Neuronale Netze und maschinelles Lernen

Wolfgang Maass

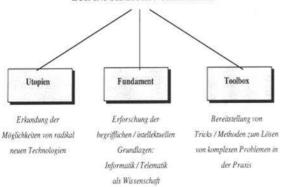
Institut für Grundlagen der Informationsverarbeitung, Technische Universität Graz, Klosterwiesgasse 32/2, A-8010 Graz, Austria, e-mail: maass@igi.tugraz.ac.at



o. Univ.-Prof. Dipl.-Math.
Dr. rer. nat.
Vorstand des Institutes für Grundlagen der Informationsverarbeitung der Technischen Universität Graz, Studium der Mathematik an der Ludwig-Maximilians-Universität in München,
Diplom 1973,
Promotion zum Dr. rer. nat 1974

Die Theoretische Informatik leistet in drei verschiedenen Bereichen einen Beitrag zur Forschung und Ausbildung in Informatik / Telematik an unserer Universität:

BEITRÄGE DER THEORETISCHEN INFORMATIK ZUR INFORMATIK / TELEMATIK



Damit sich der Leser ein genaueres Bild von unserer Arbeit verschaffen kann, möchte ich hier kurz zwei Beispiele für gegenwärtige Forschungsprojekte unseres Instituts vorstellen. Diese Projekte stellen Beispiele für unsere recht vielfältige Arbeit in den Bereichen "Neuronale Netzwerke" und "Maschinelles Lernen" dar, auf die die Referenzen am Ende dieses Artikels verweisen. * Aus Gründen der Platzersparnis kann ich in den Referenzen nur Arbeiten nach 1994 anführen.

Zusätzlich zu diesen beiden Schwerpunkten gibt es an unserem Institut eine Arbeitsgruppe in den Bereichen geome-

* Mit Unterstützung durch das Bundesministerium für ESPRIT Projekt NeuroCOLT der Europäischen Union. trische und kombinatorische Algorithmen (Prof. Aurenhammer).

1. Beispiel: INFORMATIONSVERARBEITUNG IN BIOLOGISCHEN NEURONALEN NETZWERKEN

Aktuelle Forschungsprobleme:

- Wie rechnet/kommuniziert ein Netzwerk von Neuronen in einem lebenden Organismus?
- Kann man Tricks/Methoden der Informationsverarbeitung in biologischen Neuronalen Netzwerken zur Entwicklung von leistungsfähigeren Systemen der Informatik/Telematik benutzen?

Es ist inzwischen bekannt, daß sich die Informationsverarbeitung in biologischen Neuronalen Netzwerken in radikaler Weise von der in unseren gegenwärtigen Computern unterscheidet. Das ist auch für die Technik von Interesse, weil die biologischen Systeme unseren künstlichen Rechnern bisher noch in einigen wichtigen Bereichen (z.B. Sprachverarbeitung, Bildverarbeitung, Bewegungssteuerung) weit überlegen sind. Zusätzlich sind biologische Neuronale Netzwerke die bisher in der Praxis erfolgreichsten Beispiele für massiv parallele Rechner.

Der wichtigste Parameter zur Beschreibung des gegenwärtigen Zustandes eines biologischen Neurons v ist das elektrische Potential in dessen Zellkörper (im Ruhezustand ca. -70 mV). Ein Neuron v erhält als Input von anderen Neuronen eine sehr große Anzahl (bis zu 10.000) von positiven und negativen Potentialänderungen, die EPSP's (excitatory postsynaptic potentials) bzw. IPSP's (inhibitory postsynaptic potentials) genannt werden. Die einzelnen Potentialänderungen haben den in Abb. 1 gezeigten zeitlichen Verlauf.

Diese EPSP's und IPSP's sind das Resultat des "Feuerns" von anderen Neuronen, die über eine Synapse mit dem Neuron ν verbunden sind. Diese zu verschiedenen Zeiten eintreffenden EPSP's und IPSP's überlagern sich am Zellkörper des Neurons ν , wobei genau gleichzeitig eintreffende EPSP's und IPSP's dazu neigen, sich gegenseitig aufzuheben. Vereinfacht kann man sagen, daß ein Neuron ν "feuert", sobald das elektrische Potential im Zellkörper von ν , das sich durch die Überlagerung dieser EPSP's und IPSP's ergibt, eine gewisse Schwelle (-50 mV) überschreitet. Sobald Neuron ν "feuert", wandert ein Spannungsimpuls ("Aktionspotential" oder "spike") von ca. 1-2 msec Dauer vom Zellkörper des Neurons ν weg entlang einer oft recht langen Faser ("Axon"). An den Enden des in der Regel stark verzweigten Axons löst solch ein Aktionspotential mittels "Synapsen" ein EPSP oder IPSP in anderen Neuronen aus.

Diese hier kurz angedeutete Arbeitsweise eines biologischen Neurons kann man recht gut mit dem mathematischen Modell eines spiking neurons beschreiben ([2], [3]). Berücksichtigt man noch verschiedene "noise"-Faktoren, die diese Arbeitsweise beeinflussen, gelangt man zum Modell eines noisy spiking neurons ([5]).

