

# Neuronale Netze in der Chemie

Von Johann Gasteiger\* und Jure Zupan\*

Die Leistungen des menschlichen Gehirns haben seit jeher fasziniert und Anlaß gegeben, ihre Grundlagen zu erforschen. In den letzten fünfzig Jahren wurde eine Reihe von Modellen entwickelt, die Teifunktionen des Gehirns nachvollziehen sollen. Die parallel dazu verlaufende Entwicklung von Computern ging ganz andere Wege. Dies hat zur Folge, daß die derzeitigen Computerarchitekturen und Betriebssysteme sowie die Programmabarbeitung sehr wenig mit der Informationsverarbeitung im Gehirn gemeinsam haben. Gegenwärtig findet eine Rückbesinnung auf die Leistungen des Gehirns statt, und die Modelle für die Informationsverarbeitung im Gehirn wurden in Algorithmen gefaßt und breit zugänglich gemacht. Zentraler Bestandteil dieser Gehirnmodelle, der neuronalen Netze, ist eine Informationsverarbeitungseinheit, das Modell eines Neurons. In einem solchen künstlichen Neuron werden recht einfache mathematische Operationen ausgeführt; die Leistungsfähigkeit eines neuronalen Netzes kommt erst durch das Zusammenschalten vieler Neuronen in einem Netzwerk zustande. So wie die einzelnen neuronalen Netze verschiedene Leistungen des Gehirns nachvollziehen, so können sie auch zur Lösung unterschiedlicher Problemtypen eingesetzt werden, zur Klassifikation von Objekten, zum Modellieren funktionaler Zusammenhänge, zur Speicherung und zum Rückruf von Information sowie zur Abbildung einer Menge von Daten. Dieses Potential bietet auch für chemische Informationen viele Verarbeitungsmöglichkeiten; die bisherigen Anwendungen überstreichen bereits ein weites Feld: Analyse spektroskopischer Daten, Vorhersage von Reaktionen, Kontrolle chemischer Prozesse und Analyse elektrostatischer Potentiale stehen stellvertretend für viele andere Anwendungen.

## 1. Einleitung

In vielen Bereichen, in denen komplexe Informationen verarbeitet werden müssen – vom Aktienmarkt über die medizinische Diagnostik bis hin zur Chemie –, wird plötzlich über „neuronale Netze“ diskutiert. Neuronale Netze erscheinen als Geheimwaffe zur Lösung einer Vielzahl von Problemen. Um dies zu unterstreichen, zeigt Abbildung 1 den dramatischen Anstieg an Veröffentlichungen über den Einsatz neuronaler Netze in der Chemie in den letzten Jahren. Was macht neuronale Netze so attraktiv? Sind sie wirklich ein Allheilmittel in der Informationsverarbeitung?

Neuronale Netze wurden als Modelle für die Informationsverarbeitung im menschlichen Gehirn entwickelt. Das

erklärt einen Teil der Faszination, die neuronale Netze ausstrahlen: Das menschliche Gehirn hat eine phänomenale Leistungsfähigkeit, die selbst von Supercomputern noch längst nicht erreicht wird. Offensichtlich verarbeitet das menschliche Gehirn Information ganz anders als die meisten der heute üblichen Computer, die nach der von-Neumann-Architektur konstruiert sind. Ein solcher von-Neumann-Computer arbeitet eine Rechenvorschchrift, einen Algorithmus, Schritt für Schritt, d.h. sequentiell ab.

Das menschliche Gehirn hingegen operiert in hohem Maße parallel: Die eingehende Information wird gleichzeitig über viele Verarbeitungseinheiten geschickt. Das lehrt schon das „100-Schritte-Paradoxon“: Aus der Neurophysiologie weiß man, daß sich eine Nervenzelle, ein Neuron, etwa eine Millisekunde, nachdem es „gefeuert“ hat, wieder erholt hat. Nun kann das menschliche Gehirn aber in ungefähr einer Zehntelsekunde intelligente Prozesse ausführen, wie etwa das Gesicht eines Freundes erkennen oder auf eine Gefahr reagieren. Das Gehirn kann also schwierige Aufgaben in weniger als 100 aufeinanderfolgenden Schritten lösen. Diese geringe Zahl an Schritten reicht natürlich zur Lösung komplexer Probleme nicht aus, so daß in Wirklichkeit vom Gehirn viele Aufgaben gleichzeitig nebeneinander gelöst werden müssen.

Künstliche neuronale Netze sind heute meistens als Softwarepakete realisiert, die auf von-Neumann-Rechnern laufen und bei denen die Parallelität nur simuliert wird. Echte Parallelverarbeitung ist nur mit entsprechender Hardware (Transputer) möglich und heute noch selten. Die Softwarelösung ermöglicht den Einsatz desselben Programms für ganz unterschiedliche Wissensbereiche. Derselbe Algorithmus kann zum Studium der Zusammenhänge zwischen chemischer Struktur und IR-Spektrum, zur Simulation eines Tennisspiels oder zur Vorhersage von Bewegungen in Aktienkurven verwendet werden.

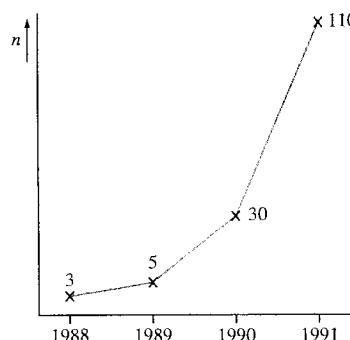


Abb. 1. Zunahme der Zahl  $n$  der Veröffentlichungen über den Einsatz neuronaler Netze in der Chemie in den Jahren 1988–1991.

[\*] Prof. Dr. J. Gasteiger  
Organisch-chemisches Institut der Technischen Universität München  
Lichtenbergstraße 4, D-8046 Garching

Prof. Dr. J. Zupan  
National Institute of Chemistry  
Hajdrihova 19, SI-61116 Ljubljana (Slowenien)

In der herkömmlichen Programmietechnik wird die Beantwortung einer Frage durch einen *problembezogenen Algorithmus* angestrebt, bei dem Anweisung genau auf das zu studierende Problem zugeschnitten ist. Oder man setzt zur Problemlösung ein Expertensystem ein, bei dem eine klare Trennung zwischen dem aufgabenspezifischen Wissen und den Mechanismen zur Schlußfolgerung und Entscheidungsfindung gezogen wird. Algorithmen neuronaler Netze beziehen sich dagegen nicht auf ein bestimmtes Wissensgebiet, sondern können ganz allgemein zur Lösung bestimmter *Typen von Problemen* eingesetzt werden, wie sie in den verschiedensten Wissensbereichen auftreten.

Nicht mehr die Rechenvorschrift ist aufgabenspezifisch, sondern die Art der Information, die in das neuronale Netz eingegeben, und die Weise, wie sie repräsentiert wird, schneiden eine Studie mit einem neuronalen Netz auf die spezielle Aufgabenstellung zu.

Neuronale Netze können zur Lösung folgender Problemtypen herangezogen werden:

– Klassifikation

Ein Objekt, charakterisiert durch eine Reihe von Eigenschaften, wird einer bestimmten Kategorie zugeordnet.

– Modellierung

Neuronale Netze können sowohl binäre als auch reelle Werte ausgeben. Damit können durch die Kombination gewisser Meßergebnisse an einem Objekt andere Eigenschaften dieses Objekts ermittelt werden. Statistische Verfahren stellen solche Zusammenhänge mit einer expliziten mathematischen Gleichung her. Neuronale Netze dagegen können solche Beziehungen implizit ausdrücken, beson-

ders auch dann, wenn eine explizite Gleichung nicht mehr aufgestellt werden kann.

– Assoziation

Neuronale Netze lassen sich zum Vergleich von Informationen einsetzen, denn sie können ähnlich geartete Informationen erfassen, beispielsweise erkennen, daß zwei Gesichter zur gleichen Person gehören, auch wenn eines der Bilder verzerrt ist (Autoassoziation). Darüber hinaus können sie auch für Zuordnungen eingesetzt werden, bei denen ein Objekt mit einem ganz bestimmten anderen Objekt in Beziehung steht (Heteroassoziation).

– Abbildung

Komplexe Information kann in eine einfachere Darstellung überführt werden (z.B. durch Projektion in eine Ebene), wobei die wesentlichen Informationsinhalte erhalten bleiben.

In dieser Übersicht werden zunächst die Grundzüge und die Typen neuronaler Netze vorgestellt, bevor ein Überblick über und ausgewählte Beispiele für die Anwendung neuronaler Netze in der Chemie gegeben werden<sup>[1, 2]</sup>.

## 2. Neuronen und Netzwerke

Wenn hier von neuronalen Netzen gesprochen wird, müßte es eigentlich immer „künstliche neuronale Netze“ heißen, denn sie wurden ja entwickelt, um die biologischen neuronalen Netze im menschlichen Gehirn nachzubilden. Der Einfachheit halber wird der Vorsatz „künstlich“ aber weggelassen.



Johann Gasteiger wurde 1941 in Dachau geboren. Er studierte Chemie an den Universitäten München und Zürich und promovierte 1971 an der Universität München bei R. Huisgen mit mechanistischen Untersuchungen an Reaktionen des Cyclooctatetraensystems. Als Postdoktorand (1971–1972) führte er an der University of California in Berkeley bei A. Streitwieser, Jr., ab-initio-Rechnungen an Carbanionen durch. 1972 wechselte er an die Technische Universität München und entwickelte in einer Gruppe von I. Ugi den Prototyp eines Syntheseplanungsprogramms. Er habilitierte sich 1978 für das Fach Chemie mit einer Arbeit über Modelle und Algorithmen zur Behandlung chemischer Reaktionen und der Reaktivität. Seit 1989 ist er apl. Professor an der TU München. Seine Forschungsschwerpunkte sind die Entwicklung von Methoden und Computerprogrammen zur Reaktionsvorhersage und Syntheseplanung, zur Auswertung und Simulation von Massenspektren und zur dreidimensionalen Modellierung von Molekülen. Er war von 1987–1991 Projektleiter für das Fachinformationszentrum Chemie zum Aufbau einer Reaktionsdatenbank auf der Basis von ChemInform und erhielt 1991 die Gmelin-Beilstein-Denk-münze der Gesellschaft Deutscher Chemiker für seine Leistungen auf dem Gebiet der Computer-chemie.



Jure Zupan wurde 1943 in Ljubljana, Slowenien, geboren. Er studierte Physik an der Universität von Ljubljana, und promovierte dort 1972 bei D. Hadzi mit einer Arbeit über die Energiebänder der Bornitride. Bis 1973 arbeitete er am Institut Josef Stefan im Bereich der Quantenchemie und der magnetischen Eigenschaften von keramischen Materialien. Seit 1974 leitet er eine Gruppe am National Institute of Chemistry in Ljubljana. Seine Arbeitsgebiete umfassen Chemometrie, künstliche Intelligenz, Expertensysteme und die Entwicklung von Computeralgorithmen für chemische Anwendungen. Als Gastprofessor war er 1982 an der Arizona State University, USA, 1988 an der Vrije Universiteit, Brüssel, und 1990–1992 an der TU München in Garching tätig. Für seine Forschung (ca. 150 Originalpublikationen, 2 Monographien und 3 Bücher) erhielt er im Jahr 1991 die höchste slowenische Forschungsauszeichnung. Er ist seit 1975 habilitiert und seit 1988 Professor für Chemometrie an der Universität von Ljubljana.

## 2.1. Ein Neuronenmodell

Neuronale Netze sind Systeme aus zu Netzwerken verknüpften Untereinheiten, den Neuronen. Das künstliche Neuron soll die Funktionen einer biologischen Nervenzelle, des Neurons, modellieren. Wenn es auch mindestens fünf physiologisch unterschiedliche Nervenzellen gibt, so genügt es doch hier, nur einen Typ vorzustellen (Abb. 2), da hier nur der Grundaufbau eines Neurons besprochen wird; auf die physiologischen Prozesse – und die sie verursachenden chemischen Vorgänge<sup>[3]</sup> – kann nicht näher eingegangen werden.

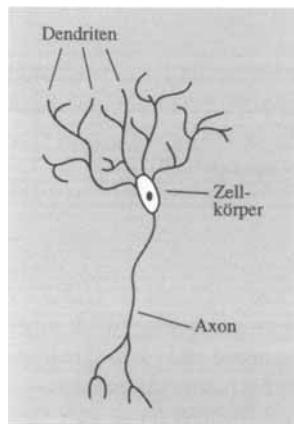


Abb. 2. Stark vereinfachte schematische Darstellung einer Nervenzelle. In Wirklichkeit ist die Zahl der Dendriten und die Zahl der Verzweigungen in den Dendriten und im Axon sehr viel höher.

Der Nervenzellkörper weist eine große Zahl von Verästelungen auf, die Dendriten, die Signale aufnehmen und zum Zellkörper leiten. Dort werden die Signale gesammelt und führen, wenn ein bestimmter Schwellenwert überschritten wird, zum „Feuern“ des Neurons, d.h. es wird ein elektrischer Reiz über das Axon weitergeleitet. An seinem Ende hat ein Axon Kontakt zu Dendriten der Nachbarneuronen; diese Kontaktstellen werden Synapsen genannt. Über sie werden die Neuronen miteinander vernetzt.

Die Synapsen sind zugleich Barrieren, die die Intensität eines Signals beim Weiterleiten ändern. Das Ausmaß dieser Änderung hängt von der Synapsenstärke ab. Das Eingabesignal mit der Stärke  $x_i$  an der Synapse  $i$  mit der Synapsenstärke  $w_i$  hat nach deren Passieren die Intensität  $s_i$  [Gl. (a), Abb. 3]. Die Synapsenstärke kann sich ändern, und das sogar zwischen zwei aufeinanderfolgenden Impulsen.

$$s_i = w_i x_i \quad (a)$$

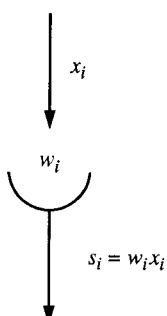


Abb. 3. Änderung eines Eingabesignals  $x_i$  beim Durchgang durch eine Synapse mit der Stärke  $w_i$ .

Jedes Neuron hat eine große Zahl von Dendriten und erhält damit gleichzeitig viele Signale. Diese  $m$  Signale vereinigen sich zu einem kollektiven Signal. Die Art und Weise, wie sich dieses kollektive Signal  $Net$  aus den Einzelsignalen zusammensetzt, ist noch nicht genau bekannt.

Bei der Entwicklung eines künstlichen Neurons werden zwei Annahmen gemacht:

1. Das Nettosignal  $Net$  ist eine Funktion aller Signale, die in einem bestimmten Zeitintervall im Neuron ankommen, und aller Synapsenstärken.
2. Diese Funktion wird meist gleich der Summe der Signale  $s_i$  gesetzt, die sich als Produkte der Eingabesignale  $x_i$  ( $i = 1, \dots, m$ ) und der Synapsenstärken  $w_i$  ( $i = 1, \dots, m$ ), jetzt Gewichte genannt, ergeben [Gl. (b)]. Abbildung 4 gibt das bis jetzt entwickelte Modell eines Neurons wieder.

$$\begin{aligned} Net = & s_1 + s_2 + \dots + s_m = w_1 x_1 + w_2 x_2 \\ & + \dots w_i x_i + \dots w_m x_m \end{aligned} \quad (b)$$

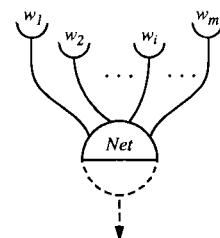


Abb. 4. Erste Stufe eines Neuronenmodells.

Das Nettosignal ist nun noch nicht das Signal, das weitergegeben wird, denn dieser kollektive Wert  $Net$  kann sehr groß und er kann vor allem auch negativ werden. Gerade diese letzte Eigenschaft kann die Wirklichkeit nicht gut wiedergeben: Ein Neuron feuert oder auch nicht, aber was soll ein negatives Nettosignal bedeuten? Um ein realistischeres Modell zu erhalten, wird der Wert von  $Net$  deshalb durch eine Übertragungsfunktion modifiziert. Am häufigsten wird dabei eine sigmoide Funktion, auch logistische oder Fermi-Funktion genannt, verwendet. Mit dieser Übertragungsfunktion wird der Wertebereich des Ausgabesignals  $out$  [Gl. (c)] zwischen Null und Eins gezwungen, ob nun  $Net$

$$out = \frac{1}{1 + e^{-(\alpha Net + \beta)}} \quad (c)$$

groß oder klein oder auch negativ ist, und vor allem, man hat nun einen nichtlinearen Zusammenhang zwischen Eingabe- und Ausgabesignal und kann damit nichtlineare Beziehungen zwischen Eigenschaften darstellen, was mit statistischen Verfahren oft schwierig ist. Darüber hinaus besitzt man mit  $\alpha$  und  $\beta$  zwei Parameter, mit denen man die Funktion des Neurons beeinflussen kann (Abb. 5).

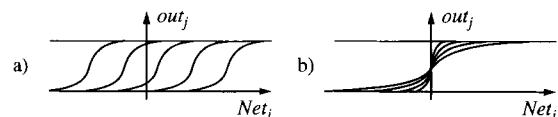


Abb. 5. Einfluß der Parameter  $\alpha$  (a) und  $\beta$  (b) auf das gemäß Gleichung (c) definierte Ausgabesignal  $out_j$ .

Die Übertragungsfunktion vervollständigt das Modell des Neurons. In Abbildung 6 a sind die Synapsenstärken oder Gewichte  $w$  noch wie in Abbildung 4 dargestellt; im weiteren werden sie, wie in Abbildung 6 b, nicht mehr gezeichnet, sind aber selbstverständlich noch zu verwenden.

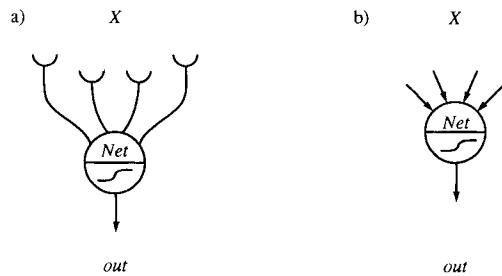


Abb. 6. Vollständiges Neuronenmodell a) mit, b) ohne explizite Angabe der Synapsenstärken  $w$ .

## Symbole und Konventionen

In der Literatur über neuronale Netze wird eine verwirrende Vielfalt an Bezeichnungsweisen und Symbolen verwendet. Um die einzelnen neuronalen Netze besser vergleichen zu können, wählen wir für den gesamten Artikel eine einheitliche Nomenklatur.

- Größen, die aus einem einzigen Wert bestehen (skalare Größen), werden mit Kleinbuchstaben in Kursivschrift wiedergegeben:  $x_i$ . (Die einzige Ausnahme ist *Net*, das großgeschrieben wird, um es – im Englischen – nicht mit dem Ausdruck für Netz zu verwechseln.)
- *Net* ist übrigens wie auch *out* Symbol für eine *einige* Größe, das ausnahmsweise zum klareren Verständnis mit *drei* Buchstaben geschrieben wird.
- Größen, die aus mehreren zusammengehörigen Werten bestehen (Vektoren oder Matrizen), werden mit Großbuchstaben in fetteter Kursivschrift dargestellt:  $X$ .
- Ein Eingabeblobjekt, das durch mehrere Einzeldaten (z.B. Meßwerte) beschrieben ist, wird demnach durch  $X$  dargestellt, die Einzelwerte durch  $x_1, x_2, \dots, x_m$ . Ein einzelner Eingabewert hieraus wird mit dem Index  $i$ , also mit  $x_i$ , beschrieben.
- Ein einzelnes Neuron aus einer Gruppe (Schicht) von  $n$  Neuronen wird mit dem Index  $j$  bezeichnet, die Gesamtheit der Ausgabesignale dieser  $n$  Neuronen mit *Out* ( $out_1, out_2, \dots, out_n$ ): Das Ausgabesignal eines beliebigen dieser  $n$  Neuronen hat demnach den Wert  $out_j$ .
- In einer Schicht mit  $n$  Neuronen, die  $m$  Eingabedaten erhalten, gibt es  $n \times m$  Gewichte, die in einer Matrix  $W$  ( $w_{11}, w_{12}, \dots, w_{nm}$ ) organisiert sind. Ein einzelnes Gewicht hieraus wird durch  $w_{ji}$  bezeichnet.
- Gibt es mehrere Eingabeblobjekte, dann werden sie mit dem Index  $s$  unterschieden:  $X_s$ ; die einzelnen Daten heißen dann  $x_{si}$ .
- In einem Mehrschichtennetz werden die Schichten durch einen hochgestellten Index  $l$  bezeichnet, z.B.  $out_j^l$ .
- Iterationen in einem neuronalen Netzwerk werden durch hochgestellte Indices  $t$ , die in Klammern geführt werden, charakterisiert, z.B.  $W^{(t)}$ .

## 2.2. Vernetzung von Neuronen

Das 100-Schritte-Paradoxon hat gelehrt, daß die Stärke des menschlichen Gehirns in der parallelen Verarbeitung von Information begründet ist. Das soeben vorgestellte Neuronenmodell ist sehr einfach gestaltet, aber selbst viel kompliziertere Modelle führen zu keiner sonderlich höheren Leistungsfähigkeit. Die eigentlichen Kapazitäten und die Flexibilität neuronaler Netze kommen erst durch die Vernet-

zung der einzelnen Recheneinheiten, der künstlichen Neuronen, zustande.

Viele Arten der Vernetzung von Neuronen wurden untersucht, und in den folgenden Abschnitten werden einige Netzwerkmodelle und -architekturen vorgestellt. Am häufigsten sind Schichtenmodelle, und an dieser Netzwerkarchitektur soll daher die Funktion eines neuronalen Netzes erläutert werden.

In einem Schichtenmodell werden die Neuronen in Gruppen, den Schichten, zusammengefaßt. Die Neuronen einer Schicht haben untereinander keine Verbindung, sondern nur zu den Neuronen der darüber- und der darunterliegenden Schicht. In einem Einschichtennetzwerk gehören alle Neuronen einer einzigen Schicht an (Abb. 7). Jedes Neuron  $j$  erhält Zugang zu allen Eingabedaten  $X$  ( $x_1, x_2, \dots, x_i, \dots, x_m$ ) und erzeugt daraus einen für dieses Neuron spezifischen Ausgabewert  $out_j$ .

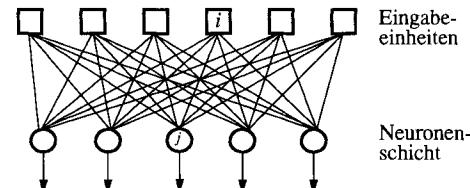


Abb. 7. Neuronales Netz mit den Eingabeeinheiten (Quadrate) und einer Schicht aktiver Neuronen (Kreise).

