

WISSENERWERB ÜBER DYNAMISCHE SYSTEME: BEFUNDE KONNEKTIONISTISCHER MODELLIERUNG

Anette Standfuss[†], Knut Möller[†] & Joachim Funke[‡]

[†]Institut für Informatik & [‡]Psychologisches Institut
Universität Bonn¹

ZUSAMMENFASSUNG

Die vorliegende Arbeit untersucht Möglichkeiten und Grenzen der Verwendung von einfachen konnektionistischen Systemen als Modelle für den Erwerb und die Repräsentation von Wissen über zeitdiskrete lineare dynamische Systeme in der Kognitionspsychologie. Ein ausgewähltes dynamisches System namens SINUS wird in Form eines "pattern associators" repräsentiert und dessen Lernverhalten untersucht. Es wird versucht, daraus Annahmen über den Wissenserwerb von Probanden im Umgang mit solchen dynamischen Systemen abzuleiten, um insbesondere Hinweise darauf zu erhalten, was "gute" von "schlechten" Probanden unterscheidet. Ein weiterer hier betrachteter Aspekt ist die Steuerung eines dynamischen Systems in einen vorgegebenen Zielzustand, der unter Beibehaltung des konnektionistischen Modells durch einen variierten Lernalgorithmus modelliert wird. Die abschließende Diskussion geht auf die Bedeutung der Modellierung für die kognitionspsychologische Theoriebildung ein.

1 EINFÜHRUNG

In aktuellen kognitionspsychologischen Arbeiten spielen Überlegungen zum Einfluß von Wissen, dessen Erwerb und dessen Anwendung eine wichtige Rolle (vgl. Mandl & Spada, 1988). Auch innerhalb der neueren KI-Forschung wird diesem Konzept verstärkt Aufmerksamkeit zugewendet, was etwa in der Beschäftigung mit wissensbasierten Systemen zum Ausdruck kommt. Für Psychologie und Informatik hat sich mit dem Aufkommen neokonnektionistischer Modellvorstellungen eine mögliche Alternative zum herkömmlichen Standpunkt der Symbolverarbeitungstradition ergeben (vgl. zur Übersicht: Kemke, 1988; McClelland & Rumelhart, 1986; Rumelhart & McClelland, 1986).

Die vorliegende Arbeit demonstriert an einem konkreten Beispiel, dem Wissenserwerb über zeitdiskrete lineare dynamische Systeme, einige Möglichkeiten und Grenzen eines konnektionistischen Lernverfahrens.

¹ Anschrift der Verfasser: Römerstr. 164, D-5300 Bonn 1. Die Arbeit im DYNAMIS-Projekt wurde durch eine Sachbeihilfe der DFG unterstützt (Az. Fu 173/1).

1.1 Darstellung der Wissenserwerbssituation

Innerhalb des Forschungsprojekts DYNAMIS (vgl. Funke, 1986a) werden Probanden (Pbn) mit folgender, aus zwei Teilanforderungen bestehender Aufgabe konfrontiert: (1) in direkter Interaktion mit einem Computer die Parameter eines ihnen zunächst unbekanntes dynamischen Systems zu identifizieren und (2) anschließend auf der Basis des angenommenen Wirkungsmodells das System zu kontrollieren. Die verwendeten Systeme lassen sich als zeitdiskrete lineare Strukturgleichungsmodelle bzw. als autoregressive Prozesse erster Ordnung darstellen. Es wird unterstellt, daß Pbn im Laufe längerer Interaktionen mit dem zunächst unbekanntes System ein zunehmend besseres Modell dieses Systems erwerben (vgl. Funke, 1985).

