

IZ-Arbeitsbericht Nr. 20

**Einsatz neuronaler Netze als Transfer-
komponenten beim Retrieval in
heterogenen Dokumentbeständen**

Thomas Mandl

November 2000



InformationsZentrum
Sozialwissenschaften

Lennéstraße 30
D-53113 Bonn
Tel.: 0228/2281-0
Fax.: 0228/2281-120
email: mandl@rz.uni-hildesheim.de
Internet: <http://www.gesis.org>

ISSN: 1431-6943

Herausgeber: Informationszentrum Sozialwissenschaften der Arbeits-
gemeinschaft Sozialwissenschaftlicher Institute e.V. (ASI)

Druck u. Vertrieb: Informationszentrum Sozialwissenschaften, Bonn
Printed in Germany

Das IZ ist Mitglied der Gesellschaft Sozialwissenschaftlicher Infrastruktureinrichtungen e.V. (GESIS), einer
Einrichtung der Wissenschaftsgemeinschaft Gottfried Wilhelm Leibniz (WGL)

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | EINLEITUNG..... | 5 |
| 2 | HETEROGENITÄT UND IHRE BEHANDLUNG IM INFORMATION RETRIEVAL..... | 6 |
| 2.1 | Heterogene Erschließung und Qualität..... | 8 |
| 2.2 | Lösungsansatz: Behandlung von Heterogenität durch Transformationen | 9 |
| 2.3 | Vage Verfahren für Transformationen | 10 |
| 2.4 | Text Categorization | 11 |
| 2.5 | Latent Semantic Indexing für Transformationen | 13 |
| 2.6 | Fazit: Heterogenität im Information Retrieval..... | 15 |
| 3 | GRUNDLAGEN NEURONALER NETZE | 17 |
| 3.1 | Natürliche neuronale Netze | 17 |
| 3.2 | Kognitionswissenschaftliche Aspekte | 18 |
| 3.3 | Das Backpropagation-Modell: ein erster Überblick..... | 19 |
| 3.4 | Aufbau und Funktionsweise neuronaler Netze | 21 |
| 3.4.1 | <i>Neuronen</i> | 21 |
| 3.4.2 | <i>Vernetzung</i> | 22 |
| 3.4.3 | <i>Lernregel</i> | 22 |
| 3.4.4 | <i>Schnittstelle zur Umgebung</i> | 23 |
| 3.5 | Modelle | 23 |
| 3.5.1 | <i>Kohonen-Netze</i> | 24 |
| 3.5.2 | <i>Adaptive Resonance Theory (ART)</i> | 25 |
| 3.5.3 | <i>Assoziativspeicher</i> | 25 |
| 3.5.4 | <i>Backpropagation-Netze</i> | 27 |
| 3.6 | Simulationssoftware DataEngine..... | 33 |

| | |
|---|----|
| 3.7 Fazit: Grundlagen neuronaler Netze..... | 33 |
|---|----|

4 NEURONALE NETZE IM INFORMATION RETRIEVAL... 34

| | |
|--|----|
| 4.1 Historischer Überblick | 34 |
| 4.2 Retrieval mit Assoziativspeichern | 35 |
| 4.2.1 Hopfield-Netzwerke..... | 35 |
| 4.2.2 Boltzmann-Maschine | 37 |
| 4.2.3 Hetero-assoziative Systeme | 39 |
| 4.3 Spreading-Activation-Modelle..... | 41 |
| 4.3.1 Funktionsweise eines Spreading-Activation-Netzwerks..... | 42 |
| 4.3.2 Beispiele für Spreading-Activation-Netzwerke | 50 |
| 4.3.3 Hopfield- und Spreading-Activation-Netzwerke für Transformationen | 69 |
| 4.3.4 Vergleich der Spreading-Activation-Netzwerke mit dem Vektorraum-Modell70 | |
| 4.3.5 Fazit: Spreading-Activation-Modelle | 72 |
| 4.4 Kohonen-Netze im Information Retrieval..... | 75 |
| 4.4.1 Grundprinzip | 75 |
| 4.4.2 Systeme..... | 76 |
| 4.4.3 Fazit: Kohonen-Netze im Information Retrieval..... | 80 |
| 4.5 Adaptive Resonance Theory-Modelle..... | 81 |
| 4.6 Backpropagation-Architekturen für Information Retrieval..... | 83 |
| 4.6.1 Lernen als Gradientenabstieg..... | 83 |
| 4.6.2 COSIMIR-Modell..... | 84 |
| 4.6.3 COSIMIR für heterogene Repräsentationen | 86 |
| 4.6.4 Transformations-Netzwerk..... | 87 |
| 4.7 Neuronale Netze bei TREC | 90 |
| 4.8 Fazit: Neuronale Netze im Information Retrieval | 93 |

5 EVALUIERUNG DES TRANSFORMATIONS-NETZWERKS 95

| | |
|--|----|
| 5.1 Transformations-Netzwerk: Thesaurus zu Klassifikation | 95 |
| 5.1.1 Datenbanken des Informationszentrum Sozialwissenschaften | 95 |
| 5.1.2 Transformations-Netzwerk und LSI..... | 96 |
| 5.1.3 Ergebnisse | 98 |

| | |
|---|------------|
| 5.2 Transformations-Netzwerk: Kölner Bibliotheks-Thesaurus zu IZ-Repräsentationen | 103 |
| 5.2.1 <i>USB-Thesaurus zu IZ-Klassifikation</i> | 104 |
| 5.2.2 <i>USB-Thesaurus zu IZ-Thesaurus</i> | 105 |
| 6 FAZIT | 108 |
| LITERATURVERZEICHNIS | 109 |

Danksagung:

Ich danke der Firma Telcordia (vorher Bellcore), die die LSI Software unentgeltlich zur Verfügung gestellt hat.

Zusammenfassung:

Information Retrieval befasst sich mit vagen Anfragen und der vagen Modellierung von Benutzerverhalten. Die zunehmende weltweite Vernetzung und der Aufbau von digitalen Bibliotheken führt zu neuen Möglichkeiten bei der gleichzeitigen Suche in mehreren Datenbeständen. Die semantische Heterogenität der Suchterme in unterschiedlichen Datenquellen und die dafür notwendigen Transferkomponenten stellen eine besondere Herausforderung dar. Meist liegt für Transformationen nur unvollständiges und unsicheres Wissen vor, so dass vage Verfahren aus dem Paradigma des „Soft Computing“ eingesetzt werden müssen.

Neuronale Netze sind eine Methode zur vagen Informationsverarbeitung und zur Implementierung kognitiver Fähigkeiten. Dieser Arbeitsbericht gibt einen umfassenden Überblick über den state-of-the-art zu neuronalen Netzen im Information Retrieval und analysiert, gruppiert und bewertet zahlreiche Systeme.

Das Transformations-Netzwerk ist ein erfolgversprechendes System, das auf dem Backpropagation-Algorithmus basiert und Retrieval bei heterogenen Daten ermöglicht. In mehreren Experimenten wird das Transformations-Netzwerk mit realen Daten getestet. Je nach Datengrundlage ergaben sich unterschiedliche Ergebnisse im Vergleich mit alternativen Verfahren. Das optimale Verfahren hängt also vom Anwendungsfall ab. Bei gleicher Qualität ist die Überschneidung der Ergebnisse verschiedener Verfahren relativ gering, so dass Fusionsverfahren erprobt werden sollten.

1 Einleitung

Die Modellierung im Information Retrieval (IR) geht bisher kaum von den kognitiven Eigenschaften des Menschen oder dessen Bedürfnissen aus, sondern von mathematischen Modellen. Die mangelnde kognitive Modellierung liegt natürlich vor allem daran, dass über die kognitive Informationsverarbeitung noch nicht genügend Erkenntnisse vorliegen. Um aber IR-Systeme an den Menschen anzupassen und für den Benutzer optimal zu gestalten, muss sich die Modellbildung stärker an den kognitiven Eigenschaften orientieren. In diesem Arbeitsbericht werden lernende Systeme eingesetzt, um menschliche Entscheidungen zu modellieren. Dabei geht es insbesondere um die steigenden Anforderungen an Information Retrieval Systeme durch die Entwicklung digitaler Bibliotheken und die zunehmende weltweite Vernetzung. Information Retrieval muss immer stärker auf semantisch heterogene Daten reagieren können. Über die Zusammenhänge zwischen heterogenen Suchtermen liegt in der Regel nur unsicheres Wissen in Form von menschlichen Urteilen vor. Somit wird der empirisch zugängliche Ausdruck der kognitiven Fähigkeiten und nicht ihre interne Struktur zum Gegenstand.

Neuronale Netze sind ein Modellierungsverfahren, das von der Funktionsweise des menschlichen Gehirns inspiriert ist. Neuronale Netze zeichnen sich durch Fehlertoleranz, Effizienz und die Fähigkeit zur vagen Verarbeitung von Informationen aus. Sie zählen zu dem Paradigma „Soft Computing“ (cf. Zadeh 1994). Verfahren zur „weichen“ Informationsverarbeitung sehen bewusst von der exakten Modellierung eines Problems ab. Exaktheit ist oft wegen hoher Komplexität oder inhärenter Vagheit nicht erreichbar. „Soft Computing“ erlaubt es, mit nicht exakten Modellen zu agieren.

Zahlreiche Information Retrieval Modelle basieren bereits auf neuronalen Netzen, jedoch lösen sie die Schwächen bestehender Information Retrieval Systeme nicht umfassend und schöpfen die Möglichkeiten des „Soft Computing“ nicht vollständig aus. Das in diesem Arbeitsbericht diskutierte Transformations-Netzwerk basiert auf dem mächtigen und häufig eingesetzten Backpropagation-Algorithmus und lernt aufgrund von Beispielen, die Repräsentation eines Objekts von einem Repräsentations-Schema in ein anderes zu übertragen.

Kapitel 2 führt kurz in die Problematik heterogener Daten im Information Retrieval ein und diskutiert vage Verfahren für Transformationen als Lösungsansatz. Das folgende Kapitel gibt einen Überblick über neuronale Netze. Kapitel 4 führt diese beiden Ansätze zusammen und berichtet über Information Retrieval Systeme auf der Basis neuronaler Netze. Die Diskussion zeigt, dass das Transformations-Netzwerk gut für die Heterogenitätsbehandlung geeignet ist. Die Experimente in Kapitel 5 bestätigen die Leistungsfähigkeit des Transformations-Netzwerks.

2 Heterogenität und ihre Behandlung im Information Retrieval

Die zunehmende weltweite Vernetzung schafft für die Informationssuche völlig neue Möglichkeiten. Unterschiedlichste Datenquellen werden virtuell integriert und ein System kann auf diese Weise sehr große Mengen von Daten anbieten. Gleichzeitig stellen diese Möglichkeiten das Information Retrieval vor die Herausforderung, die Suche in stark heterogenen Umgebungen zu erleichtern.

Die Trennung der Informationsangebote und ihre Heterogenität wird aus politischen und inhaltlichen Interessen bestehen bleiben, da viele Informationsanbieter sich durch ihr individuelles Angebot profilieren. Auch werden sich kaum alle Datenbankproduzenten auf eine gemeinsame Basis bei der Inhaltserschließung verständigen. Zudem führt die Aufteilung in Spezial-Datenbanken zu einer höheren Kompetenz bei der Aufarbeitung dieser Bereiche und ist somit für viele Benutzerabsichten sehr sinnvoll.

In der Realität ist es oft noch so, dass ein Benutzer mehrere potenziell relevante Quellen kennt, wie etwa einzelne Datenbanken oder Internet-Angebote, die er dann sukzessive abfragt. Der Benutzer überführt also das gleiche Informationsbedürfnis mehrfach in eine Anfrage. In der Regel entstehen so während eines Informationsprozesses abhängig von den Zwischenergebnissen iterativ mehrere Anfragen. Dieser gesamte Prozess wird für jedes der Informationsangebote durchlaufen, da die Zwischenergebnisse in den verschiedenen Quellen unterschiedlich sind. Somit entsteht ein erheblicher zusätzlicher Aufwand.

Bei einer erschöpfenden Suche sollten so viele Quellen wie möglich durchsucht werden. In der Praxis kann ein Benutzer aber nicht beliebig viele Datenbanken abfragen. Eine mögliche Ausweichstrategie, die automatische Auswahl von vielversprechenden Datenquellen für eine Anfrage, hat sich in der Praxis als sehr schwierig erwiesen (cf. Rittberger 1995).

Für die Benutzer ist es also wünschenswert, viele getrennt erfasste und verwaltete Informationsangebote mit einer Anfrage zu durchsuchen. Dieser Wunsch erklärt einen Teil des Erfolgs von Internet-Suchmaschinen. Suchmaschinen liefern Daten verschiedenen Typs, wenn diese innerhalb einer Seite im Format HTML (Hypertext Markup Language) kodiert sind. Diese beinhalten Bilder, Filme, Tabellen und meist Text, wobei verschiedenste natürliche Sprachen vorkommen. Die technischen Probleme der Heterogenität beim Internet löst weitgehend die gemeinsame Basis HTML. Gerade diese Chance zur Heterogenität gilt als ein Grund für den großen Erfolg des Internet, das nur die Darstellungssprache HTML vorgibt, aber keine standardisierte Inhaltserschließung oder -beschreibung (cf. z.B. Kuhlen 1999:138).

