



Neuronale Computer

Grundlagen, Stand der Forschung und erste Ergebnisse

Heinz Werntges, Rolf Eckmiller

'Neuronal Computer' – dieses Schlagwort geistert immer häufiger durch die (Fach-)Zeitschriften und erweckt hochfliegende Erwartungen ebenso wie Ängste und Skepsis. Wird hier der alte Traum vom 'wirklich intelligenten' Computer im neuen Gewand präsentiert? Sind neuronale Netzwerke ein industriell attraktiver Weg, 'harte' Probleme aus der Praxis zu lösen? Wie weit sind die Forscher, was ist (noch) übertrieben?

'Neuronal' – das hat etwas mit 'Neurobiologie' zu tun, also mit dem Gehirn und seinen Nervenzellen, den Neuronen. An diesem Wort scheiden sich oft die Geister, wenn es in einem Atemzug mit 'Computer' genannt wird.

Einige empfinden es geradezu als Sakrileg, wenn der Mensch versucht, sein Gehirn, also den vermuteten Sitz seiner Seele, zu analysieren. Soll schließlich doch der 'künstliche Mensch' geschaffen werden? Sind gar dunkle Kräfte am Werk, die seelenlose, perfekt kontrollierbare Kampfmaschinen bauen wollen?

Weniger pessimistisch denkende Zeitgenossen sehen endlich den richtigen Weg beschriftet, einen wirklich intelligenten Rechner zu schaffen, mit dem man sich zum Beispiel in Um-

gangssprache unterhalten kann, der Simultanübersetzungen beherrscht, der kreativ handelt, kurz: der all die Versprechungen erfüllt, nach denen die symbolisch arbeitenden Methoden der Künstlichen Intelligenz (KI) schon seit Jahrzehnten weitgehend vergeblich streben.

In dieser Einführung möchten wir nicht solche extreme Standpunkte diskutieren, sondern einige der vielen naheliegenden Gründe skizzieren, die unserer Ansicht nach die Beschäftigung mit Neurocomputern lohnend – und faszinierend – machen.

Neuronale Netze – die völlig anderen Computer

Die heute dominierenden seriellen Rechner mit Von-Neumann-Architektur sind in ihrer

Arbeitsweise und ihren Stärken und Schwächen geradezu komplementär zu denen unseres Gehirns. Ist es dann nicht naheliegend, einen Rechner dem Gehirn nachzubauen, wenn man von ihm Leistungen erwartet, die typisch für Gehirne sind?

Tatsächlich sind die Leistungen der biologischen datenverarbeitenden Systeme beeindruckend – von der kleinsten Fliege bis hin zum Menschen. Bildverarbeitung in Echtzeit, Orientierung und Bewegung in unbekanntem Gebiet, sensorische und motorische Meisterleistungen, die sich durch Training selbst organisieren und verbessern, sind Beispiele, die die Fähigkeiten der heutigen Industrieroboter noch weit übersteigen. Der mit Bällen jonglierende Roboter aus der netten Amiga-Grafikdemo ist immer noch ein Traum – erst

SHAMROCKs SOFTWARE-Menü für PCs ...

GRAFIK

EAGLE Grafik-Editor zur manuellen Leiterbahn-Entflechtung, auch für Multilayer geeignet. Vor- und Nachbearbeitung von Autorouter-III-Dateien möglich, Ausgabe auf Drucker + Plotter; EGA + Farbmonitor erforderlich **764 DM**

Autorouter III Automatische Entflechtung ein- oder zweiseitiger Platinen, Ausgabe auf Drucker + Plotter, kompatibel mit Shamrock-CAD und EAGLE, für CGA/EGA/VGA/Hercules **764 DM**

Shamrock-CAD Ideal zum Schaltbilder-Zeichnen, umfangreiche Bauelemente-Bibliothek (u. a. mit allen TTL-Symbolen); kompatibel mit Autorouter III, monochrome Darstellung, Ausgabe auf Drucker + Plotter; für CGA/EGA/Hercules (bitte angeben) **495 DM**

InkPen Universelles, vektororientiertes Zeichenprogramm für EGA (16 Farben) und CGA. Symbol-Bibliotheken und selbst-ablaufende „Dia-Shows“ programmierbar; Ausdruck auf Matrix-Printer **198 DM**

Picpaint Einfaches, pixelorientiertes Zeichenprogramm, ideal für Präsentations-Grafik, für EGA (16 Farben) und CGA **98 DM**

PC-DIA Anzeige und Druck von x-y-Diagrammen aus Meßwerte-Dateien; für CGA, EGA (in Farbe) und Hercules ... **198 DM**

KOMMUNIKATION

PCMAIL Professionelles Mailbox-System für Telefonmodem oder Datex-P20H, frei installierbare Haupt- und Untermenüs, vielfältig konfigurierbar **485 DM**

TELESERVICE Fernwartungs-Programm, erlaubt Fernbedienen aller MS-DOS-Kommandos und von Anwenderprogrammen; für Akustikkoppler und Modems geeignet **485 DM**

PCTERM Terminal-Programm für PCs, ATs, PS/2, bis 9600 Bd, Dateien senden/empfangen, DIN-/ASCII-/IBM-Code, XON/XOFF **98 DM**

NET.24 Low-Cost-Netzwerk für Bürokommunikation und Datei-Transfer für bis zu 26 PCs, ATs, Siemens-PCD1, Atari-ST. Starter-Kit für 2 PCs inkl. Server **303.98 DM**

PROGRAMMIER-TOOLS

EDI Superschneller Bildschirm-Editor, nur 7 KByte groß, zwei Texte mit je 64 KByte gleichzeitig bearbeitbar, alle Funktionen wie Suchen/Ersetzen, Blöcke kopieren/verschieben/laden/speichern ... **98 DM**

STRUKTOR Erstellt Struktogramme aus Pascal-ähnlichem Quelltext **398 DM**

V.24-OEM Treiberprogramm für bis zu vier serielle Schnittstellen, 110 bis 38400 Bd, mit Beispielprogrammen in Assembler, C, Pascal, Basic **198 DM**

PC-COMM INT-14H-Puffer für COM1 oder COM2. Erlaubt den Empfang bis 9600 Bd ohne Zeichenverlust über den AUX-Kanal von MS-DOS, mit Beispielprogrammen in Assembler und Pascal **98 DM**

RS2FILE Empfängt Daten über die serielle Schnittstelle und schreibt diese in eine Datei; läuft vollständig im Hintergrund, der PC kann weiterbenutzt werden **198 DM**

NET.24-OEM Treiberprogramm zum Einbinden von NET.24 in eigene Pascal-, Basic- oder C-Programme, mit Beispielen in Assembler, C, Pascal, Basic; inklusive zwei Netzwerk-Adapter **198 DM** (5 m Koax-Kabel hierzu: 7.98 DM)

Utility-Disk Drucker-Umleitung von LPT1 auf LPT2/3, COM1/2, in Datei oder auf Bildschirm; EGA-Hardcopy; Zeichenkonvertierung; schnelles Diskcopy u. v. a. **98 DM**

A51/MSDOS 8051/8052-Crossassembler für PCs, liefert Hex- oder Binärdateien, kompatibel zum Intel-ASM51 **485 DM**

AUTO-65 Erzeugt 6502-Objektcode aus Befehlen der Siemens-NC-Sprache STEP5, mischbar mit Assembler-Quellcode **198 DM**

DATENVERWALTUNG

BRAIN Literatur-Datenbank; vielfältige Ausgabeoptionen, sehr schnelle Suche, komfortable Menüführung **425 DM**

RDB Universelle Datenbank, z. B. für Adressenverwaltung; bis zu 30000 Einträge mit je bis zu 16 Feldern pro Datei **485 DM**

RECHNUNG Programm zum Schreiben von Rechnungen, dateikompatibel mit RDB **198 DM**



Wollen Sie mehr wissen? Bitte fordern Sie unseren kostenlosen Gesamtkatalog an – Anruf genügt.

SHAMROCK SOFTWARE Vertrieb GmbH

Karlstraße 35, 8000 München 2, Telefon 0 89/59 54 68 + 69

Biokybernetik in Düsseldorf

Die Abteilung Biokybernetik im Institut für Physikalische Biologie der Universität Düsseldorf bildet einen der Schwerpunkte, in denen in Deutschland Grundlagenforschung im Bereich neuronaler Netzwerke betrieben wird. Sie umfaßt zur Zeit fünfzehn Mitarbeiter. Einer der Schwerpunkte der Arbeiten liegt bei der Anwendung neuronaler Netze in der Robotik. Derartige 'intelligente' Roboter sollen dann beliebige Bewegungsabläufe lernen und sich selbständig auf Hindernisse im Raum einstellen können.

recht der Roboter, der sich dies selbst beibringt.

Die Hardware-Entwicklung stößt heute an Komplexitätsgrenzen. Es ist jetzt zwar möglich, Rechner mit Tausenden von Prozessoren zu bauen, aber eine effektive Auslastung – und damit echte Parallelität – erfordert völlig neue Rechnerkonzepte und Organisationsprinzipien. Biologische Nervensysteme arbeiten ständig hochgradig parallel.

Je komplexer die Hardware wird, desto mehr potentielle Fehlerquellen existieren auch. Lassen sich fehlertolerante Systeme entwerfen, die nicht gleich zusammenbrechen, wenn eine Komponente ausfällt, ohne schlichte Redundanz, die Bauteile verschwendet? Unser Gehirn kann beträchtliche Ausfälle von Nervengewebe verkraften. Erst Strukturen mit solchen Redundanz-Eigenschaften ebneten den Weg zur 'wafer scale integration', also dem Bau von Riesen-Chips mit 10 cm Durchmesser, welche sich wohl niemals fehlerfrei herstellen lassen werden.

Auch die Software hat heute eine Komplexität erreicht, die nicht mehr jedes Detail überschaubar macht. Kannten die Apple-Fans noch jedes Bit 'persönlich', so stehen heutige UNIX-Anwender vor einer Vielzahl von Bibliotheksaufrufen, Utilities, Shell-Eigenheiten... deren detaillierte Funktionen nicht mehr interessieren dürfen, wenn noch Zeit für die eigentliche Aufgabe übrigbleiben soll – von den zahlreichen kommerziellen Programmpaketen ganz zu schweigen. Aber warum sollte – rein praktisch gesehen – jedes Detail interessieren, solange das System tut, was es soll? Die Schwelle zur nächsthöheren Organisationsebene ließe sich überschreiten mit flexiblen, lernfähigen 'Universalprogrammen', die für ihre spezifische Aufgabe nicht in allen Einzelheiten programmiert,

sondern anhand vieler Beispiele TRAINIERT werden und sich dadurch selbst organisieren.

Wer sagte da gerade 'Science Fiction'? – Genau solche Leistungen sind die Stärken neuronaler Strukturen, und aus solchen Gründen beschäftigt sich eine schnell wachsende Zahl von Forschern weltweit mit diesen Konzepten.

Dieser Artikel will eine Einführung in die Begriffswelt der 'Neuroinformatiker' geben, und ein paar Paradebeispiele sollen veranschaulichen, wie die Ergebnisse aussehen können. Ein Überblick zur momentan eingesetzten Hard- und Software wird diesen Artikel abrunden.

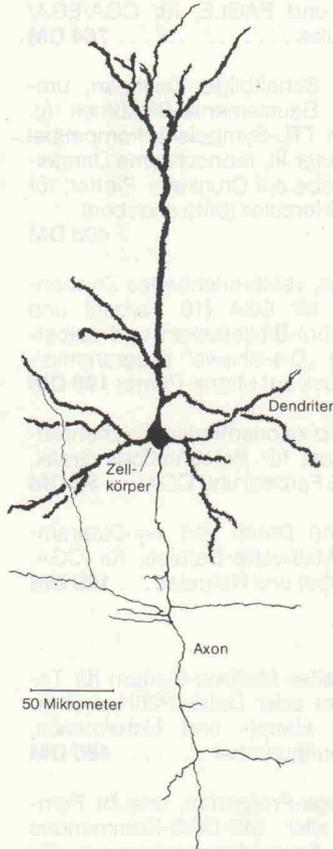
Das biologische Vorbild

Neuronale Netze weisen Gemeinsamkeiten mit vielen Wissenschaftsgebieten auf, aber – wie der Name schon verrät – die biologischen Nervensysteme waren ihr Ursprung. Ohne den folgenden kurzen Blick auf das biologische 'Vorbild' sind daher viele Begriffe und Konzepte schwer nachvollziehbar.