Man hatte anfänglich gemeint, daß es genügt, die **Häufigkeit** des Feuerns ("Feuerrate") eines Neurons v in einem gewissen Zeit-

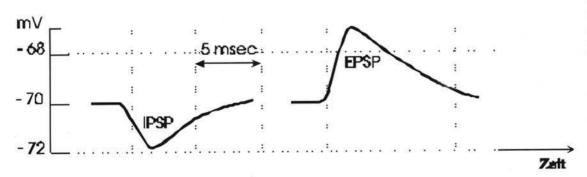


Abb. 1: Zeitlicher Verlauf der Potentialänderungen

intervall zu betrachten, um seine Rolle in der Informationsverarbeitung in einem biologischen Neuronalen Netz zu verstehen. Feinere experimentelle Methoden in der Neurobiologie haben aber in den letzten Jahren gezeigt, daß diese Annahme für viele Bereiche der Kortex nicht zutrifft. Insbesondere feuern die Neuronen in den höheren Bereichen der Kortex relativ selten (meist bedeutend weniger als 100 mal in der Sekunde) und in sehr unregelmäßigen Abständen.

Die beiden fetten Striche deuten zum Vergleich die Länge eines Zeitraums von 150 msec an. Es ist bekannt, daß ein biologisches Neuronales Netz in der Lage ist, in einem Zeitraum dieser Länge schwierige Aufgaben der Informationsverarbeitung durchzuführen. Dieser Vergleich zeigt, daß eine Informationsübertragung mit-

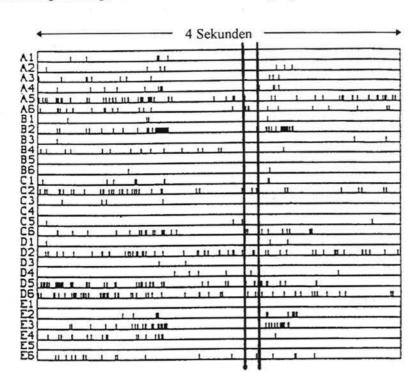


Abb. 2: Typisches Protokoll der Feuerzeiten von 30 Neuronen in der Kortex: Für jedes der 30 Neuronen ist der Zeitpunkt des Feuerns durch einen Strich angedeutet. Krüger und Aiple, 1988

tels Feuerraten viel zu langsam wäre, um einem Netzwerk von Neuronen komplexe Berechnungen in nur 150 msec zu ermöglichen. Ferner hat man nachgewiesen, daß bei einer Wiederholung derselben mentalen Aufgabe gewisse sehr präzise zeitliche Muster im Feuern einzelner Neuronen wie-

derholt werden (siehe [1]).

Andererseits ist auch bekannt, daß die meisten biologischen Neuronalen Netzwerke nicht mit einer präzisen universellen "clock" arbeiten, die es ihnen ermöglichen würde, das Auftreten bzw. Nichtauftreten eines Aktionspotentials in einem kurzen Zeitfenster als einen digitalen Output 1 bzw. 0 eines Neurons v zu verstehen (vergleichbar mit der Arbeitsweise eines digitalen Rechners). Daher neigt man in den letzten Jahren dazu, Neuronen in höheren Bereichen der Kortex als komplexe analoge Prozessoren aufzufassen, die analoge Variablen in der Form von Zeitdifferenzen zwischen ihrem Feuern übertragen. Ferner spielen kurzfristige Synchronisationen zwischen wechselnden Teilgruppen des Netzwerkes für deren Informationsverarbeitung eine wichtige Rolle. Daher erfordert die Analyse der Informationsverarbeitung in biologischen Neuronalen Netzwerken, daß es uns gelingt, einen Typ von "Rechner" zu verstehen, der sich in radikaler Weise sowohl von allen herkömmlichen theoretischen und praktischen

Rechnertypen unterscheidet als auch von den herkömmlichen Modellen für "künstliche Neuronale Netzwerke".

Es stellt sich zum Beispiel die konkrete Frage, welche Funktionen $F: [0,1]^n \to [0,1]^k$ ein Netzwerk von biologischen Neuronen im Prinzip berechnen kann, falls dessen Input und Output durch "Zeit-Kodierung" in der Form von "delays" von Feuerzeiten von n Input-Neuronen bzw. k Output-Neuronen kodiert sind (wobei sich der "delay" im Vergleich der Feuerzeit dieser Neuronen mit der eines "Referenz-Neurons" ergibt, das zur Zeit "0" und "T" feuert).

Wir haben an unserem Institut nachgewiesen ([6]), daß in der Tat jede (!) stetige und beschränkte Funktion durch eine von einem Netzwerk von noisy spiking neurons in dieser Art berechnete Funktion F beliebig genau approximiert werden kann. Daher kann man Netzwerke von spiking neurons mit Zeit-Kodierung als universelle analoge Rechner betrachten. Dabei ist die erforderliche Rechenzeit unabhän-

gig von der Komplexität der gegebenen Funktion F und der verlangten Genauigkeit der Berechnung (sie beträgt » 15 msec für realistische Werte der Parameter für ein biologisches Neuron).