Das im folgenden verwendete System SINUS ist auf einem fremden Planeten angesiedelt. Der Pb soll die Wirkungen der drei exogenen Variablen (X) auf die drei endogenen Variablen (Y) sowie die Wirkungen der endogenen Variablen untereinander erkunden. SINUS kann durch die folgenden drei Gleichungen beschrieben werden, wobei t den Zeittakt beschreibt:

$$y1(t+1)=1.0*y1(t)+10.0*x1(t),$$

$$y2(t+1)=1.0*y2(t)+ 0.2*y3(t)+ 3.0*x3(t),$$

$$y3(t+1)=0.9*y3(t)+ 2.0*x2(t)+ 0.5*x3(t).$$

Hat der Pb das System vollständig durchschaut, so müßte sich sein internes Modell durch die folgende Parametermatrix repräsentieren lassen:

	x1(t)	x2(t)	x3(t)	y1(t)	y2(t)	y3(t)
y1(t+1)	10	0	0	1	0	0
y2(t+1)	0	0	3	0	1	0.2
y3(t+1)	0	2	0.5	0	0	0.9

Da es sich bei dem System SINUS um ein fiktives Szenario handelt, kann davon ausgegangen werden, daß der Pb so gut wie kein Vorwissen darüber besitzt, seine subjektive Parametermatrix also als Nullmatrix beschrieben werden darf.

1.2 Darstellung der konnektionistischen Modellierung

Im folgenden soll das konnektionistische Modell eines Pb konstruiert werden, der sich mit dem System SINUS befaßt. Es soll nd über folgende Eigenschaften verfügen: (a) es soll lernfähig sein in dem Sinn, daß es die Verbindungsstärken zwischen den SINUS-Variablen identifiziert, und (b) es soll Wissen in Form von Matrizen repräsentieren, deren Elemente Verbindungsstärken darstellen.

Abstrakt kann die Aufgabenstellung als Musterassoziation (=prediction problem) beschrieben werden: Gelernt werden soll die Verbindung zwischen den Variablenwerten (=exogene und endogene Werte) zum Zeitpunkt t und den Zustandswerten (=endogene Werte) des Systems zum Zeitpunkt $t+1$. Für die Modellierung von SINUS werden somit sechs Input-Einheiten mit drei Output-Einheiten verknüpft:

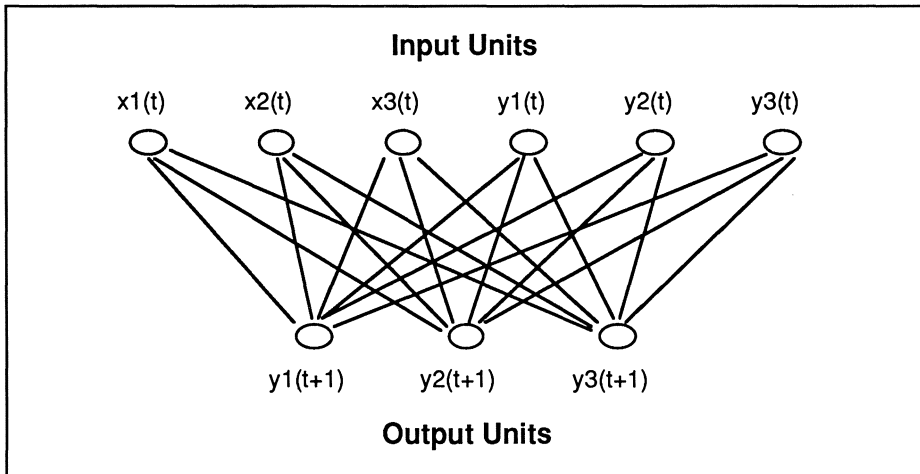


Abbildung 1. Vernetzung der sechs Inputs mit den drei Outputs

Dabei realisiert die Struktur des Netzwerks die zeitliche Komponente und die Gewichte der Kanten die Parameter der zu lernenden Matrix. Eine Zwischenschicht von Einheiten ("hidden units") wird nicht benötigt, da die dem System SINUS zugrundeliegenden Gleichungen linear sind.