Unser Gehirn besteht aus circa 10¹⁰ (10 Milliarden) Nervenzellen. Im Mittel ist jede dieser Zellen mit etwa 1000 bis 10 000 anderen verbunden (siehe die Abbildungen im Beitrag 'Assoziative Speicher' in diesem Heft). Auf dieser enormen Parallelität und Konnektivität beruhen einerseits unsere mentalen Leistungen, andererseits aber auch die großen Probleme, die Arbeitsweise des Gehirns zu 'verstehen'.

Der hier verfolgte Ansatz versucht gar nicht erst, den 'Schaltplan' des Gehirns zu ergründen, sondern betrachtet zunächst die Eigenschaften der Einzelkomponenten, der Nervenzellen (Neuronen). Die dort abgesehenen Prinzipien werden dann formal erfaßt, so daß wir

ein vereinfachtes mathematisches Modell eines Neurons erhalten. Unsere (berechtigte) Hoffnung ist, daß eine geschickte Verschaltung vieler dieser elementaren Einheiten zu einem System führt, das mehr ist als die Summe seiner Teile und die oben angekündigten Eigenschaften wie Selbstorganisation, Fehlertoleranz, Trainierbarkeit sowie Fähigkeit zum Verallgemeinern aufweist.



Eine Nervenzelle mit ihren Reizleitungen. 10 Milliarden dieser Neuronen arbeiten in unserem Gehirn – sie sollten es zumindest.

Aus dem Biologieunterricht ist manchem Leser vielleicht noch bekannt, daß ein Neuron elektrische Eigenschaften besitzt. Ionenpumpen in der Zellmembran transportieren Ladungsträger durch die Membran und sorgen dadurch für eine elektrische Spannung zwischen 'außen' und 'innen', das sogenannte Ruhepotential. Wird es lokal über eine gewisse Schwelle erhöht, so öffnen sich – wie kleine Schleusen – bestimmte Ionenkanäle in der Zellmembran, und für circa 1 ms kehrt sich das Vorzeichen des lokalen Membranpotentials um – ein Aktionspotential entsteht. Ausgleichsströme aus der Nachbarschaft fließen herbei und pflanzen dabei die Erregung der Membran fort: Das Aktionspotential wandert als einzelner Spannungspuls entlang der Nervenfasern (des Axons), gabelt sich in die vielen Äste am Ende der Faser und setzt schließlich biochemische Prozesse an den Kontaktstellen zu anderen Neuronen, den Synapsen, in Gang.

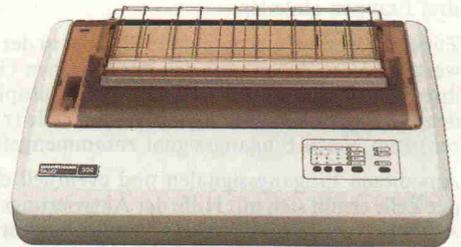
Normalerweise sind Neuronen nicht direkt elektrisch miteinander gekoppelt. Vielmehr sorgt das Eintreffen eines Aktionspotentials an einer Synapse für die Freisetzung einer Substanz (eines Neurotransmitters) in den sehr schmalen Spalt der Synapse zwischen Sender- und Empfängerzelle. Er beeinflusst durch Wechselwirkung mit Rezeptormolekülen auf der Membran der Zielzelle für eine kurze Zeit ihr Membranpotential, und damit schließt sich der Kreis. Solche Beeinflussungen sind es, die ein Neuron erregen können, also sein Membranpotential über die Schwellenspannung heben, so daß es ebenfalls Aktionspotentiale erzeugt.

Neben den 'erregenden' Synapsen gibt es auch 'hemmende'. Diese erniedrigen das Membranpotential der Zielzelle und erschweren so deren Ausbil-

Eigenschaft	Gehirn	Heutiger Computer
Parallelität	hoch	niedrig
Präzision	mäßig	hoch
Fehlertoleranz	hoch	niedrig
Speicherzugriff	global	lokal
Erkennung von Mustern & Ähnlichkeiten	gut	schlecht
Numerische, präzise Berechnungen	schlecht	gut
Fehlerloses Speichern von Daten	schlecht	gut
Rekonstruieren teilweise zerstörter Daten	gut	schlecht
Verallgemeinern von Beispielen auf implizite Regeln	gut	schlecht
Selbstorganisation	ja	bisher nicht

Die Eigenschaften von Gehirn und seriellen Computer sind genau entgegengesetzt.

„Sind Sie Tallyaner?“



Weltweit gibt es immer mehr Tallyaner. Kein Wunder. Denn bei der Breite unseres Programms bieten wir jedem den richtigen Drucker. Zum Einsteigen, Aufsteigen und Umsteigen. Passend zu fast jedem System. Wir bauen Drucker, die von sich reden machen, weil man so wenig dB (A) von ihnen hört, und solche, mit denen man es bunt treiben kann – in 12 Farben. Sie haben die Wahl zwischen Druckern, die besonders schön, besonders schnell und besonders schnell schön drucken. Und wenn Sie häufig die zu verarbeitenden Papiere wechseln müssen – wir bauen sogar Drucker, die sich vollautomatisch umstellen. Um nur einige Vorteile zu nennen.

Werden Sie doch auch Tallyaner.

mannesmann technologie 

*Für Hobby
und Büro-
Tallyssimo!*



Orgatechnik in Köln
vom 20. – 25. 10. 1988
Halle 4.1, Gang A9/B10

Mannesmann Tally GmbH
Postfach 29 69, D-7900 Ulm

Schicken Sie mir Ihr Händlerverzeichnis
und Ihre Typenübersicht über Nadel-, Tinten-
strahl-, Laser- und Hammerbankdrucker.

Name _____

Firma _____

Straße _____

PLZ _____ Ort _____

Telefon _____

CT/H10

MWG 3866-000

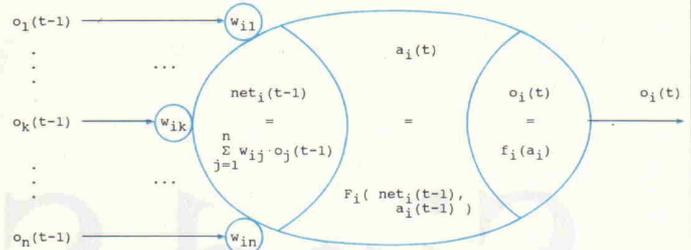
Die Zellen eines neuronalen Netzes entsprechen den Neuronen des biologischen Vorbildes. Eine Zelle führt Rechenoperationen aus und besitzt lokale Speicher, meist in Form der synaptischen Gewichte. Der Rechengang innerhalb einer Zelle läßt sich in drei Etappen einteilen:

Zunächst werden ankommende Signale, in der Regel Ausgangswerte anderer Zellen, mit den synaptischen Gewichten am Ort ihres Eintreffens modifiziert, also etwa multipliziert. Dann werden die Ergebnisse aller Synapsen der Zelle (z. B. durch Addieren) zum Netto-Eingangssignal zusammengefaßt.

Aus diesen Eingangssignalen und eventuell der Vorgeschichte der Zelle ergibt sich mit Hilfe der Aktivierungsfunktion $F()$ ihre Aktivität, die sich als Analogon zum Membranpotential eines Neurons auffassen läßt. Viele Modelle unterscheiden nicht zwischen Eingangssignal und Aktivität. Dann entfällt dieser Schritt.

Schließlich bestimmt eine Ausgangswertfunktion $f()$ aus der Aktivität der Zelle, welches Signal sie an andere Zellen – oder nach 'außen' – sendet. Die vier Beispiele zeigen, wie das allgemeine Schema in der Praxis aussehen kann.

Ideal wäre eine zeitkontinuierliche Beschreibung. Heute werden die meisten neuronalen Netze jedoch auf Digitalrechnern simuliert, und deshalb zeigt die Abbildung eine zeitdiskrete Variante einer Zelle, bei der die Zeit in Einheitsschritten erhöht wird.



Legende:

- $t-1, t$: Zwei aufeinanderfolgende Zeitpunkte in der Simulation
- $o_1 \dots o_n$: Ausgangswerte der Zellen 1 ... n
- ϱ : Vektor (o_1, \dots, o_n) der Ausgangswerte, die die Zelle i erreichen
- w_{ij} : Synaptisches Gewicht zwischen Zelle j (Sender) und Zelle i (Empfänger)
- w_i : Vektor (w_{i1}, \dots, w_{in}) der synaptischen Gewichte von Zelle i
- net_i : Gewichtete Summe aller Eingänge in Zelle i
 $net_i = \langle w_i, \varrho \rangle$ (Skalarprodukt)
- a_i : Aktivität der Zelle i
- $F_i()$: Aktivierungsfunktion von Zelle i
- $f_i()$: Ausgangswertfunktion von Zelle i

dung eines Aktionspotentials. Beide Synapsenarten variieren stark in der Größe des Effektes, den sie beim Eintreffen eines Aktionspotentials in der Zielzelle bewirken. Synapsen haben also recht 'individuelle' Eigenschaften und sind keineswegs untereinander austauschbar. Viele Forscher glauben heute, daß die Synapsen die elementaren Einheiten des biologischen Gedächtnisses sind – die Speicherstellen unseres Gehirns sozusagen. Dies ist plausibel, denn hier entscheidet sich, wie ankommende Signale weitergeleitet werden. Ferner könnten die Lerninhalte hier eine kompakte molekulare, nicht-flüchtige Realisierung beispielsweise in Form von Rezeptormolekülen in der Membran finden. Inzwischen mehren sich die experimentellen Befunde, daß diese Stellen tatsächlich im Laufe der Zeit veränderbar, also potentiell lernfähig sind.

Zur Warnung sei noch gesagt, daß längst nicht alle Vorgänge in biologischen Neuronen heute bekannt oder verstanden sind; auch wurden hier nur einige der wesentlichen Eigenschaften erwähnt. So gibt es etwa auf den sogenannten Dendriten – das sind die feinen Verästelungen des Zellkörpers, an denen die meisten Synapsen anderer Zellen anliegen – weitere Abläufe, über deren Bedeutung bei der Informationsverarbeitung man heute noch spekuliert.

Jedes Neuron ist also ein (gar

nicht so simpler) Prozessor, der die vielen ankommenden Signale 'verrechnet' und eine Ausgabe erzeugt, die über seine Nervenfasern an viele andere Zellen weitergeleitet wird. Technisch gesehen ist ein Neuron ein asynchron arbeitender Prozessor (es gibt keinen 'Systemtakt' für die Neuronen) mit hohem Fan-in und Fan-out (circa 10 000), der nach dem Prinzip der Impulsfrequenz-Kodierung 'kommuniziert': Alle Aktionspotentiale sind gleich, die Information steckt in der Frequenz und Phase ihres Auftretens.

Allgemeiner Ansatz

Was ist nun wesentlich an der Funktion von Neuronen, und welche biologischen Vorgänge brauchen in neuronalen Netzwerken nicht nachgebildet zu werden? Diese Frage ist einer der Forschungsgegenstände der Neuroinformatik. Soll die Phaseninformation der Aktionspotentiale ausgenutzt werden, so muß man den recht aufwendigen Weg gehen und die Impulsfrequenz-Kodierung nachahmen. Dies geschieht bei einigen Arbeitsgruppen tatsächlich, unter anderem auch bei uns in Düsseldorf.