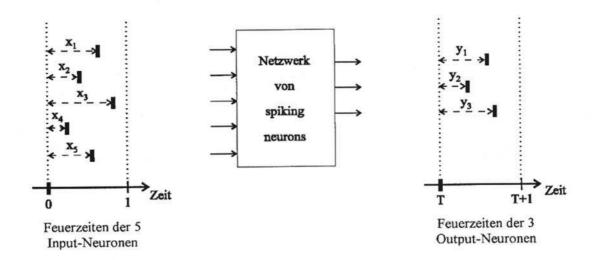


Abb. 3: Paralleles analoges Berechnen einer Funktion $F: [0,1]^5 \rightarrow [0,1]^3$ mit spiking neurons mittels "Zeit-Kodierung":

$$F(x_1, x_2, x_3, x_4, x_5) = (y_1, y_2, y_3)$$

Es stellt sich natürlich auch die Frage, wie viele Neuronen jeweils für die Berechnung einer solchen Funktion F notwendig sind. Für eine solche Komplexitätsanalyse ist es zweckmäßig, die benötigte Anzahl von spiking neurons mit der Anzahl der Prozessoren bzw. Gatter in "herkömmlichen" Modellen für digitale oder analoge Rechner zu vergleichen, die dieselbe Funktion F berechnen. In der folgenden Kurzdarstellung der bisher bekannten Ergebnisse enthält die Spalte ganz rechts insbesondere einen Vergleich mit der PRAM (parallel random access machine), dem in der Informatik am häufigsten betrachteten idealisierten Modell für einen massiv parallelen Rechner. Der Begriff "polynomial size" bedeutet hier, daß die Anzahl der benötigten spiking neurons durch ein Polynom in der Größe der "control" einer Turing machine, bzw. der Anzahl der Gatter eines sigmoidal neural nets oder einer PRAM (dargestellt als Schaltkreis) abgeschätzt werden kann. Die hier vorgestellten Resultate zeigen, daß die Rechenstärke von Netzwerken von spiking neurons mit Zeit-Kodierung mindestens so groß ist, wie die einer Reihe etwa gleich großer herkömmlicher Rechnermodelle. In unserer gegenwärtigen Forschungsarbeit (siehe [6]) weisen wir nach, daß sie für einige Typen von Berechnungen sogar weitaus größer ist.

In den einzelnen Zeilen der Tabelle 1 werden Ergebnisse für die (Mindest-)Rechenstärke von Netzwerken von biologischen Neuronen wiedergegeben, unter jeweils etwas anderen Annahmen über die Gestalt der EPSP's sowie die Art des noise, dem die Neuronen unterliegen. Durch "Runden" des mittels Zeit-Kodierung übertragenen Input und Output kann ein Netzwerk von spiking neurons im Prinzip auch mit **digitalem** Input und Output arbeiten.

Für Details dieser Ergebnisse verweisen wir auf die Publikationen [2], [3], [4], [5], [6], [9]. Die unausgefüllten Felder in der Tabelle weisen auf noch offene Forschungsprobleme hin.

In der ersten Zeile der Tabelle 1 ("rectangular EPSP's") haben wir Ergebnisse angeführt, die sich auf die in den letzten Jahren entwickelten Silicon-Modelle für Netzwerke von biologischen Neuronen ("pulse-stream VLSI", siehe [7]) beziehen, bei denen man die in der Natur auftretenden EPSP's und IPSP's durch die in dieser Technik naheliegenderen Rechteckimpulse ersetzt. Es haben sich bisher schon vielversprechende Anwendungen solcher neuartiger Systeme im Bereich der Inter-Chip Kommunikation für analoge VLSI-Chips ergeben, bei der es (ähnlich wie in der Natur) darum geht, analoge Werte sehr schnell in fehlertoleranter Weise über eine größere Entfernung zu übermitteln (siehe [8]). Das "no" in der dritten Spalte der obigen Tabelle enthält das unerwartete Ergebnis, daß die theoretische Rechenstärke eines Systems mit Rechteckimpulsen geringer ist als die eines gleich großen Systems, das mit EPSP's und IPSP's arbeitet, deren Gestalt denen von biologischen Systemen ähnelt (siehe die Zeile darunter in der obigen Tabelle, bei der sich das Modell nur durch die Form der EPSP's und IPSP's unterscheidet). Für Details verweisen wir auf [9].

Die hier angeführten Untersuchungen machen deutlich, daß wir am Beginn einer spannenden neuen Entwicklung stehen: Der gegenseitigen Durchdringung der bisher rein technisch orientierten Disziplinen Informatik und Telematik mit der gegenwärtig rasant wachsenden neuen Disziplin der Computational Neuroscience. Im Rahmen dieser Entwicklung werden die Informatik und Telematik zum einen ein neues naturwissenschaftliches Anwendungsfeld erhalten, bei dem sie ihre vielfältigen Techniken dazu benutzen können, um neue Kenntnisse über die Informationsverarbeitung und Kommunikation in biologischen Neuronalen Netzwerken zu gewinnen. Zum anderen wird das Entschlüsseln der Arbeitsweise dieser biologischen Systeme neue Anregungen zur Entwicklung von leistungsfähigeren (insbesondere massiv parallelen, fehlertoleranten und lernfähigen) Systemen in verschiedenen Anwendungsbereichen der Informatik und Telematik liefern.