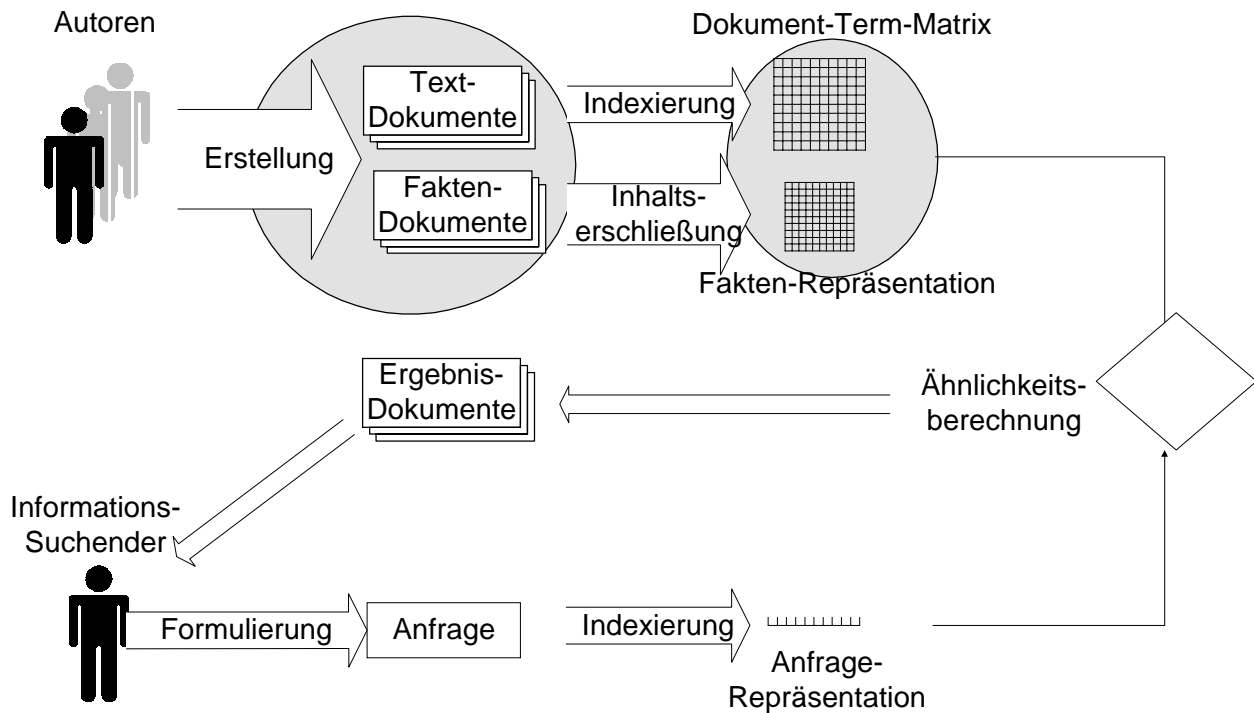


Abbildung 2-1: Der Information Retrieval Prozess bei heterogenen Datenbeständen (hier Texte und Fakten) und heterogenen Repräsentationen

Vor allem entstehen durch die Heterogenität aber semantische Probleme, die bis heute weitgehend ungelöst sind und von bestehenden Systemen wie den Internet-Suchmaschinen ignoriert werden. Das Grundproblem sind unterschiedliche Begriffsschemata, die unterschiedliche Daten repräsentieren. Häufig erfordern Fachgebiete unterschiedliche Spezialthesauri, da Fachtermini unterschiedlich gebraucht werden. Je nach Kontext ändern Begriffe ihre Semantik. Ein spezieller Thesaurus verspricht daher eine bessere Retrievalqualität in Fachgebieten, in denen die Semantik eines Terms klar definiert ist und konsistent benutzt wird.

Bestrebungen, die Heterogenität durch Standardisierung zu lösen, haben wenig Aussicht auf Erfolg. Dazu gehört die immer wieder geforderte Verwendung von weltweiten Thesauri zur Verschlagwortung von Internet-Dokumenten durch die Autoren. Es ist fragwürdig, ob Autoren überhaupt eine Indexierung vornehmen. Wenn ja, dann haben sie in der Regel keine dokumentarische Ausbildung und kennen nicht alle den gesamten Thesaurus, um ihr Dokument richtig einzuordnen. Zudem spielen wirtschaftliche Interessen bei der Auffindbarkeit von Dokumenten eine große Rolle und beeinflussen die Vergabe von Termen.

Semantische Probleme treten auch bei Wortlisten aus der automatischen Indexierung auf. Ein Term tritt in verschiedenen Korpora mit unterschiedlichen Verteilungshäufigkeiten auf. Aus Sicht des Information Retrieval, in denen sich Bedeutung auf Vorkommenshäufigkeit reduziert, ändert sich dadurch die Bedeutung. Die Bedeutung eines Terms konstituiert sich aus den Dokumenten, auf die er verweist. Damit hat ein Term in jeder Kollektion eine unterschiedliche Bedeutung. Eine automatische Analyse aller Texte ist organisatorisch unmöglich und auch nicht unbedingt sinnvoll, da empirische Untersuchungen gezeigt haben, dass in unterschiedlichen Kollektionen unterschiedliche Varianten der Inhaltserschließung

zur optimalen Retrievalqualität führen (cf. Womser-Hacker 1997:212 ff.). Dies ist plausibel, da ein Begriff in verschiedenen Fachgebieten oft unterschiedlich gebraucht wird.

Das Kernproblem der Heterogenität besteht also darin, Abbildungen zwischen verschiedenen Begriffs-Schemata - seien es Thesauri oder Wortlisten aus der automatischen Indexierung – zu erreichen. Diese Probleme gewinnen im Kontext digitaler Bibliotheken und unternehmensweiter DataWarehouses verstärkt an Bedeutung.

Abbildung 2-1 zeigt den Information-Retrieval-Prozess bei heterogenen Quellen. Die Situation des Benutzers ist nach wie vor identisch mit der im Standardfall, das System muss die Anfrage jetzt aber mit heterogenen Dokumenten vergleichen. Dadurch ergeben sich die oben besprochenen Problembereiche, die in Abbildung 2-1 mit Ellipsen markiert sind.

2.1 Heterogene Erschließung und Qualität

Heterogenität kann auch durch homogene Objekte entstehen, wenn diese z.B. unterschiedlich inhaltlich erschlossen werden oder unterschiedlicher Qualität sind. Selbst die Anwendung des gleichen Indexierungsverfahrens führt bei verschiedenen Dokument-Mengen zu heterogenen Termlisten. Unterschiedliche Verfahren wiederum extrahieren aus gleichen Dokumenten verschiedene Terme, was etwa Fusionsansätze ausnutzen. In jedem Fall repräsentieren unterschiedliche Term-Räume oder Repräsentationsmechanismen die Dokumente.

Ein weiterer Aspekt ist die Qualität, die sich auf die Dokumente oder auf die Inhaltsererschließung beziehen kann. Häufig erscheinen Benutzern Dokumente aus bestimmten Quellen besonders glaubwürdig, während sie z.B. Dokumenten von unbekanntem Informationsanbietern aus dem Internet zunächst wenig Vertrauen schenken. Die Mechanismen für das Vertrauensmanagement in Informationssystemen wie z.B. Autoritätsbeweis oder Zertifizierung untersucht Kuhlen 1999. Anbietern mit höherer Glaubwürdigkeit traut man in vielen Fällen auch eine bessere Inhaltsererschließung zu, bzw. man weiß, dass die Inhaltsanalyse professionell betrieben wird. Inhaltliche Kompetenz führt besonders bei der manuellen Indexierung eher zu besseren Ergebnissen. Dagegen sind die Dokumente kleiner Informationsanbieter teilweise nur schlecht oder gar nicht erschlossen.

In vielen Situationen wird ein Benutzer jedoch auch ein Dokument mit niedriger Qualität akzeptieren, wenn er ansonsten keine relevanten Dokumente erhält. Das Schalenmodell (cf. Krause 1996a, 1998) ist ein dezentraler Ansatz zur Bearbeitung dieser Problematik aus Sicht eines Fachinformationszentrums. Zentrale Informationsstellen haben bisher häufig monolithische Dokumenten-Kollektionen mit zentralen Thesauri aufgebaut. Die Forderung nach hoher Konsistenz und Homogenität der Daten führte zum Ausschluss potenziell relevanter Dokumente, die hinter diesen formalen Anforderungen zurückbleiben.

Das Schalenmodell berücksichtigt in einem Datenpool verschiedene Schalen von Dokumenten, die mit verschiedener Qualität erschlossen sind. Von einem inhaltlich möglichst detailliert erschlossenen Kernbereich nimmt die Qualität der Daten und die Übereinstimmung mit den Vorschriften zur Inhaltsererschließung in äußeren Schalen ab. Je nach Möglichkeit oder Wunsch siedeln sich kleinere, externe Anbieter auf verschiedenen Schalen an. Das Schalenmodell führt damit Vagheit ein und lockert die

bisher exakt verwaltete Konsistenz. Es ermöglicht dadurch auch die Integration automatisch und intellektuell indexierter Dokumente in einer Kollektion. Wird im Schalenmodell z.B. aus Kostengründen der Kernbereich verkleinert, kann die zentrale Stelle eine neue Schale mit automatisch erschlossenen Dokumenten einführen, um den bisherigen Umfang zu sichern.

Die zentrale Einrichtung erhält die Rolle eines Moderators, der den Kernbereich betreut und die Schalen definiert. Der Benutzer hat die Möglichkeit, beim Retrieval gezielt den Kernbereich oder bestimmte Schalen anzusteuern, je nachdem welche Qualitätskriterien er für diese Anfrage anlegt.

2.2 Lösungsansatz: Behandlung von Heterogenität durch Transformationen

Heterogenität führt meist zu heterogenen Beschreibungssprachen oder Indexierungsschemata für die Beschreibung der Retrieval-Objekte. Information-Retrieval-Systeme berechnen die Ähnlichkeit zwischen Anfrage und Dokument, um dem Benutzer die passendsten Dokumente für seine Anfrage zu liefern. Werden nun Anfrage und Dokument unterschiedlich repräsentiert, können die meisten Systeme keinen Vergleich durchführen. Das Ziel der Heterogenitätsbehandlung besteht in einer Vereinheitlichung der heterogenen Repräsentationen und damit gewissermaßen in einem Übersetzungsprozess zwischen verschiedenen Beschreibungssprachen. Wie bei natürlichen Sprachen beinhaltet dieser Prozess syntaktische und semantische Probleme. Bei den meisten Indexierungsschemata können die syntaktischen Probleme vernachlässigt werden.

Dieser Arbeitsbericht beschäftigt sich mit den semantischen Problemen. Die Beschreibungssprache besteht im Information Retrieval meist aus einer Liste von Termen und Gewichten. Das Vektorraum-Modell interpretiert die Terme als Achsen eines Raumes. Demnach spannen die Terme einen Merkmalsraum auf, in dem die Dokumente platziert sind, wobei die Achsen oder Terme prinzipiell gleichberechtigt sind. Jedes weitere Indexierungsverfahren führt zu einem weiteren Merkmalsraum. Die Anfrage ist bei heutigen Systemen von den Benutzern frei formuliert, ohne dass die Anfrageterme auf eine bestimmte erlaubte Menge beschränkt sind. Aber auch restringierte Zugänge kommen in der Praxis vor. Erlaubt eine Benutzungsoberfläche nur die Auswahl von Termen aus einem Thesaurus oder einer Klassifikation, dann entsteht die Anfrage in diesem Term-Raum mit weniger Dimensionen.

Der Vergleich einer Anfrage mit den Dokumenten erfolgt bei den meisten Systemen auf einer einheitlichen Basis und damit innerhalb einer Repräsentationsform. Ein Retrieval-System muss dazu die heterogen repräsentierten Objekte in ein Repräsentationsschema bringen.

Einen Lösungsansatz bieten Transformationen, die den Umstieg von einer Beschreibungssprache in eine andere und damit eine Abbildung von einem Term-Raum in einen anderen leisten. Sie erstellen ausgehend von der Repräsentation eines Objekts in einem Raum seine Repräsentation in einem anderen. Kuhlen etwa spricht von einer „postkoordinierenden Ordnung durch transformierende Anpassung“ (Kuhlen 1999:138). Präkoordination wäre in diesem Fall eine Standardisierung der Inhaltserschließung, wie sie in der dokumentarischen Tradition oft angestrebt wird, die aber in weltweiten Datennetzen nicht durchsetzbar ist.

Grundsätzlich kann die Anfrage in alle Beschreibungssprachen transformiert werden, dann müssen die Ähnlichkeitswerte aus verschiedenen Term-Räumen verglichen werden. Sinnvoller ist es daher meist, alle Retrieval-Objekte in einem Termraum zu repräsentieren.

Folgende Verfahren für Transformationen sind bekannt:

- Exakte Verfahren
- Vage Verfahren
 - Statistische Verfahren
 - Neuronale Netze
 - Hopfield- und Spreading-Activation-Netzwerke
 - Transformations-Netzwerk

Die statistischen Verfahren sind verbreitet und werden häufig eingesetzt. Daneben erscheint besonders das Transformations-Netzwerk als erfolgversprechend. Es besteht aus einem Backpropagation-Netzwerk, das mächtiger ist als neuronale Netze ohne versteckte Schichten (siehe Abschnitt 3.5.4.1).

Eine Unterscheidung der Verfahren für Transformationen ergibt sich auch aus der Art des benutzten Wissens:

- Exakte Verfahren nutzen bekanntes und explizit vorhandenes Wissen, das von Menschen formuliert und gepflegt wird (Terminologielisten, Regelwerke, Expertensysteme).
- Dagegen nutzen vage Verfahren Wissen, das in der benutzten Form nicht der Mensch formuliert, sondern das z.B. aus statistischen Zusammenhängen abgeleitet wird. Bei den meisten Verfahren gibt der Mensch Wissen in Form von Beispielen für die gewünschte Abbildung vor. Das System selbst implementiert daraus die Abbildungsfunktion.

2.3 Vage Verfahren für Transformationen

Vage Transformationen ergänzen exakte Verfahren wie Thesaurus-Erweiterungen und Konkordanzen und führen von einem oder mehreren Begriffen aus einem Begriffs-Schemata zu einem gewichteten Vektor von Begriffen in einem anderen Begriffs-Schemata.

Text-Retrieval setzt vermehrt vage Verfahren ein. Z.B. haben statistische Verfahren das traditionelle Boolesche Modell ersetzt und bilden heute die Basis zahlreicher kommerzieller Systeme. Auch neuronale Netze wurden in den letzten Jahren in Information-Retrieval-Systemen erfolgreich eingesetzt, wie der Überblick in Kapitel 4 zeigt.

Wie im Text-Retrieval läuft auch die Entwicklung der Heterogenitätsbehandlung hin zu Verfahren, die verstärkt Vagheit zulassen. Die folgenden Abschnitte zeichnen diese Entwicklung von erprobten statistischen Verfahren hin zu noch weitgehend experimentellen Verfahren auf der Basis neuronaler Netze. Bei diesen Verfahren ist kein intellektueller Aufwand für die Erstellung der Transformations-Funktion erforderlich. Statt dessen ist ein sogenanntes Doppelkorpus mit Trainings- oder

Lerndokumenten erforderlich, die nach beiden zu verbindenden Schemata erfasst wurden. Anhand dieser Datengrundlage erlernt oder modelliert das Verfahren die entsprechende Funktion.

Die Transformations-Ansätze zur Behandlung von Heterogenität eignen sich neben der Transformation von Anfragen für das Retrieval auch als Vorschlagmodus für die intellektuelle Inhaltserschließung. Dabei wird entweder eine Anfrage transformiert und der Benutzer erhält Vorschläge für andere Anfrage-Terme (cf. Schatz et al. 1996) oder ein Indexierer erhält ausgehend vom Volltext eines Dokuments einen Vorschlag für Indexterme aus einem kontrollierten Vokabular (cf. Chung et al. 1998).