In Abbildung 7 sind oben die Eingabeeinheiten gezeichnet. Sie stellen keine Neuronenschicht dar, da sie nicht die typischen Rechenoperationen der Neuronen, Bildung des Nettosignals *Net* und dessen Umwandlung durch die Übertragungsfunktion in das Ausgabesignal *out*, ausführen. Zum Unterschied von Neuronen, die in den folgenden Abbildungen von Netzen durch Kreise symbolisiert sind, werden Eingabeeinheiten durch Quadrate dargestellt.

Die Eingabeeinheiten dienen nur dazu, die einzelnen Eingabedaten auf sämtliche Neuronen der darunterliegenden Schicht zu verteilen, wobei in den Neuronen unterschiedliche Werte ankommen, da jede Verbindung einer Eingabeeinheit  $i$  zu einem Neuron  $j$  ein ganz spezifisches Gewicht  $w_{ji}$  hat, das einer bestimmten Synapsenstärke entspricht. Die Größe der Gewichte muß in einem Lernprozeß ermittelt werden; dies ist jedoch das Thema von Abschnitt 4.

Der Ausgabewert eines Neurons,  $out_j$ , wird durch die Gleichungen (d) und (e) bestimmt, die Verallgemeinerungen der Gleichungen (b) bzw. (c) sind. Der Index  $j$  läuft dabei über alle  $n$  Neuronen, der Index  $i$  über alle  $m$  Eingabewerte.

$$Net_j = \sum_{i=1}^m w_{ji} x_i \quad (d)$$

$$out_j = \frac{1}{1 + e^{-(\alpha Net_j + \beta)}} \quad (e)$$

In einem Einschichtennetzwerk sind die Ausgabesignale  $out_j$  der einzelnen Neuronen bereits die Ausgabewerte des neuronalen Netzes.

Die Gleichungen (d) und (e) legen eine formalere Repräsentation eines Neurons und eines neuronalen Netzes nahe:

Die Eingabedaten können als Vektor  $X$  ( $x_1, x_2, \dots, x_i, \dots, x_m$ ) aufgefaßt werden, der durch die Matrix der Gewichte,  $W$ , mit den Einzelementen  $w_{ji}$  und die Übertragungsfunktion in den Vektor der Ausgabewerte,  $Out$  ( $out_1, out_2, \dots, out_j, \dots, out_n$ ), umgewandelt wird (Abb. 8).

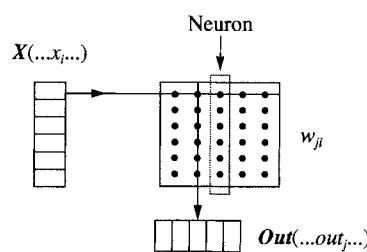


Abb. 8. Matrizendarstellung eines Einschichtennetzwerks, das die Eingabedaten  $X$  mit Hilfe der Gewichte  $w_{ji}$  in die Ausgabedaten  $Out$  umwandelt.

Jedes Neuron entspricht einer Spalte in der Matrix der Abbildung 8. In dieser Matrizendarstellung ist ebenfalls deutlich zu sehen, daß jeder Eingabewert in jedes Neuron geschickt wird. Aus der Matrizenrepräsentation eines neuronalen Netzes ist auch die algorithmische Realisierung eines Schichtenmodells deutlich abzulesen.

Eine einzige Schicht aus Neuronen heißt auch Perzeptronmodell. Damit hat man noch wenig Flexibilität für die Umwandlung der Eingabedaten in Ausgabewerte. Diese Begrenzungen des Perzeptronmodells können durch das Hintereinanderschalten mehrerer Schichten überwunden werden.

In einem Mehrschichtenmodell wird meist eine Architektur gewählt, bei der die Neuronen einer Schicht mit *allen* Neuronen der darüberliegenden Schicht und *allen* Neuronen der darunterliegenden Schicht verknüpft sind. Abbildung 9 stellt ein neuronales Zweischichtennetzwerk dar (wie bereits erwähnt, werden die Eingabeeinheiten nicht mitgezählt, da sie ja keine Neuronen sind, sondern nur zur Aufteilung der Eingabewerte auf die darunterliegende Neuronenschicht dienen).

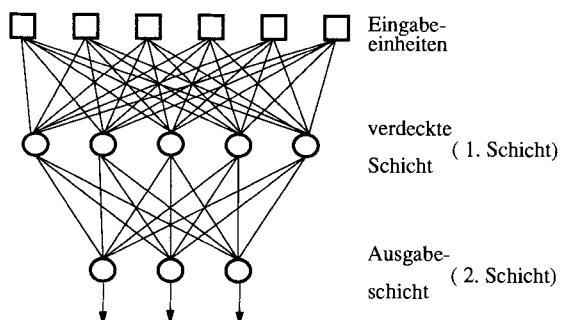


Abb. 9. Neuronales Netz mit den Eingabeeinheiten und zwei Schichten aktiver Neuronen.

Die erste Schicht an Neuronen hat keine direkten Verbindungen nach außen und heißt deshalb auch verdeckte Schicht (hidden layer); die sie bildenden Neuronen werden auch als innere Neuronen bezeichnet. Die Ausgabewerte  $Out^1$  der ersten Neuronenschicht sind die Eingabewerte  $X^2$  der zweiten Neuronenschicht. Jedes Neuron der oberen

Schicht gibt dabei seinen Ausgabewert an alle Neuronen der darunterliegenden Schicht weiter. Aufgrund der unterschiedlichen Gewichte  $w_{ji}$  in den einzelnen Verbindungen (Synapsen) hat der gleiche Ausgabewert  $Out^1 = X^2$  in den einzelnen Neuronen unterschiedliche Wirkungen [Gl. (d)]. Erst die Ausgabewerte der letzten Neuronenschicht, hier  $Out^2$ , sind das Resultat eines neuronalen Netzes. Abbildung 10 gibt ein Zweischichtennetzwerk in der Matrixnotation wieder.

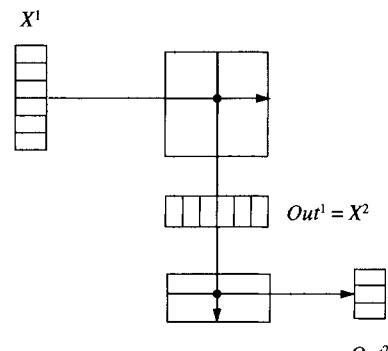


Abb. 10. Matrizendarstellung eines Zweischichtennetzwerks.

### 3. Geschichtliches

Obwohl erst in den letzten Jahren das Interesse an neuronalen Netzen enorm zugenommen hat, können sie bereits auf fast fünfzig Jahre Entwicklung zurückblicken<sup>[4]</sup>.

Bereits in den vierziger Jahren entwickelten McCulloch und Pitts<sup>[5, 6]</sup> ein Neuronenmodell als logisches Schwellenwertelement mit zwei möglichen Zuständen. Die Frage, wie neuronale Netze lernen, mußten sie offen lassen; dazu machte dann Hebb 1949 einen Vorschlag<sup>[7]</sup>. Seine Lernregel besagt, daß sich die Synapsenstärke eines Neurons proportional zur Aktivität vor und hinter der Synapse ändert.

Rosenblatt<sup>[8]</sup> organisierte einige Neuronen in einer Weise, die im wesentlichen einem Einschichtennetzwerk (Abb. 7) entspricht, und nannte diese Anordnung ein Perzeptron.

Die ursprüngliche Begeisterung bei der Entwicklung von Modellen für die Informationsverarbeitung biologischer Nervensysteme begann in den sechziger Jahren aus mehreren Gründen zu schwinden. Die Ergebnisse bei der Lösung echter Probleme durch die damaligen Netzwerke waren nicht sehr ermutigend, und mit dem Aufkommen der Computer verlagerte sich das Interesse mehr auf die Entwicklung direkt programmierter Problemlösungsmethoden. Das dafür gewählte Schlagwort „Künstliche Intelligenz“ (Artificial Intelligence) unterstreicht nur zu deutlich, welcher Anspruch erhoben wurde, obwohl von Anfang an klar war, daß die gewählte Vorgehensweise sehr wenig mit der Informationsverarbeitung im menschlichen Gehirn zu tun hatte.

Es ist daher nicht verwunderlich, daß die Erforschung neuronaler Netze einen schweren Schlag gerade von einem der Hauptvertreter der „künstlichen Intelligenz“, Marvin Minsky, erhielt. Minsky publizierte 1969 zusammen mit Papert<sup>[9]</sup> eine sehr scharfe, aber wohl berechtigte Kritik an den damaligen Netzwerkmodellen. Sie zeigten in ihrer theoretischen Studie, daß Perzeptoren – zumindest wie sie damals entwickelt waren – nur sehr begrenzte Möglichkeiten bieten.

Darüber hinaus spekulierten sie, daß eine Ausdehnung der Architektur von Perzeptronen auf mehrere Schichten keine wesentlich besseren Ergebnisse bringen würde. Diese Kritik einer so einflußreichen Persönlichkeit wie M. Minsky hatte auch zur Folge, daß kaum noch Forschungsgelder für die Modellierung biologischer Nervensysteme zu erhalten waren.

In den folgenden Jahren wurde daher nur wenig an Modellen für neuronale Netze gearbeitet; dennoch gab es auch in dieser Zeit einige wichtige Fortschritte. Erwähnt seien die Arbeiten von Albus<sup>[10]</sup>, Amari<sup>[11]</sup>, Grossberg<sup>[12]</sup>, Kohonen<sup>[13]</sup>, von der Malsburg<sup>[14]</sup> sowie Widrow und Hoff<sup>[15]</sup>.

Ein entscheidender neuer Anstoß kam dann 1982 durch den Physiker Hopfield<sup>[16]</sup>, der zeigen konnte, daß bestimmte Netzwerkmodelle aus binären Neuronen formal Spinsystemen entsprechen und mit den dort entwickelten Methoden behandelt werden können.

Den ganz großen Aufschwung in der Anwendung neuronaler Netze leiteten dann Rumelhart, Hinton und Williams durch die Publikation des „Backpropagation“-Algorithmus zum Lernen in Mehrschichtenmodellen ein<sup>[17, 18]</sup>. Wenn auch der Backpropagation-Algorithmus schon früher vorgeschlagen worden war<sup>[19]</sup>, so ist es doch das Verdienst der Forschungsgruppe über „Parallel Distributed Systems“<sup>[20]</sup>, ihn in das Blickfeld der Öffentlichkeit gerückt zu haben.

Trotz aller Erfolge bei der Entwicklung von Modellen zur neuronalen Informationsverarbeitung muß klar erkannt werden, daß wir noch weit entfernt von einem Verständnis der Arbeitsweise des menschlichen Gehirns sind. Die Leistungen künstlicher neuronaler Netze sind im Vergleich zu denen ihres biologischen Vorbilds immer noch als sehr rudimentär zu bezeichnen. Dennoch haben bereits die bisherigen Modelle neuronaler Netze neue Wege zur Verarbeitung von Information aufgezeigt. Den sich dadurch eröffnenden Möglichkeiten, insbesondere im Bereich der Chemie-Information, wollen wir uns in diesem Beitrag bevorzugt widmen.

#### 4. Architekturen und Lernverfahren

Im Laufe der Jahre entwickelte eine Reihe von Forschergruppen ganz charakteristische künstliche neuronale Netze. Diese Modelle kommen den biologischen Vorbildern unterschiedliche nahe; manche scheinen sehr gut gewisse Prozesse im menschlichen Gehirn nachzuahmen, andere haben nur recht entfernte Ähnlichkeit mit ihren biologischen Vorbildern.

Uns geht es hier aber nicht in erster Linie um die Realitätsnähe der einzelnen Modelle, sondern wir wollen vor allem zeigen, welche Art von Problemen man mit den verschiedenen Modellen bearbeiten kann und welche Leistungen diese neuronalen Netze zur Verarbeitung von Information bieten können. Ein künstliches neuronales Netz kann durchaus große Bedeutung für die Informationsverarbeitung haben, auch wenn es nur entfernt biologischen Systemen ähnelt.

In der Einleitung wurde bereits erwähnt, daß neuronale Netze zur Lösung einer ganzen Reihe von Problemen eingesetzt werden können: Klassifikation, Modellierung, Assoziation, Abbildung.

Die einzelnen neuronalen Netze eignen sich in unterschiedlichem Maß zur Behandlung dieser Problemtypen, haben also jeweils besondere Stärken und Schwächen. Dies soll

hier dargelegt werden, und wir wollen ein Gefühl dafür entwickeln, welches Netzwerkmodell bei welcher Aufgabenstellung eingesetzt werden soll.

Drei Elemente sind die wesentlichen Charakteristika jedes Modells eines neuronalen Netzes:

1. die Rechenoperationen in einem Neuron
2. die Architektur eines Netzes, also wie die einzelnen Neuronen verknüpft sind
3. das Lernverfahren, das die Gewichte so anpaßt, daß die richtige Antwort erhalten wird.

Das im Abschnitt 2 vorgestellte Modell eines Neurons ist nur eines von vielen, wenn auch ein sehr weit verbreitetes, und auch die Anordnung von Neuronen in Schichten muß nicht immer so gegeben sein, wie in Abschnitt 2 gezeigt. Das Lernverfahren allerdings ist sehr eng an die Architektur des neuronalen Netzes gekoppelt. Es gibt zwei Typen von Lernprozessen: Lernen mit und Lernen ohne Unterweisung („supervised“ und „unsupervised learning“).

Beim Lernen mit Unterweisung werden dem neuronalen Netz eine Reihe von Objekten präsentiert und ihm zu den Eingabedaten  $X$  dieser Objekte die zu erwartenden Ausgabewerte  $Y$  vorgegeben. Die Gewichte im neuronalen Netz werden dann so angepaßt, daß für den Satz an  $p$  bekannten Objekten die Ausgabewerte des neuronalen Netzes möglichst gut mit den Erwartungswerten  $Y$  übereinstimmen (Abb. 11).

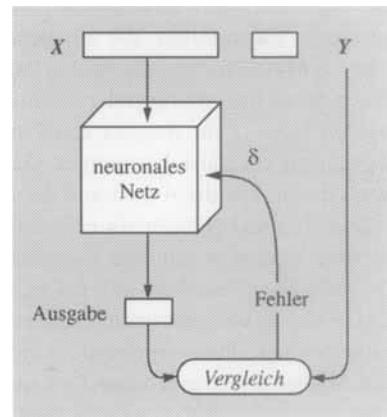


Abb. 11. Lernen mit Unterweisung. Ein Vergleich mit den Erwartungswerten liefert einen Fehler von der Größe  $\delta$ , die über die Notwendigkeit weiterer Anpassungscyclen entscheidet.

Beim Lernen ohne Unterweisung werden die Eingabedaten so lange über das Netzwerk geschickt, bis sich die Ausgabedaten stabilisiert haben und die Eingabewerte ein Objekt in bestimmte Bereiche des neuronalen Netzes abbilden (Abb. 12).

##### 4.1. Das Hopfield-Modell

Der amerikanische Physiker Hopfield brachte 1982 mit seinem Modell neues Leben in die Forschung über neuronale Netze<sup>[16]</sup>. Er wies auf Analogien zwischen neuronalen Netzen und Spinsystemen hin und konnte damit eine ganze Reihe mathematischer Methoden aus der Theoretischen Physik in die Erforschung neuronaler Netze einbringen. Außerdem führte er nichtlineare Übertragungsfunktionen ein.

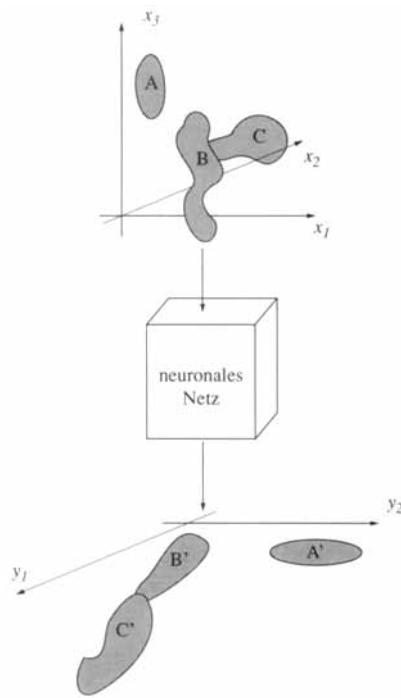


Abb. 12. Lernen ohne Unterweisung.

Das Hopfield-Netz führt eine der interessantesten Tätigkeiten des menschlichen Gehirns aus: Es kann *assoziiieren*. Das bedeutet, daß gespeicherte Bilder (oder jede andere komplexe Information, die als mehrdimensionaler Vektor oder als Matrix dargestellt werden kann) bereits anhand von Teilen dieser Information oder einem verzerrten Bild erkannt werden kann. Zum Beispiel kann in einer Sammlung von Gesichtern ein ganz bestimmtes Gesicht erkannt werden, wenn davon nur die Augen und die Nase gezeigt werden.

Das Hopfield-Netz ist ein Einschichtenmodell, das genau so viele Neuronen hat, wie Eingabedaten vorhanden sind. Da jede Eingabeeinheit mit jedem Neuron verbunden ist (vgl. Abb. 7), hat man bei  $m$  Eingabedaten  $m \times m$  Gewichte zu bestimmen. Das ursprüngliche Hopfield-Modell arbeitet mit bipolaren Eingabedaten (+1 oder -1) [Gl. (f)]. Das

$$x_i \begin{cases} +1 \\ -1 \end{cases} \quad (f)$$

Nettoergebnis im Neuron  $j$  wird, wie in Gleichung (d), durch Multiplikation aller Eingabesignale  $x_i$  mit den Gewichten  $w_{ji}$  dieses Neurons erhalten. Die Übertragungsfunktion ist hier eine einfache Stufenfunktion (Abb. 13), wie sie durch die Vorzeichenfunktion (sign) verwirklicht werden kann. Das Ausgabesignal eines Neurons in einem Hopfield-Netz ergibt

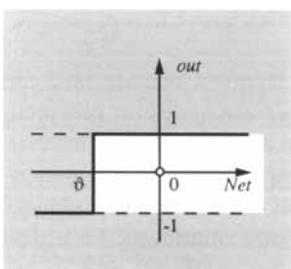


Abb. 13. Die Stufenfunktion als Übertragungsfunktion.

sich damit gemäß Gleichung (g), wobei, wie bereits erwähnt,  $x_i$  nur die Werte +1 und -1 annehmen kann.

$$out_j = \text{sign}(Net_j) = \text{sign} \sum_{i=1}^m w_{ji} x_i \quad (g)$$

Um das Lernen in einem Hopfield-Netz zu verstehen, wollen wir gleich ein Beispiel anführen: Ein Hopfield-Netz soll die vier einfachen Bilder der Abbildung 14 lernen. Jedes Bild

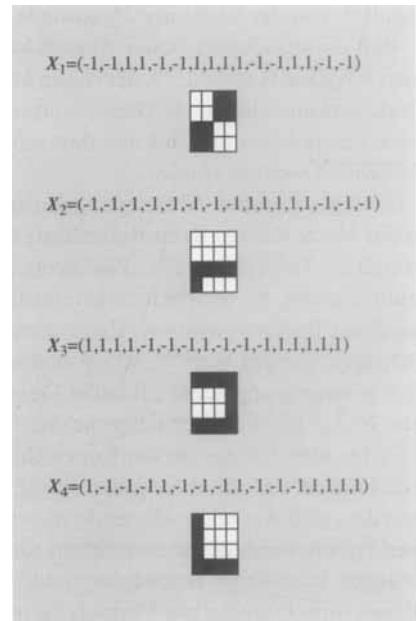


Abb. 14. Vier Bilder,  $X_1$ ,  $X_2$ ,  $X_3$  und  $X_4$ , mit denen ein Hopfield-Netz trainiert wird.

besteht aus  $4 \times 4$  Feldern (Pixeln), die entweder schwarz (+1) oder weiß (-1) sind, und läßt sich somit auch als  $4 \times 4 = 16$ dimensionaler Vektor  $X$  darstellen. In Abbildung 15 ist die Architektur des entsprechenden Hopfield-Netzes und die Eingabe des ersten Bildes mit Werten von -1 und +1 dargestellt. Für ein zum Training verwendetes Bild sind die Ausgabewerte genau gleich den Eingabewerten.

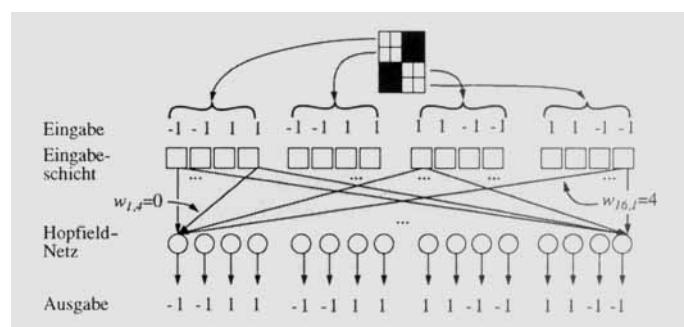


Abb. 15. Architektur des Hopfield-Netzes sowie Ein- und Ausgabe des ersten Bildes der Abbildung 14. Die angegebenen  $w_{ji}$ -Werte ergeben sich für die vier Bilder aus Abbildung 14 mit Gleichung (h).

Die Gewichte  $w$  in einem Hopfield-Netz müssen nun nicht in einem aufwendigen, iterativen Lernverfahren ermittelt werden, sondern lassen sich aus allen zu lernenden Bildern

direkt berechnen: Werden dem Hopfield-Netz  $p$  Bilder präsentiert, so lassen sich die Gewichte  $w_{ji}$  für das Neuron  $j$  aus den Eingabewerten der einzelnen Bilder  $s$ , also  $X_s$ , ermitteln nach den Gleichungen (h) und (i).

$$w_{ji} = \sum_{s=1}^p x_{sj} x_{si} \quad \text{für } j \neq i \quad (\text{h})$$

$$w_{ii} = 0 \quad \text{für } j = i \quad (\text{i})$$

Das bedeutet, daß das Gewicht  $w_{ji}$  um den Wert 1 zunimmt, wenn in einem bestimmten Bild die Felder  $j$  und  $i$  beide schwarz oder beide weiß sind, und um den Wert 1 abnimmt, wenn in einem Bild das  $i$ -te und  $j$ -te Pixel verschiedene Farben haben. Je mehr Bilder in den Pixeln  $j$  und  $i$  übereinstimmen, umso größer ist das Gewicht  $w_{ji}$ . Die vier Bilder der Abbildung 14 werden demnach in einem Hopfield-Netz aus 16 Neuronen mit 16 Gewichten in einer  $16 \times 16$ -dimensionalen Gewichtematrix gespeichert.