Ein Aspekt, der biologische informationsverarbeitende Systeme zur Informationsverarbeitung unseren künstlichen Systemen weit überlegen macht, ist die Fähigkeit der biologischen Systeme, ohne

A network of spiking neurons of the type specified below, with	can compute any boolean function	can approximate any bounded continuous function	can simulate in real-time with a network of polynomial size:		
			Turing machines	sigmoidal neural nets (approximatio n only)	PRAM's
network input/output in time-coding, and	digital	analog	digital	analog	digital
rectangular EPSP's ("pulse stream VLSSI"), no noise	✓	1	no		1
EPSP's with linear increasing and decreasing segments, no noise	1	1	1	✓	✓
EPSP's with linearly increasing initial segment, with noise	√	1	4	1	1
no significant conditions on shape of EPSP's, with noise and persi- stent firing failures	1	1			✓

Tab. 1

ein genaues "Programm" auszukommen und ihre Arbeitsweise durch "Lernen aus Beispielen" selbständig zu verbessern. Dies führt uns zum zweiten Forschungsbereich, Maschinelles Lernen, den ich ebenfalls anhand eines Beispiels kurz vorstellen möchte:

2. Beispiel: MASCHINELLES LERNEN MIT ENTSCHEIDUNGSBÄUMEN

Forschungsprobleme im Maschinellen Lernen, an denen in unserem Institut gearbeitet wird:

- Kann man den Begriff des "Lernens" genauso exakt mathematisch charakterisieren wie den Begriff des "Rechnens"?
- Kann man Methoden des Lernens in biologischen Neuronalen Netzwerken für künstliche Neuronale Netzwerke (d.h. Computer Modelle) adaptieren?
- Inwieweit ist es für Computer möglich, aus Beispielen zu lernen?
- Gibt es für Maschinen prinzipielle Grenzen für ihre "Lernfähigkeit"?

Für einige Ergebnisse zu diesen Fragen verweise ich auf [10], [11], [12], [13], [14], [15], [16], [17], [18].

Maschinelles Lernen ermöglicht eine radikal neue Betrachtungsweise von Computern und Computersoftware. Es erlaubt uns, anstelle eines vollständigen Programmes der Maschine nur gewisse Grundzüge des intendierten Programmes mitzugeben, sowie zusätzlich einen Lernalgorithmus, der die Maschine in die Lage versetzt, die Einzelheiten des intendierten Programmes selbständig aus Beispielen zu erschließen.

Es ist offensichtlich, daß Maschinelles Lernen den Einsatzbereich von Computern stark erweitert, weil es den Computer in die Lage versetzt, dem Benutzer auch bei sehr komplexen Anwendungsproblemen nützliche Ratschläge zu erteilen, für die noch keine gute Theorie (und daher auch kein gutes Programm) existiert. Daneben öffnet Maschinelles Lernen auch eine neue Perspektive für die Entwicklung und den Einsatz von extrem schnellen Rechnern mit einer großen Anzahl von parallelen Prozessoren. Der bisher überraschend geringe Markterfolg solcher Rechner ist nicht zuletzt der Tatsache zuzuschreiben, daß die Software für jeden dieser Rechner umgeschrieben werden muß. Zusätzlich wachsen mit der Anzahl der Prozessoren auch die Fehlermöglichkeiten in Software und Hardware. In Anlehnung an biologische Modelle hat sich nun die neue Möglichkeit ergeben, einen parallelen Rechner ebenfalls nur mit den Grundzügen eines Programmes sowie

einem Lernalgorithmus auszustatten, der es dem Rechner ermöglicht, aus Beispielen ein vollständig spezifiziertes Programm zu erlernen, das auf das konkrete Anwendungsproblem und die gegenwärtige konkrete Architektur des Rechners (die sich im Laufe der Zeit ändert, wenn z. B. einer der Prozessoren ausfällt) abgestimmt ist. Dies klingt sehr utopisch, aber es ist in der Form von künstlichen Neuronalen Netzen (simuliert in Software, oder speziellen massiv parallelen Computer Chips) schon Teil unserer Realität geworden (siehe z.B. [7], [19]). Im Bereich der Theorie ist der Entwurf von effizienten Lernalgorithmen und die Analyse der Anzahl der benötigten Trainingsbeispiele für künstliche Neuronale Netze eine besonders interessante Herausforderung für die Theoretische Informatik geworden (siehe z. B. [19], [20], [21], [22], [23], [24], [25], [26]).

Abgesehen von künstlichen Neuronalen Netzen gibt es aber auch viele andere Zugänge zum Maschinellen Lernen, insbesondere mit Methoden der Künstlichen Intelligenz. Ich möchte hier einen solchen alternativen Ansatz kurz an einem konkreten Beispiel erläutern. Wir betrachten ein Anwendungsproblem, das (wie viele andere) von der Universität von Irvine (USA) gespeichert wurde, und dort, wie Dutzende andere "Benchmark-Lernprobleme", für jedermann zugänglich ist (FTP site ics.uci.edu, directory pub/machine-learning-databases). Es handelt sich hier um ein Klassifikationsproblem aus der Medizin: der Diagnose einer spezifischen Krankheit (pima-indians diabetes). In der Datenbank hat man Zahlenreihen gespeichert, die für 768 Personen jeweils 8 Meßdaten sowie die (im Nachhinein) richtige Diagnose angeben.