Statistische Verfahren auf Basis des Vektorraum-Modells werden am häufigsten für vage Transformationen eingesetzt. Grundlage sind Assoziationswerte zwischen den Termen, die sich aus der Analyse von Kookkurrenzen ergeben. Die Gewichtungformeln für Assoziationen ähneln den Ansätzen zur Gewichtung von Termen in Dokumenten. Eine Assoziationsmatrix fasst das gemeinsame Auftreten von Termen in Dokumenten zusammen. Auch IR Systeme für homogene Daten greifen auf solche Werte zurück, um Anfragen um zusätzliche Terme zu erweitern (Query-Expansion) oder um Cluster von Dokumenten zu bestimmen. Die Beziehungen zwischen den Termen lassen auch auf Beziehungen zwischen den Dokumenten schließen. Ein Überblick über Assoziationsmaße im Information Retrieval bietet Ferber 1997.

Ein typisches Beispiel präsentieren Chen/Martinez et al. 1998 mit der Analyse einer Untermenge der Literaturdatenbank INSPEC mit 400.000 Dokumenten. Der aus den Kookkurrenzen automatisch generierte Thesaurus wurde mit dem intellektuell erstellten Thesaurus der INSPEC Datenbank verglichen. In dem Experiment beurteilten Testpersonen wie gut die Thesauri bei Eingabe eines Terms damit in Beziehung stehende Terme findet. Dabei stellten Chen/Martinez et al. 1998 fest, dass der automatisch generierte Thesaurus beim Retrieval von Termen einen besseren Recall erreicht als der intellektuell erstellte Thesaurus, während die Precision auf gleichem Niveau liegt.

2.4 Text Categorization

Der Begriff Text Categorization steht meist Ansätze, die zwischen dem Vokabular aus einer automatischen Indexierung und einem kontrollierten Thesaurus abbilden. Damit bildet Text Categorization eine Mischform aus automatischer und intellektueller Indexierung. Der Text wird zwar automatisch analysiert, den Texten werden aber Terme aus einer kontrollierten Liste zugewiesen, wie es sonst bei intellektueller Vorschlagwortung üblich ist. Text Categorization kann folgende Aufgaben erfüllen:

- Vorschlagmodus
Die Indexierer erhalten einen automatisch erstellten Vorschlag, den sie bei ihrer Arbeit berücksichtigen können.
- Information Filtering
In diesem Anwendungsfall kehrt sich die Rolle von Dokumenten und Anfragen um. Die Anfragen sind stabile, lange gültige Profile, die ein Benutzer festlegt. Die Dokumente sind flüchtige Elemente eines Stroms, aus dem nur die zu dem Profil passenden ausgefiltert werden. Solche Anwendungen sind etwa bei Presse-Agenturen, Nachrichtentickern oder anderen Arten von flüchtigen Dokumenten sinnvoll.

- **Behandlung von Heterogenität**
Aus Kostengründen wird oft selbst bei Institutionen, die intellektuell indexieren, diese Art der Inhaltserschließung nicht auf alle oder nicht auf neue Datenbestände angewandt. In solchen Fällen weist ein Text-Categorization-System die Thesaurus-Terme automatisch zu.

In den Bereich Vorschlagmodus fällt das System AIR/PHYS, das im Rahmen des Darmstädter Indexierungs-Ansatzes entstand (cf. Biebricher et al. 1988). Ziel war es, die Indexierungsqualität der intellektuellen Indexierung durch einen automatisch erstellten Indexierungsvorschlag zu erhöhen. Während die automatische Indexierung fast alle im Text vorkommenden Begriffe benutzt, ist die intellektuelle Indexierung auf einen Thesaurus beschränkt. Dessen Begriffe tauchen häufig in dieser Form nicht in den Texten auf. AIR/PHYS versucht daher Beziehungen zwischen dem automatischen Indexierungsergebnis und der intellektuellen Indexierung herzustellen. Dazu werden bereits intellektuell verschlagwortete Dokumente zusätzlich automatisch indexiert. Zwischen den beiden Repräsentationen ergeben sich Assoziationsfaktoren:

$$z(t,s) = \frac{h(t,s)}{f(t)} \quad (\text{Biebricher et al. 1988:334})$$

| | |
|----------|--|
| t | <i>Freitext-Term</i> |
| s | <i>Thesaurus-Deskriptor</i> |
| $f(t)$ | <i>Zahl der Dokumente, die t enthalten</i> |
| $h(t,s)$ | <i>Zahl der Dokumente aus f(t), denen s zugeordnet wurde</i> |

Der Thesaurus umfasst ca. 22.000 Begriffe. AIR/PHYS berechnete dafür 800.000 Assoziationsfaktoren, von denen es die 350.000 wichtigsten ins System übernahm. Diese Faktoren beschreiben quasi die Eigenschaften von Dokumenten, denen bestimmte Thesaurus-Begriffe zugeordnet sind. Die Übergangsfunktion enthält um weitere Thesaurus-Beziehungen, wie z.B. um 50.000 USE-Relationen und 170.000 BROADER TERM-Relationen. In vielen Fällen wurden so über die automatische Indexierung zufriedenstellende Deskriptoren aus dem Thesaurus zugewiesen.

AIR/PHYS ist ein gutes Beispiel für statistische Abbildungen zwischen heterogenen Repräsentationen. Es treten alle typischen Probleme auf, wie die Erstellung von doppelten Repräsentationen für viele Dokumente, als Basis für statistische Verfahren. Pro Deskriptor wurden nur durchschnittlich siebzehn Assoziationsfaktoren gespeichert, die auch nur aus dem Abstract gewonnen wurden, was ein zu geringer Ausschnitt aus den relevanten Eigenschaften sein kann.

Einen analogen Anwendungsfall stellt Ferber 1997 vor. Die dort geschilderten Experimente unterstreichen die Wichtigkeit der empirischen Überprüfung der angewandten Formel. Die verwendete Formel enthielt zwei Parameter, die den Einfluss der Häufigkeit von Thesaurus-Termen und Freitext-Begriffen im Korpus regelten. Die Veränderung dieser Parameter wirkte sich sehr stark auf die Qualität der Abbildung aus (Ferber 1997).

2.5 Latent Semantic Indexing für Transformationen

Latent Semantic Indexing (LSI) komprimiert die Dokument-Term-Matrix und nutzt dazu das mathematische Verfahren Singular Value Decomposition (SVD). Dabei wird der ursprüngliche Termraum auf in der Regel zwischen 100 und 300 Variablen reduziert. LSI hat in empirischen Tests insbesondere im Rahmen der TREC-Konferenz (siehe Abschnitt 4.7) positive Ergebnisse erzielt (Dumais 1994).

Latent Semantic Indexing arbeitet ähnlich wie eine Faktorenanalyse (z.B. *principal component analysis*), wie sie oft bei der statistischen Datenanalyse benutzt wird (cf. Rodeghier 1997:174ff.). Für die Erstellung einer Faktorenanalyse wird zunächst eine Korrelationsmatrix aller Merkmale erstellt, die analysiert, wie stark die beteiligten Merkmale korrelieren. Im ersten Schritt bestimmt die Faktorenanalyse wieviel jede Variable einer Matrix zur Erklärung der Varianz in einer Korrelationsmatrix beiträgt. Im zweiten Schritt wählt der Anwender die n wichtigsten Variablen in der originalen Matrix. Die Faktorenanalyse fasst dann alle Variablen zu n Faktorengruppen oder Hintergrundvariablen zusammen, wobei der Einfluss zueinander ähnlicher Faktoren in einer Gruppe gebündelt wird. Nachträglich lässt sich der Anteil jedes originalen Faktors an einer Hintergrundvariablen bestimmen und auch visualisieren (cf. Rodeghier 1997:174ff.). LSI produziert zueinander orthogonale oder rechtwinklige Faktoren. Dadurch bilden diese wieder ein Koordinatensystem. Die entstehenden Eigenschaften oder LSI-Dimensionen lassen sich nicht inhaltlich interpretieren, noch lässt sich ein Original-Term eindeutig einer Dimension zuordnen.

LSI nutzt aus, dass es sich bei der Dokument-Term-Matrix um eine spärlich besetzte Matrix handelt, bei der nur ein kleiner Teil der Zellen mit einem Gewicht belegt ist und die somit relativ wenig Information enthält. Die vorhandene Informationmenge fasst auch eine kleinere Matrix. Durch das mathematische Verfahren der Single Value Decomposition wird aus der ursprünglichen eine neue und kleinere Matrix gewonnen. Dabei werden die Singular Values und zwei weitere Matrizen bestimmt, von denen eine einen reduzierten Term-Raum und die andere einen reduzierten Dokumenten-Raum darstellt. Die zweite Matrix stellt eine komprimierte Beschreibung der Terme durch die Dokumente dar und. Aus der reduzierten Form kann die ursprüngliche vollständige Repräsentation wieder gewonnen werden. Die Berechnung erfolgt nach der Formel in der folgenden Abbildung:

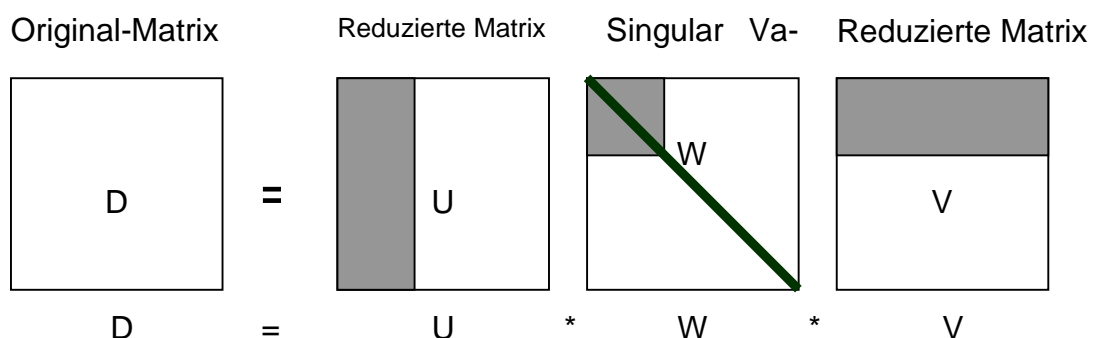


Abbildung 2-2: Schematische Darstellung der Reduktion mit Latent Semantic Indexing (cf. Syu et al. 1996)

Die Matrix W enthält die Singular Values, die alle auf der Diagonale liegen. Ihre stetig fallenden Werte sind ein Maß für die Wichtigkeit dieser Singular Value für die gesamte Matrix. Wichtigkeit bedeutet hier, wieviel diese Dimension zur Erzeugung der originalen Matrix beiträgt. Alle Singular Values reproduzieren die ursprüngliche, nicht

komprimierte Matrix. Außer der Matrix W entstehen die Matrizen U und V , wobei U einen komprimierten Term-Raum mit allen Dokumenten enthält, während V alle Terme mit weniger (Pseudo-) Dokument-Profilen beschreibt. Die Größe der beiden Matrizen ergibt sich zum einen aus der ursprünglichen Matrix und zum anderen aus der Anzahl der verwendeten Singular Values und damit der LSI-Dimensionen.

Hinter dem Einsatz von LSI Information Retrieval steht die Überlegung, dass unwichtige Dimensionen sehr wenig zur Semantik der Kollektion beitragen und bei der Berechnung der Ähnlichkeiten nur störend wirken. Zahlreiche weniger wichtige Dimensionen werden weggelassen und nur die ca. 100 bis 300 (cf. Berry 1992:4, Berry et al. 1995) wichtigsten repräsentieren die Dokumente. Da die LSI-Dimensionen letztlich komplexe Kombinationen realer Terme darstellen, wird durch das Weglassen der weniger dominanten Dimensionen der Einfluss der Terme auf das Ergebnis gewichtet. Dabei ist anzunehmen, dass der Einfluss vieler Terme sehr gering wird. Es entsteht eine gleichmäßiger verteilte Repräsentation wie bei allen Reduktionsverfahren. Im eigentlichen Retrievalprozess ersetzt die reduzierte Matrix über Dokumente und LSI-Dimensionen die ursprüngliche Dokument-Term-Matrix. Die Anfragen werden nachträglich mit den gleichen Singular Values bearbeitet und so gewissermaßen im reduzierten Raum platziert. Beim Vergleich zwischen Dokument und Anfrage werden die Ähnlichkeiten zwischen reduzierten Vektoren berechnet.

Diese Grundannahme hinter Latent Semantic Indexing ist nicht völlig plausibel, da auch ein Faktor oder Term, der wenig zur gesamten globalen Struktur der Matrix beiträgt, für einzelne Anfragen ausschlaggebend sein kann. Trotzdem zeigt LSI in empirischen Untersuchungen gute Resultate und hat sich auch für Massendaten bewährt.

Das folgende Beispiel veranschaulicht die Berechnung der neuen Matrix anhand einer Menge von vier Dokumenten mit sechs Termen. Es wurde mit einer experimentellen Software der Firma Bellcore berechnet, die auf dem SVD-Algorithmus von Lanczos beruht (cf. Berry et al. 1993). Tabelle 2-1 zeigt die ursprüngliche Dokument-Term-Matrix. Das LSI-Verfahren berechnet dafür die folgenden vier Singular-Values: 1,682 1,507 0,907 0,268. Die folgende Tabelle 2-2 zeigt die reduzierte Matrix, die sich daraus ergibt.

Tabelle 2-1: Beispiel für eine kleine Dokument-Term-Matrix

| | Term1 | Term2 | Term3 | Term4 | Term5 | Term6 |
|-------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Dok1 | 1 | 1 | 2 | | | |
| Dok2 | | 2 | 1 | 1 | | |
| Dok3 | | | | 1 | 1 | 1 |
| Dok4 | | 1 | | | | |

Tabelle 2-2: Beispiel: die mit LSI reduzierte Matrix

| | LSI-Dim1 | LSI-Dim2 | LSI-Dim3 | LSI-Dim4 |
|-------------|-----------------|-----------------|-----------------|-----------------|
| Dok1 | -0,435 | 0,724 | 0,533 | 0,025 |
| Dok2 | -0,493 | 0,281 | -0,772 | -0,283 |
| Dok3 | -0,747 | -0,620 | 0,230 | 0,058 |
| Dok4 | -0,089 | 0,101 | -0,256 | 0,956 |

In diesem Raum lassen sich nun auch wieder die Terme darstellen. Dabei zeigt sich, wie Latent Semantic Indexing die Verteiltheit der Repräsentation erhöht. Während in der originalen Matrix sowohl Term- als auch Dokument-Vektoren hauptsächlich aus Nullen bestehen und die Information in wenigen Einsen lokalisiert ist, verteilt sie sich bei der LSI-Matrix über alle Dimensionen. Anders ausgedrückt verteilt sich die vorher lokale Repräsentation eines Terms auf mehrere andere Dimensionen.

LSI wird in der Regel für Retrieval benutzt, kann aber auch für Transformationen eingesetzt werden, so z.B. für multilinguales Retrieval (cf. Young 1994). Dabei erfolgt die Transformation nach der Reduktion auf den Raum mit niedriger Dimensionalität. Wie bei LSI als IR-Verfahren besteht die Hoffnung, dass durch die Reduktion unwichtige Teile der ursprünglichen Matrix verlorengehen und nur die semantisch relevantesten Anteile übrig bleiben. Die Transformation soll dann auf Basis der wichtigen Strukturen besser gelernt werden.