Zunächst soll getestet werden, ob ein Hopfield-Netz stabilisiert ist. Dazu wird ein Eingabevektor (z.B. eines der vier Bilder der Abb. 14) in das Hopfield-Netz eingegeben und mit Gleichung (g) die Ausgabewerte des Netzes bestimmt. Diese Ausgabewerte werden mit den Eingabewerten verglichen. Sind sie gleich, kann abgebrochen werden, andernfalls werden die Ausgabewerte als neue Eingabewerte auf das Netz gegeben und der Vorgang wiederholt (Abb. 16). Erhält man nach wenigen Cycles als Ausgabe die ursprünglichen Eingabewerte, so kann man von einem stabilisierten Netz sprechen. Natürlich ist ein Hopfield-Netz nicht dazu da, als Ausgabe die ursprünglichen Eingabewerte zu produzieren; dies war nur der Stabilitätstest.

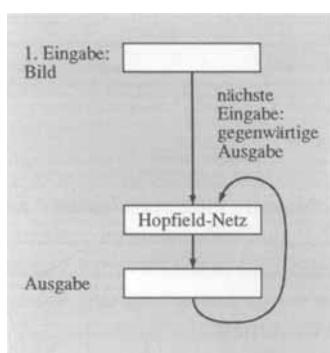


Abb. 16. Test eines Hopfield-Netzes auf Stabilität.

Der wahre Wert eines Hopfield-Netzes kommt zum Tragen, wenn mit unvollständigen oder gestörten Daten die ursprünglich gespeicherten Daten abgerufen werden können, wenn also z.B. ausgehend von einem verwaschenen oder trüben Bild das Originalbild im Hopfield-Netz gefunden werden kann.

Wir wollen dies mit den vier Bildern der Abbildung 14 und einem Hopfield-Netz, in dem diese vier Bilder – in Form von  $16 \times 16 = 256$  Gewichten – gespeichert sind, untersuchen. Gestörte Bilder stellen wir uns dadurch her, daß wir in den ursprünglichen Bildern einige Pixel ändern, also Felder von schwarz nach weiß und umgekehrt färben. In Abbildung 17 sind die Ergebnisse aufgeführt, die erhalten werden, wenn

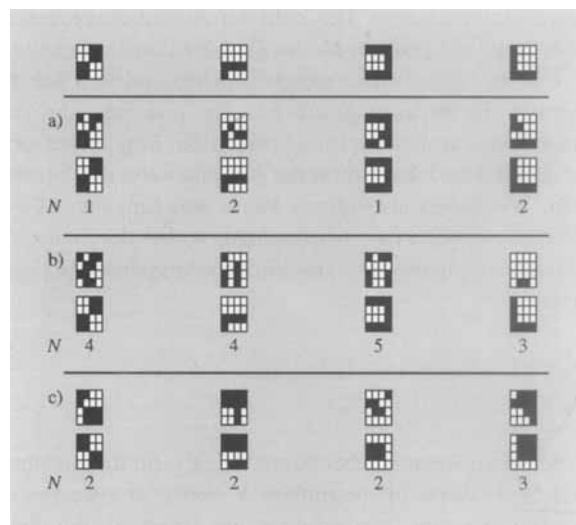


Abb. 17. Suche nach gespeicherten Bildern (ganz oben gezeigt) in einem Hopfield-Netz bei Eingabe verrauschter Bilder mit a) zwei, b) fünf und c) dreizehn veränderten Feldern.  $N$  gibt die Zahl der Iterationen an, die nötig waren, um aus der jeweils oben gezeigten fehlerhaften Eingabe das jeweils darunter abgebildete Ergebnis zu erhalten.

jedes einzelne Bild um zwei, fünf oder sogar dreizehn Pixel verändert wurde.

Man kann sehen, daß bei einer Störung in zwei Feldern die Originalbilder nach 1–2 Iterationen bereits richtig ausgegeben werden. Nach einer Änderung von fünf Feldern (31 % Verrauschtung!) werden immer noch die ursprünglichen Bilder gefunden, allerdings erst nach 3–5 Iterationen. Dies muß nicht immer so sein; wir haben bei 5-Pixel-Fehlern auch schon erlebt, daß ein falsches Bild oder ein falsches Bild in negativer Form (schwarz und weiß vertauscht) ausgegeben wird oder eine Oszillation zwischen zwei Mustern, die keinem gespeicherten Bild entsprechen, auftritt, so daß abgebrochen werden muß.

Werden dreizehn Pixel in den Bildern verändert (81 % Störung), werden nach 2–3 Iterationen die ursprünglichen Bilder als Negative erhalten. Wie kommt das zustande? Nun, immerhin ist bedeutend mehr als die Hälfte der Felder in ihrer Farbe verändert worden. Aber es ist doch bemerkenswert, daß immer nur das Negativ des ursprünglichen Bildes und nicht das Negativ eines anderen Bildes erhalten wird.

Wir haben also gesehen, daß ein so einfaches Modell wie das Hopfield-Netz eine der interessantesten Leistungen des menschlichen Gehirns, die Assoziation, nachvollziehen kann. Einen Nachteil hat aber ein Hopfield-Netz: Die Zahl der Muster (Bilder, Vektoren), die abgespeichert werden können, ist sehr begrenzt. Mit mehr Bildern braucht man auch immer mehr Neuronen, und damit wird die Gewichtematrix schnell sehr groß.

#### 4.2. Ein adaptiver bidirekionaler Assoziativspeicher (ABAM)

Wir haben am Hopfield-Netz gesehen, daß ein neuronales Netz assoziieren kann. Ein adaptiver bidirekionaler Assoziativspeicher (*Adaptive Bidirectional Associative Memory = ABAM*)<sup>[21]</sup> kann dies ebenfalls. Wir werden hier aber noch zeigen, daß ein ABAM darüber hinaus auch Muster zusammensetzen kann. Ein ABAM ist wie ein Hopfield-Netz

ein Einschichtennetz. Die Zahl der Ausgabeneuronen  $n$  ist aber meist viel geringer als die Zahl der Eingabeeinheiten  $m$ .

Um die Schreibweise zu vereinfachen und weil sich beim ABAM die Bedeutung von Eingabe und Ausgabe verwischen kann, wollen wir im folgenden den Eingabevektor nur noch mit  $X$  und den Vektor der Ausgabewerte mit  $Y$  bezeichnen. Wir haben also immer Paare von Eingabe- ( $X_s$ ) und Ausgabewerten ( $Y_s$ ) zu betrachten, wobei der Index  $s$  die einzelnen Eingabe- und zugehörigen Ausgabewerte charakterisiert:

$$X_s = (x_{s1}, x_{s2}, \dots, x_{sm})$$

$$Y_s = (y_{s1}, y_{s2}, \dots, y_{sn})$$

Mit einer Reihe solcher Paare  $\{X_s, Y_s\}$ , für die bekannt ist, welche  $Y_s$ -Werte für bestimmte  $X_s$ -Werte zu erwarten sind, werden in einem Lernverfahren mit Unterweisung die Gewichte bestimmt.

Zunächst werden nach Gleichung (j) Startwerte für die Gewichte festgelegt. Die Gewichtematrix ist nun nicht mehr

$$w_{ji} = \sum_{s=1}^p x_{sj} y_{si} \quad (j)$$

quadratisch wie beim Hopfield-Netz, sondern rechteckig; sie hat die Dimension  $m \times n$ .

Die Grundidee des Lernens in einem ABAM-Netz ist, daß man eine  $m \times n$ -Matrix auf zweierlei Weise multiplizieren kann: Standardmäßig durch Multiplikation mit einem  $m$ -dimensionalen Vektor, wodurch sich ein  $n$ -dimensionaler Vektor ergibt, oder in transponierter Form durch Multiplikation mit einem  $n$ -dimensionalen Vektor, wodurch sich ein  $m$ -dimensionaler Vektor ergibt (Abb. 18).

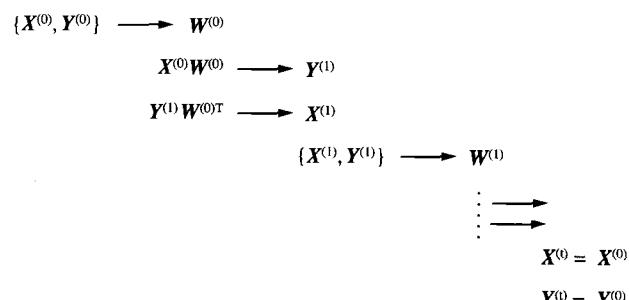


Abb. 18. Lernvorgang in einem adaptiven bidirektionalen Assoziativspeicher (ABAM).

Das Lernverfahren spielt sich nun so ab, daß aus den vorgegebenen Paaren  $\{X^{(0)}, Y^{(0)}\}$  eine Anfangsgewichtematrix  $W^{(0)}$  berechnet wird; aus  $X^{(0)}$  und  $W^{(0)}$  ergeben sich Ausgabewerte  $Y^{(1)}$ , die noch nicht mit den vorgegebenen Zielwerten übereinstimmen. Daher werden diese  $Y^{(1)}$ -Werte mit der transponierten Matrix  $W^{(0)T}$  multipliziert, wodurch  $X^{(1)}$ -Werte erhalten werden. Aus den Paaren  $\{X^{(1)}, Y^{(1)}\}$  wird nach Gleichung (j) eine neue Matrix  $W^{(1)}$  berechnet. Dieses Verfahren wird so lange fortgesetzt, bis das Paar  $\{X^{(t)}, Y^{(t)}\}$  mit den vorgegebenen Werten  $\{X^{(0)}, Y^{(0)}\}$  übereinstimmt.

Bei diesem Vorgang werden die  $Y$ -Werte in der üblichen Weise berechnet, wobei auch hier die einzelnen  $x$ - und  $y$ -Werte bipolar sind ( $+1, -1$ ) [Gl. (k) und (l)].

$$Net_j = \sum_{i=1}^m w_{ji} x_i \quad (k)$$

$$y_i = \text{sign}(Net_j) \quad (l)$$

Der Einsatz eines ABAM sei wie beim Hopfield-Netz anhand einfacher Bilder illustriert. Als Eingabeinformation dienen Bilder aus  $5 \times 5$ -Feldern, die schwarz (+1) oder weiß (-1) sein können, die also durch einen 25dimensionalen Vektor dargestellt werden können, so daß 25 Eingabeeinheiten benötigt werden. Wir verwenden nur fünf derartige Bilder und identifizieren sie auf der Ausgabeseite durch einen fünfdimensionalen Vektor. Jedem der fünf Muster (Bilder) wird eine Position in diesem Vektor zugeordnet, also beim ersten Bild ist die erste Position eine Eins, alle anderen Positionen enthalten eine Null (10000), das zweite Bild wird als (01000) repräsentiert usw. (Eigentlich haben wir es mit den bipolaren Werten +1 und -1 zu tun, aber der Übersichtlichkeit halber schreiben wir diese im folgenden in binärer Notation (1,0)). Abbildung 19 gibt die fünf Muster und ihre Identifikation wieder. Das ABAM hat eine  $25 \times 5$ -Architektur, also auch  $25 \times 5$  Gewichte.

Mit diesen fünf Bildern und den ihnen zugeordneten fünfdimensionalen Vektoren (Abb. 19) wurde ein ABAM trainiert. Dann wurde die Fähigkeit des ABAM untersucht, ge-

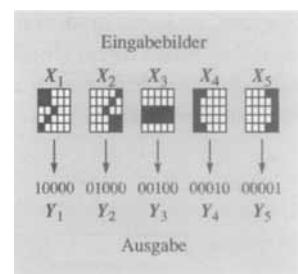


Abb. 19. Fünf Bilder, mit denen ein ABAM trainiert wurde, und ihre Identifikation.

störte Bilder mit den richtigen, eingespeicherten zu assoziieren. Alle möglichen 1-Pixel-Fehler wurden erzeugt; bei fünf Bildern mit  $5 \times 5$  Pixeln sind dies 125 verschiedene gestörte Bilder. In allen Fällen wurde das richtige ungestörte Ausgabebild von dem ABAM ermittelt.

Nun wurde untersucht, ob das ABAM fähig ist, Einzelinformationen zusammenzusetzen, d. h. zu erkennen, wenn ihm Bilder aus zwei Mustern präsentiert werden, welche Kombination von Mustern dies ist, obwohl es vorher nur die einzelnen Muster kennengelernt hatte.

In Abbildung 20 sind alle zehn Zweierkombinationen der fünf Muster als oberes Dreieck einer Matrix, die von den ursprünglichen Mustern aufgespannt wird, gezeichnet, und unter ihnen ist jeweils die Antwort des ABAM als fünfdimensionaler Vektor aufgeführt. In acht Fällen gibt das ABAM genau an, an den richtigen zwei Positionen eine Eins zurück. So liefert es z.B. für das erste kombinierte Muster, das aus den Mustern 1 und 2 zusammengesetzt wurde, die Antwort (11000), sagt also, daß das erste und das zweite Muster darin enthalten sind. Bei zwei Musterkombinationen (1 + 3 sowie 3 + 5) wird ebenfalls erkannt, aus welchen Einzelmustern sie zusammengesetzt sind, zusätzlich wird aber

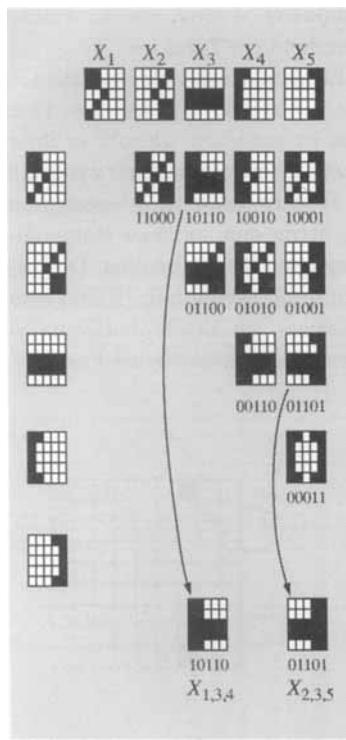


Abb. 20. Alle Kombinationen aus zwei Bildern der Abbildung 19. Die zwei Fälle, bei denen vom ABAM fälschlicherweise drei Bits aktiviert werden, werden mit der in der untersten Reihe gezeichneten Kombination dreier Bilder verglichen. Einzelheiten siehe Text.

fälschlicherweise noch jeweils ein drittes Muster als daran beteiligt angesehen. Werden diese Dreierkombinationen tatsächlich erzeugt, wie in der untersten Zeile der Abbildung 20 geschehen, so erkennt man, daß sie sich nur sehr wenig (1 Pixel Unterschied) von den Zweierkombinationen unterscheiden. Die Antwort des ABAM in diesen zwei Fällen liegt also nur sehr wenig von der Wirklichkeit entfernt.

Die Fähigkeit des ABAM, Muster zusammenzusetzen, wurde hier aus zwei Gründen etwas ausführlicher diskutiert. Zum einen ist natürlich die Fähigkeit zu erkennen, daß eine Information aus Einzelinformationen zusammengesetzt (kombiniert) ist, eine wichtige Leistung des menschlichen Gehirns. Wenn das ABAM dies auch kann, so ist damit wiederum ein wichtiger Aspekt der biologischen neuronalen Informationsverarbeitung nachgeahmt worden.

Zum anderen wird die Fähigkeit zum Zusammensetzen von Einzelinformationen bei vielen Problemstellungen in der Chemie benötigt, vor allem beispielsweise zum Erkennen der Zusammenhänge zwischen Struktur und spektroskopischen Daten. Würde einem neuronalen Netz dieser Zusammenhang anhand von Paaren aus Spektren und Strukturen angeleert, so sollte es aus einem neuen Spektrum alle vorhandenen Substrukturen ableiten können, auch wenn es eine bestimmte Kombination an Substrukturen vorher noch nie in einer einzigen Struktur zusammen „gesehen“ hatte.

### 4.3. Das Kohonen-Netz

#### 4.3.1. Grundlagen

T. Kohonen<sup>[22, 23]</sup> hat ein neuronales Netz entwickelt, das von allen Modellen die größte Ähnlichkeit mit dem biologi-

schen Vorbild hat<sup>[24]</sup>. Insbesondere gilt dies dafür, wie das Gehirn sensorische Signale verarbeitet.

In der Gehirnrinde befindet sich ein breiter Streifen, der auf die Wahrnehmung von Tastreizen spezialisiert ist und als somatosensorischer Kortex bezeichnet wird. Er ist in Bereiche gegliedert, die jeweils für bestimmte Körperteile zuständig sind, wobei zu Körperoberflächen mit besonders vielen Sinnesrezeptoren entsprechend große und zusammenhängende Felder gehören, während die Felder zu Hautregionen mit wenigen Tastnerven klein sind, selbst wenn der Körperteil im Verhältnis eigentlich sehr groß ist. Zudem sind benachbarte Körperteile auch benachbarten Regionen im somatosensorischen Kortex zugeordnet, so daß im Gehirn für den Tastsinn ein verzerrtes Abbild der Körperoberfläche vorliegt (Abb. 21).

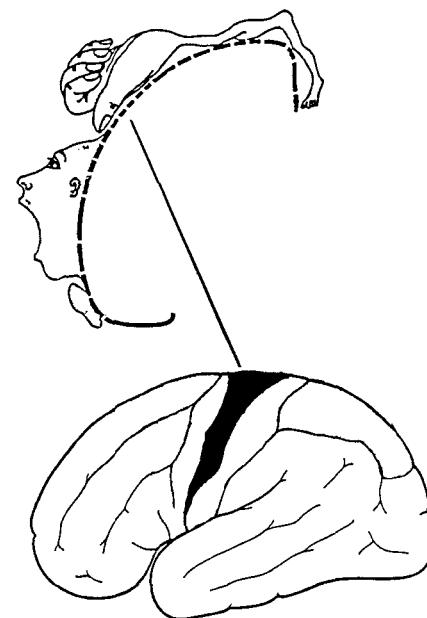


Abb. 21. Karte des menschlichen Körpers (oben) im somatosensorischen Kortex des Gehirns (unten markiert).

Kohonen hat das Konzept der „selbstorganisierten topologischen Karten“ (self-organized topological feature maps) eingeführt, das solche Abbildung erzeugen kann. Es handelt sich dabei um zweidimensionale Anordnungen von Neuronen, die die Topologie einer Information, d. h. die Beziehungen zwischen einzelnen Daten und nicht deren Größe, möglichst gut wiedergeben sollen. Mit dem Kohonen-Modell läßt sich die Abbildung einer mehrdimensionalen Information in eine Ebene aus Neuronen erzielen, wobei die wesentlichen Inhalte (Beziehungen) der Information erhalten bleiben; der Vorgang stellt also eine Abstraktion dar. Abbildung 22 zeigt

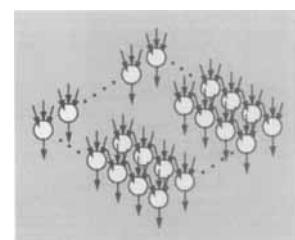


Abb. 22. Zweidimensionale Anordnung der Neuronen eines Kohonen-Netzes.

die zweidimensionale Anordnung der Neuronen eines Kohonen-Netzes.

Abbildung einer Information bedeutet in diesem Zusammenhang, daß sich die Ähnlichkeit zweier Signale in der Nachbarschaftsbeziehung der durch sie aktivierten Neuronen ausdrückt: Je ähnlicher zwei Signale sind, desto näher sollen sich die von ihnen angeregten Neuronen sein. Da es sich hier aber um eine topologische und nicht um eine euklidische Distanz handelt, hat ein Neuron in einer quadratischen Anordnung (Abb. 23 a) acht Nachbarn in der ersten Sphäre, da acht Neuronen direkt zu ihm benachbart sind. In einem Kohonen-Netz mit quadratisch angeordneten Neuronen wachsen die Nachbarschaftssphären wie in Abbildung 23 b gezeigt durch das Netz.

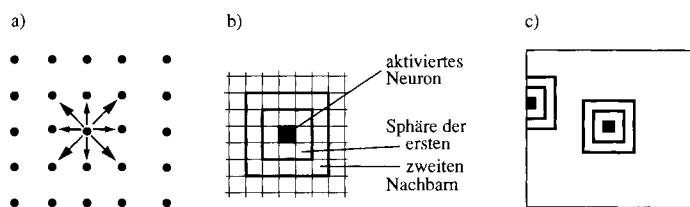


Abb. 23. Nachbarschaftsbeziehungen für die Neuronen in einem Kohonen-Netz. a) Erste Nachbarschaftssphäre; b) Wachstum der Nachbarschaftssphären; c) Neuronen am Netzrand.

Wir müssen die Diskussion der Topologie des Kohonen-Netzes noch etwas fortführen, bevor zum Lernalgorithmus übergegangen werden kann, denn die Topologie ist der entscheidende Begriff beim Kohonen-Netz.

Wenn jedes Neuron gleich viele Nachbarn haben soll, ist eine quadratisch-planare Anordnung schlecht geeignet, denn Neuronen an den Rändern haben dann weniger Nachbarn als solche im Zentrum des Netzes (Abb. 23 c). Wir können aber aus einer quadratischen oder rechteckigen Anordnung von Elementen (Neuronen) leicht eine Anordnung erzeugen, bei der jedes Element genau gleich viele Nachbarn hat. Dazu muß man nur die Fläche umwölben und an jeweils zwei gegenüberliegenden Kanten verbinden („zusammenkle-

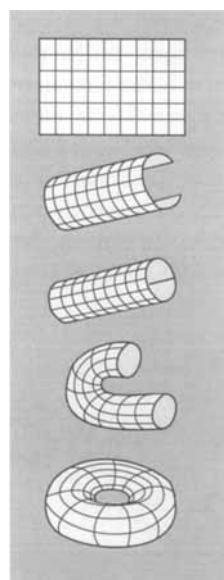


Abb. 24. Überführung einer rechteckigen Anordnung von Neuronen in einen Torus, bei dem jedes Neuron gleich viele Nachbarn hat.

ben“). So entsteht, wie Abbildung 24 zeigt, aus der Fläche ein Zylinder und aus diesem dann ein Torus.

In einem Torus hat jedes Element gleich viele Nachbarn, 8 in der ersten Sphäre, 16 in der zweiten Sphäre usw. Eine Abbildung auf einem Torus ist natürlich schlecht in ihrer Gesamtheit sichtbar zu machen. Wir werden daher weiterhin Kohonen-Netze als ebene Flächen darstellen, wissen nun aber, daß sich diese Fläche, wenn man an einer Kante ankommt, an der gegenüberliegenden Kante fortsetzt. Die beiden voll ausgefüllten Quadrate in Abbildung 25 sind also direkt benachbart, und Gleiches gilt in der horizontalen Richtung für die durch Kreuze gekennzeichneten Felder.

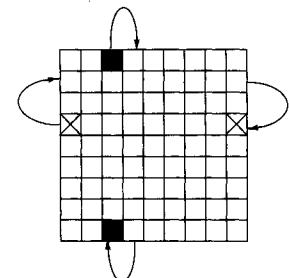


Abb. 25. Darstellung der Fläche eines Torus in einer Ebene. Die obere Kante setzt sich unten fort, die linke Kante schließt direkt an die rechte an.