Als Einheiten eignen sich hier lineare Einheiten, also solche, die ihre Aktivität als Summe der gewichteten Eingaben bestimmen. Entsprechen die Kantengewichte der optimalen Matrix, sind dadurch die Aktivitäten der Output-Einheiten den endogenen Werten des Systems SINUS äquivalent, die bei Eingabe der Daten zum Takt t als Werte im Takt $t+1$ resultieren würden.

Bei der Bestimmung der Lernregel wurde wie folgt vorgegangen. Das eben beschriebene Netz soll das Modell eines Pb darstellen, der über die exogenen Variablen in das System eingreifen kann und auf diese Art Information gewinnt. Plant dieses Netz einen Eingriff in einen bestimmten Systemzustand, so stellt die Netzausgabe die Prognose über den folgenden Systemzustand dar auf der Grundlage der bisherigen Repräsentation des Simulationssystems. Wird der Eingriff in die Tat umgesetzt, gibt das Simulationssystem eine Rückmeldung über die tatsächlich resultierenden Werte der endogenen Variablen. Dieses Feedback stellt somit den "Lehrer" des Systems dar. Aufgrund dieser Tatsache wie auch aufgrund der Tatsache, daß keine verborgenen Einheiten benötigt werden, bietet sich die einfache Delta-Regel als Lernregel an. Bei dieser wird als Fehlersignal e_i die Differenz der Zielaktivität für die Output-Einheit i (=der "Prognose") und der aktuellen Aktivität der Output-Einheit i (=der tat-

sächlich resultierende Wert) gewählt, d.h. $e_j = t_j - a_j$ (e für "error", t für "target" und a für "activity"; die Indizes bezeichnen Einheiten). Die Delta-Regel lautet dann:

$$\delta w_{ij} = \varepsilon * e_i * a_j,$$

wobei ε die Lernrate (=Schrittweite des Gradientenverfahrens) und w_{ij} das Gewicht der Kante von Einheit j (der Input-Einheit) zur Einheit i (der Output-Einheit) darstellen.

2 REALISIERUNG UND METHODISCHES VORGEHEN

Wie das eben beschriebene Modell realisiert wurde und welche Tests zur Überprüfung der Leistungsfähigkeit des Modells durchgeführt wurden, beschreiben die folgenden Abschnitte.

2.1 Algorithmen

Für die Realisierung des beschriebenen Modells wurde das Modul PA (pattern associator) mit linearen Einheiten und Delta-Regel aus der PDP-Simulationssoftware von McClelland & Rumelhart (1988) ausgewählt. Von Vorteil dabei ist, daß die Aktivitäten nicht zwischen Null und Eins liegen müssen, so daß keine Veränderungen am Algorithmus notwendig waren.

Die benötigte Pattern-Datei mit den Trainingsmustern wurde mit dem DYNAMIS-Programm (Funke, Fahrenbruck & Müller, 1986) erzeugt. Um Fehlermeldungen wegen Überschreitung des Bereichs zwischen MaxInt und MinInt zu vermeiden, wurden Start-, Ziel-, Maximal- und Minimalwerte der SINUS-Steuerdatei durch 10000 dividiert.

2.2 Vorgehen

Zwei verschiedene Validierungsstrategien sollen Verwendung finden. Zum einen wird eine selbstkonstruierte Trainingsmenge als Pattern-Datei verwendet, von der anzunehmen ist, daß sie hinsichtlich der Lernmöglichkeiten maximal viel Information enthält. Zum anderen sollte aber auch geprüft werden, inwiefern empirisch realisierte Trainingsmengen, d.h. von Pbn erzeugte Systemzustände, zu akzeptablen Ergebnissen des Simulationsmodells führen können. Hierfür wurden exemplarisch Daten von zwei Pbn aus dem DYNAMIS-Projekt ausgewählt, die nach den dort entwickelten Gütekriterien ("Güte der Kausaldiagramme", GdK; vgl. Funke & Kleinemas, 1989) als gute bzw. schlechte Identifizierer bezeichnet werden können.