Eine einfache Termerweiterung innerhalb des gleichen Vokabulars mit LSI testen Schütze/Pedersen 1997 und gelangen für die Tipster Kollektion (aufgegangen in TREC, siehe Abschnitt 4.7) zu besseren Ergebnissen als mit einem Vergleichsexperiment ohne Term-Erweiterung. Die Assoziationsmatrix zwischen den 450.000 Termen wurde schrittweise auf eine Matrix mit 450.000 mal 20 Elementen reduziert. Dumais et al. 1997 beschreiben eine vergleichbare Anwendung für multilinguales Retrieval.

2.6 Fazit: Heterogenität im Information Retrieval

Informationssysteme müssen aufgrund der weltweiten Vernetzung immer häufiger heterogene Daten integrieren, um für den Benutzer tolerant zu wirken. Die Heterogenität von Erschließungsverfahren führt v.a. zu semantischen Problemen. Die Ansätze zur Behandlung von Heterogenität lassen sich in sich in exakte und vage Verfahren unterteilen. Die exakten, deduktiven Verfahren wie etwa Konkordanzen zwischen Begriffssysteme erfordern einen hohen intellektuellen Aufwand, der nicht in allen Anwendungsfällen erbracht wird.

Die folgende Übersicht fasst die Verfahren für vage Transformationen zusammen:

- Statistische Verfahren
 - Familie ähnlicher Verfahren basierend auf Kookkurrenzen
 - statistische Verfahren im IR allgemein sehr erfolgreich
 - für Transformationen in realen Anwendungen bereits erfolgreich eingesetzt
- Hopfield- und Spreading-Activation-Netzwerke
 - einfache neuronale Netze ohne sub-symbolische Fähigkeiten
 - den statistischen Verfahren sehr ähnlich
- Transformations-Netzwerk
 - basiert auf dem Backpropagation-Algorithmus
 - eine experimentelle Anwendung für Transformationen innerhalb gleichem Vokabular bekannt
- COSIMIR-Netzwerk für Transformationen

- experimentelles IR-Verfahren basierend auf dem Backpropagation-Algorithmus
- für Transformationen adaptiert
- erste Experimente mit kleinen Datenmengen erfolgreich

Die statistischen Verfahren sind erprobt und in den meisten Projekten mit großer Sicherheit einsetzbar. Aber auch das Transformations-Netzwerk hat eine hohe Plausibilität und wurde bereits in kleinen Experimenten getestet. Nach einem Überblick über die Fähigkeiten neuronaler Netze folgt in Kapitel 4 eine Diskussion von Information Retrieval Systemen auf der Basis neuronaler Netze.

3 Grundlagen neuronaler Netze

Künstliche neuronale Netze sind lernfähige Systeme, die Information tolerant und robust verarbeiten. Wie ihr natürliches Vorbild - die Nervensysteme von Lebewesen - bestehen sie aus zahlreichen einfachen, miteinander verknüpften Prozessoren. Über ihre Verknüpfungen tauschen diese Prozessoren oder Neuronen numerische Informationen in Form von Aktivierung aus. Folgendes Zitat formuliert dieses Prinzip sehr prägnant:

"Die Informationsverarbeitung geschieht durch eine große Anzahl von relativ einfachen Prozessoren, die in einem dichten Netzwerk miteinander verbunden sind. Diese Prozessoren (auch Units genannt) arbeiten lokal, jeder für sich allein, und kommunizieren mit anderen Units nur via Signale, die sie über die Verbindungen senden." (Dorffner 1991: 16)

Künstliche neuronale Netze eignen sich für die Simulation menschlicher Fähigkeiten im Bereich Perzeption und Kognition, bei denen starre, regelverarbeitende Systeme nicht zum Erfolg führen. Im folgenden werden sie einfach als neuronale Netze bezeichnet, da in dieser Arbeit zwischen natürlichen und künstlichen Netzen keine Verwechslungsgefahr besteht. In der Regel ist immer von künstlichen neuronalen Netzen die Rede.

Einige Beispiele für die Anwendung neuronaler Netze sind Spracherkennung, Gesichtererkennung (cf. Zell 1994), Erkennung von handgeschriebenen Postleitzahlen (cf. de Waard 1994), Vorhersage von Aktienkursen (cf. Refenes/Azema-Barac 1994), Analyse von Herztönen (cf. Alonso-Betanzos et al. 1999) und Bestimmung der Toxizität von Flüssigkeiten (cf. Grauel et al. 1999).

Dieses Kapitel bietet eine kurze Einführung in die Architektur und Funktionsweise neuronaler Netze. Nach einer Darstellung des natürlichen Vorbilds und der kognitionswissenschaftlichen Aspekte folgt eine Vorstellung von Backpropagation-Netzwerken. Backpropagation ist der am häufigsten eingesetzte Netzwerktyp und ein typischer Vertreter lernfähiger Systeme. Das COSIMIR-Modell und das Transformations-Netzwerk beruhen auf dem Backpropagation-Modell. Daran schließt eine systematische Einführung der Grundlagen neuronaler Netze und verschiedener Modelle an. Dabei liegt der Schwerpunkt auf den Modellen, die für den state-of-the-art Neuronale Netze im Information Retrieval (siehe Kapitel 4) wichtig sind. Im weiteren Verlauf wird das Backpropagation-Modell erneut aufgegriffen und ausführlicher diskutiert.

Als Einführung in neuronale Netze eignet sich Scherer 1997. Einen ausführlichen systematischen Überblick bietet Zell 1994. Rojas 1993 diskutiert formale und algorithmische Grundlagen.

3.1 Natürliche neuronale Netze

Das Vorbild für die Informationsverarbeitung in (künstlichen) neuronalen Netzen ist die Architektur biologischer Nervenzellenverbände, die aus sehr vielen stark

vernetzten Einheiten besteht. Diese Neuronen können Impulse von anderen Neuronen über Nervenbahnen aufnehmen und eventuell selbst *feuern*, also elektrische Impulse an andere Neuronen weiterleiten. Die folgende Darstellung greift auf Scherer 1997 und Zell 1994 zurück.

Nervenzellen bestehen aus dem Zellkern, der Nervenfasern (Axon) und den Dendriten, länglichen Verbindungsleitungen. Signale kommen über die Dendriten in der Zelle und laufen über das Axon an andere Zellen weiter. Die Verbindungsstelle zwischen Axon und Dendrit ist die Synapse. Entlang der Nervenbahnen Axon und Dendriten verlaufen Signale oder Impulse in elektronischer Form durch Veränderung des Potentials zwischen den Zelleninneren und dem Zellenäußeren. Die Stärke eines Signals ist dabei immer gleich. Die Stärke eines Reizes wird also nur durch die Frequenz kodiert.

An den Synapsen verbreitert sich das Axon. Es endet kurz vor der nächsten Nervenzelle, wodurch ein Spalt entsteht. Diese Lücke überbrückt das elektrische Signal auf chemischem Weg durch Auslösung sogenannter Neurotransmitter. Diese Moleküle werden am Ende des Axons ausgelöst und auf der anderen Seite des Spalts von Rezeptoren aufgenommen. In der anderen Nervenzelle läuft das Signal in elektrischer Form weiter. Erreichen die in einer Zelle ankommenden Signale einen Schwellenwert, so schickt sie einen Impuls ausgehend vom Ausgangspunkt des Axons entlang einer Nervenbahn. Die Veränderung der Durchlässigkeit von Synapsen bildet eine der Grundlagen für das Lernen von Lebewesen.

Messungen zur Geschwindigkeit dieser biologischen Vorgänge zeigen, dass das Gehirn im Vergleich zu einem seriellen Computer äußerst langsam arbeitet (cf. Kinnebrock 1992:11). Jedoch aktiviert das Gehirn immer sehr viele Neuronen gleichzeitig. Aufgrund dieser Parallelverarbeitung übertrifft das menschliche Gehirn den Computer bei den meisten kognitiven Fähigkeiten.

3.2 Kognitionswissenschaftliche Aspekte

Auch wenn die künstlichen Neuronen stark von ihrem natürlichem Vorbild abweichen, so sind neuronale Netze ein plausibleres Modell für Kognition als ein serieller Rechner. Deshalb haben neuronale Netze auch in der Kognitionswissenschaft ein neues Paradigma geschaffen: den Konnektionismus.

Bis zum Erfolgsgang der neuronalen Netze seit Mitte der 80er Jahre galt das *Physical Symbol System* (PSS) als angemessenes Modell für Kognition und damit auch für Künstliche Intelligenz. Das PSS besagt, dass Symbolverarbeitung notwendig und hinreichend für Intelligenz ist. Nach der PSS-Hypothese, wie sie in Newell/Simon 1976 manifestiert ist und etwa in Newell et al. 1989 wieder aufgegriffen wird, besteht Kognition aus der Manipulation von Symbolen. Intelligenz entsteht demnach durch die Verarbeitung von Zeichen, die symbolhaft auf etwas anderes verweisen.

Dagegen treten in neuronalen Netzen Prozesse auf, die sich einzeln nicht symbolisch interpretieren lassen, sondern erst im Zusammenhang mit vielen anderen Verarbeitungsschritten sinnvoll sind. Im Gegensatz zum PSS wird Information sub-symbolisch verarbeitet.

Smolensky 1988 formuliert einen integrativen Ansatz. Er führt das sub-symbolische Paradigma ein und setzt es als "intuitive processor" dem "conscious rule interpreter" gegenüber (Smolensky 1988:4f.). Demnach besteht Kognition nicht im Abarbeiten von symbolischen Regeln, sondern die Vorgänge im Gehirn wie auch die in neuro-

nalen Netzen laufen auf einer tieferen Ebene ab. Insbesondere intuitives Expertenwissen lässt sich nicht als Regelsystem modellieren.

Zwar ergibt sich an der Oberfläche oft regelfolgendes Verhalten, tatsächliche Intelligenz entsteht aber nur unter dem Level der Symbole. Somit sind die Abläufe in neuronalen Netzen auch nicht interpretierbar wie etwa ein Computerprogramm in einer prozeduralen Programmiersprache, und viele der Neuronen repräsentieren keineswegs Entitäten der realen Welt.

Smolensky 1988 sieht das sub-symbolische Paradigma als Basis kognitiver Vorgänge. Intuition lässt sich sub-symbolisch erklären und durch das Zusammenspiel zahlreicher einfacher Prozesse entstehen auf einer höheren Ebene regelfolgende Systeme (cf. Smolensky 1988, Dorffner 1991).

3.3 Das Backpropagation-Modell: ein erster Überblick

Um komplexe Probleme zu lösen, arbeiten serielle Computer meist mit einer Menge von Regeln. So kennt ein Schachcomputer die erlaubten Züge und einige Strategie-Regeln und berechnet daraus die besten Züge.

Viele Probleme sind jedoch nicht so gut formalisierbar wie Schach. Besonders bei Expertenentscheidungen ist oft schwer oder unmöglich, die zugrundeliegenden Regeln widerspruchsfrei zu formulieren. Dies gilt z.B. für finanzielle oder medizinische Diagnoseprobleme. Ein Finanzexperte schätzt aufgrund seiner Expertise zwar die Kreditwürdigkeit eines Kunden gut ein, er kann aber nur schwer allgemeingültige Regeln für diesen Vorgang angeben. Der Experte trifft seine Entscheidung durch eine ganzheitliche Beurteilung der Daten zu einer Person. Ebenso verlässt sich ein Arzt bei einer Diagnose kaum auf ein einzelnes Symptom oder eine einzige Messung. Vielmehr bewertet er die relevanten Daten ganzheitlich.

Experten, denen es schwer fällt, ihr intuitives Wissen in Regeln zu formulieren, erwerben es durch Erfahrung, also durch die Kenntnis vieler Beispiele. Durch das Analysieren von Beispielen ist das Wissen auch wieder zugänglich. In dieser Form erhalten auch viele lernende Systeme Expertenwissen. Anstatt wie bei einem typischen Schachprogramm Regeln anzugeben, sind diese Systeme mit zahlreichen Beispielen für die gewünschte Abbildung von Symptomen auf Entscheidungen ausgestattet.

Backpropagation-Netzwerke sind ein typischer Vertreter solcher lernender Systeme. Sie lernen anhand von Beispielen komplexe Funktionen und können dabei ähnlich wie Experten ihr Wissen nicht in Form von Wenn-dann-Regeln ausgeben. Der Benutzer eines Netzwerks weiß also in den meisten Fällen nicht, *warum* ein Netzwerk eine Entscheidung trifft. Diesen Nachteil nimmt man aber in Kauf, wenn ansonsten keine Modellierung möglich scheint. Backpropagation-Netzwerke bestehen aus zahlreichen Neuronen, die in Schichten angeordnet sind. Eine Schicht dient dem Input, dann folgen Schichten für die Berechnung von Zwischenstufen und schließlich gelangt die Aktivierung in eine Output-Schicht. Input und Output bilden definierte Schnittstellen zur Welt, dort werden Daten angelegt und abgelesen. Bei einem Netz zur Beurteilung der Kreditwürdigkeit repräsentieren die Input-Neuronen z.B. die Daten eines Kunden. Ein Neuron steht z.B. für den momentanen Kontostand, ein anderes z.B. für das Alter. Bei der Eingabe eines neuen Kunden wird das entsprechende Neuron auf den jeweiligen Wert gesetzt. Welches Wissen hier

eingeht, hängt vom Anwendungsfall ab. Oft entschieden Experten, welche Daten für das Abbildungsproblem erforderlich sind. Der Output steht für das gewünschte Ergebnis. So kann im Beispiel die Kreditwürdigkeit zwischen Null und Eins liegen, wobei ein höherer Wert eine hohe Kreditwürdigkeit bedeutet.