Die große Mehrheit der heutigen neuronalen Netzwerk-Simulationen arbeitet auf einer weniger fein auflösenden Zeitskala und verwendet zeitlich gemittelte Größen, etwa Impulsraten (wieviel Impulse pro Sekunde erzeugt das Neuron ge-

rade?) oder sogar nur binäre Zustände (Neuron inaktiv/maximal aktiv) zur Approximation der Neuronen. Solche Netze sollen nun näher betrachtet werden. Ein neuronales Netz im technischen Sinne besteht aus drei Komponenten:

- den elementaren Recheneinheiten, hier *Zellen* genannt, mit den Regeln für ihr zeitliches Verhalten (ihre 'Dynamik'),
- der *Topologie* des Netzes, also der Beschreibung seiner inneren

Formaler Aufbau einer Zelle bei zeitdiskreter Beschreibung, vereinfacht nach [7]

Beispiele für Zellen

<p>1) <u>Lineare Zelle</u></p> $F_i = F_i(net_i) = net_i,$ $f_i(a_i) = a_i \implies o_i = \langle w_i, \varrho \rangle$ <p>Die lineare Zelle führt nur ein Matrixprodukt aus.</p>	<p>3) <u>Sigmoide Zelle mit Schwelle</u></p> $F_i = F_i(net_i) = net_i,$ $f_i(a_i) = 0.5 \cdot [1 + \tanh(a_i - \theta_i)]$ <p>Die Zelle zeigt quasilineares wie auch Sättigungsverhalten</p>
<p>2) <u>Binäre Zelle mit Schwelle</u></p> $F_i = F_i(net_i) = net_i,$ $f_i(a_i) = H(a_i - \theta_i) \text{ (Heaviside-Fkt.)}$ <p>Die Zelle wird beim Überschreiten ihrer Schwelle aktiv.</p>	<p>4) <u>Halb-lineare Zelle mit Schwelle und nachwirkender Aktivität</u></p> $F_i = F_i(net_i, a_i) = net_i + \alpha \cdot a_i,$ $f_i(a_i) = (a_i - \theta_i) \cdot H(a_i - \theta_i)$ <p>Für $0 < \alpha < 1$ klingt die Zellaktivität nicht plötzlich, sondern exponentiell ab, wenn net_i auf 0 springt.</p>

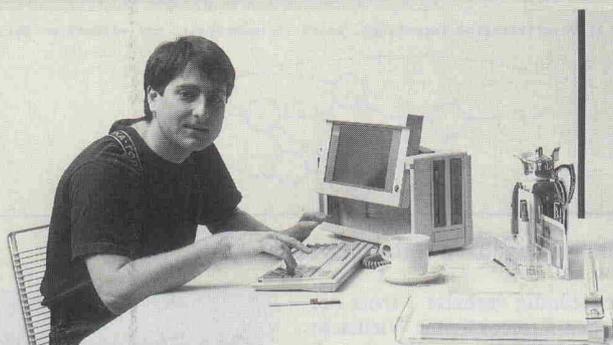
Lesen Sie hier die Geschichte von Mike Wise, Deutschlands erstem SideKick Plus Benutzer

Vor zwei Jahren habe ich mein erstes SideKick gekauft. Kennen Sie sicher: Eine total verrückte Idee. Nennt sich speicherresidentes Programm und schiebt sich auf Tastendruck in den Vordergrund, mit Editor, Rechner und wer weiß was noch alles. Egal, was man gerade macht. Das neue SideKick, SideKick Plus, hab' ich mir gleich gekauft. Und als treuer Heimsoeth-Kunde habe ich auch gleich einen dicken Rabatt bekommen. SideKick Plus hat jetzt nicht nur Editor, ASCII-Tabelle, Rechner und Kalender dabei, sondern auch noch ein Kommunikationsprogramm, einen Outliner und einen Dateimanager. Und statt einem einzigen Editor gibt es gleich neun. Alle auf einmal im Speicher, als Fenster auf dem Bildschirm. Auch die neue Kopierfunktion von SideKick Plus ist ein echter Fortschritt. Damit kann ich

Texte vom Bildschirm, Adressen in Briefe und Prozeduren in andere Programme übernehmen. Was man sonst noch damit machen kann, kön-

mer – über mein Modem. Ein Bekannter von mir ist ganz heiß drauf, der macht Telefonmarketing. Daß man heutzutage ein Modem braucht, ist klar.

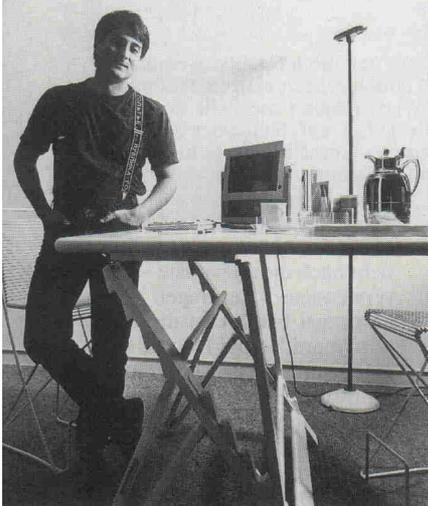
Vorige Woche habe ich meine Festplatte aufgeräumt. Aber eigentlich nur, weil ich den Dateimanager ausprobieren wollte. Habe Dateien markiert und gelöscht, ganze Verzeichnisäste kopiert, Dateien gesucht und verlagert, Verzeichnisnamen geändert. Das geht jetzt alles – SideKick-like – aus jedem Programm heraus. Also ich mag mein neues SideKick Plus. Spart Nerven, spart Geld. Zeit sowieso und macht auch noch Spaß. Was will ich mehr?«



»SideKick's Rechner war schon immer ein Geheimtip, um in Dez und Hex zu rechnen«.

nen Sie sich ja denken. Daß SideKick Plus mit allem Drum und Dran in nur 76 KByte Hauptspeicher unterkommen soll, wollte ich nicht glauben. Aber es kommt hin: SideKick's Memory-Manager lagert einfach auf Festplatte oder Above-Board aus. Adressen habe ich bisher hauptsächlich auf alten Briefumschlägen gespeichert – war keine optimale Lösung, SideKick Plus macht das besser. Ich drücke Crtl-Alt-T und habe mein Telefonbuch auf dem Bildschirm. Mit Eingabefeldern und komplettem Editor für Gesprächsnotizen. Und wenn ich jemanden anrufen will, drücke ich einfach »Return« und SideKick wählt die Num-

Mit der Kommunikationsfunktion von SideKick Plus versende ich jetzt Briefe und Telexe oder hole mir die neuesten Shareware-Programme aus den amerikanischen Mailboxen. Das macht SideKick's Script-Sprache auch noch automatisch, im Hintergrund. Der neue Terminkalender ist übrigens der erste, den ich auch benutze. Der unterbricht mich piepsend beim Programmieren, meldet sich rechtzeitig schon vor dem Termin, forscht nach freier Zeit, die er verplanen könnte, berücksichtigt Feiertage und kann nach Begriffen suchen. Man kann jetzt auch Termine verschieben, beliebig lange Notizen dazu machen – eben all das, was meinem alten SideKick gefehlt hat.



»PC ohne SideKick? Ist eine halbe Sache. Halbe Sachen kann ich mir nicht leisten.«

Für Leute, die es mit Ihrem PC ernst meinen

Orgatechnik Köln
Halle 03. 1
Stand KL 39/40

Heimsoeth software
GmbH & Co. KG
Lindwurmstraße 88
D-8000 München 2
Telefon 0 89-720 10-0



Ich bestelle SideKick Plus zum Preis von 510,72 DM.

Ich arbeite bereits mit SideKick. Bitte informieren Sie mich über die Update-Modalitäten.

Name _____

Straße _____

Ort _____

Telefon _____

Unterschrift _____

Kunden-Nr. _____

Rechner

5¼ Zoll 3½ Zoll

deutsch englisch

Scheck über 510,72 DM liegt bei.

Bitte liefern Sie per Nachnahme (zuzügl. 6 DM Gebühr).

c t 10/88

HEIMSOETH & BORLAND

Verbindungen (welche Zelle sendet an welche anderen?), und – den *Lernregeln*, die die Veränderungen der Kopplungsstärken (Gewichte) der Synapsen zwischen den Zellen bestimmen.

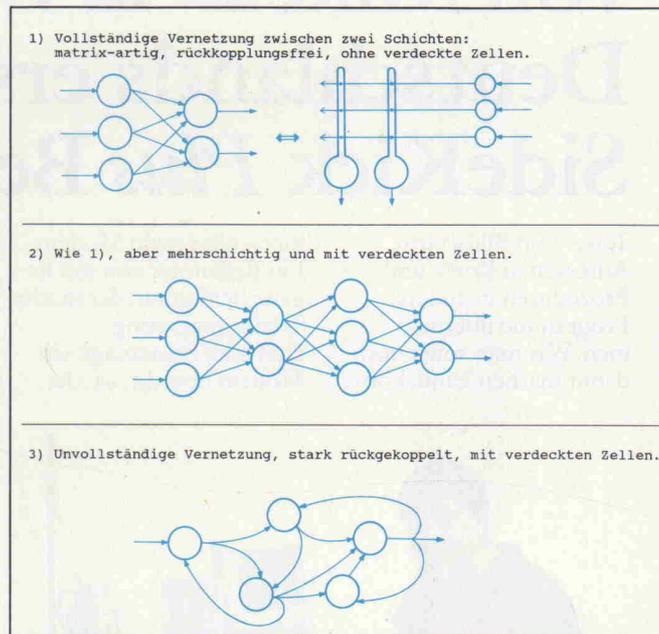
Für die Zellen gibt es eine recht allgemeine Beschreibung (siehe Kästen 'Formaler Aufbau' und 'Beispiele Aktivierungsfunktionen'). Jede Zelle besitzt lokale Variable, insbesondere die synaptischen Gewichte, eine Zellaktivität, die dem Membranpotential eines Neurons entspricht, und einen Ausgangswert, den man zum Beispiel als momentane Impulsfrequenz deuten kann. Diese Werte können REAL-Zahlen sein oder auch nur wenige Bits umfassen.

Dazu kommen lokale Rechenregeln: Eine Aktivierungsfunktion F bestimmt, wie die einlaufenden Signale (und evtl. die Vorgeschichte der Zelle) zur neuen Aktivität verrechnet werden. Eine Ausgangswertfunktion f berechnet den Ausgangswert der Zelle aus der momentanen Aktivität. Dieser wird allen Zellen mitgeteilt, an die die Zelle sendet.

Die Auswahl der Funktionen F und f bestimmt das Verhalten einer Zelle. Es kann vom rein linearen Element, das praktisch nur Skalarprodukte (aus Eingangssignal-Vektor und Synapsengewichts-Vektor) berechnet, bis zur aufwendigen Simulation biologisch plausibler Zellen reichen.

Die Frage, welche Zelle an welche anderen sendet, liegt ganz in der Hand des Netzwerk-Designers. Der Kasten zeigt einige Entwurfstypen. Jedes Netz, das zu etwas nütze sein soll, muß natürlich mit seiner Umgebung in Verbindung treten. Deshalb gibt es stets Zellen, die Eingänge von 'außen' besitzen, und solche, die Ausgänge nach 'draußen' haben. Darüber hinaus bietet vollständige Vernetzung (jede Zelle sendet an jede andere) einen allgemeinen Ansatz, denn deaktivieren lassen sich Verbindungen später immer noch, indem synaptische Gewichte schlicht auf Null gesetzt werden. Natürlich kostet vollständige Vernetzung auch den meisten Speicherplatz und schafft den meisten Verdrahtungsaufwand (bei Realisierung in Hardware).

Vollständige Vernetzung ist nicht immer die richtige Wahl. Auch unser Gehirn ist nicht



Einige denkbare Netzwerk-Topologien

vollständig vernetzt – trotz der hohen Konnektivität. Vielleicht scheut der Konstrukteur auch vor den Gefahren einer Rückkopplung im Netz zurück und entwirft daher ein reines 'feed forward'-Netz.

Als besonders wichtig und interessant erwiesen sich bei unvollständiger Vernetzung sogenannte 'verdeckte Zellen' (engl.: hidden units; sie entsprechen neurobiologisch den Interneuronen), die nur mit anderen Zellen, aber nicht nach draußen verbunden sind. Sie sind die Quellen möglicher Abstraktionsleistungen des Netzes. Dieses Thema ist aber einen eigenen Artikel wert.

Wird jetzt noch festgelegt, welche Anfangswerte die synaptischen Gewichte haben sollen und wann welche Zelle rechnet, dann kommt das ganze Netz schon in Gang. Sollen etwa – wie im biologischen Vorbild – alle Zellen gleichzeitig asynchron arbeiten oder lieber synchron, wie manche Parallelrechner und Zellularautomaten das nahelegen? Berechnet eine Netzwerk-Simulation besser die Zustände der Zellen nach einer festen Folge, oder sollte man die Reihenfolge 'auswürfeln'? Offenbar gibt es beim Entwurf eines neuronalen Netzes viel Entscheidungsfreiheit.

Beispiele

Die Definition der Zellen und die Festlegung ihrer Vernetzung führen schon zu einer Fülle in-

ständigen Information angeboten bekommen. Auch diese Strukturen lassen sich in den vorliegenden formalen Rahmen neuronaler Netze eingliedern.