Für die selbst konstruierte Lernmenge wurde eine Pattern-Datei erzeugt, wobei die gewählten Werte für die exogenen Variablen möglichst zufällig aus dem positiven wie negativen Bereich gewählt und auf dreistellige Zahlen beschränkt wurden.

3 ERGEBNISSE

Entsprechend dem eben dargelegten Vorgehen bei der Überprüfung der Leistungsfähigkeit des konnektionistischen Lernmodells werden zunächst die Ergebnisse für die optimalen

Lernbedingungen berichtet. Daran schließt sich die Darstellung der Lernergebnisse auf der Basis empirisch beobachteter Lernmengen an. Schließlich geht es um die Modellierung zielgerichteten Lernens.

3.1 Lernen bei Verwendung selbst konstruierter Lernmengen

Nach 280 Lerndurchgängen (wobei ein Durchgang aus der Präsentation aller Muster besteht) und einer Lernrate $\epsilon=1.0$ ergab sich, ausgehend von einer Nullmatrix zu Beginn des ersten Durchgangs, die folgende Gewichtsmatrix:

	$x_1(t)$	$x_2(t)$	$x_3(t)$	$y_1(t)$	$y_2(t)$	$y_3(t)$
$y_1(t+1)$	9.99	0.00	0.00	0.99	0.00	0.00
$y_2(t+1)$	0.00	0.00	2.99	0.00	0.99	0.19
$y_3(t+1)$	0.00	1.99	0.50	0.00	0.00	0.89

Ein Vergleich mit der zugrundeliegenden Gewichtsmatrix (siehe weiter oben) ergibt eine fast optimale Reproduktion der Gewichte; Abweichungen liegen nur noch auf der zweiten Nachkommastelle vor. Damit ist die Lernbarkeit der Parameter des Systems SINUS durch ein konnektionistisches Modell demonstriert.

3.2 Lernen bei Verwendung von empirisch vorkommenden Lernmengen

In einem zweiten Schritt wurde überprüft, zu welchen Lernleistungen dieses Netzwerk fähig ist, wenn anstelle einer weitgehend optimalen Lernmenge die empirisch beobachteten, von den Pbn selbst erzeugten Lernmengen verwendet werden. Hierzu konnten die Systemdaten von zwei Pbn (DI24M und XX23M mit GdK-Werten von 2.00 resp. 0.46) herangezogen werden, die zu guter bzw. schlechter Identifikation fähig waren. Das Maß GdK skaliert die Qualität der von einem Pbn korrekt erkannten Kausalstruktur des Systems SINUS auf einem Bereich von 0 (Min) bis 2 (Max) unter Verwendung von Diagrammen, in denen der Pb die von ihm vermuteten Wirkungen auf unterschiedlichen Präzisionsgraden eintragen kann.

DI24M und XX23M haben sich jeweils fünf Durchgänge lang mit dem System SINUS beschäftigt. Während DI24M das System nach fünf Durchgängen vollständig durchschaut hatte (er gibt im Kausaldiagramm - dem Diagnostikum für die subjektive Kausalstruktur - die korrekten Parameter an), gelingt dies XX23M nicht.

Lernvorgang mit den Daten von DI24M: Die Lernrate wurde $\epsilon=1.0$ gewählt. Aufgrund der geringeren Anzahl von Trainings-Patterns im Vergleich zu den selbstgenerierten Trainings-Patterns war hier eine größere Zahl von benötigten Lernepochen zu erwarten. Nach 3500 Epochen ergab sich die folgende Matrix:

	x1(t)	x2(t)	x3(t)	y1(t)	y2(t)	y3(t)
y1(t+1)	9.99	0.00	0.01	1.00	0.00	-0.01
y2(t+1)	0.00	0.02	2.95	0.00	0.98	0.23
y3(t+1)	0.00	1.98	0.52	0.00	0.01	0.88