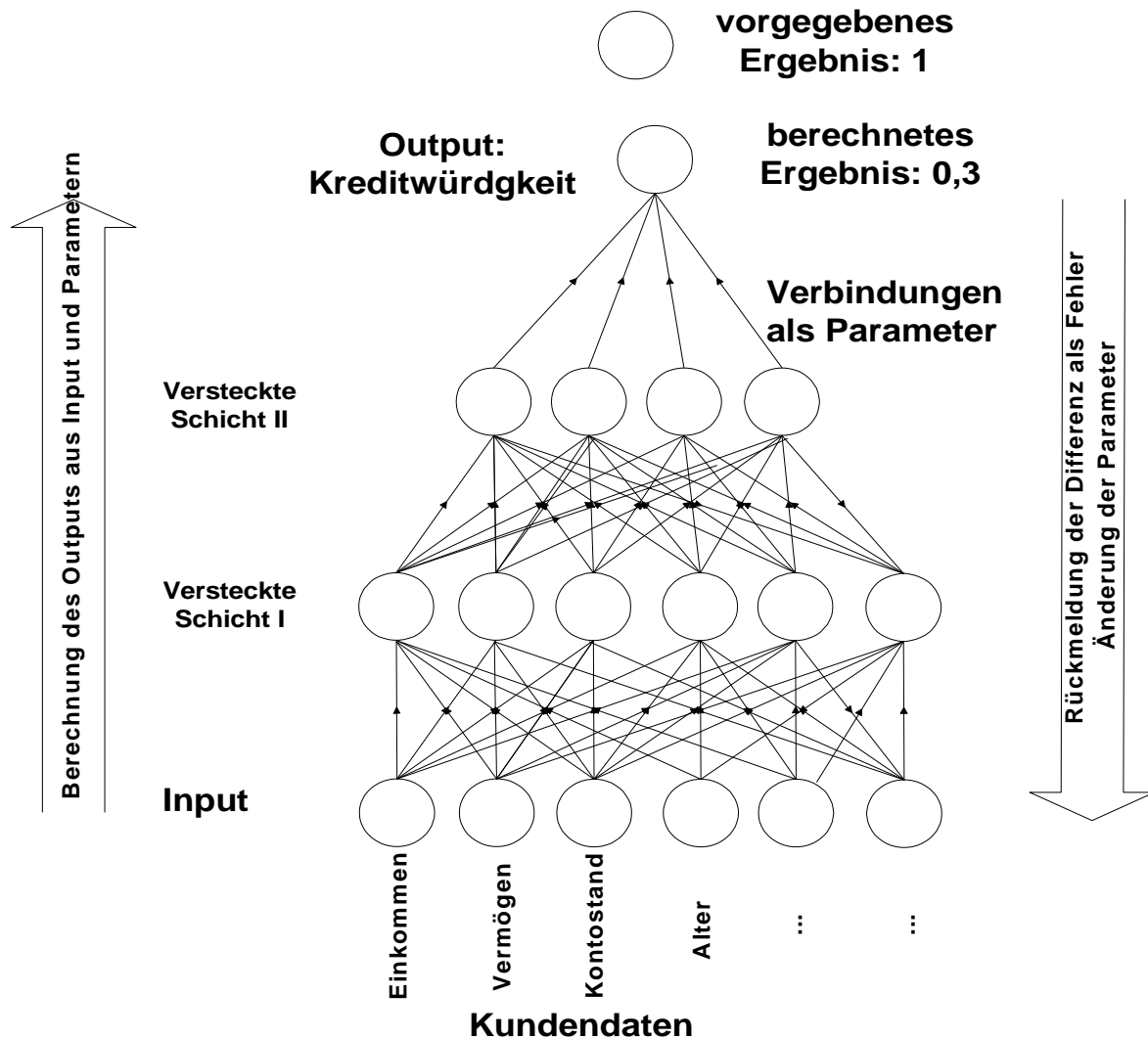


Abbildung 3-1: Funktionsweise eines Backpropagation-Netzwerks

Zwischen den Neuronen der verschiedenen Schichten befinden sich gewichtete Verbindungen. Sie bilden die Parameter des Netzes, die anfangs zufallsgesteuert initialisiert und beim Lernen richtig eingestellt werden. Das Lernen verläuft in zwei Schritten. Zunächst berechnet das Netz nach Eingabe der Daten ein Ergebnis im Output. Da noch nichts gelernt wurde, ist dieses Ergebnis sicher falsch. D.h., es stimmt nicht mit dem Wert überein, den ein Experte als Beispiel vorgegeben hat. Die Differenz zwischen Ergebnis und Vorgabe misst den Fehler des Netzes.

Im zweiten Schritt wird dieser Fehler vom Output in Richtung Input, also gewissermaßen rückwärts ins Netz gespeist. Nun verändern sich die Werte der Verbindungen. Sie stellen sich so ein, dass sich der Fehler für dieses Trainingsbeispiel etwas verringert. Dies wird nun für alle Beispiele häufig wiederholt. Bei Erfolg zeigt sich, dass der Fehler immer kleiner wird und das Netz schließlich die gewünschte Funktion lernt.

Ein Backpropagation-Netzwerk ist demnach eine Funktion mit sehr vielen Parametern, die aus einem Input einen Output berechnet. Diese Parameter werden zu Beginn zufallsgesteuert eingestellt. Anders als bei einem Schachprogramm ist weder die Anzahl, die Bedeutung noch der *richtige* Wert dieser Parameter bekannt. Durch die kontinuierliche Präsentation von Beispielen, stellt das Netz die Parameter so ein, dass sich aus den Input-Daten der richtige Output ergibt. Jede Verbindung kann als eine Mikro-Regel betrachtet werden, die jedoch für sich alleine keinen Sinn ergibt. Nur im Zusammenspiel aller Verbindungen entsteht die richtige Funktion.

Nach dem Erlernen einer Funktion anhand von vorliegenden Beispielen, erfolgt der Einsatz des Netzes. Nun berechnet es aus unbekanntem Input-Daten den Output ab. Im Beispiel bestimmt es die Kreditwürdigkeit neuer Kunden.

3.4 Aufbau und Funktionsweise neuronaler Netze

Die Bestandteile neuronaler Netze fasst ein erstmals von Rumelhart/McClelland 1986 vorgestellter Rahmen zusammen, der sich gut für die Kategorisierung eignet. Die folgende Darstellung orientiert sich an diesem Rahmen.

3.4.1 Neuronen

Jedes Modell besteht aus einer Menge von Prozessoren, die in der Gehirnmetapher den Neuronen entsprechen. Diese Prozessoren werden auch Knoten oder Units genannt. Sie gruppieren sich häufig in Mengen oder Schichten mit ähnlichen Eigenschaften. Neuronen repräsentieren Konzepte, Eigenschaften oder *Micro-Features* und sind symbolisch nicht interpretierbar.

Aktivierungszustände

Die Neuronen oder Knoten besitzen Aktivierungszustände (a_n in Abbildung 3-2). Jedes Neuron hat zu einem Zeitpunkt eine bestimmte numerische Aktivierung. Die Aktivierungszustände aller Knoten und damit der Aktivierungszustand des gesamten Netzes beschreibt bei einem Netz mit n Neuronen ein n -stelliger Vektor. In der Praxis werden die Aktivierungszustände jedoch häufig auf das Intervall von 0 bis 1 normalisiert. In der Gehirnmetapher entspricht der Aktivierungszustand des Netzes dem Kurzzeitgedächtnis. Die Aktivierung entspricht meist dem Output (o_n in Abbildung 3-2) eines Neurons.

- **Aktivierungsfunktion**
Jedes Neuron berechnet lokal seine Aktivierung (a_n in Abbildung 3-2) nach einer definierten Aktivierungsfunktion aus der alten Aktivierung und aus dem Input, den das Netz liefert. Teilweise verwenden die Modelle Schwellwertfunktionen. Ein Neuron leitet in der Regel nur dann Aktivierung weiter („feuert“), wenn der gesamte Input (net-input_n in Abbildung 3-2) eine bestimmte Schwelle überschreitet. Für viele Netzwerkmodelle wie das Backpropagation-Modell (siehe Abschnitt 3.5.4) sind komplexe Aktivierungsfunktionen erforderlich.
- **Ausbreitungsfunktion**
Diese Funktion errechnet das Netzeingangssignal eines Knoten aus den Ausga-

besignalen der mit ihm verbundenen Knoten und den Gewichten der dazwischenliegenden Verbindungen. Das Eingangssignal für alle Units lässt sich wieder als n -stelliger Vektor interpretieren und wird meist einfach als Summe der Produkte zwischen Gewichtung und Output-Signal berechnet.

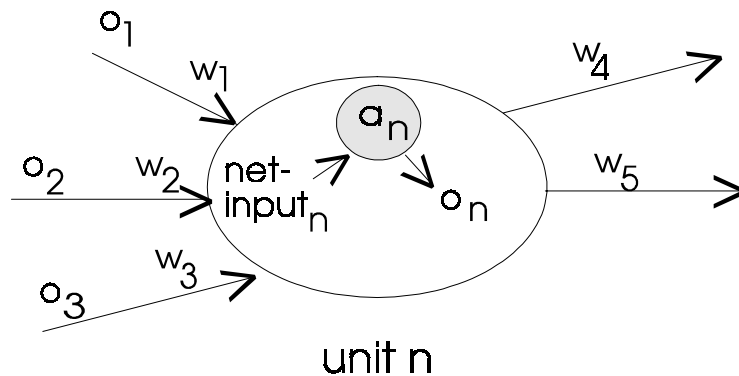


Abbildung 3-2: Die Funktionsweise eines künstlichen Neurons: (cf. Dorffner 1991: 17)

- o_i : Output der Unit i
- w : Gewicht einer Verbindung
- $net-input_n$: Input in Unit n
- a_n : Aktivierung der Unit n

3.4.2 Vernetzung

Die Neuronen sind durch gewichtete Verbindungen vernetzt. Bei einem Netz mit n Neuronen entsprechen diese Gewichtungen einer $n \times n$ -Matrix. Die Verbindungsmatrix legt die Architektur des Netzwerks fest. Im Hopfield-Modell sind alle Werte der $n \times n$ -Matrix belegt. In vielen anderen Modellen sind die Neuronen in Schichten gruppiert, wobei innerhalb der Schichten keine Verbindungen erlaubt sind. Manche Architekturen beschränken Verbindungen auf eine Richtung, andere verfügen über bidirektionale Links. Beim Backpropagation-Netzwerk laufen die Verbindungen nur in eine Richtung, damit steht für die Gewichte auch nur eine Hälfte der $n \times n$ -Matrix zur Verfügung.

Die Werte der Verbindungsgewichte können in der Regel sowohl positive als auch negative Werte annehmen.

3.4.3 Lernregel

Die wichtigste Eigenschaft neuronaler Netze ist die Selbstorganisation oder das Lernen. Dabei verändert das Netz die Stärke der Verbindungen und damit ihre Durchlässigkeit. Die Aktivierungsausbreitung berechnet aus einem Input den Output, während das Lernen diese Abbildungsfunktion verändert.

Lernverfahren sind das wichtigste Unterscheidungsmerkmal für neuronale Netze. Die häufigsten Typen sind überwachte und unüberwachte Verfahren. Bei unüberwachten Modellen analysiert das Netz nur die Daten selbst, während beim überwachten Lernen der gewünschte Output als Teacher benutzt wird. Dieser kann in Form von beispielhaften Zuordnungen vorliegen.

Ein dritter Typ ist das Reinforcement Lernen, das zwischen den beiden anderen liegt. Der gewünschte Output wird nicht exakt, sondern nur als graduelle oder vage Entscheidung vorgegeben. Das Netz erfährt gewissermassen nur die Richtung, in die es sich entwickeln soll.

Die Hebb'sche Lernregel ist eine typische Vertreterin des unüberwachten Lernens. Sie besagt, dass häufig benutzte Verbindungen gestärkt und selten benutzte geschwächt werden (*Use it or lose it!*). Bei überwachten Lernverfahren muss die erwünschte Ausgabe für einige Eingaben bekannt sein. Aus der Differenz zwischen tatsächlichem und erwünschtem Output ergibt sich ein Fehler. Anhand dieses Fehlers versucht das Netz nun seine Verbindungen so einzustellen, dass der Fehler kleiner und die gewünschte Ausgabe somit besser erreicht wird. Die entsprechende Lernregel wird als Delta-Regel bezeichnet, da die Differenz die entscheidende Rolle spielt. Die allgemeinste Form bestimmt die notwendige Änderung an der Gewichtungstärke zwischen zwei Units und berücksichtigt neben dem Fehler die Aktivierung des empfangenden Neurons vor der Verbindung:

Allgemeine Delta-Regel

$$\Delta w_{ij} = \eta \text{ Output}_i (\text{Teacher}_j - \text{Aktivierung}_j)$$

η Lernrate

Zell 1994:85

Überwachtes Lernen ähnelt Näherungsverfahren, die Funktionen nach Vorgabe einiger Punkte möglichst genau eingrenzen. Das am häufigsten benutzte Modell ist der Backpropagation Algorithmus (siehe 3.5.4).

3.4.4 Schnittstelle zur Umgebung

Neuronale Netze besitzen mit den Input- und Output-Neuronen Schnittstellen nach aussen. Bei der Eingabe wird die Aktivierung der Input-Neuronen von aussen gesetzt. Die Aktivierungszustände von Output-Neuronen bilden die Ausgabe.

Input-Daten werden meist vorverarbeitet und durch lineare oder nicht-lineare Transformationen dabei auf das Intervall zwischen Null und Eins normalisiert. Die adäquate Vorverarbeitung entscheidet oft über die Qualität eines Netzes.

3.5 Modelle

Der im letzten Abschnitt aufgespannte Rahmen erlaubt als Ausprägungen die verschiedensten neuronalen Netzwerkmodelle.

3.5.1 Kohonen-Netze

Kohonen-Netze sind ein unüberwachter Lernalgorithmus, der Strukturen in Daten aufdeckt. Sie werden auch als selbstorganisierende Karten (Self Organizing Maps, SOM) bezeichnet (cf. Kohonen 1984, Scherer 1997:93ff., Zell 1994:179ff.). Kohonen-Netze bestehen aus zwei Schichten, einer Eingabe-Schicht und einer Kohonen- oder Ausgabe-Schicht, in der eine topologische Karte der Daten entsteht. Die Anzahl der Neuronen in der Eingabe-Schicht ergibt sich aus der Dimensionalität der Eingangsdaten. Die Struktur der Kohonen-Schicht wird vom Anwender vorgegeben. Eine zwei- oder dreidimensionale Anordnung unterstützt die spätere Visualisierbarkeit der Daten. Abbildung 3-3 zeigt ein Kohonen-Netzwerk mit zweidimensionaler Anordnung der Output-Neuronen. Jede Unit der Eingabe-Schicht ist mit allen Units der Kohonen-Schicht verbunden, so dass jedem Kohonen-Neuron ein Gewichtsvektor von der Größe des Eingabe-Vektors zugeordnet ist. Zudem sind die Kohonen-Neuronen untereinander verbunden.