Bildverarbeitung in Echtzeit ist zur Zeit noch eines der anspruchsvollsten Einsatzgebiete für herkömmliche Computer, bedingt durch sowohl große Datenmengen als auch hohe Anforderungen an die Rechengeschwindigkeit. Im nächsten Heft wird eine Forschergruppe aus Mainz über den Einsatz neuronaler Netze zur massiv parallelen Bildverarbeitung berichten, wobei die Nachahmung der primären Sehvorgänge im Gehirn durch 'festverdrahtete' Zellen eine zentrale Rolle spielt.

Der Clou: Lernfähigkeit

Die dritte Komponente neuronaler Netze – und unserer An-

Das Skalarprodukt in Neuronalen Netzen

Wenn ein Netz arbeitet, läuft in den Zellen ständig eine Rechenoperation ab, die die eintreffenden Erregungen bewertet und zusammenfaßt – das Skalarprodukt. Es ist eine Operation, die zwei Vektoren durch komponentenweises Multiplizieren und anschließendes Addieren zu einem Skalar verrechnet. Kurzschreibweisen dafür sind:

$$net_i = \langle \underline{w}_i, \underline{o} \rangle = \sum_{j=1}^h w_{ij} \cdot o_j$$

Ausgeschrieben bedeutet das:

$$net_i = w_{i1} \cdot o_1 + w_{i2} \cdot o_2 + \dots + w_{in} \cdot o_n$$

Die o_j sind die Signale, die eine Zelle von ihren Nachbarn empfängt, und die w_j sind die synaptischen Gewichte. Setzt man ein bestimmtes w_{ij} auf einen hohen positiven Wert, reagiert die Zelle besonders empfindlich auf Reize der Zelle j . Ein auf Null oder auf einen negativen Wert gesetztes synaptisches Gewicht entspricht einer gekappten oder einer hemmenden Reizleitung.

Interessanter Anwendungen. Bereits 1943 haben McCulloch und Pitts [6] gezeigt, daß einfache neuronale Verschaltungen nach dem hier vorgestellten Muster in der Lage sind, sich wie digitale Gatter zu verhalten. Der Kasten 'Binäre Zelle' veranschaulicht, wie aus dem allgemeinen Ansatz einer Zelle die wichtigsten booleschen Funktionen folgern. Dies bedeutet nicht weniger, als daß im Formalismus neuronaler Strukturen die gesamte heutige Digitaltechnik enthalten ist.

An anderer Stelle wurde in c't schon über assoziative Speicher berichtet [8], die sich an früher (in ihren Synapsengewichten) Gespeichertes 'erinnern', wenn sie nur Bruchstücke der voll-

sicht nach die reizvollste – ist die verwendete Lernregel. Eine Lernregel beschreibt die Langzeitdynamik eines Netzes. Während zum Beispiel die Aktivierungsfunktionen (siehe Kästen) die Dynamik (die zeitliche Entwicklung) der Zellen, insbesondere ihrer Aktivitäten, bestimmen, beschreiben die Lernregeln die zeitliche Änderung der synaptischen Gewichte. Im allgemeinen ist diese Änderung ein viel langsamerer Vorgang als die Aktivitätsänderung der Zellen (siehe die Kästen 'Klassifizierung' und 'Lernregeln' für fortgeschrittene Netzwerke).

Was ist denn nun so reizvoll an diesem Formelkram? – Durch die Lernregeln wird das Netz erst programmiert! Bisher

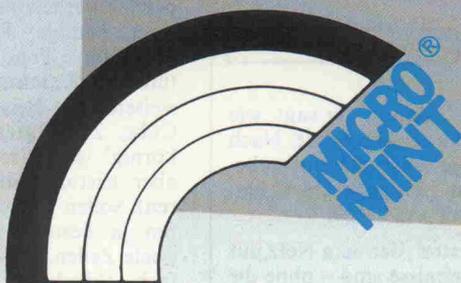
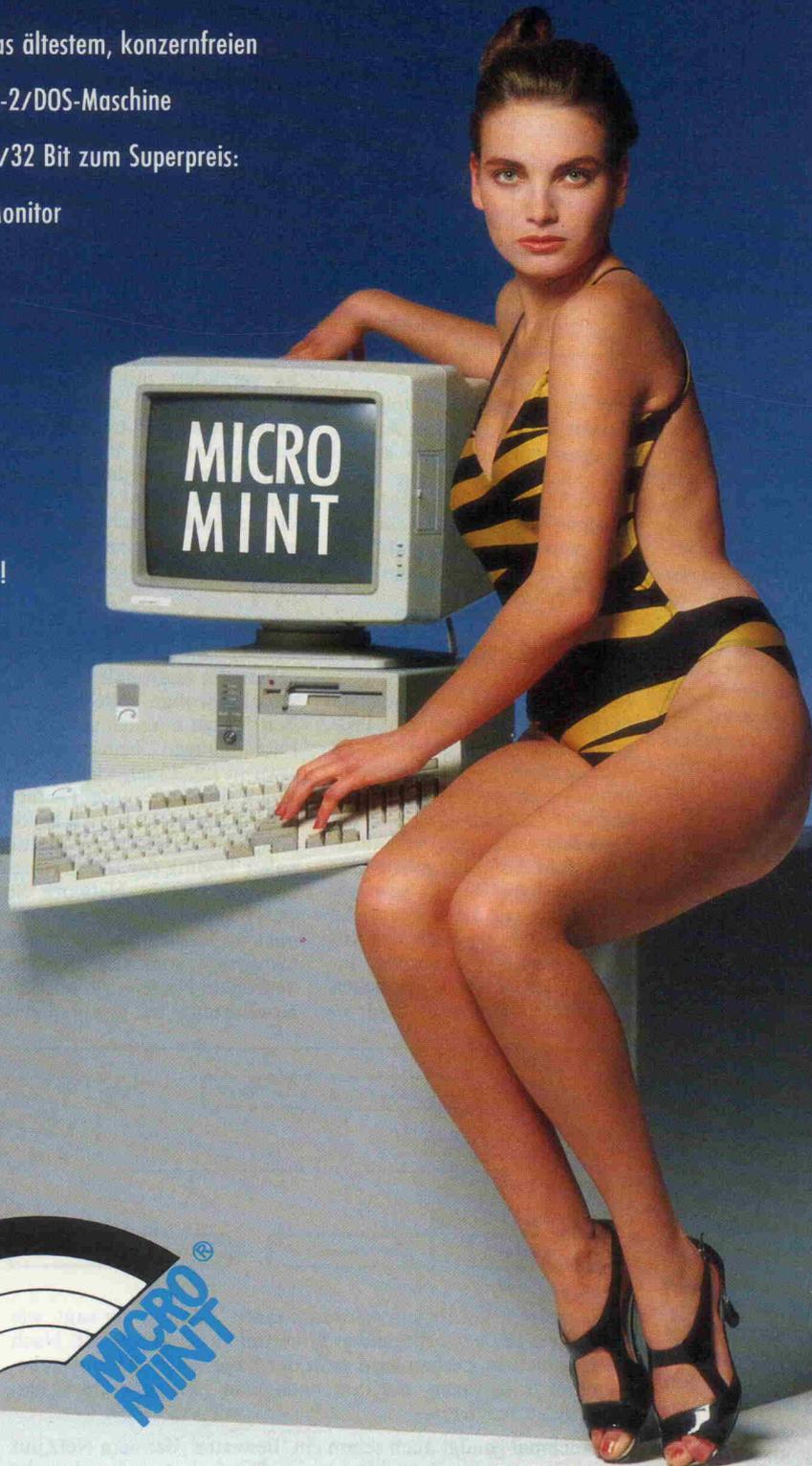
Kühl bleiben, auch bei High-Speed

Zukünftiges heute, von Europas ältestem, konzernfreien
PC-Hersteller: UNIX/XENIX/OS-2/DOS-Maschine
mit 16 MHz/16 Bit bis 33 MHz/32 Bit zum Superpreis:
2.699.- Komplett-AT incl. Monitor

Ständig Sonderpreise
für Wiederverkäufer
bei Komponenten und
neuen, innovativen Produkten!

8 Tage Rückgaberecht!

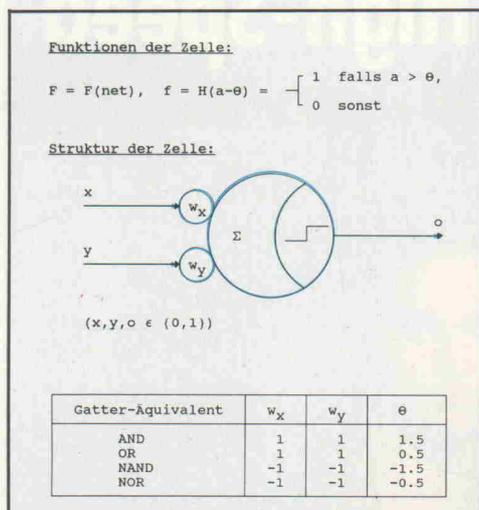
Händlerpreisliste anfordern!



MICROMINT Computer GmbH · Hochdahler Str. 151 · D-4006 Erkrath 2 (Hochdahl) · FAX 0 21 04/30 61 02 · Telex 8 589 305 MCM



0 21 04/3 30 24



Binäre Zelle nach McCulloch und Pitts [6]

wurde nämlich noch nicht ver-raten, woher denn die Werte der synaptischen Gewichte stam-men, die das Netz erst zu dem befähigen, was es tun soll, son-derne sie wurden schlicht voraus-gesetzt. Neben der Wahl der ge-eigneten Netz-Topologie ist die Einstellung der synaptischen Gewichte aber die Hauptauf-gabe bei der 'Programmierung' eines neuronalen Netzes. Will man unbedingt Parallelen zur üblichen Programmierung zie-hen, so könnte die Definition der Netz-Topologie vielleicht der Strukturierung einer Auf-gabe in Unterprogramme ent-sprechen, während die Festle-gung der synaptischen Gewichte dem Schreiben des eigentlichen Programmcodes ähnelt.

Ganz im Gegensatz zu gewöhn-licher Programmierung werden diese Gewichte – die 'Essenz des Netzes' – aber nicht explizit ver-langt. Die Lernregel sorgt viel-mehr dafür, daß sich sinnvolle Werte sozusagen 'bei der Ar-beit' herausbilden. Während das Netz Daten verarbeitet, än-dert es ständig seine synapti-schen Gewichte ein wenig. Wie ein Kleinkind, das laufen lernt, produziert es anfangs sehr viele Fehler, die aber nach und nach weniger werden.

Wie gänzlich anders die 'Pro-grammierung' neuronaler Netze im Vergleich zu sequentiellen Rechnern verläuft, zeigt schon die Wortwahl bei ihrer Beschrei-bung. Sie hat mehr gemein mit Kindererziehung als mit her-kömmlicher Informatik. Es wird nicht allgemeingültig ana-lysiert, definiert, programmiert und korrigiert, sondern am Bei-spiel probiert, vorgemacht, geübt und wiederholt.

Wo liegen nun die typischen Einsatzgebiete für diese Art der 'Programmierung'? So sugges-tiv die Beschreibung ihrer Durchführung auch sein mag, so schwer vorstellbar ist der Ein-satz neuronaler Netze auf den typischen Einsatzgebieten heu-tiger Rechner.

Anwendungsgebiete

In der Tat geht es nicht etwa darum, die sequentielle Pro-grammierung zu ersetzen. Es wird in nächster Zeit kein neu-ronales Netz geben, das darauf trainiert ist, sich etwa wie ein Text-Prozessor zu verhalten. Für solche Aufgaben gibt es klar formulierbare Regeln, also ge-nau das, was die sequentielle Programmierung braucht. An-ders ist die Lage, wenn zwar 'in-tuitiv' klar ist, was geschehen soll (Motto: Das kann doch je-

des Kind . . .), aber niemand die zugrundeliegenden Regeln kennt.

Jeder weiß etwa, wie ein Baum aussieht, obwohl es sehr un-terschiedlich aussehende Baum-arten gibt. Kein Kind käme auf die Idee, einen Baum etwa mit ei-nem Telegraf- oder Hoch-spannungsmast zu verwechseln. Trotzdem fällt es schwer, klar zu definieren, was den Baum aus-macht. Wie lernen wir denn den Unterschied zwischen 'Baum' und 'Nicht-Baum'? – Durch Beispiele, nicht durch Regeln.

Ein neuronales Bilderken-nungs-System bekommt also zu den Bild-Daten die Namen der dargestellten Objekte genannt. Es verändert mit Hilfe einer Lernregel seine synaptischen Gewichte so, daß es die Namen der Objekte lernt und sie auch dann korrekt wiedergibt, wenn das Objekt einmal etwas ver-ändert aussieht. Technisch reali-siert ist das etwa bei der Erkennung handgeschriebener Texte [3, 4].