Ein konkretes "real-world-Problem" des Maschinellen Lernens:

Datensatz PI (pima-indians diabetes): 768 Fälle

6,148,72,35,0,33.6,0.627,50,1 1,85,66,29,0,26.6,0.351,31,0 8,183,64,0,0,23.3,0.672,32,1 1,89,66,23,94,28.1,0.167,21,0 0,137,40,35,168,43.1,2.288,33,1 5,116,74,0,0,25.6,0.201,30,0 3,78,50,32,88,31.0,0.248,26,1 10,115,0,0,0,35.3,0.134,29,0 2,197,70,45,543,30.5,0.158,53,1 8,125,96,0,0,0.0,0.232,54,1 4,110,92,0,0,37.6,0.191,30,0 10,168,74,0,0,38.0,0.537,34,1 10,139,80,0,0,27.1,1.441,57,0 1,189,60,23,846,30.1,0.398,59,1 5,166,72,19,175,25.8,0.587,51,1 7,100,0,0,0,30.0,0.484,32,1 0,118,84,47,230,45.8,0.551,31,1 7,107,74,0,0,29.6,0.254,31,1 1,103,30,38,83,43.3,0.183,33,0 1,115,70,30,96,34.6,0.529,32,1 3,126,88,41,235,39.3,0.704,27,0 usw.

Die letzte Zahl (0 oder 1) jedes "Falles" gibt die richtige Diagnose an.

Jeder Fall wird durch <u>8 Meßwerte</u> ("attributes", oder "features" genannt) charakterisiert:

Npregnant: continuous Plasma_glucose: continuous

Diastolic_bllod_pressure: continuous

Skin_thickness: continuous

Insulin: continuous Mass: continuous Pedigree: continuous Age: continuous

Ziel:

Vorhersage der richtigen Diagnose für einen neuen Fall, z.B.: 4,103,60,28,0,17.3,0.422,48,?

Eine der fruchtbarsten Methoden, um aus Beispielen gewonnene Kenntnisse für ein Klassifikationsproblem exakt niederzuschreiben, ist die, einen Entscheidungsbaum aufzustellen. Das ist ein Baum, bei dem an der Wurzel und an jedem inneren Knoten einer der Meßwerte (z.B. "Age") eines Falles abgefragt wird, und bei dem man gemäß dem Intervall, in den dieser Meßwert bei einem konkreten Fall fällt, auf einer diesem Intervall zugeordneten Kante zu einem Knoten auf der nächsten Ebene des Baumes weitergeht. Jedes Blatt des Entscheidungsbaumes ist mit der Zahl 1 oder 0 beschriftet, je nachdem, ob der Entscheidungsbaum vorhersagt, daß die Krankheit vorliegt oder nicht vorliegt bei einem Patienten dessen Meßwerte (angefangen bei der Wurzel des Entscheidungsbaumes) zu diesem Blatt hinführen. Die Qualität eines Entscheidungsbaumes ("accuracy") wird gemessen mittels dem Prozentsatz von neuen Fällen desselben Klassifikationsproblems, bei denen er eine richtige Vorhersage liefert.

Ein Lernalgorithmus mit Entscheidungsbäumen bekommt als Input eine Liste von "Trainingsbeispielen" (z.B. den ganzen Datensatz PI oder auch nur einen Teil davon). Er muß dann vollkommen selbständig die in diesen Daten verborgenen Strukturen aufspüren und die von ihm entdeckten Gesetzmäßigkeiten in der Form eines Entscheidungsbaumes (dem **Output** des Lernalgorithmus) niederschreiben.

Wir haben an unserem Institut einen neuen Lernalgorithmus mit Entscheidungsbäumen (genannt T2) entwickelt [27], der darauf spezialisiert ist, sehr kleine aber dafür "optimale" Entscheidungsbäume zu vorhandenen Listen von Trainingsbeispielen zu errechnen. Der von T2 errechnete Entscheidungsbaum ist optimal in der Hinsicht, daß es keinen anderen Entscheidungsbaum der Tiefe 2 gibt, der die in den Trainingsbeispielen vorhandenen Gesetzmä-Bigkeiten genauer wiedergibt. Ein wichtiger Vorteil von kleinen Entscheidungsbäumen ist der, daß sie dem Benutzer zusätzlich wichtige heuristische Kenntnisse dafür liefern, welche Meßwerte am wichtigsten sind, und welches deren kritischen Werte sind. Das läßt sich am besten an einem Beispiel erläutern: Wir geben zuerst den Entscheidungsbaum an, den der bisherige state-of-theart Lernalgorithmus mit Entscheidungsbäumen C4.5 [28] für das hier betrachtete Klassifikationsproblem PI liefert. Es ist offensichtlich sehr schwer, aus diesem Entscheidungsbaum heuristische Kenntnisse über die Diagnose der Krankheit pima-indians diabetes zu gewinnen.