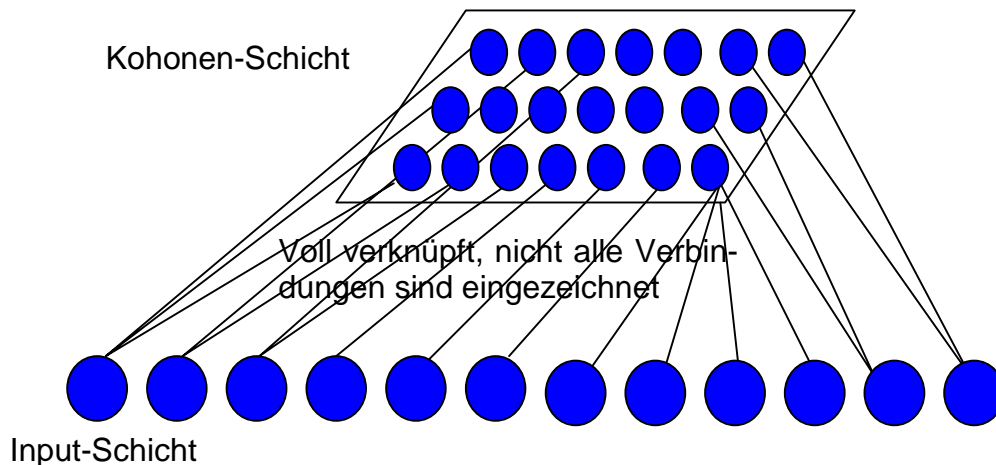


Abbildung 3-3: Schematisches Kohonen-Netzwerk

Die Eingabe-Vektoren werden mit den Gewichtsvektoren verglichen. In der Regel ist die Euklidische Distanz das Ähnlichkeitsmaß. Das Neuron mit der geringsten Distanz oder der höchsten Ähnlichkeit zum Eingabemuster gewinnt und erhält die gesamte Aktivierung. Die Gewichtungen des Gewinner-Neurons in die Eingabe-Schicht werden so modifiziert, dass die Ähnlichkeit weiter steigt. Geometrisch betrachtet verschiebt der Algorithmus den Gewichtsvektor in Richtung des Eingabevektors. Soweit arbeitet das Kohonen-Netzwerk wie andere Clustering-Verfahren. Um die topologische Struktur zu erzeugen, verändern sich auch die Gewichtsvektoren der Nachbar-Neuronen des Gewinners. Dies erfordert eine Definition von Nachbarschaft, die durch verschiedene Funktionen wie etwa die Gauss-Funktion oder der Kosinus implementieren. Diese Funktionen liefern ein Maß für die Entfernung jedes Neurons in der Kohonen-Schicht vom Gewinner-Neuron, das die Intensität der Gewichtsänderung beeinflusst. Je näher ein Neuron dem aktivierten Neuron ist, desto stärker wird sein Gewichtsvektor adaptiert. Die Vektoren sehr naher Neuronen werden somit immer in ähnliche Richtungen verschoben. Dadurch entstehen Cluster, in die ähnliche Muster abgebildet werden.

Kohonen-Netzwerke werden u.a. in der Spracherkennung für Sprechererkennung, für Bildverarbeitung in der Medizin und für die Extraktion der Eigenschaften von Chromosomen eingesetzt (cf. Kohonen 1997a).

3.5.2 Adaptive Resonance Theory (ART)

Adaptive Resonance Theory (ART) ist eine Familie von mathematisch elaborierten Netzwerken, die Inputmuster in Cluster einteilen. Ein besonderes Merkmal von ART gegenüber Kohonen-Netzen ist die zweifache Überprüfung der Ähnlichkeit zwischen Input und Zielcluster. ART I verarbeitet nur binäre Vektoren. ART II ist eine Erweiterung für Muster mit reellen Zahlen. ART III zieht biologische Vorgänge beim Übergang eines Nervensignals auf eine andere Zelle mit in die Modellierung ein und versucht so, kognitive Vorgänge adäquater zu modellieren. Parameter sind etwa die prä- und postsynaptischen Mengen von Neurotransmittern. ARTMAP ist eine überwacht lernende Variante des Clustering-Verfahrens, bei der die korrekte Einteilung der Muster vorgegeben wird. Mehr Informationen zu ART bietet Zell 1994.

Die Grundidee von ART beruht auf der Beobachtung, dass alle anderen Lernverfahren nicht angemessen auf Änderungen in einer Lernmenge reagieren. Dies ist biologisch nicht plausibel, da natürliche Nervensysteme neu auftretenden Muster flexibel verarbeiten, sie in den Wissensbestand integrieren und die bisherigen Erfahrungen dabei nicht vollständig überschreiben.

Zunächst klassifiziert ART einen Input-Vektor im wesentlichen wie im Kohonen-Netz in einer Winner-take-all-Schicht. Dann überprüft eine weitere Schicht von Neuronen (die Vergleichsschicht), inwieweit die Zuordnung angemessen ist, indem sie mit einer anderen Ähnlichkeitsfunktion prüft, ob die Ähnlichkeit einen vorgegebenen Schwellenwert übersteigt. Erreicht die Ähnlichkeit nicht das geforderte Maß, wird das Muster zur erneuten Klassifikation an die Erkennungsschicht zurückverwiesen. Findet das ART-Netz kein befriedigendes Cluster, so fügt es ein neues Neuron in der Erkennungsschicht hinzu, dessen Cluster für den Eingabe-Vektor zuständig ist. Wie im Kohonen-Netz wird nach jeder Zuordnung der Clustervektor in Richtung des Eingabemusters verändert.

Dieses Verfahren erlaubt es, bereits gelernte Zuordnungen von Mustern zu Clustern nicht zu überschreiben und sie so nicht zu *vergessen*.

3.5.3 Assoziativspeicher

Hopfield-Netze sind unüberwacht lernende Netzwerke, die meist als assoziativer Speicher dienen. Sie reagieren auf unvollständige oder gestörte Muster mit sinnvollen Ausgaben (cf. Zell 1994:197ff., Scherer 1997:125ff.). Ein assoziativer Speicher ruft ein gespeichertes Muster direkt über den Inhalt ab und nicht über einen Index wie traditionelle Datenbanken. Hopfield-Netze verarbeiten meist binäre Muster.

Hopfield-Netze bestehen aus einer untereinander voll vernetzten Schicht von Neuronen und besitzen symmetrische Verbindungen. Alle Neuronen dienen sowohl als Input als auch als Output. In der Retrievalphase wird ein Muster angelegt und einem der gespeicherten Muster zugeordnet. Dabei läuft die Aktivierungsausbreitung wie in anderen Netzen und das Netz konvergiert zu einem Zustand, der einem der gespeicherten Muster entspricht. Als Stoppkriterium wird die Änderung der

Aktivierungsmatrix zwischen zwei Schritten berechnet und mit einem Schwellwert verglichen.

Um zu erreichen, dass das Netz immer zu einem der gespeicherten Muster konvergiert, stellt ein Algorithmus die Verbindungen entsprechend ein. Das Hopfield-Netz weicht hierbei von den meisten anderen Netzwerk-Modellen ab und lernt die Gewichte nicht schrittweise, sondern berechnet sie aus den zu speichernden Mustern. Dazu ist eine Energie-Funktion definiert, die das Energie-Niveau des Netzes ausdrückt:

$$E = -\frac{1}{2} \sum_i \sum_{j \neq i} w_{ij} akt_i akt_j + \sum_i akt_i \Theta_i$$

Scherer 1997:129

Dieses Niveau wird durch die Aktivierungsausbreitung minimiert. Dabei summiert jedes Neuron die gewichteten Output-Werte jedes anderen Neurons, mit dem es in Verbindung steht. Dieser Input löst die binäre Aktivierungsfunktion ein, wenn ein bestimmter Schwellenwert überschritten wird.

Die Initialisierung stellt sicher, dass jedes Minimum dieser Energie-Funktion eines der Muster repräsentiert.

$$w_{ij} = \sum_s w_{ij}^s = \sum_s x_i^s x_j^s$$

s Index über die Muster

Scherer 1997:131

Das Hopfield-Netz konvergiert nicht immer zu einem gespeicherten Muster, sondern läuft oft in ein lokales Minimum der Energie-Funktion. Da das Lernverfahren immer versucht, die Energie zu minimieren, kann es das lokale Minimum nicht mehr verlassen. Dem versucht eine Variante des Hopfield-Netzwerks entgegenzuwirken, die sogenannte Boltzmann-Maschine (cf. Zell 1994: 207 ff.).

Boltzmann-Maschine und Hopfield-Netz unterscheiden sich nur durch die Aktivierungsfunktion und das Lernverfahren, das bei der Boltzmann-Maschine nicht deterministisch sondern stochastisch abläuft. Die Lernfunktion enthält einen zufalls-gesteuerten Faktor. Damit wird die Energie des Netzwerks in einem Schritt nicht unbedingt verkleinert, sondern kann auch steigen. Erreicht das Verfahren ein lokales Minimum, dann erreicht die Hopfield-Regel keine Verbesserung mehr. Der zufalls-gesteuerte Beitrag zur Aktivierungsfunktion ermöglicht auch die Erhöhung des Fehlers bei einem einzelnen Lernschritt. Steckt ein Netz in einem lokalen Fehlerminimum, erlaubt der Boltzmann-Lernalgorithmus durch die vorübergehende Erhöhung des Fehlers das Verlassen des Minimums. Diese Hoffnung ist durchaus plausibel, da die Wahrscheinlichkeit einer Erhöhung des Fehlers mit der Anzahl der Trainingsschritte steigt.

Das Lernverfahren minimiert wie beim Hopfield-Netz die gesamte Energie im Netz. Die Boltzmann-Netze entwickeln aber die Energie-Metapher weiter und übertragen sie auf das Auskühlen eines glühenden Metalls. Das Lernverfahren wird dieser Metapher folgend als *simulated annealing* bezeichnet. Das Kühlen eines Metalls erfolgt

bei einem Kristallisationsvorgang sehr langsam, um allen Molekülen die Möglichkeit zu geben, sich optimal auszurichten.

Die Energie des Netzes berechnet sich nach der Summe aller Aktivierungen und aus den paarweise zwischen den Neuronen herrschenden Energien, die sich aus den beiden Aktivierungen und dem Gewicht der Verbindung ergibt:

$$E = - \sum w_{ij} o_i o_j + \sum \theta_i o_i$$

o_i Output = Aktivierung von Neuron i

w_{ij} Gewicht der Verbindung zwischen Neuron i und j

θ_i Schwellenwert von Neuron i

Zell 1994:208

Die Wahrscheinlichkeit, dass ein Neuron aktiviert ist, sinkt mit der im System enthaltenen Gesamtenergie. Demnach wird ein Neuron mit folgender Wahrscheinlichkeit aktiviert:

$$p_k = p(o_k \equiv 1) = \frac{1}{1 + e^{-\Delta E/T}}$$

ΔE Energiedifferenz zwischen zwei Zuständen

T Temperatur-Parameter

Zell 1994:209

Der Parameter T wird zwischen den Schritten langsam reduziert. Das Lernverfahren für die Boltzmann-Maschine unterscheidet zwischen sichtbaren und versteckten Neuronen. Dabei sind sichtbare Neuronen am Input und Output beteiligt, während versteckte Neuronen nur der internen Verarbeitung dienen. Beim Lernen wird zunächst ein Mustervektor angelegt, wobei die Aktivierung der Eingabe-Neuronen konstant bleibt. Nachdem ein Energiegleichgewicht im gesamten Netz erreicht ist, spielen die Aktivierungen der versteckten Neuronen die entscheidende Rolle für das Lernen. Nun verändern sich die Verbindungsgewichte so, dass diese Aktivierungen auch ohne externen Input das gleiche Aktivierungsmuster in den sichtbaren Neuronen induzieren. Dazu modifiziert eine lokale Lernregel die Verbindungen zwischen sichtbaren und versteckten Neuronen.

Die Forschung im Bereich Assoziativspeicher befasst sich v.a. mit den Fragen, wie gut das Retrieval bei unvollständigen Mustern arbeitet und mit ihrer Speicherkapazität (cf. Jagota et al. 1995, Schwenker et al. 1996).

3.5.4 Backpropagation-Netze

In den 60er Jahren galt das Perzeptron als ein erfolgversprechendes Netz, das beliebige Beziehungen zwischen Input- und Output-Mustern lernen kann.

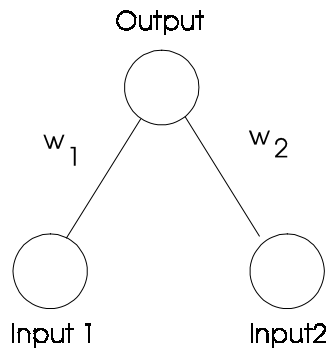


Abbildung 3-4: Aufbau eines Perzeptrons (McClelland/Rumelhart 1988:124)

Ein Perzeptron besteht aus zwei verbundenen Schichten. Beim Lernen wird an der Input-Schicht ein Vektor angelegt und daraus der Output berechnet. Dieser Ausgabevektor wird mit dem gewünschten Zielvektor verglichen. Durch Anwendung der Delta-Regel (siehe Abschnitt 3.4.3) versucht das Perzeptron die durch die Beispiele vorgegebenen Funktionen zu lernen. Alle Muster liegen dabei mehrmals am Input des Perzeptron an. Eine Phase, in der alle Muster einmal präsentiert werden, heisst Epoche. Dieses einfache Verfahren findet immer eine Lösung, wenn eine existiert. Die Lösung besteht in einer Menge von Gewichtungen, die dafür sorgen, dass das Netz den Input richtig auf den Output abbildet (cf. McClelland/Rumelhart 1988:123).

3.5.4.1 Vom Perzeptron zum Backpropagation-Netzwerk

Gegen Ende der 60er Jahre analysierten Minsky/Papert (1969) in ihrem Buch *Perceptrons* formal die Mächtigkeit des Lernverfahrens und konnten zeigen, dass das Perzeptron nur linear trennbare Funktionen lernen kann. Da bereits eine so einfache Funktion wie das logische XOR (Entweder-Oder) nicht linear trennbar ist, lernt das Perzeptron nur eher einfache Funktionen.

Lineare Trennbarkeit bedeutet anschaulich, dass sich in einem zweidimensionalen Koordinatensystem eine Gerade zwischen die Punkte der verschiedenen zu lernenden Klassen ziehen lässt. Abbildung 3-5 zeigt, dass zwar die logischen Funktionen AND und OR linear trennbar sind, aber nicht XOR.

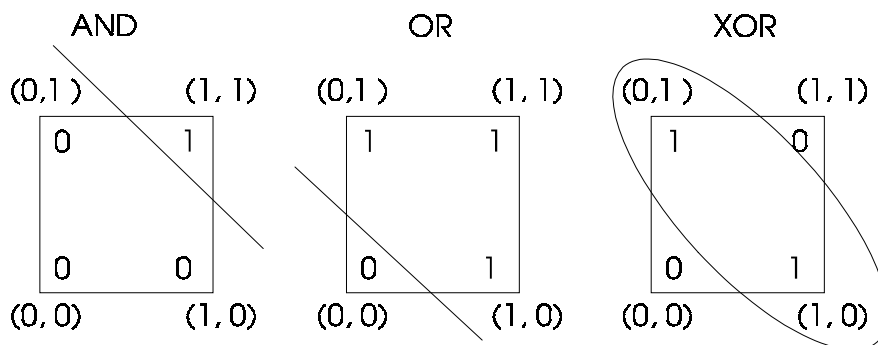


Abbildung 3-5: Lineare Trennbarkeit (McClelland/Rumelhart 1988:124)

Vermutlich sind zahlreiche Probleme nicht linear trennbar (cf. Zell 1994:100f.). Minsky/Papert (1969) hatten bereits erkannt, dass eine Zwischen-Schicht im Perzeptron diese Beschränkung aufhebt. Cybenko (1989) zeigt, dass eine versteckte Schicht prinzipiell ausreichend ist, um jede kontinuierliche Funktion anzunähern.