Die Topologie eines Kohonen-Netzes wurde hier etwas ausführlicher diskutiert, um Anwendungsbeispiele besser verstehen zu können (siehe Abschnitte 10.1 und 10.2).

#### 4.3.2. Das Lernverfahren

Im Kohonen-Netz wird kompetitiv gelernt: Alle Neuronen des Netzes konkurrieren um die Stimulation durch das Eingabesignal. Das Eingabesignal ist dabei ein Objekt, das durch  $m$  Einzelwerte beschrieben wird; es kann also als Punkt in einem  $m$ -dimensionalen Raum aufgefaßt werden, der auf eine Ebene abgebildet werden soll. Nur ein einziges Neuron wird schließlich als bestes – als das zentrale Neuron  $c$  – ausgewählt („The winner takes all“), wobei unterschiedliche Kriterien verwendet werden; häufig wählt man dasjenige Neuron, dessen Gewichte insgesamt dem Eingabesignal am ähnlichsten sind [Gl. (m)].

$$out_c \leftarrow \min \left\{ \sum_{i=1}^m (x_i - w_{ji})^2 \right\} \quad (m)$$

Für dieses zentrale Neuron werden die Gewichte  $w_{je}$  so korrigiert, daß sein Ausgabewert dem Eingabesignal noch ähnlicher wird. Auch die Gewichte der anderen Neuronen werden korrigiert, allerdings um so weniger, je weiter sie von dem am stärksten erregten (zentralen) Neuron entfernt sind. Hierbei kommt es nur auf die topologische Distanz an, d. h. auf die Nachbarschaftssphäre (vgl. Abb. 23), in der sich das betrachtete Neuron bezüglich des zentralen Neurons befindet.

Anschließend wird der Prozeß mit den nächsten Eingabedaten wiederholt. Jedes Objekt, also jeder Punkt aus dem  $m$ -dimensionalen Raum, stimuliert ein ganz bestimmtes

Neuron, so daß jedes Objekt einem definierten Punkt im Kohonen-Netz zugeordnet wird.

Ein einfaches Beispiel, die Abbildung einer Kugeloberfläche auf ein Kohonen-Netz, soll die Arbeitsweise und die Ergebnisse eines Kohonen-Netzes näher erläutern (Abb. 26). Die Kugeloberfläche wurde dazu in acht Kugeldreiecke eingeteilt; ein Punkt auf der Oberfläche ist durch seine drei Koordinaten ( $x$ -,  $y$ -,  $z$ -Wert) charakterisiert. Als Kohonen-Netz wurde eine Anordnung von  $15 \times 15 = 225$  Neuronen festgelegt. Das heißt, drei Eingabeeinheiten geben ihre Daten auf 225 Neuronen, so daß  $3 \times 225 = 675$  Gewichte bestimmt werden müssen.

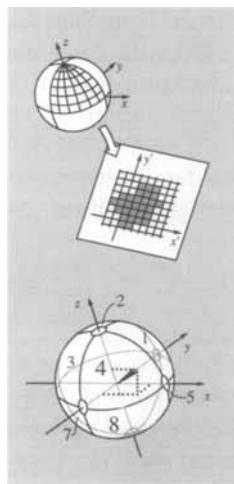


Abb. 26. Abbildung einer Kugeloberfläche auf ein Kohonen-Netz. Die Kugeloberfläche wurde in acht Kugeldreiecke eingeteilt; jeder Punkt auf der Kugeloberfläche wird durch seine Zugehörigkeit zu einem dieser Kugeldreiecke charakterisiert. Der in der unteren Kugeldarstellung markierte Punkt gehört beispielsweise zum Dreieck mit der Nummer 4.

Auf der Kugeloberfläche wurden willkürlich 2000 Punkte ausgewählt und deren  $x$ -,  $y$ - und  $z$ -Koordinaten der Reihe nach zum Training des Kohonen-Netzes verwendet. Zur grafischen Darstellung des Netzes, das sich nach dem Lernen dieser 2000 Punkte herausgebildet hatte, wurden die Punkte mit der Nummer des Kugeldreiecks, aus dem sie stammen, identifiziert (diese Information wurde aber im Lernverfahren nicht verwendet, sondern diente nur am Schluß zur Kennzeichnung der Punkte!). Da 2000 Punkte auf 225 Felder abgebildet werden, muß natürlich ein Feld jeweils mehrere Punkte repräsentieren. Aber es ergibt sich, daß am Ende des Lernens in ein Feld immer Punkte aus dem gleichen Kugeldreieck gelangen und daß in benachbarte Felder Punkte aus benachbarten Regionen der Kugeloberfläche projiziert werden.

Abbildung 27 zeigt das resultierende Kohonen-Netz. Felder mit gleichen Ziffern, also mit Punkten aus dem gleichen

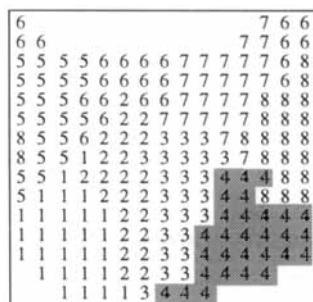


Abb. 27. Ergebnis der Abbildung einer Kugeloberfläche auf ein Kohonen-Netz.

Kugeldreieck, bilden zusammenhängende Flächen im Kohonen-Netz. Dabei ist natürlich zu beachten, daß die dargestellte Fläche eigentlich einen Torus bildet (Abb. 24), daß also Punkte am linken Rand ihre Fortsetzung am rechten Rand haben (Abb. 25). So haben die beiden Felder am linken Rand mit der Ziffer 8 tatsächlich direkten Anschluß an die übrigen Felder, die mit der Ziffer 8 markiert sind. Einigen Feldern (Neuronen) des Kohonen-Netzes wurde kein einziger Punkt der Kugeloberfläche zugeordnet; diese Felder sind als weiße Fläche dargestellt. Daß die Kugeloberfläche unter Erhalt der Nachbarschaftsbeziehungen der Punkte auf der Kugel auf das Kohonen-Netz abgebildet wurde, läßt sich auch aus den in Abbildung 28 dargestellten Details des Kohonen-Netzes ablesen. Dreiecke auf der Kugel, die an Längengraden oder am Äquator zusammenstoßen, sind in der Projektion auf das Kohonen-Netz auch benachbart und haben ganze Linienzüge gemeinsam. In mehreren Regionen stoßen vier Felder zusammen (in Abb. 28 eingekreist); diese Regionen oder Punkte entsprechen den Durchstoßpunkten der Koordinatenachsen durch die Kugeloberfläche. So stoßen z.B. in Abbildung 28 unten Mitte die Felder 1, 2, 3 und 4 zusammen; diese Region entspricht dem „Nordpol“ der Kugel von Abbildung 26.

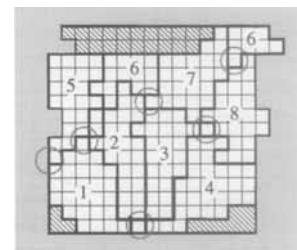


Abb. 28. Markierung der Bereiche in der Kohonen-Karte von Abbildung 27, in denen vier Kugeldreiecke zusammenstoßen, durch gestrichelte Kreise.

Die Abbildung einer Kugel auf ein Kohonen-Netz wurde so ausführlich dargelegt, um zu zeigen, wie ein einfacher dreidimensionaler Körper auf eine Ebene abgebildet wird. Damit wird die Arbeitsweise eines Kohonen-Netzes als Topologie-erhaltende Abbildung gut illustriert und die Grundlage für das Verständnis der Anwendung des Kohonen-Netzes auf Beispiele aus der Chemie geschaffen.

#### 4.4. „Backpropagation“

Die Mehrzahl der Anwendungen neuronaler Netze verwendet den „Backpropagation“-Algorithmus. Dieser Algorithmus repräsentiert keine spezielle Netzwerkarchitektur – meist wird ein Mehrschichtennetz eingesetzt –, sondern ist ein besonderes Lernverfahren. Obwohl diese Methode erst 1986 durch Rumelhart, Hinton und Williams<sup>[17]</sup> eingeführt wurde, gewann sie sofort große Popularität und trug entscheidend zum Siegeszug der neuronalen Netze bei. So ergab sich bei einer vor gut zwei Jahren durchgeföhrten Analyse aller Veröffentlichungen über den Einsatz neuronaler Netze in der Chemie, daß der Backpropagation-Algorithmus bei 90 % der Arbeiten verwendet wurde<sup>[22]</sup>.

Die Attraktivität des Lernens durch Backpropagation kommt dadurch zustande, daß die Korrekturen der Gewichte des neuronalen Netzes mit wohldefinierten Gleichungen berechnet werden können. Allerdings existiert zwischen die-

ser Prozedur zur Fehlerkorrektur und den Prozessen, die bei der Änderung von Synapsenstärken im biologischen System ablaufen, kaum eine Analogie.

Der Backpropagation-Algorithmus wird bei Ein- und Mehrschichtennetzen angewendet und ist ein Lernverfahren mit Unterweisung (supervised learning). Die Eingabedaten werden über die Schichten hinweggeschickt; die Ausgabedaten einer Schicht  $l$ ,  $\text{Out}^l$ , sind die Eingabedaten  $X^{l+1}$  der Schicht  $l+1$ . Aus der letzten Schicht sollten eigentlich die für die Eingabedaten zu erwartenden Ergebnisse erhalten werden. Zunächst wird dies noch nicht der Fall sein. Die Ausgabedaten  $\text{Out}^{\text{last}}$  der letzten Schicht werden deshalb mit den Erwartungswerten  $Y$  verglichen und daraus ein Fehler ermittelt. Dieser Fehler wird nun verwendet, um die Gewichte der Ausgabeschicht zu korrigieren. Dann werden die Gewichte der vorletzten Schicht unter Berücksichtigung des Fehlers in der letzten Schicht korrigiert, und so werden die Fehler schichtweise von unten nach oben durchgeschleust und zur Berechnung der Korrekturen der Gewichte herangezogen (Abb. 29). Der Fehler wandert also gegenläufig zu den Eingabedaten, daher der Name Backpropagation oder, wie bereits salopp formuliert, „Fehlerkorrektur im Rückwärtsgang“<sup>[25]</sup>. Im folgenden werden nur die Grundzüge des Backpropagation-Algorithmus aufgezeigt; für eine ausführliche Ableitung muß auf die Literatur verwiesen werden<sup>[1, 17, 18, 25]</sup>.

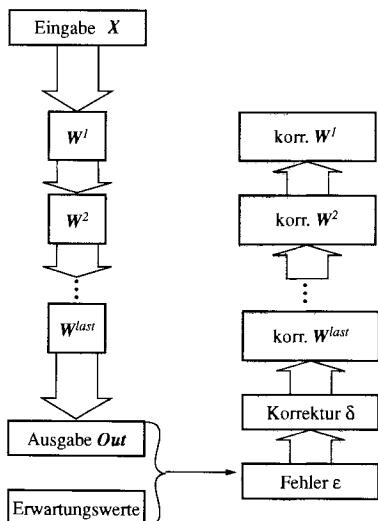


Abb. 29. Das Lernverfahren des Backpropagation-Algorithmus. Die Gewichte werden durch Rückführung des Fehlers korrigiert.

Der Backpropagation-Algorithmus soll die Gewichte so verändern, daß der Fehler für die Ausgabewerte  $\text{Out}$  minimal wird, d. h. daß diese möglichst gut mit den vorgegebenen Werten  $Y$  übereinstimmen.

In der letzten Schicht kann der Fehler direkt bestimmt werden, denn der Wert  $Y$ , der in der Ausgabeschicht erwartet wird, ist ja bekannt. Die Gewichtsänderungen der letzten Schicht,  $\Delta w_{ji}^{\text{last}}$ , werden durch Ableiten des Fehlers  $\varepsilon$  [Gl. (n)] nach den einzelnen Gewichten bestimmt. Dies führt aufgrund der Kettenregel zu den Gleichungen (o) und (p).  $\eta$  ist dabei ein Parameter, die *Lernrate*, dessen Wert vorgegeben werden kann. Er bestimmt, wie rasch ein neuronales Netz lernt. Gewöhnlich wird für ihn ein Wert zwischen 0.1 und 0.9 gewählt.

$$\varepsilon = \sum_{j=1}^m (y_j - \text{out}_j^{\text{last}})^2 \quad (\text{n})$$

$$\delta_j^{\text{last}} = -\partial \varepsilon / \partial \text{Net}_j^{\text{last}} \quad (\text{o})$$

$$\Delta w_{ji}^{\text{last}} = \eta \delta_j^{\text{last}} \text{out}_j^{\text{last}} - 1 \quad (\text{p})$$

In den Zwischenschichten ist der Fehler dagegen nicht direkt bekannt, denn man weiß ja gar nicht, welche Ausgabewerte  $\text{Out}^l$  aus den Zwischenschichten kommen sollen. Hier wird nun angenommen, daß sich der Fehler aus der darunterliegenden Schicht gleichmäßig über alle Verbindungen zur darüberliegenden Schicht verteilt hat. Damit kann der Fehler in einer Schicht jeweils aus dem Fehler der darunterliegenden Schicht berechnet werden. Dies ist die Grundlage des Backpropagation-Algorithmus: Der Fehler wird durch die einzelnen Schichten zurückgetragen („backpropagation of errors“), und damit werden die Gewichte der einzelnen Schichten korrigiert.

Insgesamt ergibt sich für die Korrektur der Gewichte einer Schicht die geschlossene Form (q). Häufig werden bei der

$$\Delta w_{ji}^l = \eta \sum_{k=1}^r \delta_{jk}^{l+1} w_{jk}^{l+1} f'(\text{Net}_j^l) \text{out}_i^{l+1} \quad (\text{q})$$

Korrektur der Gewichte auch noch die Änderungen der Gewichte im vorhergehenden Cyclus (previous) mitberücksichtigt. Dies geschieht dadurch, daß die Gleichung (q) noch um den additiven Beitrag  $\mu \Delta w_{ji}^{l(\text{previous})}$  erweitert wird. Der Parameter  $\mu$  ist das *Vergessensmoment*. Er bestimmt, wie stark vorhergehende Gewichteänderungen mitberücksichtigt werden sollen, d. h. er verleiht dem Lernvorgang ein gewisses Beharrungsvermögen. Je kleiner  $\mu$  ist, um so rascher werden vorhergehende Änderungen der Gewichte wieder vergessen. Man kann zeigen, daß die Summe von  $\eta$  und  $\mu$  ungefähr bei 1 liegen soll<sup>[26]</sup>.

## 5. Andere Modelle neuronaler Netze

Neben den in Abschnitt 4 vorgestellten Verfahren zur Datenanalyse durch neuronale Netze gibt es noch eine Reihe anderer, von denen hier nur zwei noch kurz erwähnt werden sollen.

Das *Counterpropagation-Netz* besteht aus einer Kohärenzschicht, die mit einem Speicher kombiniert ist, um Antworten zu geben<sup>[27]</sup>.

Der *verteilte assoziative Speicher* (*Associative Memory System* = AMS) wurde als Modell für die Informationsverarbeitung im Kleinhirn entwickelt, das vor allem für die Verarbeitung motorischer Reize zuständig ist<sup>[10]</sup>. Dieses AMS-Modell wurde unter anderem zur Vorhersage der chemischen Reaktivität, d. h. der Brechbarkeit von Bindungen, eingesetzt<sup>[28]</sup>.

## 6. Überblick über Anwendungen neuronaler Netze in der Chemie

Bei einer Vielfalt von Problemen aus den verschiedensten Bereichen der Chemie wurde bereits versucht, Lösungen un-

ter Verwendung neuronaler Netze zu finden: Das geht von so umfassenden Themen wie Struktur-Spektren-Beziehungen, chemische Reaktivität, Sekundär- und Tertiärstruktur von Proteinen sowie Prozeßkontrolle bis hin zu ganz speziellen Einzelfragen wie der Klassifizierung der Energieniveaus des Curiumatoms, dem Erkennen und Klassifizieren von Aerosolteilchenverteilungen sowie dem Zusammenhang zwischen der physikalischen Struktur und den mechanischen Eigenschaften von Polyethylenterephthalatgarnen. Diese Vielfalt unterstreicht, daß neuronale Netze ganz allgemeine, breit einsetzbare Methoden zur Lösung von Problemen darstellen. Die konkrete Anwendung wird erst durch die Art der Daten, die in ein neuronales Netz eingegeben werden, bestimmt.

Dieser Artikel will ein Gefühl dafür vermitteln, welche Art von Problemen mit neuronalen Netzen bearbeitet werden kann, und damit den Leser in die Lage versetzen zu entscheiden, ob er diese Methoden für seine Aufgabenstellungen verwenden kann. Ziel ist also, die Leistungsfähigkeiten der Methoden aufzuzeigen und Anreize zu weiteren Anwendungen zu geben, und nicht so sehr, einen umfassenden Überblick über die bisherigen Arbeiten zu vermitteln.

Die im folgenden präsentierte Auswahl typischer Anwendungen neuronaler Netze in der Chemie ist nicht nach Anwendungsbereichen, sondern nach Problemtypen gegliedert, d.h. die Arbeiten sind danach eingeteilt, ob es sich um Aufgaben zur Klassifikation, Modellierung, Assoziation oder Abbildung handelt.

Es gibt natürlich eine Reihe von Methoden, die alternativ zu neuronalen Netzen eingesetzt werden können – und seit vielen Jahren auch erfolgreich verwendet werden. Viele Aufgaben, für die heute neuronale Netze herangezogen werden, können auch mit statistischen und Mustererkennungs(Pattern recognition)-Methoden wie Regressionsanalyse, Clustermethoden und Hauptkomponentenanalyse gelöst werden. In vielen Fällen dürften diese Methoden zu ebenso guten Ergebnissen führen, und man vermißt leider in den meisten Arbeiten über neuronale Netze eine Gegenüberstellung der Leistungsfähigkeit dieser Netze und der etablierter Methoden. Solche Vergleiche könnten die spezifischen Vorteile neuronaler Netze stärker herausarbeiten. Dazu gehören:

- die mathematische Form der Beziehung zwischen den Eingabe- und Ausgabedaten muß nicht vorgegeben werden
- neuronale Netze können auch nichtlineare Beziehungen aufstellen

In jedem Fall sollte man beim Einsatz neuronaler Netze die gleiche Sorgfalt bei der Problemformulierung, der Repräsentation der Information, der Auswahl der Daten und der Aufteilung der Daten in Trainings- und Testdatensatz verwenden, wie sie bei statistischen und Mustererkennungsmethoden notwendig ist. Es muß ganz deutlich gesagt werden, daß die Qualität der Ergebnisse mit neuronalen Netzen entscheidend davon abhängt, welche Mühe und Arbeit man in diese Teilspekte steckt.

Die wesentlichen Gesichtspunkte, die man vor dem Einsatz eines bestimmten neuronalen Netzes erwägen soll, sind:

- Welcher Typ einer Aufgabenstellung liegt vor? Klassifikation, Modellierung, Assoziation oder Abbildung?
- Welches Lernverfahren kann gewählt werden? Mit oder ohne Unterweisung? Sind also für einen Satz an Objekten

die zu erwartenden Ergebnisse bekannt, oder soll die Struktur einer Information erst gefunden werden?

In Tabelle 1 sind die Netzvarianten mit den Problemtypen und Lernverfahren verknüpft, damit leichter entschieden werden kann, welches neuronale Netz für eine konkrete Problemlösung eingesetzt werden soll.

Tabelle 1. Verwendungsmöglichkeiten neuronaler Netze.

	Hopfield-Netz	ABAM	Kohonen-Netz	Backpropagation
Klassifikation		x	x	x
Modellierung				x
Assoziation	x	x		x
Abbildung			x	
Lernverfahren [a]	ohne	ohne + mit	ohne	mit

[a] ohne = ohne Unterweisung, mit = mit Unterweisung (siehe Abschnitt 4).

In der Chemie stellen sich meistens Klassifikations- und Modellierungsaufgaben. Dies ist einer der Gründe, weshalb Mehrschichtennetze, die mit dem Backpropagation-Algorithmus trainiert werden, dominieren. Eine Zusammenstellung der bis Ende 1990 erschienenen Anwendungen neuronaler Netze in der Chemie ergab, daß über 90 % der Arbeiten den Backpropagation-Algorithmus einsetzen<sup>[2]</sup>. Ein Hopfield-Netz wurde nur zweimal, ein adaptiver bidirektionaler assoziativer Speicher nur einmal und ein Kohonen-Netz, gekoppelt mit dem Counterpropagation-Lernverfahren, ebenfalls nur einmal verwendet. Diese Verteilung sollte aber für die Zukunft nicht unbedingt so bleiben. ABAM und Kohonen-Netz sowie der Counterpropagation-Algorithmus bieten Möglichkeiten, die bei weitem noch nicht ausgeschöpft wurden.

Bleibt man beim Mehrschichtenmodell mit dem Backpropagation-Algorithmus, so findet man ein breites Spektrum in der Komplexität der eingesetzten Netze: von Netzen mit 20 Gewichten bis zu solchen mit 40 000, in einem Fall sogar mit 500 000 Gewichten! Die Zahl der zum Training eines Netzes verwendeten Daten sollte beim Backpropagation-Algorithmus mindestens so groß sein wie die Zahl der Gewichte. Diese Regel wurde bei weitem nicht immer eingehalten. Auch die Zahl der Iterationen, mit der der gesamte Datensatz zum Training über das Netz geschickt wurde (Zahl der Epochen), schwankt beachtlich: von 20 bis zu 100 000 Iterationen. Es läßt sich denken, daß die Trainingszeiten mit der Zahl der Gewichte und der Zahl der Epochen stark zunehmen: von wenigen Minuten auf einem Personal Computer bis zu Stunden auf einem Cray-Supercomputer.

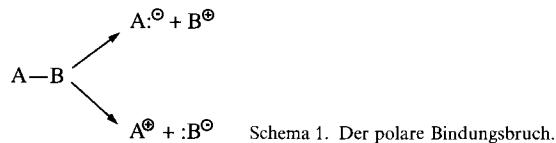
## 7. Klassifikation

Klassifikationsprobleme sind eines der verbreitetsten Einsatzgebiete neuronaler Netze. Ein Objekt wird durch Meßdaten charakterisiert und aufgrund dessen einer bestimmten Kategorie zugeordnet, oder es wird festgestellt, daß es eine bestimmte Eigenschaft nicht hat oder nicht zu der betrachteten Klasse gehört. Die Ausgabedaten sind also binärer Natur – eine Eigenschaft ist vorhanden oder nicht, das Objekt gehört einer bestimmten Klasse an oder nicht. Die Eingabeda-

ten, die das Objekt charakterisieren, können binärer Natur oder auch reelle Werte (Meßdaten) sein. Klassifikation ist ein traditionelles Gebiet für den Einsatz von statistischen oder Mustererkennungsmethoden. Neuronale Netze bieten den Vorteil, daß sie auch bei sehr komplizierten Zusammenhängen zwischen den Objektdaten und der gesuchten Klassenzugehörigkeit noch eingesetzt werden können. Auch für Zusammenhänge, die sich nicht oder nur schwer in explizite Gleichungen fassen lassen oder die eindeutig nichtlinear sind, sind neuronale Netze noch geeignet. Im folgenden wollen wir Beispiele aus mehreren Bereichen der Chemie geben, wobei entweder eine einzige Eigenschaft zu diagnostizieren (Abschnitt 7.1), eine Kategorie aus einer ganzen Reihe von Klassen auszuwählen (Abschnitt 7.5) oder ein Objekt gleichzeitig mehreren Klassen zuzuordnen ist (Abschnitte 7.2–7.4).