Lernvorgang mit den Daten von XX23M: Hier wurde die Lernrate ebenfalls $\epsilon=1.0$ gewählt. Nach 14500 (!) Epochen ergab sich die folgende Matrix

	x1(t)	x2(t)	x3(t)	y1(t)	y2(t)	y3(t)
y1(t+1)	9.98	0.00	0.00	0.99	0.00	0.00
y2(t+1)	0.00	0.00	3.00	0.00	0.99	0.20
y3(t+1)	0.00	1.99	0.49	0.00	0.00	0.90

Es zeigt sich also, daß die Korrektheit der resultierenden Matrix unabhängig von der Güte der Systemidentifikation durch die Versuchsperson ist. Demnach ist D124M im Gegensatz zu XX23M in der Lage, die durch seine Eingriffe gewonnenen Informationen richtig auszuwerten. Daß allerdings die von XX23M erzeugten Daten "schwerer" auswertbare Informationen enthalten, macht die wesentlich erhöhte Zahl von Lernepochen deutlich.

3.3 Zielgerichtet Lernen

Ein weiterer Aspekt, der bisher nicht betrachtet wurde, betrifft die Erreichung eines Ziels, d.h. die Steuerung des Systems SINUS in einen vorgegebenen Systemzustand. Für die Pbn stellt sich diese Anforderung zusätzlich zu derjenigen der Identifikation; während der Identifikationsstätigkeit kann der Pb spielerisch "ausprobieren", wie gut ihm die Erreichung eines vorgegebenen Zielzustands gelingt (vgl. 1.1). Diese Teilaufgabe der Steuerung kann mit demselben Modell wie bisher, jedoch unter Verwendung eines erweiterten Lernalgorithmus, realisiert werden (vgl. Linden, 1989).

Beschreibung des Algorithmus: Die Einheiten sind wie bisher lineare Einheiten und die Gewichte werden mit der Delta-Regel modifiziert. Folgende sechs Schritte sind zu leisten: (1) Anlegen eines Input-Patterns; (2) Berechnung der Ausgabe der Output-Einheiten; (3) Berechnung des Target-Patterns, d.h. der sich aufgrund der Eingabe ergebenden tatsächlichen Systemzustands; (4) Berechnung des Fehlers für jede Output-Einheit (=Target-Output); (5) Veränderung der Gewichte mit der Delta-Regel; (6) Ermittlung des Eingriffs, der basierend auf

dem aktuellen Modell des Systems, gegeben durch die aktuellen Gewichte, das Target-Pattern als Ausgabe der Output-Patterns erzeugt.

- (a) Der unter (4) berechnete Fehler wird über die Kanten an die Input-Einheiten geschickt, die den exogenen Variablen entsprechen, d.h. mit dem Kantengewicht multipliziert, und dort aufsummiert (entspricht der Netzeingabe bei einem inversen Durchgang.)
- (b) Diese Eingabe für die Input-Einheiten, die den exogenen Variablen entsprechen, wird dazu benutzt, deren Aktivität zu verändern. Die Aktivität der Input-Einheiten, die den endogenen Variablen entsprechen, bleibt erhalten. Das so bestimmte modifizierte Input-Pattern wird als neue Eingabe verwendet.
- (c) Die Ausgabe der Output-Einheiten wird berechnet.
- (d) Der Fehler für die Output-Einheiten wird mit dem alten Target-Pattern berechnet.
- (e) Goto (a), falls der Fehler größer ist als ein Schwellenwert, sonst goto (1).

Bei Verwendung dieses Lernalgorithmus konnte ein vorgegebener Zielzustand erreicht und über mehrere Schritte gehalten werden. Inwiefern dieser Algorithmus menschlichen Lernprozessen zugrundegelegt werden kann, behandelt die folgende Diskussion.