Jedoch existierte für ein dreistufiges Netzwerk keine Lernregel. In den 70er und 80er Jahren wurde ein Verfahren entdeckt, das den Fehler auch über eine Zwischen-Schicht zurückverfolgt. Dieser Backpropagation-Algorithmus löst auch nicht linear trennbare Probleme wie etwa die XOR-Funktion. Ein entsprechendes Backpropagation-Netz besteht also aus mindestens drei Schichten, wobei die Schichten zwischen Input und Output versteckte oder *Hidden Layer* heißen.

Für Gewichtsveränderungen von der Zwischen-Schicht in Richtung Input-Schicht wird der gewünschte Output aus der Summe der Produkte aus Fehler und entsprechender Gewichtung in der darüberliegenden Schicht gemittelt. Das Fehler-Signal für alle Neuronen, die nicht Output-Neuronen sind, folgt aus den Fehlern der oberen Schichten:

$$\delta_{pj} = f'_j(\text{net}_{pj}) \sum_k \delta_{pk} w_{kj}$$

Rumelhart et al. 1986b:329

Ein Faktor ist die Ableitung der Aktivierungsfunktion, die die Vorwärtsschritte definiert. Für ein Backpropagation-Netzwerk muss die Aktivierungsfunktion damit an jeder Stelle ableitbar sein. Im Perzeptron wird dagegen die neurobiologisch plausible Schwellenwertfunktion benutzt, die die Ableitungsbedingung nicht an jeder Stelle erfüllt. Backpropagation-Netzwerke setzen z.B. die Sigmoid-Funktion ein:

$$\text{aktivierung}_i = \frac{1}{1 + e^{-\text{netinput}_i \cdot \text{const.}}}$$

cf. Rumelhart et al. 1986b:329

Konvergiert der Fehler in einem Netz während des Trainings, so hat es die gewünschte Abbildung gelernt. Der Fehler zwischen Output und gewünschtem Ergebnis dient als Maß für die Qualität der Abbildung.

3.5.4.2 Probleme und Lösungsansätze

Der Backpropagation-Algorithmus besitzt einige Schwächen, die bei der praktischen Anwendung berücksichtigt werden müssen.

- Die Trainingszeiten sind oft sehr lang.
- Es ist nicht garantiert, dass der Algorithmus das globale Fehlerminimum findet. Da der Fehler bei jedem Schritt minimiert wird, ist es möglich, dass nur ein lokales Minimum erreicht wird (cf. Zell 1994:112f.). Das gleiche Problem tritt beim Hopfield-Modell auf (siehe Abschnitt 3.5.3). Dies lässt sich für die Veränderung nur eines Gewichts gut zweidimensional veranschaulichen. In Abbildung 3-6 steuert der Algorithmus nur berab, und bleibt im

lokalen Minimum B stecken, obwohl das globale Minimum des Fehlers bei C eine wesentlich bessere Lösung darstellt.

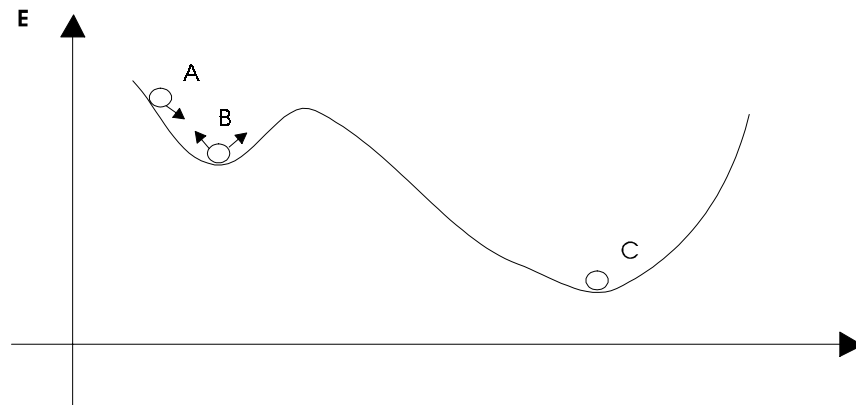


Abbildung 3-6: Fehlerverlauf in Abhängigkeit von einem Gewicht (cf. Dorffner 1991:111)

- Die Entwicklung und der Lernprozess erfordern zahlreiche heuristische Parameterseetzungen, von denen im folgenden einige diskutiert werden:
 - Ende des Trainings (*Overlearning*)
Verfolgt man die Fehlerentwicklung in der Trainings- und Testmenge während des gesamten Trainings, so ergibt sich etwa folgendes Bild:

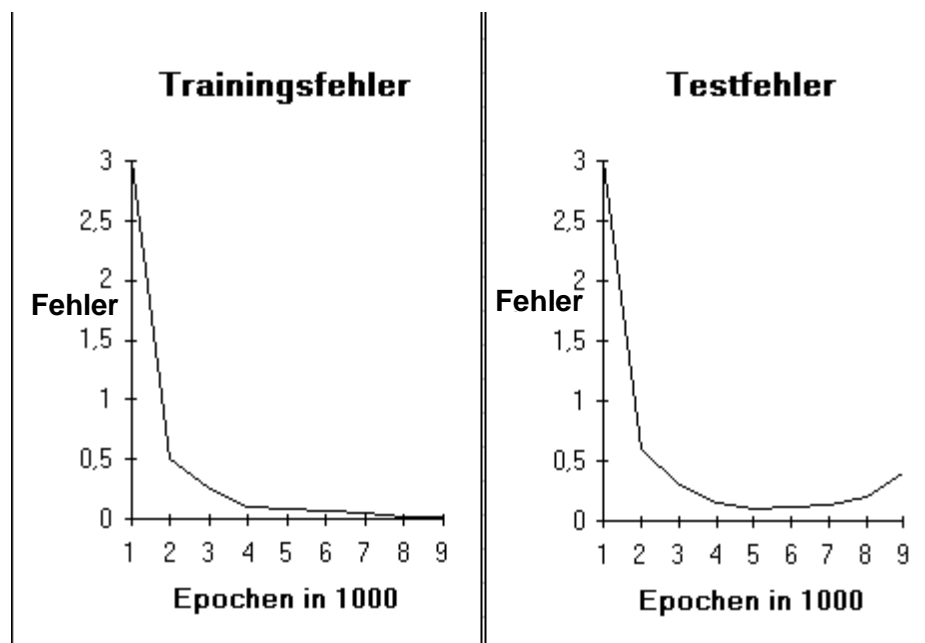


Abbildung 3-7: Fehler in Trainingsmenge und Testmenge in Abhängigkeit von der Anzahl der Trainings-Epochen

Der Fehler in der Trainingsmenge konvergiert mit steigender Epochenzahl. Der Testfehler sinkt nur bis zu einem bestimmten Zeitpunkt und steigt dann

wieder an. Durch längeres Training sinkt die Generalisierungsfähigkeit. Diesen Effekt nennt man *overlearning*. Eine plausible Erklärung dieses Phänomens ist, dass das Netz sich bei längerer Lernzeit immer mehr auf die Eigenheiten in den Trainingsfällen spezialisiert. Das Training wird daher meist bei einem Minimum in der Testmenge beendet.

- **Anzahl der Neuronen**
Die Daten bestimmen die Zahl der Units in Input- und Output-Schicht. Die Zahl der versteckten Neuronen dagegen wird durch Experimentieren optimiert. Sie hängt von der Komplexität der anzunähernden Funktion ab. In der Regel wird sie zwischen der Zahl der Input- und Output-Neuronen liegen.
- **Die Anzahl der Verbindungen**
Die Zahl der Verbindungen hängt natürlich stark von der Zahl der Neuronen ab. Meistens sind Feed-Forward-Netze nur zwischen den Schichten vernetzt. Es besteht aber die Möglichkeit, Verbindungen einzuführen, die Schichten überspringen (*shortcut connections*) und die z.B. vom Input direkt zum Output laufen und eine versteckte Schicht auslassen. Andererseits kann man auch geringere Grade der Vernetzung festlegen, indem man Maße für die maximale Zahl von Verbindungen angibt, die von einer Unit ausgehen oder bei dieser ankommen (*fan-in, fan-out*). Allein zufälliges Eliminieren von Gewichten führte in Experimenten von Le Cun 1989 zu Verbesserungen der Performanz. Die Einführung eines *weight decay*, der bei jedem Schritt alle Verbindungen leicht schwächt, unterstützt die Suche nach unwichtigen Verbindungen, die dann aus dem Netz entfernt werden (cf. z.B. Moody 1992).

3.5.4.3 Weitergehende Parameter

Zahlreiche Verbesserungen und Varianten für den Backpropagation-Algorithmus versuchen, diesen Schwächen zu begegnen. Die meisten versuchen, die Konvergenz durch mathematische Verfahren zu beschleunigen. Verschiedene empirische Studien zeigen, dass die Auswirkungen der einzelnen Parameter sehr stark vom Problem abhängen (cf. z.B. Cherkassky/Vassilas 1989, Jervis/Fitzgerald 1993, Schiffmann et al. 1993, Cichocki/Unbehauen 1993).

- **Einsatz alternativer Aktivierungsfunktionen**
Die Aktivierungsfunktion steuert die Aktivierung eines Neurons in Abhängigkeit von der ankommenden Aktivierung. Eine Veränderung kann die gesamte Dynamik eines Netzes verändern.
- **Änderungen an den Variablen der Aktivierungsfunktion**
Viele Aktivierungsfunktionen benutzen Variablen, die ihren Verlauf beeinflussen. Veränderungen betreffen wiederum das gesamte Netzwerk.
- **Initialisierung der Gewichte**
Alle Gewichte eines Netzes werden vor dem Training zufällig initialisiert. Das Training beginnt damit an einem zufälligen Punkt der Fehlerfunktion. Der wichtigste Parameter ist dabei das Intervall für die Initialisierungsgewichte. In der Regel

wird das Intervall $[-1; 1]$ benutzt. Meist wird ein Netz mehrfach initialisiert und mit den gleichen Parametern trainiert. Das beste Ergebnis wird dann verwendet.

- **Multitask-Learning**
Bei Multitask-Learning wird einem Backpropagation-Netzwerk zusätzliche Information im Output gegeben. Das Netz lernt neben der erwünschten Abbildung noch weitere Abbildungen, die für den Anwendungsfall aber nicht primär interessant ist. Damit erhält der Lernalgorithmus mehr Fehlerinformation, um die Verbindungen einzustellen (cf. Caruana 1995/97). Dies widerspricht der Intuition zusätzliche Information als Input einzuleiten.
- In mehreren empirischen Untersuchungen konnte dadurch die Qualität für die erwünschte Abbildung verbessert werden. So lernte z.B. ein Netz von Bartlmae 1998 die Kreditwürdigkeit von Länder zu bestimmen. Das Lernen zusätzlicher Informationen über diese Länder im gleichen Netz erhöhte die Qualität für die eigentliche Aufgabe.
- **Varianten des Lernverfahrens**
 - *Learning Rate Adaption*
Die Erhöhung der Lernrate ist die einfachste Möglichkeit, Backpropagation zu beschleunigen, sie führt aber häufig zu unerwünschten Nebenwirkungen. Die Lernrate wird selten größer 0,3 gewählt. Es könnte zum einen zu Oszillationen (cf. Cichocki/Unbehauen 1993:146) aufgrund konkurrierender Muster kommen. Diese Probleme haben zu Überlegungen geführt, variable Lernraten zuzulassen, die zu Beginn des Lernens hoch sind und gegen Ende hin abnehmen (cf. Cichocki/Unbehauen 1993:144).
 - **Stochastische Lernverfahren**
Mathematisch weniger gut zu erfassen sind Methoden, die stochastische Elemente in den Backpropagation-Prozess einbringen. Dies kann zum einen bei der Aktivierung und zum andern beim Lernen geschehen. Am einfachsten wird dies durch zufallsgesteuerte Elemente in der Aktivierungs- und Lernfunktion realisiert. Durch eine derartige *noise injection* kann das Lernverfahren eventuell aus einem lokalen Minimum entkommen (cf. Dorffner 1991:271). Dies ähnelt dem Algorithmus der Boltzmann-Maschine (siehe Abschnitt 4.2).
- **Momentum**
Die Einführung eines zusätzlichen Terms versucht, Backpropagation zu beschleunigen. Das sehr häufig eingesetzten Backpropagation mit Momentum addiert bei jeder Gewichtsveränderung ein Bruchteil der letzten Gewichtsveränderung hinzu. Dadurch soll die aktuelle Lerngeschwindigkeit und -richtung beibehalten werden. Durch den Momentum-Term würden Oszillationen verhindert und flache Strecken in der Fehlerfunktion könnten schneller durchschritten werden.

3.6 Simulationssoftware DataEngine

Künstliche neuronale Netze werden meist nicht als Hardware mit physikalisch vorhandenen Neuronen implementiert, sondern durch Software simuliert. Anstelle der parallelen Verarbeitung tritt eine serielle Implementierung, bei der ein zentraler Prozessor die Berechnungen für jedes Neuron nacheinander ausführt. Dies beeinflusst lediglich die Geschwindigkeit des Netzes, während die Vorteile wie Robustheit, Fehlertoleranz und bei Backpropagation-Netzwerken subsymbolische Verarbeitung von Information. Einen Überblick über vorhandene Software bietet Searle 2000.

DataEngine¹ ist ein kommerzielles Produkt, das unter Windows läuft (cf. z.B. Zimmermann 1995). Es erlaubt die Realisierung von Backpropagation-Netzwerken, von Kohonen- und von Fuzzy Kohonen-Netzen. Abbildung 5-3 zeigt die Benutzungsoberfläche von DataEngine.

Die besondere Stärke von DataEngine liegt in der Kombination von Fuzzy Logik und neuronalen Netzen in einem Programm. Durch das Bilden von Blöcken können Verarbeitungsprozesse realisiert werden, an denen beide Techniken beteiligt sind. Eine weitere Stärke liegt in den Bearbeitungsmöglichkeiten für Daten, was meist den größten Teil der Arbeit mit neuronalen Netzen ausmacht. Ein mächtiger Spreadsheet-Editor erlaubt typische Vorverarbeitungsschritte und die Anwendung statistischer Funktionen.