## 7.1. Chemische Reaktivität

Der Chemiker leitet seine Erkenntnisse über die Reaktivität von Bindungen und funktionellen Gruppen aus einer Vielzahl von Beobachtungen an einzelnen Reaktionen ab. Wie kann nun dieser Prozeß auf ein neuronales Netz übertragen werden? Behandeln wir das Problem des polaren Bindungsbruchs (Schema 1), den einleitenden Schritt vieler organischer Reaktionen<sup>[29]</sup>.



Die Reaktivität soll hier sehr grob nur darüber charakterisiert werden, ob eine Bindung leicht oder schwer heterolytisch gespalten wird. Dazu genügt ein einziges Neuron als Ausgang des neuronalen Netzes, das auf Eins gesetzt wird, wenn die Bindung leicht gespalten werden kann, und den Wert Null erhält, wenn der Bindungsbruch schwer zu realisieren ist. Nun muß noch der polare Bindungsbruch durch Faktoren charakterisiert werden, die diesen Vorgang beeinflussen. Dazu wurde eine Reihe energetischer und elektronischer Effekte verwendet: Bindungsdissoziationsenergie BDE, Differenz der Gesamtladung  $\Delta q_{tot}$ , Differenz der  $\pi$ -Ladung  $\Delta q_\pi$ , Differenz der  $\sigma$ -Elektronegativität  $\Delta \chi_\sigma$ ,  $\sigma$ -Polarität  $Q_\sigma$ , Bindungspolarisierbarkeit  $\alpha_b$  und Ausmaß der Resonanzstabilisierung  $R^\pm$  der Ladungen, die beim polaren Bindungsbruch entstehen. Werte für diese Größen wurden mit empirischen Verfahren berechnet<sup>[30–34]</sup>. Für diese sieben Parameter benötigt man sieben Einheiten, in die die (reellen) Werte der einzelnen Größen eingegeben werden. Eine Zwischenschicht mit drei Neuronen vervollständigt die Netzwerkarchitektur dieser Studie (Abb. 30).

Ein Datensatz aus 29 aliphatischen Verbindungen, die 385 Bindungen enthielten, wurden erstellt. Da jede Bindung in zwei Richtungen heterolytisch gespalten werden kann (siehe Schema 1), sind insgesamt 770 polare Bindungsbrüche möglich. Aus diesen wurden 149 Heterolyse von Einfachbindungen ausgewählt und davon 64 zum Training des Netzes

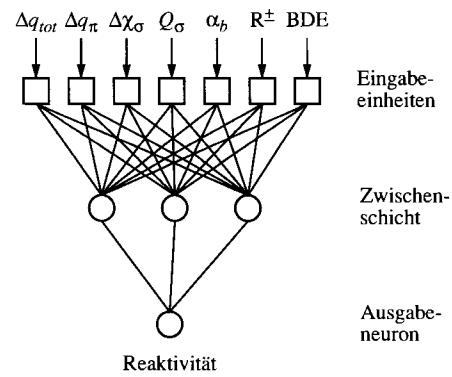


Abb. 30. Architektur und Eingabeparameter für ein neuronales Netz, das die Brechbarkeit von Bindungen vorhersagen soll. Einzelheiten siehe Text.

mit dem Backpropagation-Algorithmus verwendet; mit den restlichen 85 Bindungsbrüchen wurde das Netz getestet (die Einteilung in Trainings- und Testdatensatz wird in Abschnitt 10.1 erläutert). Abbildung 31 zeigt eine Auswahl an Molekülen aus dem Datensatz, in die die als brechbar und die als nicht brechbar klassifizierten Bindungen eingezzeichnet sind.

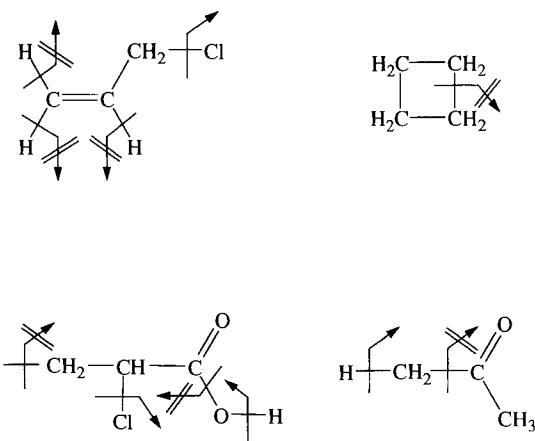


Abb. 31. Auswahl an Strukturen aus dem Trainingssatz mit der Angabe, welche Bindungen leicht (Pfeile) und welche schwer (durchgestrichene Pfeile) heterolytisch gespalten werden können. Die Pfeilrichtung gibt an, zu welchem Atom das Elektronenpaar der Bindung beim Bindungsbruch verschoben wird, welches Atom also die negative Ladung erhält.

Nach 1300 Cyclen (Epochen) hatte das Netz alle 64 Bindungsbrüche des Trainingsdatensatzes korrekt gelernt. Nun wurden die Bindungsbrüche aus dem Testdatensatz über das fertig trainierte neuronale Netz geschickt. Diese 85 Bindungsbrüche, über die das Netz also vorher noch keine Information erhalten hatte, wurden ebenfalls als korrekt klassifiziert. Die Einteilung der Bindungen in leicht und schwer brechbare wurde vom Netz genau so vorhergesagt, wie sie vom Chemiker festgelegt worden war. Das Netz hatte also die Zusammenhänge zwischen den elektronischen und energetischen Variablen und dem polaren Bindungsbruch gelernt.

Damit konnte das Netz nun auch auf Verbindungen angewendet werden, die weder im Trainings- noch im Testdatensatz enthalten sind. Und es machte korrekte Vorhersagen auch bei Bindungstypen, die Atome enthielten, die gar nicht zum Training verwendet wurden. In Abbildung 32 werden die vorhergesagten reaktiven Bindungen einer Struktur ge-

zeigt, die nicht zum Training des Netzes verwendet wurde. Als besonders reaktiv ergaben sich die allylständigen C-Br- und C-S-Bindungen sowie die C-H-Bindung des zentralen Allyl-C-Atoms und die S-H-Bindung. Die allylischen Positionen an den Enden des Systems wurden als weniger acid eingeschätzt, wobei die Position, bei der der Bromsubstuent induktiv stabilisierend wirken kann, noch die höhere Acidität erhielt. Alle diese Ergebnisse stimmen mit der chemischen Erfahrung überein.

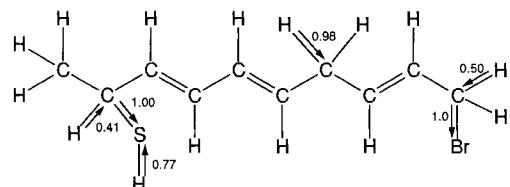


Abb. 32. Durch das für polare Bindungsbrüche trainierte neuronale Netz vorhergesagte Bindungsbrüche in einer Struktur, die nicht im Lernverfahren verwendet wurde. Die Pfeilrichtungen geben die Verschiebung der Elektronenpaare an, die Zahlenwerte die vorhergesagte Wahrscheinlichkeit des Bindungsbruchs.

Bemerkenswert ist dabei noch, daß auch die Reaktivität der SH-Gruppe richtig beurteilt wird, obwohl im Trainingsdatensatz keine einzige Struktur mit einem Schwefelatom enthalten war. Dies ermöglichen die elektronischen und energetischen Parameter, die den Einfluß eines Atoms in allgemeiner Form in den zur Berechnung eingesetzten Methoden enthalten, so daß ein Atomtyp berücksichtigt werden kann, sofern er in diesem Rechenverfahren enthalten ist, auch wenn er im aktuellen Trainingssatz nicht vorkommt.

Dieses so entwickelte neuronale Netz kann für ein breites Spektrum aliphatischer Strukturen vorhersagen, welche Bindungen leicht und welche schwer polar gebrochen werden können.

## 7.2. Prozeßkontrolle

Bei vielen chemischen Prozessen lassen sich die Beziehungen zwischen Prozeßdaten und Steuerparametern nur – wenn überhaupt – durch nichtlineare Gleichungen wiedergeben. Diese Prozesse sind deshalb schwierig zu modellieren und in ihrem Verlauf vorherzusagen. Es überrascht daher nicht, daß neuronale Netze intensiv für Aufgaben der Prozeßkontrolle eingesetzt werden<sup>[35–41]</sup>. Dabei versucht man sowohl bestimmte Ereignisse zu klassifizieren (Ja-nein-Entscheidungen) als auch Steuerparameter zu modellieren (Vorhersage eines reellen Wertes).

Ein Beispiel, bei dem zwischen verschiedenen Klassenzugehörigkeiten auszuwählen ist, soll die Einsatzmöglichkeiten illustrieren<sup>[36]</sup>. Für einen Reaktortank, kontinuierlich gerührt, in dem eine exotherme Reaktion ablief, sollten Fehlfunktionen aus sechs verschiedenen Meßdaten abgeleitet werden.

Folgende Größen wurden gemessen (vgl. Abb. 33): (1) Die Auslaßkonzentration des Edukts,  $C_e$ , (2) die Reaktortemperatur  $T_r$ , (3) das Reaktorfüllvolumen  $V_r$ , (4) die Auslaßfließgeschwindigkeit  $FR_p$ , (5) die Kühlwassertemperatur  $T_c$  und (6) die Kühlwasserfließgeschwindigkeit  $FR_c$ .

Anhand dieser Daten sollten mehrere Fehlverhalten des Reaktors diagnostiziert werden. Fehlfunktionen können durch die Einlaßkonzentration des Edukts,  $C_{eo}$ , die Einlaßtemperatur  $T_e$  und die Einlaßfließgeschwindigkeit  $FR_e$  verursacht werden. Weichen diese Größen um mehr als 5% vom Normalwert nach oben oder unten ab, liegt eine Fehlfunktion des Reaktors vor.

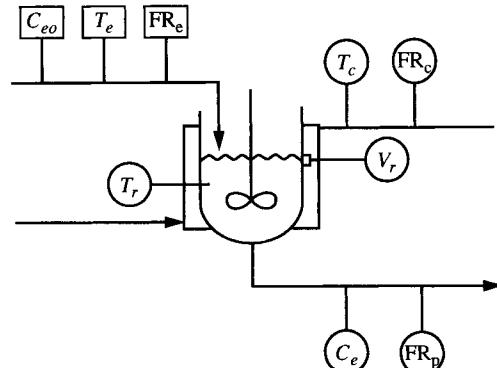


Abb. 33. Modell des Reaktortanks mit Angabe der sechs Meßgrößen  $C_e$ ,  $T_r$ ,  $V_r$ ,  $FR_p$ ,  $T_c$  und  $FR_c$ , sowie der Zustandsgrößen, die eine Fehlfunktion verursachen ( $C_{eo}$ ,  $T_e$  und  $FR_e$ ).

Jede der Fehlfunktionen beeinflußt fast alle „Symptome“, also alle sechs Meßwerte, so daß eine bestimmte Fehlfunktion nicht direkt aus einer einzigen Messung abgeleitet werden kann. Darüber hinaus können sich auch mehrere Fehlfunktionen in den Meßdaten gegenseitig kompensieren oder synergistisch verstärken. Außerdem können die Fehlfunktionen nicht nur einzeln, sondern auch zu mehreren gleichzeitig auftreten.

Welche Netzwerkarchitektur wurde nun gewählt? Einzugeben sind sechs – reelle – Meßdaten, also werden sechs Eingabeeinheiten benötigt. Ausgabeneuronen wurden ebenfalls sechs gewählt, und zwar für jede der drei entscheidenden Einlaßparameter  $C_{eo}$ ,  $T_e$  und  $FR_e$  jeweils ein Neuron für eine Abweichung vom Normalwert nach oben und ein Neuron für eine Abweichung nach unten. Das jeweilige Ausgabeneuron sollte also aktiviert werden, wenn die entsprechende Fehlfunktion eintritt. Fünf Neuronen in der Zwischenschicht vervollständigten das Mehrschichten-Netz (Abb. 34).

In diesem neuronalen Netz waren demnach  $6 \times 5 + 5 \times 6 = 60$  Gewichte zu bestimmen. Zwölf einzelne

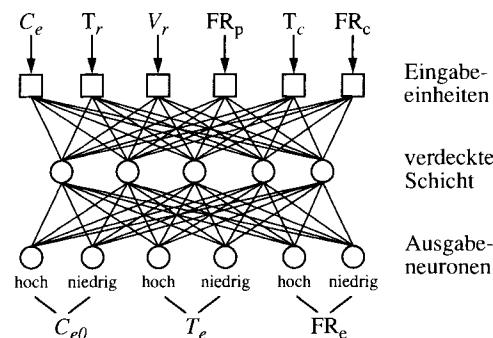


Abb. 34. Neuronales Netz zur Diagnostik von Fehlfunktionen des chemischen Reaktors der Abbildung 33.

Fehlfunktionen wurden absichtlich erzeugt und mit den dabei gemessenen Daten das neuronale Netz unter Verwendung des Backpropagation-Algorithmus trainiert. Das so trainierte Netz konnte die Reaktordaten bei Normalfunktion, die nicht im Lernprozeß verwendet wurden, als ungestörtes Verhalten identifizieren. Außerdem wurden vier Mehrfachfehlfunktionen eingestellt; das neuronale Netz konnte diese ebenfalls korrekt aus den dabei gemessenen Daten ableiten.

Neuronale Netze werden sicherlich in der Prozeßkontrolle große Bedeutung erlangen. Man könnte auch das auf einen bestimmten Prozeß eintrainierte neuronale Netz auf einem Chip fest einprogrammieren und diesen Chip in den Kontrollprozeß einbauen.

### 7.3. Zusammenhang zwischen Struktur und IR-Spektrum

In den Beispielen der Abschnitte 7.1 und 7.2 wurden noch recht einfache neuronale Netze mit relativ wenigen Gewichten verwendet. Bei der nun folgenden Anwendung wurde ein erheblich größeres Netz mit fast 10000 Gewichten entworfen.

Die moderne Strukturaufklärung basiert auf spektroskopischen Methoden. Da die Beziehungen zwischen der Struktur und den spektroskopischen Daten einer organischen Verbindung aber zu komplex sind, um sie in einfache Gleichungen fassen zu können, gibt es eine Vielzahl empirischer Regeln. Da eine Unmenge an spektroskopischen Daten zur Verfügung steht, ist eine ideale Voraussetzung für das Training neuronaler Netze gegeben. Die ersten Schritte wurden bereits unternommen, um die Zusammenhänge zwischen Struktur und spektroskopischen Daten in neuronale Netze einzuspeichern; wir werden aber sehen, daß auf diesem Gebiet noch viel Entwicklungsarbeit zu leisten ist.

Munk et al.<sup>[42]</sup> gingen der Frage nach, inwieweit mit einem neuronalen Netz aufgrund eines IR-Spektrums Aussagen über die in einer Verbindung enthaltenen Teilstrukturen gemacht werden können. Der Bereich eines IR-Spektrums von  $400\text{--}3960\text{ cm}^{-1}$  wurde in 256 Intervalle eingeteilt und jedes dieser Intervalle einem Eingabeelement zugeordnet. Befand sich in diesem Bereich eine Bande, so wurde deren Intensität in das Eingabeelement eingegeben. Das neuronale Netz hatte 36 Ausgabeneuronen, die für 36 verschiedene funktionelle Einheiten (Substrukturen, z.B. primärer Alkohol, Phenol, tertiäres Amin, Ester) zuständig waren. War eine Substruktur in der untersuchten Verbindung enthalten, so erhielt das entsprechende Neuron den Wert Eins, andernfalls den Wert Null. Weiterhin wurde eine Zwischenschicht mit 34 verdeckten Neuronen verwendet, so daß in diesem Mehrschichtenetz  $256 \times 34 + 34 \times 36 = 9928$  Gewichte festzulegen waren. Die prinzipielle Vorgehensweise und die Architektur des neuronalen Netzes ist in Abbildung 35 skizziert.

Um die Gewichte des neuronalen Netzes zu bestimmen, wurde das Netz mit 2499 IR-Spektren und den zugehörigen, in die Substrukturen zerlegten Strukturen nach dem Backpropagation-Algorithmus trainiert. 416 IR-Spektren wurden zum Test der Vorhersagefähigkeit des Netzes verwendet. Ein einziger Cyclus durch alle Spektren erforderte 10 min CPU-Zeit auf einer VAX 3500; für ein Training mit vielen Cyclen (Epochen), typischerweise waren 100 Epochen nötig, wurde ein Cray-Supercomputer verwendet.

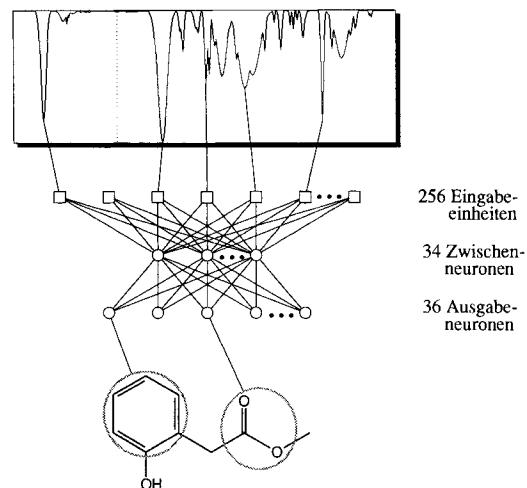


Abb. 35. Neuronales Netz, das die Zusammenhänge zwischen dem IR-Spektrum einer Verbindung und den darin enthaltenen Substrukturen lernen soll.

Für jede Substruktur wurde die Qualität der Ergebnisse durch eine Zahl, den A50-Wert, beschrieben. Dieser Wert gibt die Genauigkeit bei 50% Informationsrückgabe an, d.h. mit welcher Genauigkeit eine Substruktur bestimmt werden kann, wenn der Schwellenwert auf den Mittelwert der Verteilungskurve gesetzt wird.

Als typisches Ergebnis sind in Abbildung 36 die Resultate für primäre Alkohole angegeben. Der Schwellenwert lag bei einem Ausgabewert von 0.86. Bei diesem Wert werden von den 265 im Trainingsatz enthaltenen primären Alkoholen 132 korrekt identifiziert, aber es werden auch 34 Verbindungen fälschlicherweise als primäre Alkohole eingeordnet. Der A50-Wert für diese Gruppe ist also  $132/(132 + 34) = 79.5\%$ . Dieser Wert wurde noch als gut bezeichnet. Mit ähnlicher oder besserer Qualität konnten 30 der 36 Substrukturen bestimmt werden.

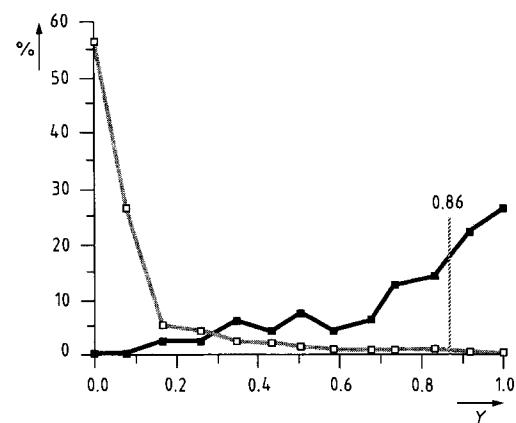


Abb. 36. Prozentuale Verteilung der Ausgabewerte  $Y$  des neuronalen Netzes für primäre Alkohole. Die durchgehogene Linie gilt für Verbindungen, die primäre Alkohole sind, die gepunktete Linie für alle anderen Verbindungen. Der Mittelwert der Ausgabewerte für primäre Alkohole lag bei 0.86.

Die Ergebnisse dieses Netzwerkes mit einer Zwischenschicht wurden auch mit der Klassifikationsfähigkeit eines Netzes ohne Zwischenschicht verglichen<sup>[43]</sup>. Es konnte dabei gezeigt werden, daß die Neuronenzwischenschicht zu einer wesentlichen Verbesserung der Ergebnisse beitrug.

Diese Untersuchung hat natürlich das Problem, die Zusammenhänge zwischen IR-Spektrum und Struktur aufzuklären, nicht gelöst. Im wesentlichen hat man sich auf einige wichtige Substrukturen konzentriert und das Gerüst außer acht gelassen. Dennoch sind die Vorhersagen nur mäßig; von 265 primären Alkoholen nur 132 zu erkennen und dazu noch 34 falsche Zuordnungen zu treffen ist enttäuschend. Setzt man den Schwellenwert höher, so kann mit großer Sicherheit gesagt werden, ob eine bestimmte Substruktur vorhanden oder nicht vorhanden ist, zugleich hat man aber einen weiten Bereich von Verbindungen, bei denen keine sicheren Vorhersagen möglich sind. In einem System zur automatischen Strukturaufklärung kann aber dank derartiger Vorhersagen der Suchraum bedeutend eingeschränkt werden. Hierin liegt der Wert dieser Ergebnisse.

Damit ist natürlich noch nicht das letzte Wort zum Zusammenhang zwischen Struktur und IR-Daten gesprochen. In weiteren Untersuchungen müßte vor allem versucht werden, die Gerüstschwingungen zuzuordnen, dazu müßten die Strukturen aber noch anders kodiert werden.

#### 7.4. Zusammenhang zwischen Struktur und Massenspektrum

Die Zusammenhänge zwischen Massenspektren und Strukturen sind noch komplizierter als die zwischen IR-Spektren und Strukturen. Dennoch wurde auch dieses Problem bereits mit neuronalen Netzen angegangen.

Auch hier wurde ein Mehrschichtennetz mit einer Zwischenschicht nach dem Backpropagation-Algorithmus trainiert<sup>[44]</sup>. Die Massenspektren wurden durch 493 Merkmale beschrieben; dazu gehörten die Logarithmen der Intensitäten der Peaks zwischen  $m/z$  40 und 219, die Logarithmen der Neutralabspaltungen zwischen  $\Delta(m/z)$  0 und 179, Autokorrelationssummen, Modulo-14-Werte, Seriennummern usw. Die Werte dieser 493 Spektrencharakteristika wurden auf ebenso viele Eingabeeinheiten gegeben.