Pruned C4.5 tree for dataset PI (accuracy: 73.4 %):	Pedigree > 0.63 :
The state of the control of the state of the control of the contr	Pedigree <= 0.652 : 0
Plasma_glucose > 166 : 1	Pedigree > 0.652 :
Plasma_glucose <= 166:	
1 Mass <= 25.4 : 0	
Mass > 25.4:	
Plasma_glucose <= 99 :	
Pedigree <= 0.787 : 0 Pedigree > 0.787 :	Mass <= 39.6 :[S4]
	Age > 58:
	Pedigree > 0.711 :
Plasma_glucose > 99 :	Pedigree <= 0.711 :
Age <= 30 :	Plasma_glucose <= 143:0
Mass <= 45.4 :	Plasma_glucose > 143 :
Diastolic_bllod_pressure <= 44 :	
Plasma_glucose <= 117 : 0	Npregnant > 4 : 0
Plasma_glucose > 117 : 1	Subtree [S1]
Diastolic_bllod_pressure > 44 :	Subucc [51]
Plasma_glucose <= 153 :	Skin_thickness > 0:0
Npregnant <= 5 :	Skin_thickness <= 0:
Pedigree <= 0.331 : 0 Pedigree > 0.331 :	1 Age <= 25 : 1
	1 Age > 25:0
	Subtree [S2]
Pedigree > 0.382 ;	
Insulin > 207:0	Pedigree <= 0.431 : 1
Insulin <= 207 :	Pedigree > 0.431 :
Pedigree > 0.955 : 1	Pedigree <= 0.503 : 0
Pedigree <= 0.955 :	Pedigree > 0.503:
Mass <= 31.3 :[S1]	Skin_thickness <= 0 : 1 Skin_thickness > 0 :
	Age > 28 : 1
	Age <= 28 :
	Diastolic_bllod_pressure > 66 : 0
	Diastolic_bllod_pressure <= 66 :
	Plasma_glucose <= 116 : 1
Mass <= 28.5 : 1	Plasma_glucose > 116:
1 Mass > 28.5 :	Plasma_glucose <= 126 : 0
Plasma_glucose <= 118:0	Plasma_glucose > 126 : 1
Plasma_glucose > 118 : 1	
Plasma_glucose > 153 :	Subtree [S3]
Pedigree <= 0.238 : 0	Diastolia bllad program de 92 :
Pedigree > 0.238 : 1	Diastolic_bllod_pressure <= 82 : Mass <= 28 :
Mass > 45.4:	Plasma_glucose <= 116 : 1
Insulin <= 230 :	Plasma_glucose > 116:0
Insulin > 230 : 0	Mass > 28 :
Age > 30 : Age <= 58 :	Skin_thickness <= 26 :
	Npregnant <= 2 : 0
Mass > 45.5 : 1	1 1 Npregnant > 2 :
Mass <= 45.5 :	1
Skin_thickness <= 37 :	Age <= 39 :
Plasma_glucose > 160 : 1	Age > 39 : 0
Plasma_glucose <= 160 :	Plasma_glucose > 106 :
Npregnant > 12 : 1	Pedigree > 0.411 : 0
Npregnant <= 12 :	Pedigree <= 0.411 : Diastolic_bllod_pressure <= 76 : 1
Npregnant <= 1 :	Diastolic_blod_pressure < 76 :
	Mass <= 34.5 : 1
Plasma_glucose > 114 : 1	Mass > 34.5 : 0
	Skin_thickness > 26 :
	Mass <= 31.6:0
	Mass > 31.6:
	Pedigree <= 0.334 : 0
Diastolic_bllod_pressure > 68:0	1 1 Pedigree > 0.334 : 1
Pedigree > 0.527 :	Diastolic_bllod_pressure > 82:
Pedigree > 1.394 : 0	Plasma_glucose <= 141:0
Pedigree <= 1.394 :	Plasma_glucose > 141 : 1
Npregnant > 7 : 1	Subtree (S4)
Npregnant <= 7:	Subtree [S4]
1 1 1 1 Insulin > 285 : 0	Diastolic_bllod_pressure <= 68:0
Insulin <= 285 :	Diastolic_bllod_pressure > 68 : 1
Pedigree <= 0.63 : 1	Diastone_onod_pressure > 00 : 1

Zum Vergleich geben wir nun den Entscheidungsbaum an, den T2 für dasselbe Klassifikationsproblem PI liefert. Dieser Baum ist sehr viel einfacher, ohne daß er an Vorhersagefähigkeit (accuracy) verliert.

T2 tree for dataset PI (accuracy: 74.8 %):

Mass in (-infinity,29.9):

1 Age in (-infinity,49): 0

1 Age in [49,53): 1

1 Age in [53,+infinity): 0

Mass in [29.9,+infinity):

l Plasma_glucose in (-infinity,56): 1

| Plasma_glucose in [56,128): 0

| Plasma_glucose in [128,+infinity): 1

Eigenschaften des in unserem Institut entwickelten Lernalgorithmus T2:

- T2 ist nicht nur ein "greedy" Algorithmus, wie alle bisher in der Praxis verwendeten Lernalgorithmen, sondern ein beweisbar erfolgreicher Lernalgorithmus in einem präzisen mathematischen Modell für Lernen: "agnostic PAC-Lernen". Damit ist ein wichtiger Schritt zur Verbindung von Theorie und Praxis des Maschinellen Lernens gemacht worden.
- T2 liefert für viele real-world Lernprobleme einen Entscheidungsbaum der nicht nur einfach zu "verstehen" ist, sondern bei vielen Lernproblemen eine höhere "accuracy" hat bei der Vorhersage (d.h. bei der Klassifikation von neuen Daten).
- T2 hat eine Rechenzeit, die fast linear ist in der Anzahl der Trainingsbeispiele. Dazu benutzt er eine raffinierte neue algorithmische Methode, die in der Arbeitsgruppe für Maschinelles Lernen in unserem Institut im letzten Jahr entdeckt wurde.