Leser, die sich einen aktuellen Überblick über die Forschungs-gebiete für neuronale Netze ver-schaffen wollen, sollten sich um Beiträge zu kürzlich gehaltenen Fachtagungen bemühen, wie etwa in [2]. Einen kleinen Ein-druck über relevante Themen-gebiete konnte man sich bei der diesjährigen ersten internatio-nalen Tagung der INNS (Inter-national Neural Network Soci-ety) verschaffen, deren The-men im letzten Kasten zusam-mengestellt sind. Offenbar sind neuronale Netze wirklich keine Konkurrenz zu gewöhnlichen

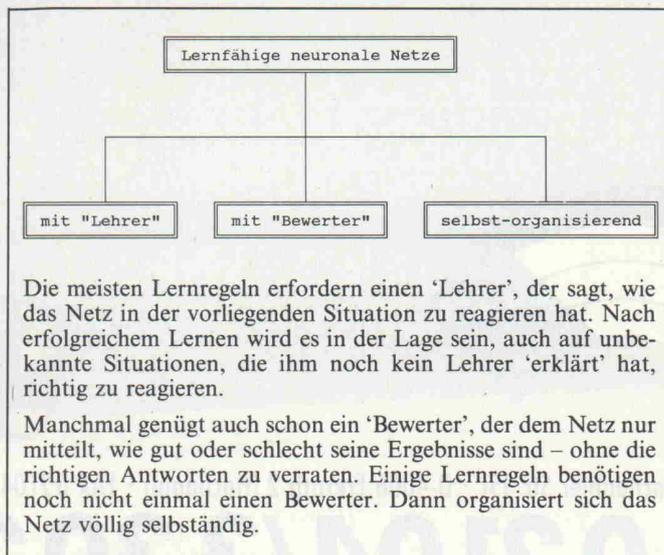
Rechnern, sondern eine Erwei-terung auf bisher schwer er-schließbare Gebiete.

Softe . . .

Realisiert sind neuronale Netze heute meist in Form von Simu-lationsprogrammen auf ge-wöhnlichen Rechnern. Eine Li-ste der dabei eingesetzten Ma-schinen ist hier nicht ange-bracht, da praktisch alle verfü-gbaren Rechner zum Einsatz kommen – vom PC über Work-stations bis zum Großrechner. Einige Arbeitsgruppen haben mit viel Mühe Simulationspa-kete entworfen, die flexible und standardisierte Werkzeuge zum Netzentwurf bieten, etwa [5]. Routinen zur Erzeugung von Zellen, zur Vernetzung, zur De-finition der Lernregeln, der Akt-ivierungs- und Ausgangswert-funktion (F und f) einer Zelle und zum 'Updating' des ganzen Netzes sind in Bibliotheken vor-handen, ebenso wie grafische Analysehilfen, die lokale Akti-vitäten eines laufenden Netzes interaktiv anzeigen. Die bevor-zugte Sprache ist zur Zeit 'C'.

Jede Simulation eines parallel arbeitenden neuronalen Netzes auf einem seriellen Rechner führt bald zu Frust: Rechenzeit und Speicherbedarf wachsen etwa mit der Anzahl Synapsen, leider nicht mit der Zahl der Zellen. Sobald Netze groß und damit interessant werden, wächst die Rechenzeit ins schier Uferlose, insbesondere wenn sie lernen sollen. Auf einem PC sind meist nur einfache Demonstra-tionen realisierbar; spezielle Co-prozessoren oder eine schnelle Workstation schaffen hier et-was Erleichterung.

Deshalb wurden und werden Rechner mit Parallel-Struktu-ren zunehmend zur Netzwerk-Simulation eingesetzt – sofern verfügbar. Problematisch ist da-bei der Vernetzungsgrad. Wäh-rend heutige Parallelrechner meist starre Topologien mit op-timierter Kommunikation auf-weisen (zum Beispiel Baum, n-Cube, 2-D-Gitter) und 'grob-körnig' sind (relativ wenige, aber leistungsfähige Prozessoren), sollen neuronale Struktu-ren ja besonders 'feinkörnig' (viele Zellen, die dann nur ein-fach aufgebaut sein können) und hochgradig vernetzt sein. Zieht man willkürlich Grenzen im Netz und ordnet die ent-stehenden Teile den Prozessoren eines Parallelrechners zu, ent-



Klassifizierung lernfähiger neuronaler Netze

Die meisten Lernregeln erfordern einen 'Lehrer', der sagt, wie das Netz in der vorliegenden Situation zu reagieren hat. Nach erfolgreichem Lernen wird es in der Lage sein, auch auf unbe-kannte Situationen, die ihm noch kein Lehrer 'erklärt' hat, richtig zu reagieren.

Manchmal genügt auch schon ein 'Bewerter', der dem Netz nur mitteilt, wie gut oder schlecht seine Ergebnisse sind – ohne die richtigen Antworten zu verraten. Einige Lernregeln benötigen noch nicht einmal einen Bewerter. Dann organisiert sich das Netz völlig selbständig.

ProSoft-Preise liegen richtig!

☎ 0261/40 47-1 • TX 862476 PSOFT • Telefax 0261/40 47-252

Wir suchen ständig günstige Einkaufsquellen für die angebotenen und neue innovative Produkte. Günstige Möglichkeit der Finanzierung durch Ratenkredit. Fordern Sie die Unterlagen an.

Olivetti Olivetti Olivetti

Bitte erfragen Sie Preis und Lieferzeit für alle Olivetti - Produkte

Commodore Commodore

PC-10 III

8088-2 mit 4,77/7,16 und 9,54 MHz Taktfrequenz, 640 KB Hauptspeicher, parallele und serielle Schnittstelle, Maus-Interface, AGA - Grafikkadpter (Monochrom und Farbe), Echtzeituhr, 2 Diskettenlaufwerke a'360KB, MF-Tastatur, Monitor, MS-DOS 3.2 und GW-Basic **1668.-**

PC-10 III 2/20 wie PC-10 III, jedoch mit 20 MB Festplatte **2298.-**

PC-10 III 2/30 wie PC-10 III, jedoch mit 30 MB Festplatte **2348.-**

PC-10 III 2/50 wie PC-10 III, jedoch mit 50 MB Festplatte **2798.-**

PC-20 III

wie PC-10 III, jedoch nur 1 Diskettenlaufwerk 360 KB und 20 MB Festplatte **2458.-**

PC AT 40/40

80288 CPU mit 8 oder 10 MHz Takt umschaltbar, 1 MB Hauptspeicher, serielle und parallele Schnittstelle, EGA-Grafikkadpter, 1 Disklaufwerk 1.2 MB, 1 Festplatte 40MB, MF-Tastatur mit separatem Cursorblock, Monitor 14", MS-DOS 3.2 und GW-Basic **4998.-**

Commodore 386 PC-60/40

80386 CPU mit 4,77/6/8/10/12 und 16 MHz Takt umschaltbar, 1 MB Hauptspeicher, 2 serielle und parallele Schnittstellen, EGA - Grafikkadpter, 1 Disk. 1.2 MB, 1 Festplatte 40 MB, MF-Tastatur, Monitor 14", MS - DOS 3.2 und GW - Basic **9696.-**

Commodore PC 60/122 wie PC 60/40 jedoch mit 122 MB Festpl. **11498.-**

Amiga 2000 mit Monitor 1084 **2298.-**

Amiga 500 **999.-**

Tandon Tandon Tandon

PC	1828.-	PCA 40 plus	5888.-
XPC 2/20	2398.-	PCA 70 plus	6998.-
XPC 2/30	2448.-	PCA 110 plus	7998.-

PCA 20 plus **4598.-**

PAC 286 **3998.-** Target 20 **4198.-**

PAC 286 plus **4598.-** Target 40 **5498.-**

PAC Floppy **898.-** Target 20 plus **4698.-**

Data PAC **748.-** Target 40 plus **5888.-**

Tandon 386-16 **13198.-** Tandon 386-20 **14698.-**

Laptop Laptop Laptop

Laptop 420 SLC 80288 mit 12,5 MHz, 640KB Hauptspeicher, 1 Diskettenlaufwerk (3,5"), 1.44 MB, 20 MB Festplatte (3,5"), parallele u. serielle Schnittstelle, Tastatur, MS - DOS 3.2/3.3 und GW - Basic **7398.-**

Amstrad Amstrad Amstrad

PC 1640 D Mono **1698.-** PC 1640 D Farbe **1998.-**

PC 1640 HD Mono **2198.-** PC 1640 HD Farbe **2498.-**

PC 1640 D EGA **2498.-**

PC 1640 HD EGA **2998.-**

PC 1512 S Mono **1128.-** PC 1512 D Mono **1398.-**

PC 1512 S Farbe **1498.-** PC 1512 D Farbe **1748.-**

Portable PC **1448.-** PPC 512 D **1698.-**

Textsysteme **948.-** PCW 8512 **1258.-**

PCW 9512 **1498.-**

Drucker **528.-** DMP 4000 **768.-**

LMP 3160 **788.-** LQ 5000 **1188.-**

DMP 2160 (f. CPC-Serie) **488.-**

Plantron Plantron Plantron

PT-XT Tower Computersystem 4,77/8 MHz, 256 KB RAM, Monochrom-Grafikkarte, Multi I/O-Karte, 1 Diskettenlaufwerk 360 KB, dt. Tastatur, u. dt. Bedienungsanleitung **1698.-**

PT-XT/64 Tower-Computersystem wie PT-XT, jedoch zus. mit 64 MB Festplatte **2498.-**

PT-AT Tower-Computersystem 8/10 MHz, 640 KB RAM, Monochrom - Grafikkarte, Multi I/O-Karte, Floppy-Disk-Contr., 1 Disklaufw. 1.2 MB, dt. Tastatur u. dt. Bedienungsanl. **2498.-**

PT-AT/64 Tower-Computersystem wie PT-AT, jedoch zus. mit 64 MB Festplatte **3548.-**

PT-286 AT Tower-Computersystem wie PT-AT/64, jedoch zus. mit 2. Disklaufwerk (3,5", 720 KB) und Super-EGA-Karte **3868.-**

PT-286 AT Tower-Computersystem + Hitachi Multi 560 oder Mitsubishi EUM-1481 A und MS-DOS 3.3 und GW-Basic **5298.-**

PT-386 HT/2 Computersystem 16 MHz, 1 MB RAM, Monochrom - Grafikkarte, Multi I/O-Karte, 1 Disklaufw. 1.2 MB, Echtzeituhr, dt. Tastatur u. dt. Bedienungsanleitung **5298.-**

PT-386 HT Computersystem wie PT-386 HT/2 jedoch mit Super-EGA-Karte 800 x 600 und Festplatte 64 MB **6998.-**

Aufpreis für PT-386 mit 20 MHz Version **1448.-**

MS-DOS 3.30 dt. + GW-Basic **208.-**

CMP CMP CMP CMP CMP

CMP-AT 80288 mit 6/12 MHz Taktfrequenz, Hauptspeicher 512KB, erweiterbar auf 4 MB on Board, Echtzeituhr, 2 x parallel und 1 x serielle Schnittstelle, 1 Diskettenlaufwerk 1.2 MB, Hercules kompatible Grafikkarte, MF-Tastatur u. engl. Bedienungsanl. **2298.-**

CMP-AT/20 **2898.-** CMP-AT/40 **3198.-**

CMP Baby-AT wie CMP-AT, jedoch mit Baby-AT Gehäuse **2198.-**

CMP Baby-AT/20 **2798.-** CMP Baby-AT/40 **3098.-**

CMP-AT/40 (Baby-AT) + incl. Genoa Super EGA Hires + Hitachi Multi 560 Autocan oder Mitsubishi EUM-1481 A incl. MS-DOS 3.3 und GW-Basic **4898.-**

CMP-Tower wie CMP-AT, jedoch mit Tower-Gehäuse **2598.-**

CMP Tower/20 **3198.-** CMP Tower/40 **3498.-**

CMP Tower Maxi wie CMP-AT, jedoch m. Tower-Maxi Gehäuse **2698.-**

Tower-Maxi/20 **3298.-** Tower-Maxi/40 **3598.-**

Atari - Atari - Atari

Atari 1040 STF Tastatur, 1024KB RAM, 192KB ROM, Integrierte Floppy 720 KB, Monochrom-Monitor SM 124, Maus und Basic **1498.-**