Auch hier wurde die Struktur einer organischen Verbindung durch 36 Substrukturen – allerdings zum Teil andere als bei der Studie mit den IR-Spektren – charakterisiert; dazu waren 36 Ausgabeneuronen nötig. Die Zahl der Neuronen in der Zwischenschicht betrug 80, so daß  $493 \times 80 + 80 \times 36 = 42\,320$  Gewichte bestimmt werden mußten.

Dementsprechend wurden auch größere Datensätze untersucht: mit 31 926 Massenspektren zum Training und 12 671 Massenspektren zum Testen der Ergebnisse. Mit solch großen Datensätzen und einem Netzwerk mit so vielen Gewichten benötigt das Lernverfahren – auch hier wurde der Backpropagation-Algorithmus verwendet – natürlich schon erhebliche Zeit: Eine Epoche, also der Vorgang, alle 32 000 Spektren einmal über das Netz zu schicken, erforderte 6 h CPU-Zeit auf einer HP-9000/370- oder SUN-4-Workstation. Typischerweise waren 50 Epochen nötig, so daß das Training ungefähr zwei Wochen an reiner Rechenzeit auf einen leistungsfähigen Arbeitsplatzrechner benötigte.

Die Klassifikationsergebnisse mit dem fertig trainierten neuronalen Netz, MSnet, wurden mit Resultaten aus STIRS<sup>[45]</sup> verglichen. STIRS aus der Gruppe von McLafferty ist ein leistungsfähiges Expertensystem, um das Vorhandensein funktioneller Gruppen aus Massenspektren abzuleiten.

Die Klassifikationsergebnisse mit MSnet waren etwas besser als diejenigen mit STIRS. MSnet bietet aber noch einige zusätzliche Vorteile: (1) Für die Zugehörigkeit einer Verbindung zu einer bestimmten Klasse kann ein Wahrscheinlichkeitswert angegeben werden. (2) Nicht nur die Anwesenheit, sondern auch die Abwesenheit einer funktionellen Gruppe kann diagnostiziert werden. (3) Die Rechenzeiten bei der Abfrage sind bei MSnet um zwei Größenordnungen niedriger als bei STIRS.

Dies muß betont werden: Mag auch das Training eines neuronalen Netzes viel Rechenzeit erfordern, mit einem fertig trainierten Netz können Vorhersagen mit einem Minimum an Rechenzeit gemacht werden.

Solch ein globaler Anspruch, den Zusammenhang zwischen der molekularen Struktur und spektroskopischen Daten für den gesamten Bereich der Organischen Chemie darzustellen oder zu lernen, muß mit einem grundsätzlichen Problem der statistischen Verteilung von Daten kämpfen. Ein Beispiel: Der Datensatz aus 32 000 Verbindungen enthielt 33 Phthalsäureester, die einen sehr charakteristischen Peak bei  $m/z$  149 ergeben. Aber die meisten Spektren, die bei  $m/z$  149 einen Peak haben, stammen gerade nicht von Phthalsäureestern, da ja im Datensatz nur sehr wenige Phthalsäureester enthalten sind, und folglich werden Phthalsäureester nicht erkannt.

In der Arbeit<sup>[44]</sup> wird ein interessanter Versuch gemacht, dieses generelle Problem anzugehen. Es wird eine Hierarchie an neuronalen Netzen vorgeschlagen (Abb. 37). Während ein übergeordnetes Netz erst eine Einteilung nach den wichtigsten funktionellen Gruppen vornimmt, werden die einzelnen Verbindungsklassen dann durch spezialisierte neuronale Netze weiter verfeinert. So wurde ein spezielles Netz entwickelt, das Verbindungen, die die O-C=O-Gruppe enthalten, auf 22 Unterklassen (gesättigte Ester, aromatische Ester, Lactone, Anhydride etc.) aufteilt. Diese Idee einer Hierarchie neuronaler Netze könnte auch bei anderen Problemen gute Dienste leisten.

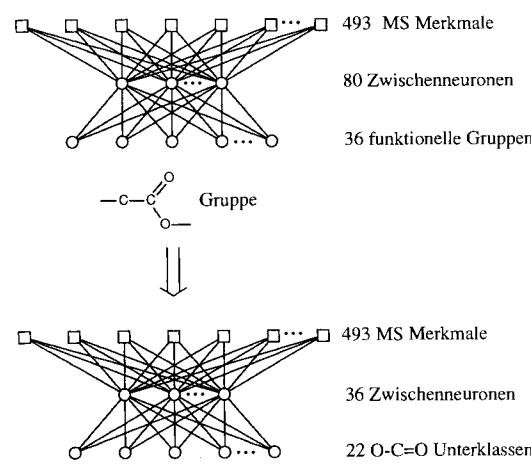


Abb. 37. Hierarchie neuronaler Netze zur Ableitung von Substrukturen aus Massenspektren.

#### 7.5. Sekundärstruktur von Proteinen

Hatte man es in dem Beispiel des vorhergehenden Abschnitts noch mit einer recht einfachen Netzwerkarchitektur

zu tun, so wird jetzt über Fälle mit einer ziemlich umfangreichen Codierung der Eingabedaten berichtet. Dementsprechend liegt ein recht komplexes neuronales Netz mit einer Vielzahl von Gewichten vor.

Um tieferen Einblick in die physiologischen Eigenschaften von Proteinen zu erhalten, muß man deren Sekundärstruktur kennen. Daher fehlt es nicht an Versuchen, die Sekundärstruktur von Proteinen aus deren Primärstruktur, also aus der Aminosäuresequenz, abzuleiten. Chou und Fasman<sup>[46]</sup> führten eine heute viel verwendete Methode ein, um aus der Aminosäuresequenz abzuleiten, welche Sekundärstruktur die Teile eines Proteins einnehmen. Dieses Verfahren kann für die einzelnen Aminosäuren eines Proteins mit einer Trefferquote von 50–53 % vorhersagen, ob sie an einer  $\alpha$ -Helix, einem  $\beta$ -Faltblatt oder einer unregelmäßigen Knäuelstruktur teilnehmen<sup>[47]</sup>. In den letzten Jahren erschienen nun in kurzer Folge einige Arbeiten<sup>[48–56]</sup> über die Vorhersage der Sekundär- oder sogar der Tertiärstruktur von Proteinabschnitten aus der Aminosäuresequenz mit Hilfe neuronaler Netze.

Hier soll anhand der Arbeit von Qian und Sejnowski<sup>[48]</sup> die prinzipielle Vorgehensweise aufgezeigt werden; die meisten anderen Untersuchungen<sup>[49–56]</sup> haben eine sehr ähnliche Strategie gewählt.

Sowohl der Methode von Chou und Fasman<sup>[46]</sup> als auch dem Einsatz neuronaler Netze liegt die Annahme zugrunde, daß die Aminosäure (AS) selbst und ihre unmittelbare Umgebung, also die Aminosäuren in der Sequenz unmittelbar vor und nach der betrachteten AS, darüber entscheiden, an welcher Sekundärstruktur diese Aminosäure teilnimmt.

Um diese Abhängigkeit der Sekundärstruktur von der Sequenz zu berücksichtigen, werden die gerade betrachtete Aminosäure und die jeweils sechs Aminosäuren, die dieser Aminosäure in der Sequenz vorausgehen bzw. nachfolgen, in das neuronale Netz eingegeben. Aus der Aminosäuresequenz wird also ein „Fenster“ von 13 Aminosäuren herausgeschnitten. Dieses Fenster muß dann in Schritten über die gesamte Aminosäuresequenz geschoben werden, damit jede AS der Sequenz sich einmal im Zentrum dieses Fensters befindet (Abb. 38).

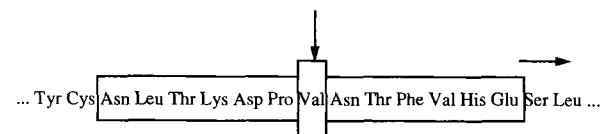


Abb. 38. Ausschnitt („Fenster“) von 13 Aminosäuren aus einer Proteinsequenz, mit dessen Hilfe bestimmt werden soll, an welcher Sekundärstruktur die betrachtete Aminosäure, in diesem Fall Valin, teilnimmt.

Wie wurden nun die einzelnen Aminosäuren codiert? Für jede AS des Fensters aus 13 AS wurde ein Bitvektor der Länge 21 verwendet. In diesem Bitvektor ist für jede der 20 natürlichen AS eine bestimmte Position reserviert, d.h. beispielsweise das 14. Bit wird auf Eins gesetzt, wenn die AS Prolin vorliegt. Eine weitere Position wird benötigt, um kennzuzeichnen, wenn das Fenster am Anfang oder am Ende des Proteins keine AS mehr enthält. Insgesamt werden für die Größe des Fensters und für die Identität einer AS also  $13 \times 21 = 273$  Eingabeeinheiten benötigt, wobei jeweils nur ein Bit, also Null oder Eins, in das neuronale Netz eingegeben wird.

Ausgabeneuronen hatte das Netz drei, eines für das Vorkommen einer  $\alpha$ -Helix, eines für  $\beta$ -Faltblatt und eines für eine geknäuelte Struktur. Als optimales Netz wurde nach Versuchen mit 0–80 Zwischenneuronen eine Architektur mit einer Zwischenschicht von 40 Neuronen gewählt, so daß  $273 \times 40 + 40 \times 3 = 11040$  Gewichte zu bestimmen waren. Auch hier wurde das Backpropagation-Lernverfahren gewählt. Die Gesamtarchitektur ist in Abbildung 39 dargestellt.

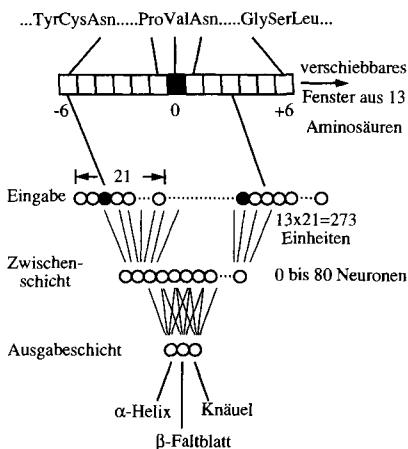


Abb. 39. Neuronales Netz zur Ableitung der Sekundärstruktur eines Proteins aus dessen Aminosäuresequenz.

Das Netz wurde mit 106 Proteinen, die insgesamt 18105 Aminosäuren enthielten, trainiert. Getestet wurde die Leistungsfähigkeit des Netzes mit 15 zusätzlichen Proteinen, die insgesamt 3520 AS umfaßten. Damit konnte eine Vorhersagegenauigkeit von 62.7 % erzielt werden.

Mit 62.7 % Korrektheit kann also angegeben werden, ob eine Aminosäure an einer  $\alpha$ -Helix, einem  $\beta$ -Faltblatt oder einer geknäuelten Sekundärstruktur teilnimmt. Dies ist eine merkliche Verbesserung gegenüber den traditionellen Methoden zur Vorhersage der Sekundärstruktur, läßt aber noch einiges zu wünschen übrig, so daß verständlich wird, weshalb auf diesem Gebiet so aktiv geforscht wird<sup>[48–56]</sup>.

## 7.6. Zusammenfassung

Die Anwendungen aus ganz unterschiedlichen Bereichen der Chemie unterstreichen die breite Einsetzbarkeit neuronaler Netze zur Klassifikation. In allen vorgestellten Beispielen wurde ein Mehrschichtennetz verwendet, das mit dem Backpropagation-Lernverfahren trainiert wurde.

Die Zahl der Neuronen in der Zwischenschicht wird meist durch systematische Versuchsreihen festgelegt. Mit zu wenigen Neuronen läßt sich eine Problemlösung nicht korrekt lernen; mit einer wachsenden Zahl an Neuronen sinkt der Fehler beim Lernen, man benötigt dann aber auch längere Trainingszeiten. Zu viele Neuronen und zu langes Training können zu einem weiteren Problem führen: Overtraining. Darunter versteht man, daß ein neuronales Netz zwar den Trainingsdatensatz ohne Fehler wiedergibt, bei neuen Daten aber zu schlechten Vorhersagen führt. Mehrschichtennetze haben meist sehr viele Gewichte, also auch sehr viele Freiheitsgrade, um sie an einen Datensatz anzupassen. Damit besteht die Gefahr, daß man beim Training in einem lokalen

Fehlerminimum landet, das wenig Vorhersagekraft bei neuen Daten hat.

Mit der Komplexität des neuronalen Netzes, vor allem also mit der Zahl der Gewichte und der Zahl der verwendeten Trainingsdaten, können die Trainingszeiten stark zunehmen. Lange Trainingszeiten sollten aber nicht zu sehr abschrecken, denn im Idealfall muß ein Netz nur ein einziges Mal trainiert werden. Hat es einmal zu Ende gelernt, so können Vorhersagen mit neuen Daten sehr rasch gemacht werden, da diese neuen Daten nur ein einziges Mal über das fertig trainierte Netz geschickt werden müssen.

Bei Klassifizierungsproblemen möchte man in den Ausgabeneuronen die Werte Eins oder Null erhalten, um eindeutig entscheiden zu können, ob ein Objekt zu einer Kategorie gehört oder nicht. In Wirklichkeit erhält man Werte zwischen Eins und Null und kann dann anhand von Schwellenwerten (z.B. 0.8 bzw. 0.2) festlegen, ob das Objekt zu einer Klasse gehört oder nicht. Die numerischen Ausgabewerte können auch als Wahrscheinlichkeiten für eine Klassenzugehörigkeit aufgefaßt werden und als solche in Entscheidungssysteme übernommen werden.

Diese numerischen Ausgabewerte zeigen auch bereits den Übergang zu Modellierungsproblemen, die im nächsten Kapitel behandelt werden. Bei Modellierungsaufgaben möchte man ja Funktionswerte, also reelle Daten erhalten. Diese können aus den Werten zwischen Null und Eins durch anschließende Umwandlung mit mathematischen Funktionen erhalten werden.

Noch einmal, die wesentlichste Aufgabe beim Einsatz eines neuronalen Netzes ist es, eine geeignete Repräsentation der Eingabe- und Ausgabedaten zu finden. Die hierarchische Anordnung mehrerer Netze (Abschnitt 7.4) und das verschiebbare Fenster im Abschnitt 7.5 sollen zeigen, daß der Phantasie keine Grenzen gesetzt sind.

## 8. Modellierung

Wir haben schon gesehen, daß selbst bei Klassifizierungsproblemen ein neuronales Netz Werte zwischen Null und Eins, also ein Kontinuum an Werten, ausgibt. Man kann aber ein neuronales Netz auch mit reellen Erwartungswerten trainieren und die Ausgabewerte direkt in ihrer reellen Größe verwenden, so wie man üblicherweise einen Funktionswert aus einer Reihe von Variablen berechnet. Für diese Aufgabe, aus Daten über ein Objekt (Verbindung, Reaktion, Spektrum etc.) eine weitere Eigenschaft dieses Objekts abzuleiten, hat sich der Ausdruck „Modellierung“ eingebürgert, und so wollen wir ihn im folgenden auch verstehen. Das neuronale Netz übernimmt also die Aufgabe, aus eingegebenen Variablen (Daten) eines Objekts den Wert einer davon abhängigen Größe (oder auch mehrerer) zu ermitteln. Ein neuronales Netz bietet dabei den großen Vorteil, daß man den Zusammenhang nicht durch eine explizite mathematische Gleichung formulieren muß. Das neuronale Netz findet den Zusammenhang selbst und drückt ihn implizit durch die Gewichte der Verbindungen zwischen den Neuronen aus.

### 8.1. HPLC-Analyse

Ein einfaches Beispiel soll stellvertretend für viele Anwendungsmöglichkeiten in der analytischen Chemie stehen.

Bei der HPLC-Analyse spanischer Weine wurde die Trennung der Komponenten, ausgedrückt als Selektivitätsfaktor SF, in Abhängigkeit vom Ethanolgehalt (10, 20, 30 Vol.-%) und vom pH-Wert der mobilen Phase (5.0, 5.5, 6.0) bestimmt. Die neun experimentellen Punkte wurden mit Standardmodellierungstechniken an eine quadratische Gleichung angepaßt<sup>[57]</sup>. Das Ergebnis ist in Gleichung (r) wie-

$$SF = 0.018 x_1^2 - 1.42 x_2^2 + 0.015 x_1 x_2 - 0.995 x_1 + 16.16 x_2 - 31.86 \quad (r)$$

dergegeben ( $x_1$  = Vol.-% Ethanol,  $x_2$  = pH-Wert); dieser funktionale Zusammenhang ist außerdem in Form von Linien mit gleichem Selektivitätsfaktor in Abbildung 40 dargestellt.

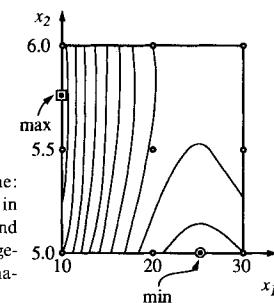


Abb. 40. HPLC-Analyse spanischer Weine: Darstellung des Selektivitätsfaktors SF in Abhängigkeit vom Ethanolgehalt  $x_1$  und vom pH-Wert der mobilen Phase  $x_2$ . Die geschwungenen Pfeile weisen auf den maximalen und den minimalen SF hin.

Mit den gleichen neun experimentellen Daten wurde ein neuronales Netz aus zwei Eingabeeinheiten – eine für den Ethanolgehalt, eine für den pH-Wert –, einem Ausgabeneuron – für den Selektivitätsfaktor – und sechs Neuronen in der Zwischenschicht nach dem Backpropagation-Algorithmus trainiert<sup>[58]</sup>.

In dieses Netz wurden dann Werte für den Ethanolgehalt und den pH-Wert eingegeben. Die Ergebnisse sind dem Diagramm der Abbildung 41 zu entnehmen. Auch hier wurden wie in Abbildung 40 Linien mit gleichem Selektivitätsfaktor gezogen.

Der Vergleich der Abbildungen 40 und 41 zeigt, daß beide Verfahren, Standardmodellierungstechniken und neuronales Netz, zu recht ähnlichen Ergebnissen kommen. Insbesonde-

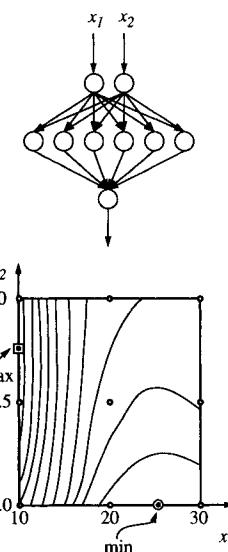


Abb. 41. Mit dem oben gezeigten neuronalen Netz erhaltenen Abhängigkeit des Selektivitätsfaktors vom Ethanolgehalt und vom pH-Wert bei der HPLC-Analyse spanischer Weine. Siehe auch Abbildung 40.

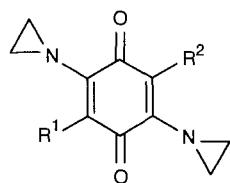
re werden die Lagen von minimalem und maximalem Selektivitätsfaktor ganz ähnlich ermittelt.

Der Vorteil des neuronalen Netzes ist klar: Beim statistischen Modellieren mußte die mathematische Form des funktionalen Zusammenhangs – hier eine quadratische Gleichung – explizit vorgegeben werden. Beim neuronalen Netz ist dies nicht notwendig, es findet den Zusammenhang implizit durch Vergabe geeigneter Gewichte.

## 8.2. Quantitative Struktur-Wirkungs-Beziehungen (QSAR)

Die Suche nach quantitativen Struktur-Wirkungs-Beziehungen (*Quantitative Structure Activity Relationships*, QSAR) ist eines der wichtigsten Anwendungsgebiete für Modellierungstechniken. Speziell zur Vorhersage pharmakologischer und biologischer Daten wird viel Arbeit und Mühe investiert. Um so überraschender ist es, daß bisher nur wenige Arbeiten über den Einsatz neuronaler Netze zur Aufstellung quantitativer Beziehungen zwischen der Struktur und der biologischen Aktivität einer Verbindung publiziert wurden. Eine typische Studie sei hier kurz erwähnt<sup>[59, 60]</sup>.

In dieser Studie wurde bewußt ein Datensatz gewählt, der bereits mit statistischen Modellierungstechniken, einer multilinearen Regressionsanalyse, untersucht worden war, um die Leistungsfähigkeit eines neuronalen Netzes mit der einer Standardmethode aus dem Bereich der QSAR vergleichen zu können. Der untersuchte Datensatz umfaßte 39 zum Teil anticarcinogene *para*-Chinone (Schema 2). Der Einfluß der



Schema 2. Doppelt aziridinsubstituierte *para*-Benzochinone, eine Verbindungsklasse, die anticarcinogene Substanzen umfaßt. R<sup>1</sup>, R<sup>2</sup> beispielsweise = CH<sub>3</sub>, C<sub>6</sub>H<sub>5</sub> etc.

Substituenten R<sup>1</sup> und R<sup>2</sup> wurde durch sechs physikochemische Parameter beschrieben: den Beitrag des Substituenten R<sup>1</sup> oder beider Substituenten zum molaren Brechnungsindeks, MR<sub>1,2</sub> bzw. MR<sub>1,2</sub>, ihr Beitrag zur Hydrophobie, π<sub>1</sub> bzw. π<sub>1,2</sub>, sowie die Substituentenkonstanten des Feld- (F) und Resonanzeffekts (R). Für diese Beschreibung der Substituenteneinflüsse waren demnach sechs Eingabeeinheiten nötig (Abb. 42).

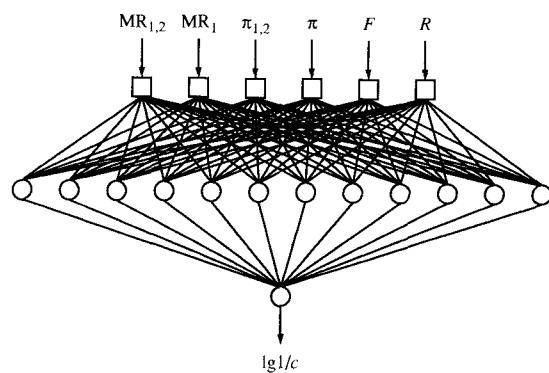


Abb. 42. Neuronales Netz zur Vorhersage der anticarcinogenen Aktivität von *para*-Benzochinonen.

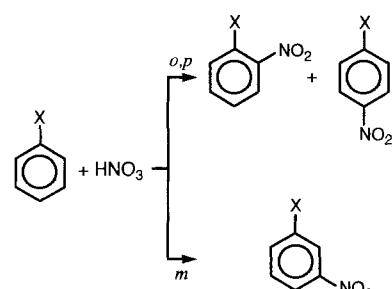
Als Ergebnis sollte das neuronale Netz die minimale effektive Dosis bei einer einzigen Injektion des Medikaments liefern. Diese minimale effektive Dosis ist diejenige Menge an Substanz (angegeben als lg 1/c), die zu einer 40proz. Verlängerung der Lebensdauer führt. Zur Ausgabe des Wertes von lg 1/c war ein einziges Neuron vorhanden. Eine Zwischenschicht mit 12 Neuronen vervollständigte die Netzwerkarchitektur (Abb. 42).