QED:

Abschließend möchte ich noch einmal auf die Ausführungen vom Anfang dieses Artikels zurückkommen. Man sieht anhand der beiden in diesem Artikel erläuterten Beispiele recht gut, daß die Arbeit in der Theoretischen Informatik sehr vielfältig ist. Sie erfordert ein präzises Durchdenken von Utopien, die Schaffung eines wissenschaftlichen Fundaments für neue Anwendungen der Informatik und Telematik sowie die Erweiterung der vorhandenen Toolbox von mathematischen und algorithmischen Methoden.

Literatur

- M. Abeles. (1991) Corticonics: Neural Circuits of the Cerebral Cortex, Cambridge University Press.
- [2] W. Maass. (1995) On the computational complexity of networks of spiking neurons, Advances in Neural Information Processing Systems vol. 7 (NIPS '94), G. Tesauro, D. S. Touretzky and T. K. Leen, eds., MIT Press (Cambridge MA), 183-190;

IIG-Report Series 393 der Technischen Universität Graz.

Electronically available at:

- ftp: cscx.cs.rhbnc.ac.uk in /pub/neurocolt/tech_reports: as nc-tr-94-021.ps.Z and
- ftp: archive.cis.ohio-state.edu in /pub/neuroprose: as maass.spiking.ps.Z
- [3] W. Maass. (1995) Lower bounds for the computational power of networks of spiking neurons, Neural Computation 8:1, 1-40;

- IIG-Report Series 398 der Technischen Universität Graz. Electronically available at:
- ftp: archive.cis.ohio-state.edu in /pub/neuroprose: as maass.spiking-details.ps.Z
- [4] W. Maass. (1995) Analog computations on networks of spiking neurons, Proceedings of the 7th Italian Workshop on Neural Nets, World Scientific Press.
- [5] W. Maass. (1995) On the computational power of noisy spiking neurons, Proceedings of the 1995 Conference on Neural Information Processing Systems (NIPS '95) in Denver.
- [6] W. Maass. (1995) An implementation of sigmodial neural nets in temporal coding with noisy spiking neurons. IIG-Report Series 442 der TU Graz.
- [7] A. Murray, L. Tarassenko. (1994) Analogue Neural VLSI: A Pulse Stream Approach, Chapman & Hall.
- [8] J. Douglas, M. A. Mahowald, K. A. C. Martin. (1994) Hybrid analog-digital architectures for neuromorphic systems, Proc. of the 1994 IEEE Conf. on Neural Networks in Orlando, 1848 - 1853.
- [9] W. Maass, B. Ruf. (1995) On the relevance of the shape of postsynaptic potentials for the computational power of networks of spiking neurons, Proceedings of the International Conference on Artificial Neural Networks ICANN'95, EC2&Cie, Paris, 515-520. Electronically available at:

ftp: archive.cis.ohio-state.edu in /pub/neuroprose: as maass.shape.ps.Z

- [10] P. Auer, P. M. Long. (1994) Simulating access to hidden information while learning, Proceedings of the 26th Annual ACM Symposium on the Theory of Computing, pages 263-272. ACM Press. Also NeuroCOLT Technical Report NC-TR-95-005.
- [11] P. Auer, N. Cesa-Bianchi. (1994) On-line learning with malicious noise and the closure algorithm. Proceedings of the 1994 Conference on Algorithmic Learning Theory, Lecture Notes in Artificial Intelligence 872, Springer, 1994, pages 229—247. Also NeuroCOLT Technical Report NC-TR-95-043 and Technical Report IIG-390.
- [12] P. Auer. (1995) Learning nested differences in the presence of malicious noise. Proceedings of the 1995 Conference on Algorithmic Learning Theory.
- [13] P. Auer, M. K. Warmuth. (1995) Tracking the best disjunction, Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science.
- [14] P. Auer, N. Cesa-Bianchi, Y. Freund, R. E. Schapire. (1995) Gambling in a rigged casino: The adversarial multi-armed bandit problem, *Proceedings of* the 36th Annual IEEE Symposium on Foundations of Computer Science. A preliminary version appeared as NeuroCOLT Technical Report NC-TR-95-001.
- [15] P. Auer, P. M. Long, W. Maass, G. J. Wöginger. (1995) On the complexity of function learning. *Machine Learning 18*, 187-230. IIG-Report Series 372 der Technischen Universität Graz.
 - Electronically available at: ftp: cscx.cs.rhbnc.ac.uk in /pub/neurocolt/tech_reports: as nc-tr-94-018.ps.Z
- [16] W. Maass. (1994) Efficient agnostic PAC-learning with simple hypotheses, Proc. of the 7th Annual ACM Conference on Computational Learning Theory, 1994, 67-75.
 - IIG-Report Series 383 der Technischen Universität Graz. Electronically available at:
- ftp: cscx.cs.rhbnc.ac.uk in /pub/neurocolt/tech_reports: as nc-tr-94-020.ps.Z
- [17] W. Maass, M. Warmuth. (1995) Efficient learning with virtual threshold gates, Proceedings of the 12th International Machine Learning Conference, Tahoe City (USA), Morgan Kaufmann (San Francisco), 378-386.
- [18] D. P. Dobkin, D. Gunopulos, W. Maass. (1994) Computing the maximum bichromatic discrepancy, with applications in computer graphics and machine learning, *Invited paper for a special issue of the Journal of Computer and System Sciences*. Electronically available at:
 - ftp: cscx.cs.rhbnc.ac.uk in /pub/neurocolt/tech_reports: as nc-tr-95-008.ps.Z
- [19] M. A. Arbib, ed.. (1995) Handbook of Brain Theory and Neural Networks, MIT-Press (Cambridge).
- [20] P. Auer, M. Herbster, M. K. Warmuth. (1995) There is no good squashing function for the square loss, Proceedings of the 1995 Conference on Neural Information Processing Systems (NIPS '95) in Denver.
- [21] W. Maass. (1994) Agnostic PAC-learning of functions on analog neural nets, Advances in Neural Information Processing Systems vol. 6, Morgan Kaufmann (San Mateo, 1994), 311-318; journal version: Neural Computation 7 (1995), 902 - 926.