EGA/VGA Grafik-Adapter

EGA Wonder Enhanced EGA mit VGA **498.-**

Orchid VGA (1024 x 768 und 512 KB) **798.-**

VIP - VGA Karte von ATI **598.-**

Video Seven VEGA VGA **698.-**

VEGA de Luxe Autoswitch EGA-Karte **598.-**

Paradise EGA Autoswitch (80 Zeichen) **288.-**

Genoa Super EGA Hires plus, mit VGA **398.-**

Genoa Spectrum **288.-**

Genoa Super VGA **648.-**

Genoa Super VGA Hires **928.-**

Tape Streamer 40 MB "Alloy APT-40" jetzt Quick Tape kompatibel incl. DC-2000 Cassette **nur 698.-**

52 MB "Wangtek FAD 3500" für XT oder AT **848.-**

Identica 60 MB Back-up-System, extern **1398.-**

Seagate Festplatten CT 10/88

20 MB Festplattenkit ST-225 incl. XT-Contr. u. Kabelsatz **598.-**

30 MB Festplattenkit ST-238 incl. XT-RLL-Contr. u. Kabelsatz **638.-**

30 MB Festplattenkit ST-138 incl. Contr. u. Kabels., 3,5", 40ms **678.-**

ST-225 (20 MB) **498.-** ST-238 (30 MB) **518.-**

ST-251/1 (40 MB) **888.-** ST-277R (65 MB) **878.-**

ST-4098 (80 MB) **1298.-** ST-4144R (122 MB) **1998.-**

ST-251-0 40 MB, 40 ms, halbe Bauhöhe **778.-**

ST-125/0 (20 MB) **528.-** ST-125/1 (20 MB) **598.-**

ST-157R-0 (50 MB) **898.-**

Priam V-185 110 MB, 18 ms, RLL-fähig **1898.-**

Filecards

20 MB Filecard **648.-**

20 MB Business Card (Tandon) **748.-**

30 MB Filecard **798.-**

Co-Prozessoren

8087 (5 MHz) **198.-** 8087 (8 MHz) **329.-**

8087 (10 MHz) **398.-** 80287 (6 MHz) **329.-**

80287 (8 MHz) **489.-** 80287 (10 MHz) **598.-**

80387-16 **978.-** 80387-20 **1498.-**

80387-25 **2598.-** Fast-Sockel 80287-8/10 88.-

Monitor, Mäuse und Scanner

NEC Multisync II **1448.-** NEC Multisync GS **548.-**

Mitsubishi Autocan EUM-1481 A **1298.-**

14" ADI kompatibler Monitor **198.-**

14" Flat Screen Monitor **228.-**

Hitachi Multi 560 **1298.-**

EIZO 9070S 16" EGA-Monitor **1998.-**

Logimouse C7 plus package deutsch **168.-**

MS-kompatible Mouse seriell **78.-**

Handy Scanner (Cameron) **678.-**

Handy Reeder (Texterkennung) **678.-**

Handy Scanner HS 2000 (DFI) mit Graustufen **498.-**

Star - Star - Star

LC-10 centr. **578.-** LC-10 centr. color **658.-**

LC-10 comm. **578.-** LC-10 comm. color **658.-**

Einzelblatteinzug für LC-10 **198.-**

LC 24-10 **838.-**

Epson - Epson - Epson

LQ-850 **1458.-** LQ-1050 **1868.-**

LQ-500 **848.-** FX-850 **1098.-** FX-1050 **1358.-**

EX-800 **1358.-** EX-1000 **1698.-** LX-800 **598.-**

LQ-2550 **2998.-** GQ-3500 **3898.-** LX-800 VC/P **878.-**

Einzelblatteinzüge für LQ-500 **198.-** LQ-850 **328.-** LQ-1050 **398.-**

NEC - NEC - NEC - NEC - NEC

P2200 **778.-**

Einzelblatteinzug für P2200 **198.-**

Neu! NEC P6 plus 265 Zeichen/Sekunde, 80 Zeichen/Zelle, incl. Traktor u. Halbautom. Einzelblatteinzug **1498.-**

Kyocera - Laserdrucker

F-1000 **5598.-** F-2200 **11798.-**

F-1200 **8298.-** F-3000 **16598.-**

Okidata Okidata Okidata

ML-182 parallel **468.-** ML-320 parallel **998.-**

ML-292 Elite **1048.-** Personality Mod.f.292 **298.-**

ML-393 **2448.-** Emulationsm. f. 393 **148.-**

ML-393 color **2648.-** Emulationsm. f. 393 **148.-**

Laserline 6 Elite **3398.-** HP-Laserjet plus **398.-**

Okimate OM-20 **398.-**

ProSoft GmbH

Bogenstraße 51-53, Postfach 207, D-5400 Koblenz-Goldgrube, Telefon (0261) 4047-1, Telex 862476, Telefax (0261) 4047-252

Alle Preise zuzügl. 10.- DM Versandkosten pro Paket. Lieferung per Nachnahme oder Vorkassenzahlung - Versandkosten Ausland DM 40.- pro Paket. Lassen Sie sich kennen! Bar auf den ProSoft Markt Original-Produkte der führenden Hersteller. Überzeugen Sie sich selbst durch Abholung der Ware in unseren Verkauf- und Vorführräumen in Koblenz. Wir gewähren Ihnen bei Barzahlung kein Scheck! 2% Skonto auf alle Preise, was vielleicht schon zur Deckung Ihrer Reisekosten ausreicht. Einige unserer Vorlieferanten liefern Produkte ohne die Seriennummer des Herstellers. In diesem Fall übernehmen wir anstelle der Herstellergarantie die unbeschränkte gesetzliche Gewährleistung.

Filiale München Theresienstraße 56, 8000 München 2, Tel. 0 89 2/80 93 89 direkt bei der technischen Hochschule. Bitte beachten Sie, daß nicht ständig sämtliche Ware in unserer Filiale München vorrätig ist. Rufen Sie an!

steht schnell ein Kommunikationsengpaß. Immerhin, schneller als rein seriell geht's allemal. So findet man vom grobkörnigen Transputersystem bis zur feinkörnigen Connection Machine mit ihren 2^{16} (= 65 536) Prozessoren viele Parallelrechner-Typen zu Netzwerk-Simulationen eingesetzt.

Eine naheliegende und preiswerte Art der schnellen Netzwerk-Simulation bieten inzwischen zwei amerikanische Firmen (HNC und SAIC, beide mit Sitz in San Diego). Da die Hauptrechenzeit einer Zelle für die Bestimmung des Netto-Eingangswertes (siehe den zweiten Kasten) draufgeht und sich dahinter eine Skalarproduktbildung verbirgt, liegt es nahe, einen preiswerten AT-kompatiblen Rechner mit einer Coprozessorkarte auszustatten, die viel Speicher (typisch 10 MByte) und schnelle Pipeline-Prozessoren (Weitek-Chipset) enthält. Die Firmen verkaufen die Karten zusammen mit Programm-Bibliotheken (C-Funktionen), die die Nutzung der Coprozessorkarte sehr effektiv und auf die Anforderungen neuronaler Netze maßgeschneidert ermöglichen. HNC ist dabei, eine Hochsprache namens AXON zu entwickeln, um den Entwurf neuronaler Netze zu erleichtern und zu standardisieren.

... und harte Fakten

Wirtschaftlich interessant für ein breites Anwendungsfeld werden neuronale Netze wohl erst dann, wenn sie in Form adäquater Hardware realisiert sind. Viele namhafte Computer- und IC-Hersteller haben schon Chip-Prototypen hergestellt oder sind dabei. Allerdings gibt es bisher kaum wirklich marktreife Produkte, vielmehr wird intensiv Grundlagenforschung betrieben. Große Probleme machen vor allem:

- Analog-VLSI-Techniken
- Nicht-flüchtige, aber variable synaptische Stärken
- Realisierung der Lernregeln auf dem Chip
- Hohe Vernetzung 'on chip' und nach außen
- Definition von Netzwerk-Modulen
- Grundlegende konzeptionelle Fragen

Sollen viele hundert Zellen pro Chip gebaut werden, kann nicht jede Zelle einen Pipeline-Prozes-

sor in Digitaltechnik zur schon erwähnten Skalarprodukt-Bestimmung besitzen. Viel eleganter ist der 'analoge' Weg übers Ohmsche Gesetz und die Kirchhoff-Regeln (siehe auch [8]). Die Zelle wichtet eintreffende Spannungen mit Eingangswiderständen (synaptische Stärken!) und summiert schlicht die Ströme. Dieser Weg in Analogtechnik ist vielversprechend, hat aber seine Tücken.

Zum einen ist die VLSI-Technik noch nicht so weit, ausreichend kleine Bauteiltoleranzen zu garantieren. Werden ferner die Synapsen-Gewichte nicht digital, sondern zum Beispiel durch Widerstände repräsentiert, stellt sich gleich die Frage: Wie ändert man sie denn? Repräsentiert man sie in Form von in Kondensatoren gespeicherten Ladungen, brauchen sie Refresh - in digitalen, dynamischen RAMs ist das kein Problem, analog aber sehr schwierig. Es gibt Lösungsansätze, aber noch nicht den 'Königsweg' zum Ziel.

Eng mit der Frage der Synapsen verbunden ist das Problem, wie eine Lernregel implementiert werden soll. Nur im Ausnahmefall wird es wohl einen brauchbaren physikalischen Effekt geben, der es 'analog' schafft.

Als Konsequenz gibt es inzwischen zwar analog arbeitende Chips, aber mit starren Synapsen. Rein digitale Systeme sind in der Entwicklung, aber zurechtgestutzt auf Spezialanwendungen. Es wird ferner experimentiert mit Hybridsystemen, die vielleicht alle Vorteile in sich vereinen können.

Noch ist nicht absehbar, welche konkreten Eigenschaften ein 'neural network chip' besitzen soll. Wahrscheinlich wird es keine universelle Lösung geben können. Die erfolgreich simulierten neuronalen Strukturen sind sehr vielfältig, aber ebenso uneinheitlich - und bisher ohne Anspruch auf Allgemeingültigkeit. Hier ist noch viel Simulationsarbeit zu leisten.

Lohnt es sich denn ... ?

Wenn jetzt der Eindruck entstanden ist, daß neuronale Netzwerke das Labor noch nicht verlassen haben, so stimmt das nur zum Teil. Wie schon erwähnt, gibt es bereits Hersteller von 'Neurocomputern', deren schnelle simulierte Netze industriell relevante Probleme lösen

Lernregeln bestimmen, wie sich die vielen synaptischen Gewichte eines neuronalen Netzes im Laufe der Zeit verändern. Da in diesen Gewichten die eigentliche Funktion eines Netzes verborgen liegt, entspricht die Anwendung einer Lernregel einer Veränderung des Netzwerk-Verhaltens, also einem Programmiervorgang.

Ganz konkret sagen Lernregeln aus, wie eine Zelle i aus der ihr lokal zugänglichen (!) Information, etwa ihrer Aktivität a_i und dem Ausgangswert o_j , den ihr die Zelle j gerade sendet, eine Korrektur δ_{ij} für ein synaptisches Gewicht w_{ij} berechnen soll. Dieser Korrekturwert wird auf w_{ij} addiert, so daß im nächsten Simulationsschritt ein verändertes Gewicht wirksam ist:

$$w_{ij}(t) := w_{ij}(t-1) + \delta w_{ij}$$

Die hier angegebenen Beispiele für Lernregeln sehen vielleicht ungewohnt aus, aber wer sich von den Indizes nicht verwirren läßt, merkt schnell, daß nur ein paar Additionen und Multiplikationen dahinterstecken. Dennoch, diese unscheinbaren Formeln leisten Erstaunliches, wenn sie - in jeder Zelle und auf jede Synapse angewandt - gemeinsam tätig werden: Die Zellen beginnen nach und nach, die ihnen gestellte Aufgabe gemeinsam zu lösen.

Mathematisch bedeutet das meist, daß die synaptischen Gewichte w_{ij} gegen Grenzwerte konvergieren. Die Sicherstellung einer schnellen Konvergenz (dann lernt das Netz schnell) gegen die 'richtigen' Grenzwerte (dann tut das Netz, was es soll) ist die große Herausforderung bei der Suche nach neuen Lernregeln.