35 Benzochinone wurden zum Training des Mehrschichtennetzes nach dem Backpropagation-Algorithmus verwendet. Die mit dem Netz erhaltenen Werte für lg 1/c wurden mit denen verglichen, die sich mit einer durch multilinear Regressionsanalyse bestimmten Gleichung errechnen lassen. In 17 Fällen waren die Ergebnisse mit dem neuronalen Netz besser, bei 6 etwa gleich gut und bei 12 schlechter, d.h. die Ergebnisse mit dem neuronalen Netz sind signifikant besser. Allerdings läßt sich das Problem durch einen linearen Ansatz schon ganz gut lösen, so daß ein neuronales Netz nicht mehr viel verbessern kann. Bei QSAR-Problemen, die nichtlineare Zusammenhänge umfassen, mag mit neuronalen Netzen bedeutend mehr gewonnen werden.

## 8.3. Chemische Reaktivität

Während wir uns im Abschnitt 7.1 mit einer Ja-nein-Aussage zur chemischen Reaktivität – bricht eine Bindung leicht polar oder nicht – begnügten, wollen wir hier quantitative Aussagen zum Ablauf einer chemischen Reaktion machen.

Die elektrophile aromatische Substitution an monosubstituierten Benzolen kann prinzipiell zu drei Isomeren führen, *ortho*-, *meta*- bzw. *para*-Produkt (Schema 3). Die Abhängigkeit der Isomerenverteilung von der Art des Substituenten X ist ein klassisches Problem der Organischen Chemie. Im wesentlichen lassen sich Substituenten in zwei Klassen einteilen: Elektronenliefernde Substituenten (induktiv oder mesomer), die bevorzugt in *o*- und *p*-Stellung dirigieren, und mesomere Elektronenacceptoren, die in *m*-Stellung lenken. Die Faktoren, die das *o/p*-Verhältnis bestimmen, sind sowohl sterischer als auch elektrostatischer Art.

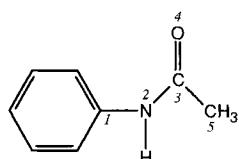


Schema 3. Isomerenverteilung bei der elektrophilen aromatischen Substitution.

In einer Untersuchung der Produktverhältnisse bei der Nitrierung monosubstituierter Benzole<sup>[61]</sup> wurden die Mengen an *ortho*- und *para*-Produkten zusammengefaßt. Daher wurde ein Ausgabeneuron für den Gehalt an *o*- + *p*-Isomer und ein zweites für den an *m*-Isomer verwendet. Wie bereits erwähnt, wird die Produktverteilung von der Art des Substituenten, insbesondere von den von ihm ausgehenden elek-

tronischen Effekten bestimmt. Um diese zu erfassen, wurden zwei Kodierungen der Eingabeinformation erprobt. Im ersten Ansatz wurden die partiellen Atomladungen auf den sechs Kohlenstoffatomen des Benzolrings herangezogen, so wie sie mit dem semiempirischen quantenmechanischen Programm MOPAC<sup>[6,2]</sup> nach der Mullikenschen Populationsanalyse berechnet werden. Dafür wurden sechs Eingabeeinheiten verwendet; hinzu kam eine Zwischenschicht mit zehn verdeckten Neuronen, so daß insgesamt  $6 \times 10 + 10 \times 2 = 80$  Gewichte zu bestimmen waren.

In einem zweiten Ansatz wurde die Struktur des Substituenten direkt in Form einer Bindungsliste (Connection Table) repräsentiert. Diese hatte die Dimension  $5 \times 5$ ; jede Zeile enthielt zunächst die Ordnungszahl des betrachteten Atoms, dann die Atomnummer, die Nummer des vorausgehenden Atoms, an das es gebunden ist, die Ordnung dieser Bindung sowie die formale Ladung des betrachteten Atoms. Für jedes Atom des Substituenten, außer für Wasserstoffatome, wurde eine Zeile ergänzt. Die erste Zeile gibt somit die Verhältnisse an dem Atom, das direkt an den Ring gebunden ist, wieder. Mit jeder folgenden Zeile wird weiter durch den Substituenten fortgeschritten (vgl. Abb. 43). Hatte der Substituent weniger als fünf Nichtwasserstoffatome, so wurde der Rest der 25 Positionen mit Nullen aufgefüllt, hatte er mehr als fünf Schweratome, so wurden die Atome, die vom Anknüpfungspunkt des Substituenten an den Ring (Atom 1) weiter entfernt waren, weggelassen. Abbildung 43 erläutert am Beispiel des Acetanilids diese Kodierung durch eine Bindungsliste. Für diese Kodierungsform wurden  $5 \times 5 = 25$  Eingabeeinheiten benötigt; die Zwischenschicht hatte fünf Neuronen, das Netz also  $25 \times 5 + 5 \times 2 = 135$  Gewichte.



Ordnungs- zahl	Bindung zwischen Atom 1 und Atom 2	Bindungs- ordnung	Ladung
7	2	1	1
6	3	2	1
8	4	3	2
6	5	3	1
0	0	0	0

Abb. 43. Beispiel für die Repräsentation monosubstituierter Benzole durch eine Bindungsliste für den Substituenten.

Das Netz wurde mit 32 monosubstituierten Benzolen nach dem Backpropagation-Algorithmus trainiert; mit 13 weiteren Benzolderivaten wurde das Netz dann getestet. In diesem Beispiel war die enorme Zahl von 100 000 Epochen, also Trainingszyklen nötig, bis der Fehler für den Trainingsdatensatz genügend klein war.

Von den beiden Kodierungsformen und den dabei verwendeten neuronalen Netzen lieferte der zweite Ansatz, die Eingabe des Substituenten durch eine Bindungsliste, die eindeutig besseren Ergebnisse. Um die Qualität der Ergebnisse zu beurteilen, genügt es, sich auf die Prozentzahlen der *meta*-Produkte zu konzentrieren (Tabelle 2). Der Trainingsdatensatz konnte bis auf einen mittleren Fehler von 0.3 % beim Gehalt an *m*-Isomeren gelernt werden; beim Netz, das mit

Tabelle 2. Ergebnisse der beiden im Text erläuterten neuronalen Netze zur Vorhersage der Menge an *meta*-Produkt bei der Nitrierung monosubstituierter Benzole und Vergleich mit den Ergebnissen des Expertensystems CAMEO sowie mit den Vorhersagen dreier Chemiker. Angegeben ist die Größe des Fehlers der Vorhersage in Prozent.

Methode	Trainingsdatensatz (32 Verbindungen)	Testdatensatz (13 Verbindungen)
neuronales Netz aufgrund von Ladungswerten	5.2	19.8
neuronales Netz aufgrund von Bindungslisten	0.3	12.1
CAMEO	18.0	22.6
Chemikerwissen	—	14.7

den Ladungswerten erhalten wurde, gelang dies nur bis zu einem mittleren Fehler von 5.2 %. Beim Testdatensatz von 13 Verbindungen, die das Netz noch nicht gesehen hatte, ergab sich mit der Bindungslistenkodierung eine mittlere Abweichung von 12.1 % im vorhergesagten Gehalt an *m*-Produkt. Bei der Ladungsrepräsentation war der Fehler mit 19.8 % merklich höher.

Die Ergebnisse dieser beiden neuronalen Netze wurden mit den Werten, die mit CAMEO<sup>[6,3]</sup>, einem Expertensystem zur Reaktionsvorhersage, erhalten wurden, verglichen. Die Vorhersagen mit den beiden neuronalen Netzen waren durchweg besser als die mit CAMEO erzielten. Und zu guter Letzt wurden die 13 monosubstituierten Benzole noch drei Organikern gegeben, damit sie den zu erwartenden Anteil an *m*-Produkt bei der Nitrierung vorhersagen sollten. Die von ihnen angegebenen Werte wurden gemittelt; es ergab sich ein Fehler von 14.7 %. Damit waren die Chemiker besser als das neuronale Netz mit der Ladungskodierung und als CAMEO, wurden aber von dem neuronalen Netz mit der Bindungslisteneingabe geschlagen!

So ermutigend die Ergebnisse über die Produktverhältnisse bei der Nitrierung monosubstituierter Benzole mit dem auf der Bindungslistenkodierung basierenden neuronalen Netz sind, so sollen sie hier doch noch näher kommentiert werden.

Zum einen ist nicht überraschend, daß die Kodierung der Benzolderivate durch die Partialladungen auf den sechs Ringatomen wenig überzeugende Ergebnisse liefert. Abgesehen von den bekannten Unzulänglichkeiten der Mullikenschen Populationsanalyse ist die Grundzustandsladungsverteilung nur einer der Faktoren – und nicht einmal der wichtigste elektronische Effekt –, der die Produktverhältnisse bei der elektrophilen aromatischen Substitution beeinflußt. So mit können die Ladungswerte die Benzolderivate nur unzureichend bei der Erklärung der Produktverhältnisse der Nitrierung repräsentieren.

Die Kodierung der Benzolderivate durch die  $5 \times 5$ -Bindungsliste, so zufriedenstellend die Ergebnisse auch scheinen mögen, kann zur Erklärung der Effekte, die dirigierend wirken, zur Aufklärung des Einflusses der Reaktionsbedingungen und zur Vorhersage der Produktverhältnisse bei di- und polysubstituierten Benzolen nichts beitragen. Dazu muß man schon eine andere Repräsentation für die Benzolderivate wählen. Und das ist tatsächlich machbar<sup>[6,4]</sup>. Man darf dabei den Substituenten nicht global beschreiben, sondern muß seinen Einfluß auf die einzelnen Positionen des aromatischen Ringes direkt repräsentieren; für jede Ringposition muß man einen Wert für den Resonanzeffekt, das lokale

elektrostatische Potential und den sterischen Effekt angeben. Da damit der Einfluß eines Substituenten individuell auf jede einzelne Ringposition wiedergegeben wird, kann von monosubstituierten Benzolen auf di- und polysubstituierte geschlossen und für diese Vorhersagen über die Isomerenverteilung bei weiterer Substitution gemacht werden. Zusätzlich läßt sich auch noch der Einfluß des Mediums berücksichtigen, indem eine weitere Eingabeeinheit für die Konzentration der Schwefelsäure zur Verfügung gestellt wird und das Netz damit, zusammen mit den Deskriptoren der Substinenteneinflüsse, trainiert wird<sup>[64]</sup>.

#### 8.4. Zusammenfassung

Modellierungsaufgaben stellen sich häufig und in vielen Bereichen der Chemie. Hier eröffnet sich ein breites Anwendungsgebiet für neuronale Netze. Zur Zeit dominieren bei diesen Anwendungen nahezu ausschließlich Mehrschichtennetze und der Backpropagation-Algorithmus. Dies muß aber nicht so sein; andere neuronale Netze, insbesondere mit dem Counterpropagation-Algorithmus<sup>[27]</sup> trainierte, können durchaus ebenso zur Modellierung verwendet werden.

Der Einsatz neuronaler Netze zur Modellierung, d. h. zur Vorhersage einer Objekteigenschaft aus einer Reihe von Parametern oder Messungen an diesem Objekt, sollte immer gegenüber der Verwendung statistischer Methoden abgewogen werden. Hat man eine relativ klare Vorstellung, welche Variablen die gesuchte Eigenschaft beeinflussen, und liegt ein weitgehend linearer Zusammenhang vor, so bieten traditionelle Methoden, z.B. eine multivariate Regressionsanalyse, eindeutige Vorteile: Man kommt schneller und mit weniger Rechenzeit zu einem Ergebnis, man kann Maßzahlen für die Güte der ermittelten Beziehungen angeben und vor allem läßt die aus der statistischen Modellierung erhaltene Gleichung die Einflüsse der einzelnen Effekte auf die gesuchte Eigenschaft gut interpretieren.

Neuronale Netze sollte man aber einsetzen, wenn anzunehmen ist, daß nichtlineare Beziehungen zwischen der abhängigen und der unabhängigen Variablen bestehen, und wenn nicht genau angegeben werden kann, welche Parameter die untersuchte Eigenschaft beeinflussen.

Ob statistische Verfahren oder neuronales Netz, der Erfolg einer Untersuchung hängt entscheidend von der Wahl des Datensatzes, von der Repräsentation der Information und von den Methoden zur Validierung der Ergebnisse ab. Auch beim Einsatz neuronaler Netze haben die folgenden Punkte große Bedeutung:

- Wahl eines homogenen Datensatzes zum Training (z.B. durch Experimental-Design-Techniken oder durch ein Kohonen-Netz (vgl. Abschnitt 10.1)
- Aufspaltung des Datensatzes in repräsentative Trainings- und Testdatensätze
- Wahl geeigneter Parameter als Eingabedaten zur Beschreibung der Objekte

### 9. Assoziation

Die Fähigkeit zur Assoziation ist eng verknüpft mit dem Erkennen von Ähnlichkeiten zwischen Objekten. Neuronale Netze können auch als Speicher aufgefaßt werden, da sie die

anhand der Trainingsdaten gelernte Information in Form der Gewichte, der „Synapsenstärken“, niederlegen. Wird ein neues Objekt in ein neuronales Netz eingegeben, dessen Eingabedaten denen eines zum Lernen verwendeten Objekts gleichen, so können manche Netzwerkarchitekturen dies erkennen und liefern als Ausgabe das zum Training verwendete, gespeicherte Objekt.

Diese Fähigkeit neuronaler Netze zur Assoziation, zum Erkennen von Ähnlichkeiten in einer Information, ist bisher noch recht wenig in der Chemie genutzt worden. Auch die beiden hier vorgestellten Untersuchungen sind im wesentlichen nur Modellstudien. An sehr kleinen, einfach gestalteten Datensätzen wird die prinzipielle Lösung eines Problems aufgezeigt; für eine echte Anwendung müßten aber größere Sätze komplexerer Daten untersucht werden.

#### 9.1. Basislinienkorrektur

Bei vielen Spektren treten systematische Abweichungen von der Grundlinie auf, sei es durch Verunreinigungen, Lösungsmittelleinflüsse oder durch apparative Probleme. Fünf Typen von Basislinien (normal, ansteigend, abfallend, konkav, konvex) wurden als einfache, schwarze Pixelmuster in einer Matrix aus  $6 \times 20$  Punkten in ein Hopfield-Netz eingegeben<sup>[65]</sup>. Dann wurde ein einfaches, simuliertes Spektrum, das eine konvexe Basislinie enthielt, durch das so trainierte Hopfield-Netz geschickt. Nach drei Iterationen wurde vom Netz das Muster der Basislinie ausgegeben (Abb. 44).

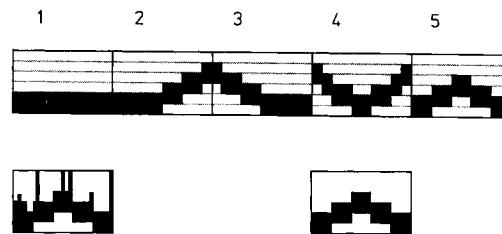


Abb. 44. Fünf Typen von Basislinien eines Spektrums (oben), ein stark vereinfachtes, simuliertes Spektrum mit einer konkaven Basislinie (unten links) und die vom neuronalen Netz nach drei Iterationen herausgefilterte konkav Basislinie (unten rechts).

Man könnte nun das Spektrum mit dieser Basislinie korrigieren und so ein bereinigtes Spektrum erhalten. Wir haben in dieser Übersicht aber bereits darauf hingewiesen, daß eine reale Anwendung natürlich eine viel höhere Auflösung benötigen würde, also die Basislinien durch eine größere Pixelmatrix (z.B.  $20 \times 250$ ) dargestellt werden müßten. Wegen der grundsätzlichen Beschränkungen in der Speicherkapazität von Hopfield-Netzen müßten dann aber, um gute Vorhersagen zu erzielen,  $2 \times 10^7$  Matrixelemente überprüft werden, eine Aufgabe, die sinnvollerweise nur mit Hardware-Implementierungen paralleler neuronaler Netze gelöst werden kann.

Wegen der Speicher- und Rechenprobleme bei Hopfield-Netzen wurde das gleiche Problem der Basislinienerkennung auch mit einem Hamming-Netz untersucht<sup>[65]</sup>. Ein Hamming-Netz<sup>[66]</sup> kommt mit bedeutend weniger Speicherplatz aus. In der Tat konnte das oben skizzierte Problem auch mit einem Hamming-Netz gelöst werden. Aber auch hier wurde

wiederum nur eine  $7 \times 20$ -Matrix zur Darstellung der fünf Basislinientypen eingesetzt; eine Anwendung auf experimentelle Daten mit höherer Auflösung steht noch aus.

## 9.2. Identifizierung von Spektren

Ebenfalls in einer Modellstudie wurde untersucht, wie leicht veränderte UV-Spektren, wie sie z.B. durch Lösungsmitteleinflüsse entstehen, erkannt werden können<sup>[67]</sup>. Prototypen von UV-Spektren wurden durch eine Abfolge von Punkten (Pixel) in zwei  $10 \times 10$ -Feldern dargestellt, wobei das erste Feld mit dem Bandenmaximum an den Eingang und das zweite Feld mit dem abklingenden Signal an den Ausgang eines adaptiven bidirektionalen assoziativen Speichers (ABAM) gelegt wurde (Abb. 45 a). Wegen der bidirektionalen Art des ABAM ist natürlich die Bezeichnung Eingang und Ausgang willkürlich.

Das ABAM wurde mit fünf derartig codierten Modell-UV-Spektren trainiert. Zur Abfrage wurden unscharf codierte und leicht veränderte Spektren verwendet (vgl. Abb. 45 b). Die Rückruffähigkeiten des ABAM hingen stark

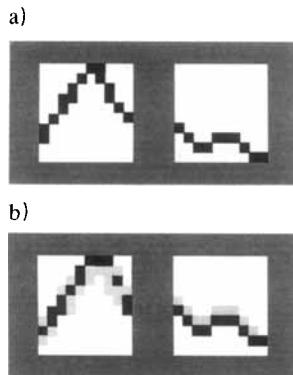


Abb. 45. Rasterartige Repräsentation eines UV-Spektrums (a) und unscharf codierte Formen dieses Spektrums (graue Punktzüge) (b).

von der Wahl mehrerer Netzparameter ab, aber schließlich wurde ein Satz gefunden, mit dem alle fünf gelernten Spektren bei Eingabe des jeweils verrauschten Spektrums eindeutig zurückgerufen werden konnten.

## 10. Abbildung

Viele chemische Phänomene werden gleichzeitig von einer ganzen Reihe von Faktoren beeinflußt, hängen von vielen Parametern ab. Diese Parameter können als Koordinaten eines vieldimensionalen Raumes aufgefaßt werden; die einzelnen Beobachtungen entsprechen dann Punkten in diesem Raum. Will man die Struktur dieser von vielen Variablen abhängigen Information erkennen, die wesentlichen Zusammenhänge zwischen den einzelnen Datenpunkten sichtbar machen, so muß man die Dimensionalität des Raumes so weit reduzieren, daß er graphisch dargestellt und somit mit dem Auge analysiert werden kann. Bei einer derartigen Abbildung, beispielsweise in eine zweidimensionale Ebene, sollen die wesentlichen Beziehungen zwischen den Datenpunkten möglichst weitgehend erhalten bleiben.

In Abschnitt 4.3 haben wir gesehen, wie ein dreidimensionaler Raum, eine Kugeloberfläche, durch ein Kohonen-Netz

so auf die Oberfläche eines Torus abgebildet werden kann, daß die Nachbarschaftsbeziehungen zwischen den Punkten auf der Kugeloberfläche bestehen bleiben. Im folgenden werden zwei Beispiele für die Abbildung mehrdimensionaler chemischer Informationen durch ein Kohonen-Netz gegeben.

## 10.1. Chemische Reaktivität

Wir wollen noch einmal den in Abschnitt 7.1 vorgestellten Datensatz untersuchen, der eine Reihe von Einfachbindungen in aliphatischen Molekülen danach einteilt, ob sie leicht oder schwer heterolytisch gebrochen werden können (vgl. Abb. 31). Jede Bindung wurde dabei durch sieben elektronische und energetische Parameter wie Ladungsdifferenz, Bindungspolarisierbarkeit und Bindungsdissoziationsenergie gekennzeichnet (siehe Abb. 30). Der Bruch einer Bindung wird also als Punkt in einem siebendimensionalen Raum dargestellt.

Die Frage ist nun, separieren sich in diesem siebendimensionalen Raum die Punkte der reaktiven Bindungen von denen der nichtreaktiven? Und weiterhin, wenn dem so ist, läßt sich diese Trennung durch ein Kohonen-Netz auf eine zweidimensionale Fläche abbilden?

Mit den 149 Bindungen wurde ein Kohonen-Netz aus  $11 \times 11$  Neuronen trainiert, wobei als Eingabedaten die Werte der sieben elektronischen und energetischen Faktoren (siehe Abb. 30) verwendet wurden. In Abbildung 46 ist das Ergebnis dargestellt. Tatsächlich finden sich reaktive Bindungen bevorzugt in bestimmten Neuronen und nichtreaktive in anderen. Die Neuronen der reaktiven Bindungen nehmen zudem einen zusammenhängenden Teil der Kohonen-Karte ein<sup>[29]</sup>.

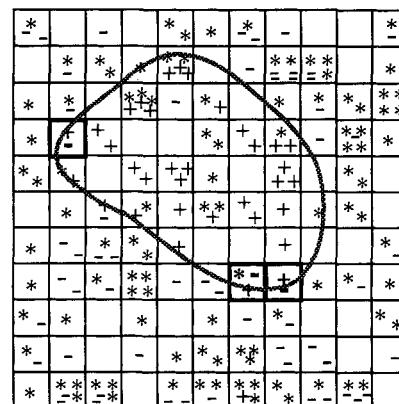


Abb. 46. Kohonen-Netz, das polare Bindungsbrüche, die durch sieben elektronische und energetische Parameter charakterisiert sind, abbildet. + bedeutet eine reaktive Bindung, - eine nichtreaktive Bindung, \* eine Bindung, deren Reaktivität nicht festgelegt wurde.

Daraus läßt sich folgern, daß die gewählten Parameter die Heterolyse einer Bindung gut charakterisieren können – weil sich reaktive und nichtreaktive Bindungen trennen – und daß ein Kohonen-Netz diese Separation auch bei der Abbildung auf eine Fläche beibehalten kann. Es sei nochmals betont, daß das Kohonen-Netz ohne Unterweisung lernt, daß also die Information, ob eine Bindung reaktiv ist oder nicht, während des Lernverfahrens nicht genutzt wurde.