4 DISKUSSION DER ERGEBNISSE

Welche psychologische Bedeutung hat der Nachweis, daß ein bestimmtes konnektionistisches Modell die Parameter eines zu identifizierenden dynamischen Systems korrekt lernt? Damit ist zunächst einmal gezeigt, daß es hinsichtlich der internen Repräsentation des dynamischen Systems ausreicht, die Parametermatrix verfügbar zu haben: ein Netzwerk aus neun Knoten sowie den zwischen den sechs Alt- und drei Neu-Werten bestehenden gewichteten Kanten genügt hierfür ("Repräsentationsannahme").

Ein interessanter Aspekt besteht in der Frage, ob das konnektionistische Modell möglicherweise nicht nur vom Ergebnis her, sondern auch vom Lernverlauf her plausible Eigenschaften besitzt. Hierzu kann auf eine "Ökonomieannahme" zurückgegriffen werden (vgl. Funke, 1986b, p. 113), wonach Pbn zu Beginn der Identifikation mit möglichst einfachen Modellen anfangen. Dabei werden direkte Wirkungen (solche, bei denen exogene Variablen auf endogene Variablen einwirken) eher erkannt als indirekte Wirkungen (Nebenwirkungen, Eigendynamik). Im folgenden sind die direkten (=D) und indirekten (=I) Wirkungen des Systems SLNUS dargestellt:

	x1(t)	x2(t)	x3(t)	y1(t)	y2(t)	y3(t)
y1(t+1)	D	-	-	I	I	-
y2(t+1)	-	-	D	-	I	I
y3(t+1)	-	D	D	-	-	I

Untersucht man den mit dem Pattern-Associator durchgeführten Lernvorgang hinsichtlich der Reihenfolge korrekt erkannter Kanten, so zeigt sich bei mehrfacher Wiederholung, daß als erstes zwischen 'Einfluß' und 'kein Einfluß' unterschieden wird, d.h. tendenziell werden die

Parameter der Matrix als erstes korrekt gelernt, deren Wert 0 ist. Erst danach wird der Einfluß genau differenziert. Dabei läßt sich keine genaue Reihenfolge angeben, vielmehr kann man diesen Vorgang als ein gleichmäßiges Ausdifferenzieren beschreiben, bei dem Rundungsfehler und ausgewählte Trainings-Patterns eine gewisse Rolle spielen.

Dieses Nichtunterscheiden-Können zwischen direktem und indirektem Einfluß ergibt sich aus der Architektur des Netzes, bei der die Werte endogener und exogener Variablen gleichbedeutende Eingaben sind. Die Begriffe direkter und indirekter Einfluß ergeben sich jedoch erst aus der Unterscheidung direkt und indirekt beeinflussbarer Variablen. Bei dieser Architektur steckt die Unterscheidung zwischen direkten und indirekten Variablen lediglich in der Erzeugung der Input-Patterns. Desweiteren ist das Lernverfahren im Pattern-Associator ein Gradientenverfahren, d.h. es beruht gerade auf der gleichzeitigen Veränderung aller Gewichte entlang des Gradienten. - Es muß also gesagt werden, daß sich die Ökonomieannahme mit diesem Modell nicht untersuchen läßt, da der Lernvorgang nicht adäquat ist.

Eine für dieses Modell sinnvolle Anwendung wäre die folgende Untersuchung: Es wird die Anzahl der Lernepochen untersucht, die für verschiedene Trainingsmengen benötigt werden. Dabei sollen die Input-Patterns und die Output-Patterns alle in demselben Winkel zueinander stehen. Zwischen den einzelnen Trainingsmengen wird dann der Winkel variiert, wobei die beiden Extrema 'linearabhängige Vektoren' und 'orthogonale Vektoren' sind.