IIG-Report Series 362 der Technischen Universität Graz.

Electronically available at:

ftp: cscx.cs.rhbnc.ac.uk in /pub/neurocolt/tech-reports: as nc-tr-95-002.ps.Z

ftp: archive.cis.ohio-state.edu in /pub/neuroprose: as maass.agnostic.ps.Z

[22] W. Maass. (1994) Neural nets with superlinear VC-dimension, Proceedings of The International Conference on Artificial Neural Networks (ICANN '94), Springer (Berlin), 581 - 584; Journal version: Neural Computation 6 (1994), 875 - 882.

IIG-Report Series 366 der Technischen Universität Graz.

Electronically available at:

ftp: cscx.cs.rhbnc.ac.uk in /pub/neurocolt/tech_reports: as nc-tr-94-019.ps.Z and

ftp: archive.cis.ohio-state.edu in /pub/neuroprose: as maass.super.ps.Z

[23] W. Maass. (1994) Perspectives of current research about the complexity of learning on neural nets, in Theoretical Advances in Neural Computation and Learning, V. P. Roychowdhury, K. Y. Siu, A. Orlitsky, editors, Kluwer Academic Publishers (Boston, 1994), pages 295 - 336. IIG-Report Series 385 der Technischen Universität Graz.

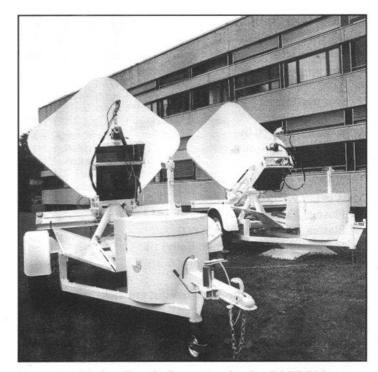
Electronically available at:

ftp: cscx.cs.rhbnc.ac.uk in /pub/neurocolt/tech_reports: as nc-tr-95-003.ps.Z and

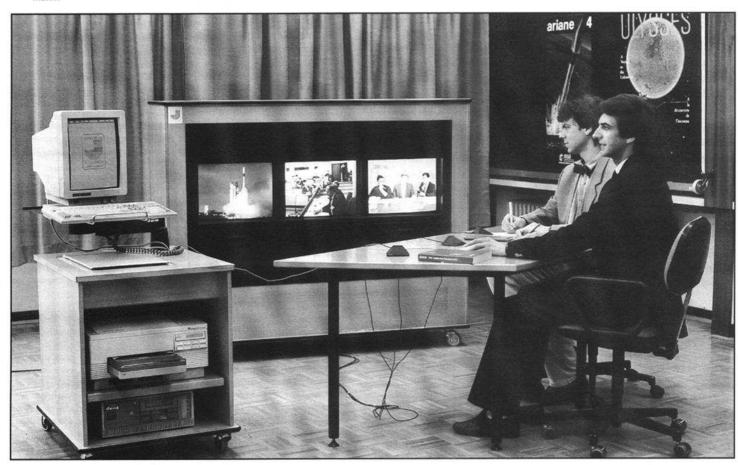
- ftp: archive.cis.ohio-state.edu in /pub/neuroprose: as maass.perspectives.ps.Z
 [24] W. Maass. (1995) Vapnik-Chervonenkis dimension of neural nets, in: The Handbook of Brain Theory and Neural Networks, M. A. Arbib, ed., MIT-Press (Cambridge), 1000-1003.
- [25] B. Ruf. (1994) A Stop Criterion for the Boltzmann Machine Learning Algorithm, Proceedings of the ESANN'94 (European Symposium on Artificial Neural Networks), De facto, Belgium. Electronically available at:

ftp://cscx.cs.rhbnc.ac.uk/pub/neurocolt/tech_reports/nc-tr-95-006.ps.Z

- [26] M. Schmitt. (1995) On Methods to Keep Learning Away from Intractability, F. Fogelman-Soulié and P. Galliani, eds., Proceedings of the International Conference on Artificial Neural Networks ICANN'95, EC2&Cie, Paris, 211-216.
- [27] P. Auer, R. C. Holte, W. Maass. (1995) Theory and applications of agnostic PAC-learning with small decision trees, Proceedings of the 12th International Machine Learning Conference, Tahoe City (USA), Morgan Kaufmann (San Francisco), 21-29.
- [28] J. R. Quinlan. (1993) C4.5: Programs for Machine Learning, Morgan Kaufmann.



Transportable Satellitenbodenstation für das DICE-Videokonferenzsystem, Experimente für LAN-Vernetzung via Satellit



Satellitenvideokonferenzsystem DICE (Direct Inter-Establishment Communications in Europe) entwickelt vom Institut für Angewandte Systemtechnik für die Europäische Weltraumagentur