Die hier abgedruckten Beispiele sind Lernregeln, die sich bewährt haben und die jeweils bei bestimmten, hier nicht näher erläuterten Netzwerk-Typen einsetzbar sind. Ihre genauere Erklärung und die Vorführung ihrer Wirksamkeit sind aber Themen, die den Rahmen dieser Einführung überschreiten.

Regel mit Lehrer:

Diese Regel heißt auch *Delta-Regel*. Sie kommt in mehreren Varianten vor. Für bestimmte Lernsituationen gibt es Konvergenzbeweise. Sie ist sehr beliebt in vielen Anwendungen.

Regel mit Bewerter:

Hier wird angenommen, daß Synapsen nur eine gewisse Zeit 'nach Benutzung' veränderbar sind. Jede Synapse besitzt außer ihrem Gewicht w noch eine zweite zeitabhängige Größe, eben ihre Veränderbarkeit v .

Selbstorganisierende Regel:

Diese Regel ist biologisch motiviert und als *Hebb-Regel* bekannt. Eine synaptische Stärke wächst mit der Korrelation zwischen einlaufendem Signal und Aktivität der Zielzelle.

Art der Lernregel	Ref	beschreibende Gleichung(en)
Regel mit Lehrer	[7]	$\delta w_{ij} = \sigma \cdot (t_i - a_i) \cdot o_j$
Regel mit Bewerter	[1]	$\delta w_{ij} = \sigma_1 \cdot b \cdot v_{ij}$ $\delta v_{ij} = \sigma_2 \cdot v_{ij} + (1 - \sigma_2) \cdot a_i \cdot o_j$
selbst-organisierend	[7]	$\delta w_{ij} = \sigma \cdot a_i \cdot o_j$

Legende:

- δw_{ij} : Änderung des synaptischen Gewichts w_{ij} beim Lernschritt
- $\sigma, \sigma_1, \sigma_2$: Lernparameter, typisch: $0 < \sigma < 1$
- b : Bewerter-Eingabe, ein Maß für den momentanen Erfolg des Netzes
- v_{ij} : Veränderbarkeit des synaptischen Gewichts w_{ij}
- δv_{ij} : Änderung von v_{ij} beim Lernschritt
- t_i : "target" der Zelle i , d.h. die erwünschte Aktivität der Zelle i ; vom "Lehrer" stammende Information.

Lernregeln beschreiben langfristige Veränderungen der synaptischen Gewichte.

Siener Soft

Wir sind einfach schnell.

ASSEMBLER / DISASSEMBLER

Advantage Disassembler	842
Macro Assembler 5.1 (OS/2)	342
Masterkey V2.2, Disassembler	169
Cross-16 Crossassembler	379
Sourcer	319
Sourcer with BIOS Preprocessor	379

BÜCHER

Inside OS/2	99
Flying Flight Simulator	55
Microsoft Quickbasic Book	75
The Peter Norton Programmer's Guide to Windows, The official Guide	99
Appl. Prog. Guide to NetWare	199
The MS-DOS Encyclopedia	375
Running MS-DOS	99
Supercharging MS-DOS	99

DATENBANKEN

Magic PC, multi-user	1.295
Magic PC, single user	650
Paradox 2.0	1.315
Paradox 386	1.950
Paradox Network Pack 6 User	1.938
Revelation Advanced	1.595
Revelation Bump Disk, 4 Users	870
Dataflex: Preise gelten nur für MS-PC-DOS Versionen.	
Dataflex Development multi-user, V2.3b	2.490
Dataflex Development single-user, V2.3b	1.350
Dataflex Runtime, multi, V2.3b	650
Dataflex Runtime, single V2.3	456
dB Fast, dBase Compiler	295
Arbeiten mit dB Fast, deutsch c-tree	49
c-tree	890
c-tree/r-tree bundle	1.350
r-tree	890
Foxbase+ 2.0	675
Foxbase+ 2.0 Multi-User	1.149
Foxbase+ 386	1.495
Foxbase+ Multi-User Runtime Unlimited	1.295
Foxbase+ Single-User Runtime Unlimited	935
C ISAM	990
R:Base for OS/2	2.250
Clipper, Summer 87	1.195
BTrieve	595
BTrieve/Net	1.195
XQL	1.698
XTrieve	595
XTrieve Report Option	395
XTrieve/Net	1.195
XTrieve/Net Report Option	950
C Scope Look & Feel	295
Oracle for 1-2-3	456
Oracle, Developers Copy	798
CB Tree, 'C' B+tree Record Indexing Syst	295
Relational Report Writer	399
Xdb Language Interface C	1.095
Xdb Language Interface COBOL	1.095
Xdb Language Interface PASCAL	1.095
Xdb-Forms	795
Xdb-SQL	1.395

DESKTOP PUBLISHING

Pagemaker 3.0	1.750
PagePerfect	1.250
InSet 2.1.C plus	399
GoScript, Postscript Emulation	499
Manuscript 2.0	890
Halo DPE	562
Pageview	145
Art Gallery	65
Newsmaster II	185
Print Master plus	115
FontGen IV+	750
SLED	590
Ventura Publisher 1.1	1.550

EDITOREN

SPF/PC 2.0	513
EDIX	350
Personal Editor II	375
K-Edit 4.0	399
Personal REXX	375
Norton Editor	250
Brief 2.1	535
PI	499

GRAPHIK

Design CADD 2D	595
Design CADD 3D	595
Boeing Graph 4.0, (3D)	650
Timeline Graphics	296
Show Partner F/X	849
CADD 3.0, Level I	149
CADD 3.0, Level II	249
CADD 3.0, Level III	399
Generic 3D, 3-dim. Graphics	499
GSS Graphic Development Toolkit	1.285
Greenleaf Data Windows for OS/2	965
PC Storyboard plus	684
3-D Graphics for Symphony	399
Gratplus	165
MathCAD 2.0	779
Halo 88, für alle Microsoft Compiler	798
Halo 88, mit 1 Library (Compiler angeben)	627
Grafmatic	369
Omniplot	675
Piotmatic	369
Printmatic	369
In-a-Vision	1.045
Chart 3.0	935
Graph-in-the-Box 2.0	249
OCAD PCB Layout	4.500
OCAD STD-III	1.695
OCAD VST (Visual Simulation)	3.750
Grasp	169
Pixie	798
Concorde Total Graphics Pack	1.650
Harvard Graphics, Version 2.1	995

HARDWARE

Copy II PC Option Board deluxe	395
Everoom 24, 2400 baud, internal Modem	499
Maynstream 60 MB, Cartridge System	3.750
Maynstream 60 MB, Cassette System	2.950
Maynstream 125 MB, Cartridge System	4.350
Maynstream Adapter für PS/2	550
Maynstream Adapter für XT/AT	695
Microsoft Bus Mouse	350
Microsoft serial Mouse, mit Windows	399
DC 600 DXL Bandkassette	150
DC 600A Bandkassette	85
Acoustic Cups für Worldport	149
Worldport 1200	475
Worldport 2400 mit Carbon Copy plus 5.0	850
Dataphon s21-23d mit Netzteil	295

KALKULATION

Quattro	399
Timeline	795
The Baler 1-2-3 Compiler	1.195
Trans for 1-2-3, konvertiert in ASCII	1.195
XYZ: Consolidate across Multible Worksh	795
XYZ: Model, Modeling Language for Symp	795
XYZ: Query across Multi Worksheets	795
XYZ: Spread, move across Symphony	795
Agenda	1.195
Freelance plus	795
Lotus 1-2-3, Version 2.01	912
Symphony 2.0	1.495
Excel 2.0	975
Excel Network User Pack	650
Works	495
Luoid 3-D	456
Harvard Total Project Manager	1.199
Password! for Symphony	195

KOMMUNIKATION

Procomm plus	249
PC Anywhere III	249
Carbon Copy plus 5.0	456
CC Express, automatische Dateiübertragung	1.195
Crosstalk Mk. 4	425
Smarterm 240	695
MIRROR II	195
Lap Link	295
Lap Link Mac	349
Lap Link plus	349
The Brooklyn Bridge 2.0 mit File-Manager	349
The Brooklyn Bridge für Victor 9000	550

NETZWERK-PRODUKTE

NTNX286, Novell-kompatibles OS	1.495
NTNX, Novell-kompatibles OS	1.095

PC Slave 16N - Multiprocessor Netz Karte	1.495
PC Slave 286 - Multiprocessor Netz Karte	2.250
LAN Administrator Kit	899
NetManager, The LAN Support Center	3.500
10-Net, Card	1.595
10-Net, Card with Operating System	1.495
PS/2 Ethernet Card für Microchannel	1.100
QDOS Network, 4 user	675
NCA Network Communication Adapter	570
NCA Server Software	1.350
LAN Shell Menu System	1.695
PC-110 Arcnet-Karte, Stern	456
PC-210 Arcnet-Karte, Bus	595
PC-250 Arcnet-Karte, twisted-pair	495
PC-500-PS, Arcnet-Karte, 16 Mbit, 16-bit	1.695
PC-500-WS, Arcnet-Karte, 16 Mbit, 8-bit	1.395
Lancard/E, Ethernet Karte	695
PS/2 Microchannel Arcnet-Board	1.100
Ethercard	798
Ethercard plus	995

NOVELL

Händleranfragen für Netzwerk-Produkte willkommen
Wir bieten Unterstützung für alle Produkte rund um Novell-Netze

Advanced 286 V2.11	6.829
Boot-Prom für Arcnet-Karte	137
Boot-Prom für Ethernet-Karte	137
ELS Level I (4 User)	1.596
ELS Level II (8 User)	
NACS	1.596
NetWare User Reference Manuals Kit	399
Novell Diskcoprocessor	1.095
Novell Keypad	275
SFT 286 V2.11	9.990
NE1000 Ethernet Karte	895
NE2000 Ethernet Karte	
NP600 Ethernet plus Karte	1.675

OBJECT ORIENTED PROGRAMMING

C_talk, object oriented C	499
Advantage C++	1.395
PforC++	679
Smalltalk/V	342
Smalltalk/V 286	570
Smalltalk/V Communications	198
Smalltalk/V EGA/VGA Support	198
Smalltalk/V Goodies II	198
Smalltalk/V Goodies III	198
Actor	1.250

OPERATING SYSTEMS

VM/386, Multitasking für 80386-Rechner	484
OS/2 Programmer's Toolkit V1.0	798
PC-MOS 386, 25 User	2.699
PC-MOS 386, 5 User	1.698
PC/NX	269
PC/VMS	269
Wendin DOS	269
Wendin DOS Application Development Kit	299

PASCAL

Turbo Asynch plus 4.0	345
Turbo Power Tools plus	313
Turbo Pascal 4.0	219
Turbo Pascal Database 4.0	219
Turbo Pascal Editor 4.0	219
Turbo Pascal Graphix 4.0	219
Turbo Pascal Numerical Methods 4.0	219
Pascal 4.0 mit OS/2 Support	684
Norton Guides Pascal	169
Norton Guides Pascal & Assembly	299
Science & Engineering Tools	299
Overlay Manager 4.0	149
TDebugPLUS	99
TDebugPLUS with Source	149
Turbo Analyst 4.0	169
Turbo Professional 4.0	199

PROGRAMMIERSPRACHEN

C TOOLS PLUS/5.0	465
Turbo C TOOLS	465
Turbo Basic 1.1	199
Turbo Basic Database Toolbox	219
Turbo C 1.5	199
Turbo Prolog 2.0	325
Turbo Prolog Toolbox	219
dBC III plus mit Source code	2.750
Lattice C Compiler	842
Basic Compiler 6.0 mit OS/2 Support	684
C Compiler 5.1 mit OS/2 Support	855
Cobol 3.0, (ehemals Microfocus COBOL)	2.052
Fortran Compiler 4.1 mit OS/2 Support	855
muMATH/muSIMP, V4.12	705
Pascal 4.0 mit OS/2 Support	684
Quick C	219
QuickBASIC 4.0	219
Real/CICS	2.850
Realia Cobol	2.850
Realia Cobol with Realmenu	3.150
RM Fortran, Version 2.42, mit Plink	1.309
Janus ADA Compiler "D" Pak	350
Janus ADA Compiler "D" Pak	2.508
Janus ADA Compiler "ED" Pak	1.254

STATISTIK

Statistix	650
BASS System (SAS clone) mit BassStat	1.095
SEGS Scientific & Engineering Graphics S	699
NCSS+	250
NCSS+ Graphics	450
Statpro	2.250
SPSS Data Entry Module	1.095
SPSS Graphics Module	1.095
SPSS Mapping Module	1.095
SPSS Tables	1.095
SPSS Trend Module	1.095
SPSS/PC Plus Version 2.0	2.590
CSS Complete Statistical System	1.350
Stats+	550
Statgraphics	1.482