Die zu testende Hypothese ist nun die folgende: Die Anzahl der benötigten Lernepochen ist umgekehrt proportional zur Größe des Winkels, wobei im Extremfall eines Winkels von 0 Grad kein korrektes Lernen mehr möglich ist. Da solche Trainingsmengen nur dann erzeugt werden können, wenn diese nicht in einem Simulationsdurchgang mit dem DYNAMIS-Programm erzeugt werden, muß anschließend untersucht werden, welche Eingaben die optimalsten Bedingungen in dem Sinne schaffen, daß die so erzeugten Vektoren ($X_1(t)$, $X_2(t)$, $X_3(t)$, $Y_1(t)$, $Y_2(t)$, $Y_3(t)$) bzw. ($Y_1(t+1)$, $Y_2(t+1)$, $Y_3(t+1)$) in einem möglichst großen Winkel zueinander stehen.

Als nächstes kann dann die folgende Hypothese an realen Versuchspersonendaten untersucht werden: Die Güte der Systemerkennung durch eine Versuchsperson ist umso größer, je optimaler, in dem eben beschriebene Sinne, die von ihr durch Systemeingriffe erzeugten Vektoren sind. Diese Hypothese wird demnächst überprüft.

Abschließend möchten wir festhalten, daß gerade bei dem zur Diskussion stehenden Aufgabentyp kritisiert werden könnte, menschliche Pbn würden stärker hypothesengeleitet (symbolisch) vorgehen. Dem sind experimentelle Befunde mit diesem Paradigma (z.B. Berry & Broadbent, 1988) entgegenzusetzen, die für die Existenz assoziativer Mechanismen sprechen könnten. Mit unserer Arbeit haben wir einen Beitrag zur Beschreibung eines von möglicherweise mehreren Mechanismen geleistet.

Literatur

- Berry, D.C. & Broadbent, D.E. (1988). Interactive tasks and the implicit-explicit distinction. *British Journal of Psychology*, 79, 251-272.
- Funke, J. (1985). Steuerung dynamischer System durch Aufbau und Anwendung subjektiver Kausalmodelle. *Zeitschrift für Psychologie*, 193, 435-457.
- Funke, J. (1986a). Ein Forschungsprogramm zur subjektiven Repräsentation dynamischer Kleinsysteme: Aufbau und Anwendung von Wissen in Abhängigkeit von Person- und Systemmerkmalen. *Berichte aus dem Psychologischen Institut der Universität Bonn*, 12, Heft 1.
- Funke, J. (1986b). *Komplexes Problemlösen - Bestandsaufnahme und Perspektiven*. Heidelberg: Springer.
- Funke, J., & Kleinemas, U. (1989). Theoretische und empirische Beiträge zur Diagnostik strukturellen Wissens im Kontext dynamischer Systeme. *Berichte aus dem Psychologischen Institut der Universität Bonn*, 15, Heft 1.
- Funke, J., Fahnenbruck, G., & Müller, H. (1986). DYNAMIS - Ein Computerprogramm zur Simulation dynamischer Systeme. *Berichte aus dem Psychologischen Institut der Universität Bonn*, 12, Heft 3.
- Kemke, C. (1988). Der neuere Konnektionismus. Ein Überblick. *Informatik-Spektrum*, 11, 143-162.
- Linden, A. (1989). *Untersuchungen von Backpropagation in konnektionistischen Systemen*. Bonn: Institut für Informatik der Universität (Diplomarbeit).
- Mandl, H., & Spada, H. (Eds.) (1988). *Wissenspsychologie*. München: Psychologie Verlags Union.
- McClelland, J.L., & Rumelhart, D.E. (Eds.) (1986). *Parallel distributed processing. Explorations in the microstructure of cognition. Volume 2: Psychological and biological models*. Cambridge, Mass.: MIT Press.
- McClelland, J.L., & Rumelhart, D.E. (1988). *Explorations in parallel distributed processing. A handbook of models, programs, and exercises*. Cambridge, Mass.: MIT Press.
- Rumelhart, D.E., & McClelland, J.L. (Eds.) (1986). *Parallel distributed processing. Explorations in the microstructure of cognition. Volume 1: Foundations*. Cambridge, Mass.: MIT Press.