Noch ein weiterer Schluß kann aus Abbildung 46 gezogen werden. Bindungsbrüche, die das gleiche Neuron aktivieren, dürfen in etwa die gleiche Reaktivitätsinformation tragen, das Zusammenspiel der sieben Parameter sollte einen ähnlichen Nettoeffekt in der Reaktivität ergeben. Danach genügt es, von den Bindungen, die dasselbe Neuron aktivieren, eine einzige auszuwählen, wenn die Zusammenhänge zwischen der Reaktivität und den elektronischen und energetischen Parametern weiter untersucht werden sollen. Zugleich sollte aus jedem belegten Neuron mindestens eine Bindung gewählt werden, um das Reaktivitätsspektrum möglichst weit zu überstreichen. In der Tat wurde der Datensatz, der mit dem Backpropagation-Algorithmus studiert wurde (siehe Abschnitt 7.1), unter diesen Gesichtspunkten ausgewählt.

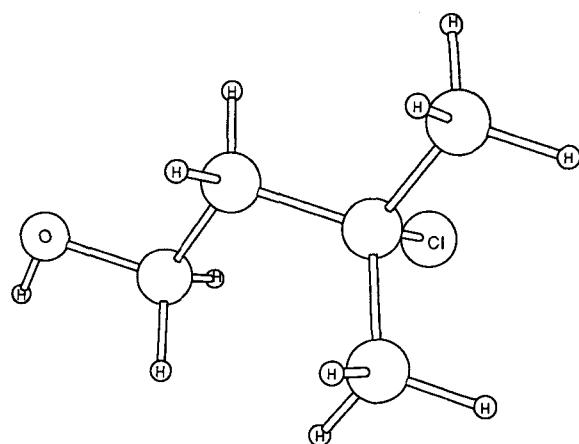
Die Kohonen-Methode kann also zur ausgewogenen Wahl von Datensätzen für Untersuchungen mit statistischen oder Mustererkennungsmethoden oder auch mit anderen neuronalen Netzen eingesetzt werden.

## 10.2. Elektrostatisches Potential

Das elektrostatische Potential um ein Molekül beeinflußt in entscheidendem Maße viele physikalische, chemische und biologische Eigenschaften der Substanz. Besonders bei der Untersuchung von Substrat- und Rezeptor-Wechselwirkungen, aber auch bei Studien zur chemischen Reaktivität werden elektrostatische Potentiale im Detail analysiert.

Bewegt man eine Probeladung, z.B. eine punktförmige positive Einheitsladung, um ein Molekül, so läßt sich für jeden Raumpunkt das elektrostatische Potential auf quantenmechanischem oder klassisch-elektrostatischem Weg bestimmen.

Schema 4 zeigt ein dreidimensionales Molekülmodell von 3-Chlor-3-methylbutan-1-ol und Abbildung 47 das elektrostatische Potential, das eine positive Punktladung auf der van-der-Waals-Oberfläche dieses Moleküls erfährt. Die



Schema 4. Molekülmodell von 3-Chlor-3-methylbutan-1-ol.

Größe des elektrostatischen Potentials wird dabei in eine Farbcodierung übersetzt: Stark negative Werte des elektrostatischen Potentials, also Positionen, von denen eine positive Ladung angezogen wird (nucleophile Stellen), werden durch rot wiedergegeben; stark positive Werte, also Stellen, an denen eine positive Ladung abgestoßen wird (elektrophile

Stellen), sind blau oder violett; Zwischenwerte werden kontinuierlich durch Farbübergänge dargestellt. Das elektrostatische Potential wurde dabei klassisch durch Summation der Coulomb-Wechselwirkungen der Probeladung mit den nach dem PEOE-Verfahren<sup>[30, 31]</sup> erhaltenen Atomladungen berechnet.

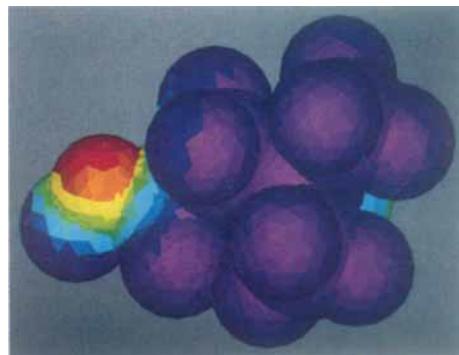


Abb. 47. Elektrostatisches Potential auf der van-der-Waals-Oberfläche von 3-Chlor-3-methylbutan-1-ol. Rote Regionen haben ein negatives Potential, ziehen also eine positive Ladung an, in blauen und violetten Regionen wird sie abgestoßen.

Abbildung 47 gibt eine Parallelprojektion der Potentialverteilung auf der van-der-Waals-Oberfläche des Moleküls in die Zeichenebene wieder. Dabei kann natürlich jeweils nur derjenige Teil des elektrostatischen Potentials, der gerade vom Betrachter einzusehen ist, dargestellt werden. So ist z.B. in Abbildung 47 das Chloratom kaum zu sehen. Einen vollständigen Einblick in die Potentialverteilung kann man sich nur durch eine Reihe solcher Abbildungen aus verschiedenen Betrachterpositionen verschaffen. Je komplexer die Gestalt der van-der-Waals-Oberfläche und die Variation im elektrostatischen Potential ist, um so mehr Abbildungen braucht man, und es wird immer schwieriger, einen ganzheitlichen Eindruck von der Potentialverteilung und den Beziehungen zwischen elektrophilen und nucleophilen Zentren zu erhalten.

Diese Unzulänglichkeit einer Parallelprojektion auf eine Ebene (Bildschirm) ließ uns nach anderen Projektionsmethoden suchen, die in einer *einzig* Abbildung auf eine Ebene die wesentlichen Aspekte des elektrostatischen Potentials auf der Moleküloberfläche wiedergeben können<sup>[68]</sup>. Ein Kohonen-Netz ist eine derartige Projektionsmethode.

Abbildung 48 zeigt die Projektion des elektrostatischen Potentials des Alkohols 3-Chlor-3-methylbutan-1-ol aus Abbildung 47 auf ein Kohonen-Netz. Um diese Abbildung zu erhalten, wurden willkürlich 20000 Punkte von der Moleküloberfläche ausgewählt und mit der  $x$ -,  $y$ - und  $z$ -Koordinate eines jeden Punktes ein Kohonen-Netz mit  $60 \times 60$  Neuronen trainiert. Nachdem alle Punkte über das Netz geschickt worden waren, wurde untersucht, welche Neuronen mit welchen Punkten besetzt waren.

In der Tat fanden sich Punkte mit gleichen oder ähnlichen Potentialwerten im gleichen oder in nahe beieinanderliegenden Neuronen. Das elektrostatische Potential auf der van-der-Waals-Oberfläche eines Moleküls wurde also auf die Kohonen-Karte abgebildet, wobei die Nachbarschaftsbeziehungen auf der van-der-Waals-Oberfläche weitgehend erhalten blieben.

Es sei nochmals daran erinnert, daß in einem Kohonen-Netz die Abbildung auf die Oberfläche eines Torus erfolgt, also auf eine Fläche ohne Anfang und Ende (vgl. Abb. 24). Die Karte der Abbildung 48 kann also beliebig nach oben oder unten oder nach links oder rechts verschoben werden.

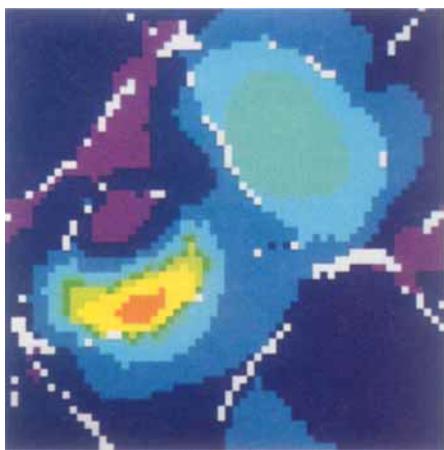


Abb. 48. Kohonen-Netz des elektrostatischen Potentials aus Abbildung 47.

Ein Vergleich der Kohonen-Karten der elektrostatischen Potentiale von Molekülen läßt wesentliche Ähnlichkeiten in den elektronischen Eigenschaften der Moleküle auf ihrer Oberfläche erkennen, also an Stellen, an denen sie in unmittelbaren Kontakt mit ihrer Umgebung treten. Das elektrostatische Potential auf der Oberfläche eines Moleküls ist ein entscheidender Faktor für die Wechselwirkung eines Substrats mit einem biologischen Rezeptor. Somit sollten Kohonen-Karten der elektrostatischen Potentiale von Molekülen, die an den gleichen Rezeptor binden, gewisse Ähnlichkeiten aufweisen.

In der Tat konnte gezeigt werden, daß die Kohonen-Karten der elektrostatischen Potentiale von Verbindungen, die an den muscarinischen Rezeptor binden, eine Reihe charakteristischer Gemeinsamkeiten zeigen, Ähnlichkeiten, die z.B. in den Kohonen-Karten der elektrostatischen Potentiale von Molekülen, die an den nikotinischen Rezeptor binden, nicht zu finden sind<sup>[68]</sup>.

Kohonen-Karten lassen also in einer einzigen Abbildung wesentliche Charakteristika elektrostatischer Potentiale erkennen, Charakteristika, die offensichtlich auch für die Bindung von Substraten an biologische Rezeptoren wesentlich sind.

## 11. Zusammenfassung und Ausblick

Obwohl die Entwicklung von Computermodellen für die Informationsverarbeitung im menschlichen Gehirn auf fast fünfzig Jahre zurückblicken kann, werden neuronale Netze erst in den letzten vier bis fünf Jahren in breiterem Maße für Problemlösungen herangezogen. Neue, leistungsfähige Algorithmen eröffneten eine Vielfalt an Anwendungen, und der Einsatz neuronaler Netze steigt, auch in der Chemie, weiter sprunghaft an.

So vielfältig wie die Leistungen des menschlichen Gehirns sind auch die Modelle neuronaler Netze und ihre Einsatz-

möglichkeiten zur Klassifikation, Modellierung, Assoziation und Abbildung. Das Potential, das neuronale Netze zur Verarbeitung chemischer Information bieten, ist bei weitem noch nicht ausgeschöpft.

In der Chemie will man häufig Objekte bestimmten Kategorien zuordnen oder Eigenschaften von Objekten vorhersagen. Dies macht die Dominanz des Backpropagation-Algoritmus bei der Anwendung neuronaler Netze in der Chemie verständlich. Es gibt jedoch eine ganze Reihe anderer Modelle neuronaler Netze, die mit Erfolg in der Chemie eingesetzt werden könnten; dies sollte in Zukunft breiter erforscht werden.

Zweifelsohne hätten sich viele der Probleme, die mit neuronalen Netzen bearbeitet wurden, auch mit statistischen oder Mustererkennungsmethoden lösen lassen. Neuronale Netze bieten aber auch Leistungen, die über die Möglichkeiten traditioneller Datenanalysemethoden hinausgehen. Besonders wichtig ist, daß für den Zusammenhang zwischen den Eingabe- und den Ausgabedaten keine mathematische Form vorgegeben werden muß, sondern er aus den Daten selbst ermittelt und implizit dargestellt wird. Dabei können auch nichtlineare Beziehungen modelliert werden.

Die Verwendung neuronaler Netze erfordert noch viel Experimentieren; Leitlinien, um möglichst rasch zu einer tragfähigen Problemlösung zu kommen, werden erst allmählich sichtbar. Von zentraler Bedeutung für den Erfolg beim Einsatz eines neuronalen Netzes ist die Art der Datenrepräsentation; je besser die zu untersuchende chemische Information codiert wird, um so leichter und besser läßt sich ein Problem lösen.

*Die Kooperation unserer beiden Arbeitsgruppen wurde entscheidend vom Bundesministerium für Forschung und Technologie der Bundesrepublik Deutschland und vom Slowenischen Forschungsministerium gefördert. Insbesondere die Gastprofessur von Jure Zupan im Rahmen des BMFT-Projekts 08G 3226 (Automatische Auswertung von Massenspektren) an der Technischen Universität München hat es uns ermöglicht, einen produktiven wissenschaftlichen Austausch aufzubauen. Wir möchten uns besonders bei unseren Mitarbeitern A. Fröhlich, P. Hofmann, X. Li, J. Sadowski, K.-P. Schulz, V. Simon und M. Novic bedanken, die mit uns die ersten Schritte in das Neuland der Anwendung neuronaler Netze zur Lösung chemischer Probleme unternommen haben.*

Eingegangen am 20. Juli 1992 [A 899]

- [1] Ein Lehrbuch zur Einführung in neuronale Netze erscheint in Kürze: J. Zupan, J. Gasteiger, *Neural Networks for Chemists: A Textbook*, VCH, Weinheim, 1993.
- [2] Für einen ausführlichen Überblick über Anwendungen neuronaler Netze in der Chemie bis 1990 siehe J. Zupan, J. Gasteiger, *Anal. Chim. Acta* **1991**, 248, 1–30.
- [3] K. J. Skinner, *Chem. Eng. News* **1991**, 69(40), 24–41.
- [4] Eine sehr lesenswerte Zusammenstellung von Veröffentlichungen, die Meilensteine in der Entwicklung neuronaler Netze darstellen, jeweils mit einer kurzen Kommentierung, findet sich in *Neurocomputing: Foundations of Research* (Hrsg.: J. A. Anderson, E. Rosenfeld), MIT Press, Cambridge, MA, USA, 1988.
- [5] W. S. McCulloch, W. Pitts, *Bull. Math. Biophys.* **1943**, 5, 115–133.
- [6] W. Pitts, W. S. McCulloch, *Bull. Math. Biophys.* **1947**, 9, 127–147.
- [7] D. O. Hebb, *The Organization of Behaviour*, Wiley, New York, 1949.
- [8] F. Rosenblatt, *Psychol. Rev.* **1958**, 65, 386–408.
- [9] M. Minsky, S. Papert, *Perceptrons*, MIT Press, Cambridge, MA, USA, 1969.
- [10] J. S. Albus, *Brain, Behavior and Robotics*, BYTE Books, Peterborough, USA, 1981.
- [11] S.-I. Amari, *Biol. Cybernetics* **1977**, 26, 175–185.

- [12] S. Grossberg, *Biol. Cybernetics* **1976**, *23*, 121–134.
- [13] T. Kohonen, *IEEE Trans. Comput.* **1972**, *C-21*, 353–359.
- [14] C. von der Malsburg, *Kybernetik* **1973**, *14*, 85–100.
- [15] B. Widrow, M. E. Hoff, *IRE WESCON Conv. Rec.* **1960**, 96–104.
- [16] J. J. Hopfield, *Proc. Natl. Acad. Sci. USA* **1982**, *79*, 2554–2558.
- [17] D. E. Rumelhart, G. E. Hinton, R. J. Williams in *Parallel Distributed Processing*, Vol. 1 (Hrsg.: D. E. Rumelhart, J. L. McClelland and the PDP Research Group), MIT Press, Cambridge, MA, USA, **1986**, S. 318–362.
- [18] D. E. Rumelhart, G. E. Hinton, R. J. Williams, *Nature* **1986**, *333*, 533–536.
- [19] P. J. Werbos, Dissertation, Applied Mathematics, Harvard University, **1974**.
- [20] *Parallel Distributed Processing*, Vol. 1 + 2 (Hrsg.: D. E. Rumelhart, J. L. McClelland and the PDP Research Group), MIT Press, Cambridge, MA, USA, **1986**.
- [21] B. Kosko, *Appl. Opt.* **1987**, *26*, 4947–4960.
- [22] T. Kohonen, *Biol. Cybernetics* **1982**, *43*, 59–69.
- [23] T. Kohonen, *Self-Organization and Associative Memory*, Springer, Berlin, **1988**.
- [24] H. Ritter, T. Martinetz, K. Schulten, *Neuronale Netze – Eine Einführung in die Neuroinformatik selbstorganisierender Netzwerke*, Addison-Wesley, Bonn, **1990**.
- [25] W. Kirchner, *c't* **1990**, *11*, 248.
- [26] X. Li, J. Gasteiger, J. Zupan, unveröffentlichte Ergebnisse.
- [27] R. Hecht-Nielsen, *Appl. Opt.* **1987**, *26*, 4979.
- [28] K.-P. Schulz, P. Hofmann, J. Gasteiger in *Software-Entwicklung in der Chemie* 2 (Hrsg.: J. Gasteiger), Springer, Heidelberg, **1988**, S. 181–196.
- [29] V. Simon, J. Gasteiger, J. Zupan, *J. Am. Chem. Soc.*, eingereicht.
- [30] J. Gasteiger, M. Marsili, *Tetrahedron* **1980**, *36*, 3219–3228.
- [31] J. Gasteiger, H. Saller, *Angew. Chem.* **1985**, *97*, 699–701; *Angew. Chem. Int. Ed. Engl.* **1985**, *24*, 687–689.
- [32] M. G. Hutchings, J. Gasteiger, *Tetrahedron Lett.* **1983**, *24*, 2541–2544.
- [33] J. Gasteiger, M. G. Hutchings, *J. Chem. Soc. Perkin Trans. 2* **1984**, 559–564.
- [34] J. Gasteiger, M. Marsili, M. G. Hutchings, H. Saller, P. Löw, P. Röse, K. Rafeiner, *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 467–476.
- [35] J. C. Hoskins, D. M. Himmelblau, *Comput. Chem. Eng.* **1988**, *12*, 881–890.
- [36] V. Venkatasubramanian, R. Vaidyanathan, Y. Yamamoto, *Comput. Chem. Eng.* **1990**, *14*, 699–712.
- [37] K. Watanabe, I. Matsura, M. Abe, M. Kubota, D. M. Himmelblau, *AICHE J.* **1989**, *35*, 1803–1812.
- [38] P. Bhagat, *Chem. Eng. Prog.* **1990**, *86*(8), 55.
- [39] V. Venkatasubramanian, K. Chan, *AICHE J.* **1989**, *35*, 1993–2002.
- [40] L. H. Ungar, B. A. Powell, S. N. Kamens, *Comput. Chem. Eng.* **1990**, *14*, 561–572.
- [41] N. Bhat, T. J. McAvoy, *Comput. Chem. Eng.* **1990**, *14*, 573–583.
- [42] M. E. Munk, M. S. Madison, E. W. Robb, *Mikrochim. Acta* **1991**, *II*, 505–514.
- [43] E. W. Robb, M. E. Munk, *Mikrochim. Acta* **1990**, *I*, 131–155.
- [44] B. Curry, D. E. Rumelhart, *Tetrahedron Comput. Methodol.* **1990**, *3*, 213–238.
- [45] F. W. McLafferty, D. B. Stauffer, *J. Chem. Inf. Comput. Sci.* **1985**, *25*, 245–252.
- [46] P. Y. Chou, G. D. Fasman, *Biochemistry* **1974**, *13*, 222–241.
- [47] W. Kabsch, C. Sander, *FEBS Lett.* **1983**, *155*, 179–182.
- [48] N. Qian, T. J. Sejnowski, *J. Mol. Biol.* **1988**, *202*, 865–884.
- [49] L. H. Holley, M. Karplus, *Proc. Natl. Acad. Sci. USA* **1989**, *86*, 152–156.
- [50] D. G. Kneller, F. E. Cohen, R. Langridge, *J. Mol. Biol.* **1990**, *214*, 171–182.
- [51] H. Bohr, J. Bohr, S. Brunak, R. M. J. Cotterill, B. Lautrup, L. Norskov, O. H. Olsen, S. B. Petersen, *FEBS Lett.* **1988**, *241*, 223–228.
- [52] H. Bohr, J. Bohr, S. Brunak, R. M. J. Cotterill, F. Fredholm, B. Lautrup, O. H. Olsen, S. B. Petersen, *FEBS Lett.* **1990**, *261*, 43–46.
- [53] H. Andreassen, H. Bohr, J. Bohr, S. Brunak, T. Bugge, R. M. J. Cotterill, C. Jacobsen, P. Kusk, B. Lautrup, *J. Acquired Immune Defic. Syndr.* **1990**, *3*, 615.
- [54] J. D. Bryngelson, J. J. Hopfield, S. N. Sothhard, *Tetrahedron Comput. Methodol.* **1990**, *3*, 129–141.
- [55] M. S. Friedrichs, P. G. Wolynes, *Tetrahedron Comput. Methodol.* **1990**, *3*, 175–190.
- [56] G. L. Wilcox, M. Poliac, M. N. Liebman, *Tetrahedron Comput. Methodol.* **1990**, *3*, 191–211.
- [57] J. Zupan, F. X. Rius, *Anal. Chim. Acta* **1990**, *239*, 311–315.
- [58] M. Tusar, F. X. Rius, J. Zupan, *Mitteilungsblatt der GDCh-Fachgruppe „Chemie – Information – Computer (CIC)“* **1991**, *19*, 72–84.
- [59] T. Aoyama, Y. Suzuki, H. Ichikawa, *J. Med. Chem.* **1990**, *33*, 905–908.
- [60] T. Aoyama, Y. Suzuki, H. Ichikawa, *J. Med. Chem.* **1990**, *33*, 2583–2590.
- [61] D. W. Elrod, G. M. Maggiola, R. G. Trenary, *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 477–484.
- [62] MOPAC, A General Molecular Orbital Program Package V4.0, QCPE.
- [63] A. J. Gushurst, W. L. Jorgensen, *J. Org. Chem.* **1988**, *53*, 3397–3408.
- [64] A. Fröhlich, J. Gasteiger, unveröffentlichte Ergebnisse.
- [65] M. Tusar, J. Zupan in *Software Development in Chemistry 4* (Hrsg.: J. Gasteiger), Springer, Berlin, **1990**, S. 363–376.
- [66] R. P. Lippmann, *IEEE ASSP Magazine* **1987**, April, 4–22.
- [67] M. Otto, U. Hörchner in *Software Development in Chemistry 4* (Hrsg.: J. Gasteiger), Springer, Berlin, **1990**, S. 377–384.
- [68] J. Gasteiger, X. Li, *Software Development in Chemistry 7* (Hrsg.: D. Ziessow), Springer, Berlin, **1993**; X. Li, J. Sadowski, C. Rudolph, J. Gasteiger, unveröffentlichte Ergebnisse.

## Nur Kopieren ist teurer...

... und zudem mühsamer! Diplomanden und Doktoranden können als studentische Mitglieder der GDCh die "Angewandte" für zehn Mark und ein paar Zerquetsche jeden Monat druckfrisch frei Haus erhalten. Das sind weniger als acht Pfennige pro Seite!

### Interessiert?

Dann rufen Sie doch einfach bei Beate Geiß an (Tel. 0 62 01 / 6 06 - 1 99) oder schicken ihr ein Fax (0 62 01 / 6 06 - 1 84). Aber natürlich können Sie ihr auch schreiben:

**VCH-Leserservice, Postfach 10 11 61, 6940 Weinheim**