Systat	1.795
Systat mit Graphic	2.250
StatPlan III, für rapid analysis of Data	690
RATS, Regression Analysis of Time Series	995
Statpac Gold	1.395

UTILITIES

Pizazz	125
Pizazz Plus	249
Free Space	175
ORG Plus	300
Eureka! The Solver	359
Sidekick plus	435
1DirPlus	169
Fastrax Diskoptimizer	125
Memory Mate	162
Homebase 2.5	265
Copy II PC	88
Copy II PC Option Board deluxe	399
PC Tools de Luxe	159
Clear for C	
Clear for dBase	295
Clear for dBase (HP-Laserjet)	450
Super Project Expert/2, OS/2 Support	1.995
SuperProject Plus 2.0	1.185
VCScreen	295
Vitamin C	590
Take Charge!	275
XTree	135
XTree PRO	219
Fastback plus	359
Fill & File	395
Sideways	185
QDOS II	149
Vopt	199
V-EMM, LIM 4.0 Emulation	342
Interactive EasyFlow	456
Fanci Console	249
Jet Plotter	285
LaserControl	399
LaserPlotter	325
Matrix Plotter (Epson oder Toshiba)	278
Bookmark	259
Bookmark plus	350
Advanced Hard Disk Diag. 2.0	189
LIMSIM, LIM 4.0 Simulation	250
Panel plus for OS/2	1.350
Tornado Notes	249
Diff-E-Q, Differential Equation Solver	1.254
Goter	165
Super PC-Kwik	249
Flowcharting II plus	499
Mace Utilities 4.10C	219
Periscope	499
SeeMore for 1-2-3	169
SeeMore for Symphony	169
Smart Notes	249

Norton Commander	149
Norton Guides Assembly	185
Norton Guides Basic	185
Norton Guides C	185
Norton Guides C & Assembly	149
Norton Guides OS/2	249
Norton Guides Pascal	249
Norton Guides Pascal & Assembly	249
Norton Utilities Advanced Edition	285
Corporate PVCs	969
Network PVCs	3.176
Personal PVCs	450
Disk Technician	199
Disk Technician+	349
Copy Write, April 88	125
PrintQ, Professional Print Spooling Soft	199
Flash up	159
Flash up Toolbox	139
Flashcode	250
Screen Sculptor	257
Flash 5.6	185
Speedstor, universal	119
Say What!	115
Masterkey Unlock	405
SQZ! plus	328
The UI Programmer	982
Pathfinder 4.0	169

WINDOWES

Windows Express (ehemals Clickstart)	395
Windows Convert	250
Windows Designer	1.585
Windows Draw	705
Windows Graph	875
Windows Graph	848
Windows 286	249
Windows 2.03 Development Toolkit	1.262
Windows 386	449
Windows Filter	399
Windows Intalk	399
WinText	895
WinTime	395
Desqview 2.01	295
Desqview Companions I	219
QEMM 50/60	149
QEMM-386, Vers 4.1	149
Windows for C, OS/2 Support	1.100
Windows for C, OS/2, mit Source code	2.250
PC Paintbrush for Windows	295
Publisher's Pictures Paks, each	250

WORDPROCESSING

Sprint	499
DisplayWrite IV	950
Word-for-Word	275
Wordstar 2000 Plus Rel. 3	695
Wordstar Pro LAN Pack	995
Wordstar Professional	695
Network Word Master Pack V4.0	1.653
Network Word Node V4.0	399
Word 4.0, single user	695
IZE	1.150
FlightWriter	195
Ask Sam	950
Q & A, 3.0	890
Software Bridge	342
Wordperfect 5.0	695

Alle Produkte werden als Originallizenzen geliefert.

Siener Soft

Ein Service der Firma:
Roland Siener Computerhandel

Bachmayerstraße 2

6200 Wiesbaden

Telefax 06121/402502

06121/404001

Ladenverkauf von 9.00 bis 18.00 Uhr

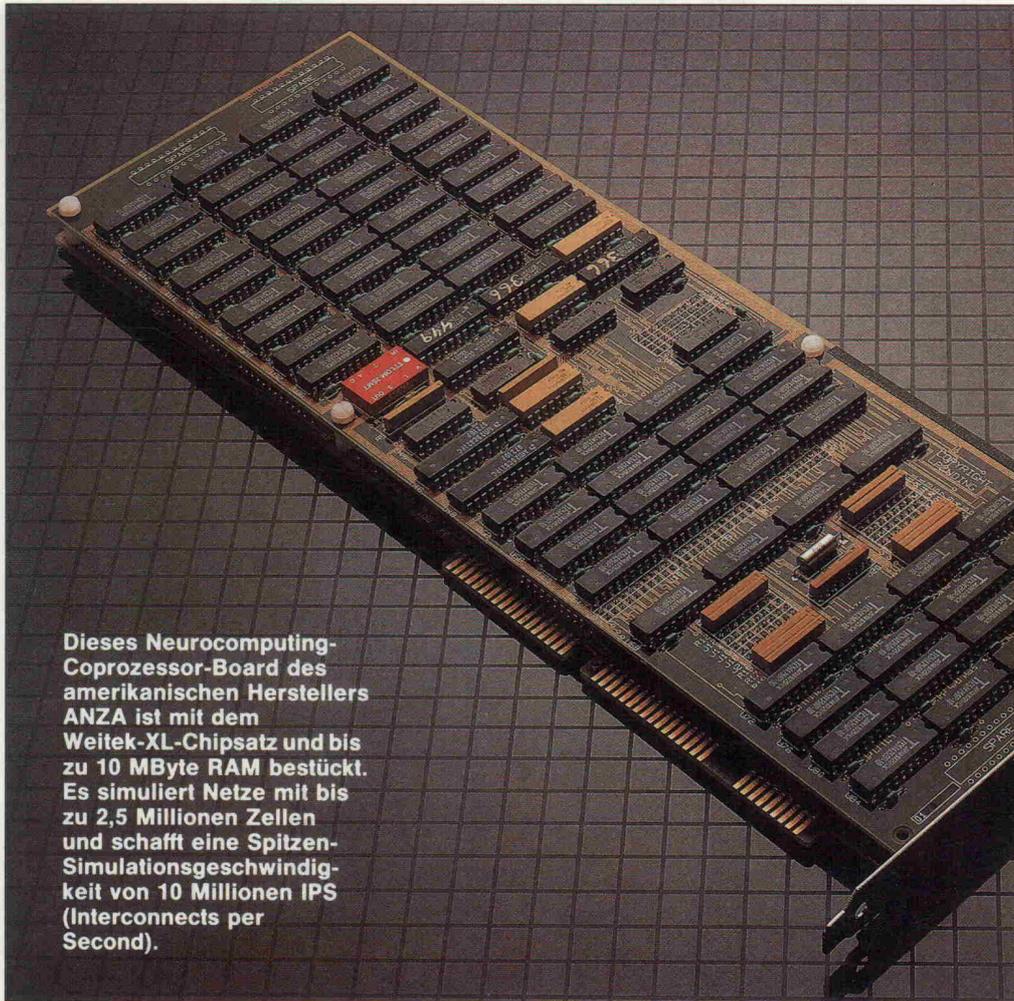
Lieferbedingungen

Alle Preise enthalten
Versandkosten. Bei Zahlung
per Vorkasse können Sie je
Produkt DM 4,- vom Preis
abziehen. Preisänderungen
und Irrtum vorbehalten.

Achtung!

Alle Modems außer
Dataphon ohne FTZ-
Zulassung, Betrieb in der
BRD verboten.

DISCOUNT SERVICE IMPROVING SOFTWARE



Dieses Neurocomputing-Coprocessor-Board des amerikanischen Herstellers ANZA ist mit dem Weitek-XL-Chipsatz und bis zu 10 MByte RAM bestückt. Es simuliert Netze mit bis zu 2,5 Millionen Zellen und schafft eine Spitzen-Simulationsgeschwindigkeit von 10 Millionen IPS (Interconnects per Second).

– und davon können die Hersteller schon leben. Der Parallelbeitrag in diesem Heft, 'Assoziative Speicher', zeigt bereits eine praktisch einsetzbare Software-Implementation.

Den wahren Forscher interessieren solch profane Gesichtspunkte nur am Rande, das Gebiet Neurocomputer steckt noch mitten in der Grundlagenforschung. Besonders faszinierend ist die Geschichte von 'NETtalk' – dem Netz, das 'sprechen' lernte [9]:

T. Sejnowski und C. Rosenberg nutzten ein paar Wochen des

Sommers 1986, um ein neuronales Netz auf einer VAX 11/750 in der Sprache C zu simulieren. Sie benutzten ein zur Zeit sehr erfolgreiches Lernverfahren namens 'back-propagation', um ihrem Netz den Zusammenhang zwischen geschriebenen englischen Wörtern und ihrer Aussprache, also ihren Phonemketten, beizubringen. Dazu fütterten sie ihr Netz mit einer Liste von einigen tausend Wörter-Beispielen und ihren Phonemketten (wie aus dem Wörterbuch) und hörten sich die vom Netz erzeugten Phonemketten via Voicebox-Ausgabe an.

Nach einem Listen-Durchlauf hörte sich das Netz an wie ein Säugling, der noch nicht sprechen kann – es lallte geradezu. Eine wiederholte Eingabe derselben Liste sorgte für deutliche Verbesserung. Bei erneuten Listen-Durchläufen wurde die Aussprache immer verständlicher, bis nach circa 50 Durchläufen kaum noch Fehler auftraten, und wenn, dann meist solche, die nicht völlig danebenlagen (etwa stimmloses mit stimmhaftem 'th' verwechselt). Nun kam die Nagelprobe: Eine Liste unbekannter Wörter wurde eingegeben. Das Netz sprach sie mit höherer Fehlerate, aber völlig verständlich aus!

Was war passiert? Nicht die Beispiele, sondern die versteckt in den Beispielen vorhandenen Ausspracheregeln hat das Netz gelernt – deshalb konnte es verallgemeinern. Bedenkt man nun, daß ein konventionelles Aussprache-Programm (wie DECTalk von Digital), das mehr als ein Mannjahr Entwicklungs-

zeit kostete und sich auf das geballte Wissen der Linguisten stützt, kaum bessere Ergebnisse erbringt, erkennt man die Leistungsfähigkeit neuronaler Netzwerke – und ihre Faszination. (be)

Literatur

[1] Barto, A. G., Sutton, R. S. and Anderson, C. W.: Neuronlike Adaptive Elements That Can Solve Difficult, Learning Control Problems, 1983, IEEE SMC-13: 834-846

[2] Eckmiller, R., v. d. Malsburg, C. (Hrsg.): Neural Computers, Springer Verlag, Heidelberg 1988

[3] Fukushima, K.: Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position, Biol. Cybern. 36 (1980), 193-202

[4] Fukushima, K.: Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition, Neural Networks 1 (1988): 119-130

[5] Goddard, N.: The Rochester Connectionist Simulator, Vol. 1: User Manual. Dept. of Computer Science, University of Rochester, Rochester, 1987, NY 14625, U.S.A.

[6] McCulloch, W. S., Pitts, W.: A Logical Calculus of the Ideas Immanent in Nervous Activity, Bull. Math. Biophys. 5 (1943): 115-133

[7] D. E. Rumelhart, J. L. McClelland: 'Parallel Distributed Processing', Vol. 1 & 2 (1986), Vol. 3 (1988), MIT Press

[8] Schreiber, S. B.: Natürliche Intelligenz. Teil 1: Neuronen und Synapsen – alles nur ein organischer Computer?, c't 4/87: 98-101

[9] Sejnowski, T. J., Rosenberg, C. R.: NETtalk: A Parallel Network that Learns to Read Aloud, Technical Report JHU/EECS-86/01, Johns Hopkins University, 1986, Baltimore



- Aktuelle Forschungsthemen**
- Sehen und Bildverarbeitung
 - Sprechen und Sprache
 - Sensor-Motorik und Robotik
 - Mustererkennung
 - Assoziatives Lernen
 - Selbst-Organisation
 - Kognitive Informationsverarbeitung
 - Neurobiologie lokaler Verschaltungen
 - Netzwerk-Dynamik
 - VLSI-Implementierungen
 - Optische Implementierungen
 - Neurocomputer
 - Anwendungen