3.7 Fazit: Grundlagen neuronaler Netze

Neuronale Netze sind eine relativ neue, aber bereits erfolgreich eingesetzte Technik zur Verarbeitung von Informationen, die zum Paradigma des Soft-Computing gehört. Neuronale Netze ermöglichen vage und tolerante Informationsverarbeitung und sind lernfähig. Schwächen bestehender Information Retrieval Systeme erfordern u.a. diese Eigenschaften. Lernen erhöht die Adaptivität von Retrieval Systemen und die Heterogenität erfordert tolerante Verarbeitungsverfahren. Beim neuronalen Backpropagation-Algorithmus steht der Benutzer im Mittelpunkt, wenn er die Trainingsbeispiele liefert. Diese Benutzerorientierung wird im Information Retrieval gefordert. Deshalb liegt es nahe, neuronale Netze im Information Retrieval einzusetzen und zu testen.

¹ <http://www.mitgmbh.de/mit/sp/index.htm>

4 Neuronale Netze im Information Retrieval

Das Potenzial neuronaler Netze als Modell für Information Retrieval schätzen viele Forscher als sehr hoch ein. Die folgenden Zitate verdeutlichen dies:

„... connectionist models provide a generic computational methodology of potential applicability to most aspects in information retrieval. To the extent that a problem can be formulated as a connectionist model, enormous computational power can be brought to bear in solving that problem through massively parallel processing.“ (Doszkocs et al. 1990:228)

„[It]... makes one wonder why the research toward neural nets in information retrieval still is restricted to such a small school“ (Scholtes 1992:638)

„The learning property of backpropagation networks and the parallel search property of the Hopfield network provide effective means for identifying relevant information items in databases.“ (Chen 1995:201)

Welche Rolle spielen neuronale Netze aber tatsächlich im Information Retrieval? In kommerziellen Systemen sind sie kaum zu finden. Der folgende state-of-the-art Bericht gibt einen ausführlichen Überblick über die vorhandenen Systeme und ihre Leistungsfähigkeit.

4.1 Historischer Überblick

Neuronale Netze erlebten in den 80er Jahren eine Renaissance, wobei das Erscheinen des Sammelbands Rumelhart/McClelland 1986 als markantes Ereignis gilt. Dieser Einschnitt etablierte vor allem der Backpropagation-Ansatz und führte zu zahlreichen Anwendungen.

Zu dieser Zeit wurden bereits vernetzte Systeme im Information Retrieval diskutiert, jedoch handelte es sich dabei fast durchwegs um semantische Netze, die Verbindungen symbolisch definieren, während die Verbindungen zwischen künstlichen Neuronen rein numerisch sind. Die semantischen Spreading-Activation-Netzwerke werden in Abschnitt 4.3.2.7 besprochen.

Bereits wenige Jahre später erschienen bei der ACM SIGIR-Konferenz (Association of Computing Machinery, Special Interest Group Information Retrieval) zwei theoretisch fundierte Ansätze, in denen die Potenziale der Spreading-Activation-Netzwerke insbesondere im Bereich Lernen bereits deutlich wurden und die sich im Laufe der Zeit weiterentwickelten (cf. Kwok 1989, Belew 1989). Einen guten Überblick über die frühe Entwicklung dieser Spreading-Activation-Netzwerke für Information Retrieval bieten Doszkocs et al. 1991. Die beiden Ansätze entwickelten sich in verschiedene Richtungen weiter. Belew bemühte sich bei der weiteren Arbeit um eine Integration semantischer und neuronaler Netze (siehe Abschnitt 4.3.2.6). Der Ansatz von Kwok 1989 ging in dem System PIRCS auf (siehe Abschnitt 4.3.2.1) und bewährte sich im

Rahmen der TREC Initiative an Massendaten (siehe Abschnitt 4.7). In den 90er Jahren folgten weitere Systeme, wobei neben PIRCS noch Mercure (siehe Abschnitt 4.3.2.4) den Schritt zur Anwendung in TREC schaffte. Diese Erfolge der Spreading-Activation-Netzwerke deuten auf eine gewisse Konsolidierung hin, die sich jedoch nicht in der kommerziellen Entwicklung niedergeschlagen hat. Nach wie vor bekennt sich keines der bekannteren und auf dem Markt erhältlichen IR-Systeme zu neuronalen Netzen. Lediglich die Firma Infoseek berichtet, in ihrer Suchmaschine Technologie der Firma HNC eingebunden zu haben¹. HNC hat mit MatchPlus auch an TREC teilgenommen (siehe Abschnitt 4.7).

4.2 Retrieval mit Assoziativspeichern

Der Begriff Retrieval ist in der Neuroinformatik bereits belegt, und wird für das Retrieval von Mustern aus Assoziativspeichern benutzt. Vor allem Hopfield-Netzwerke (siehe Abschnitt 3.5.3) wirken als assoziative Speicher (content-addressable memory). Sie speichern Muster durch das Einstellen ihrer Verbindungen. Ein zentraler Begriff bei Assoziativspeichern ist die Energie im Netzwerk, die ein Maß für die Summe aller Aktivierungen darstellt. Ein gespeichertes Muster stellt dann ein Energie-Minimum im Netz dar. Nach einem Input konvergiert das Netz zu einem solchen Energie-Minimum. Hopfield-Netze eignen sich somit vor allem für die Zuordnung von unvollständigen oder beschädigten Mustern zu einem der gespeicherten Muster. Retrieval bedeutet hier also Finden eines möglichst ähnlichen Musters. Dies gilt für die am weitesten verbreitete Form der Assoziativspeicher, die *auto-associative memories*, bei denen die Neuronen zugleich Input und Output bilden. Die hetero-assoziativen Systeme dagegen nutzen für Input und Output verschiedene Neuronen und realisieren so eine Abbildung zwischen beliebigen Räumen verschiedener Dimensionalität. Damit ähneln sie den als Spreading-Activation-Netzen im IR sehr populären Systemen, bei denen Input und Output in eigenen Schichten angeordnet sind und die in Abschnitt 4.3 diskutiert werden. Die Übergänge zwischen diesen Netzwerktypen sind fließend.

In ihrer auto-assoziativen Form eignen sich Hopfield-Netzwerke kaum für den IR-Prozess, auch wenn die Analogie zwischen Retrieval aus Assoziativspeichern und Retrieval im Information Retrieval zunächst plausibel erscheint. Als Muster speichert das Netz die Dokumente und die Anfrage bildet das unvollständige Muster, das eingegeben wird. Jedoch konvergiert das Netz dann nur zu einem einzelnen Muster und liefert somit nur ein Dokument. In einem Information Retrieval Prozess will der Benutzer aber in der Regel eine größere Menge von Dokumenten. Im Prinzip ist ein Hopfield-Netzwerk ein System zu Klassifizierung und eignet sich zumindest in Reinform nicht für Information Retrieval.

4.2.1 Hopfield-Netzwerke

Personnaz et al. 1986 schlagen ein Hopfield-Netz für Information Retrieval vor. Sie zeigen, dass darin keine Zyklen auftreten und betonen, dass das System orthogonale Muster besonders leicht findet. In einem kleinen Beispiel speichern Personnaz

¹ <http://info.infoseek.com/doc/PressReleases/aptex.html>

et al. 1986 die Titel wissenschaftlicher Zeitschriften als Folgen von Buchstaben. Auch bei Input von unvollständigen Folgen von Buchstaben konvergiert das Netz schnell zu den gespeicherten Titeln.

Chen 1995 beschreibt ein Hopfield-Netz, das nicht in Schichten aufgeteilt ist, und in dem jedes Neuron einen Thesaurusterm repräsentiert. Die Terme sind untereinander verbunden. Ihre Gewichte werden mittels eines Thesaurus initialisiert, wobei die Verbindungsstärken die Beziehung zwischen den Termen wiedergeben. Dabei kann es sich um einen automatisch, auf der Basis von statistischen Informationen erstellten und/oder um einen intellektuell erstellten Thesaurus handeln. Für einen intellektuell erstellten Thesaurus werden heuristisch numerische Werte festgelegt, die den Beziehungen wie Synonymie oder *allgemeinerer Begriff* entsprechen. Ein Hopfield-Netzwerk verstärkt häufig die genutzten Verbindungen und implementiert so unüberwachtes Lernen. Dies ist bei dem Netz von Chen 1995 nicht der Fall. Lernen findet hier nicht statt.

Ein Benutzer wählt einen oder mehrere Terme für eine Suche. Die Ausbreitung der Aktivierung erreicht mit den ursprünglichen Begriffen assoziierte Terme, die dem Benutzer dann vorgeschlagen werden. Diese Terme kann der Benutzer bewerten und je nach Relevanz werden sie wieder aktiviert. Dann läuft die Aktivierungsausbreitung weiter. Das Netz bildet also nicht den eigentlichen Retrieval-Prozess ab, sondern leistet eine assoziative Term-Expansion, die vor oder während der Interaktion mit einem IR System abläuft. Das System kann als ein Spezialfall des Spreading-Activation-Ansatzes (siehe Abschnitt 4.3) betrachtet werden, bei dem nur eine Term-Schicht vorliegt, die durch inner-layer Verbindungen verknüpft ist.

Interessant an dem Ansatz von Chen 1995 ist die Integration von mehreren verschiedenen Thesauri. Neben einem aus 3000 Dokumenten extrahierten Thesaurus wurden jeweils Teile des *ACM Computing Review Classification System* und der *Library of Congress Subject Headings* benutzt. Insgesamt besitzt das Netz 14.000 Terme und 80.000 Verbindungen. Zwischen den einzelnen Thesauri bestehen zunächst keine Beziehungen. Um diese zu definieren, wurden intellektuell Cluster von zusammengehörenden Termen erstellt. Benutzer stellten dazu für sie interessante Begriffe zusammen. Diese Cluster wurden dann als eigene Neuronen implementiert. Cluster-Neuronen, die Terme aus mehreren Thesauri beinhalten, dienen als indirekte Verbindung zwischen den Thesauri. Dabei erscheint fragwürdig, ob die von einzelnen Benutzern subjektiv erstellten Listen zu Verbindungen führen, die für alle Anwender effizient sind. Weiterhin ist unklar, wie viele solche Beziehungen nötig sind und wie viele im vorliegenden Netz integriert waren. Dies wurde dadurch umgangen, dass in jeder Anfrage Terme aus allen drei Thesauri vorhanden waren. Solche künstlichen Anfragen kommen in einer realen Umgebung natürlich nicht vor. Interessant an dem Ansatz ist, dass er widersprüchliche Aussagen über Beziehungen integrieren kann.

Die Bewertung des Ansatzes prüfte nur die Qualität der Term-Erweiterung. Der Effekt auf den Retrieval-Prozess lässt sich so nicht messen. Dazu müssten z.B. Anfragen mit und ohne Term-Erweiterung evaluiert werden.

Lin/Chen 1994 erweitern den Hopfield-Ansatz auf eine multilinguale Umgebung (siehe Abschnitt 4.3.3). Bordogna et al. 1996 erweitern ein assoziatives Hopfield-Netzwerk auf reelle Aktivierungswerte. Zusätzlich lernt das Netz aus Relevanz-Feedback.

Chung et al. 1998 realisieren ein Hopfield-Netzwerk, das den Indexierungsprozess in vernetzten Umgebungen unterstützt. Sie gehen davon aus, dass viele Autoren von Dokumenten v.a. im Internet nicht für intellektuelle Indexierung geschult sind. Ein

System, das den Volltext eines Dokuments analysiert, soll ihnen Terme für die Indexierung vorschlagen. Auch bei der Auswahl von vorgeschlagenen Termen ist grundsätzlich fragwürdig, ob Laien eine Qualität der Indexierung erreichen, die sich positiv auf den Retrievalprozess auswirkt. In der Forschung ist ohnehin umstritten, welche Art der Indexierung die besseren Ergebnisse bringt (cf. Krause 1996a). Fragwürdig ist auch, ob in einer betont nicht regulierten Umgebung hinreichend genügend Autoren von solchen Werkzeugen Gebrauch machen. Sinnvoller ist ein Mechanismus wie das Schalenmodell, das verschiedene Schalen mit unterschiedlicher Qualität von Dokumenten und Indexierung zulässt und sie verwaltet (siehe Abschnitt 2.1).

Das System von Chung et al. 1998 indexiert eine Kollektion und analysiert die Kookkurrenzen für alle Termpaare. Daraus wird unabhängig von der interaktiven Indexierungskomponente mit einer Kohonen-Karte eine zwei- und dreidimensionale Visualisierung erstellt. Vergleichbare Systeme diskutiert Abschnitt 4.4. Die Kookkurrenz-Werte bilden die Gewichtungen der Verbindungen im Hopfield-Netzwerk. Chung et al. 1998 verwenden somit nicht das Hopfield-Lernverfahren (siehe Abschnitt 3.5.3), sondern setzen die Werte der Verbindungen direkt aus der Indexierung ein, um Muster im Netz zu speichern. Das Retrieval oder die Suche nach einem ähnlichen Muster zum Input verläuft dann wieder wie in einem Hopfield-Netzwerk. Das Muster wird als Aktivierungsmuster auf die betroffenen Neuronen übertragen. Die Spreading-Activation verläuft nach einer der üblichen Aktivierungsfunktionen (siehe Abschnitt 3.4.1). Nachdem ein stabiler Zustand erreicht ist, bei dem mehrere Schritte wenig an den Aktivierungszuständen der einzelnen Neuronen ändern, endet der Prozess. Die am stärksten aktivierten Terme bilden den Output und werden dem Benutzer, in diesem Fall dem Autor des Dokuments als Index-Terme vorgeschlagen.

Chung et al. 1998 berichten von Experimenten mit einer Menge von 290 Dokumenten. In einem Test mit Studenten als Benutzer evaluieren sie die Übereinstimmung der ausgewählten Terme mit den von professionellen Indexierern vergebenen Schlagwörtern. Die Übereinstimmung messen sie mit den Maßen Precision und Recall, die im Information Retrieval üblich sind. Hier liefern sie aber keine Aussage über das Retrieval. Die Übereinstimmung von Index-Termen zwischen verschiedenen Indexierungsverfahren sagt noch nichts über die zu erwartende Qualität des Retrievals. Ähnliche Verfahren, die als Vorschlagsmodus Terme aus kontrolliertem Vokabular automatisch bestimmen, stellt Abschnitt 2.4 im Kontext der Heterogenitätsbehandlung vor.

Die Problematik der Bewertung greift auch Abschnitt 5.1.2 auf. Für die Übertragung des Verfahrens auf große Kollektionen erwarten die Autoren Performanzprobleme, so dass sie an einer Beschleunigung durch Parallelisierung arbeiten. Gegenüber der algorithmischen Optimierung sollte die Integration des Verfahrens in den Information Retrieval Prozess allerdings höhere Priorität besitzen.

4.2.2 Boltzmann-Maschine

Die Boltzmann-Maschine (siehe Abschnitt 3.5.3) ist ein Hopfield-Netzwerk mit spezifischer Aktivierungs- und Lernregel. Die Aktivierung verläuft als *simulated annealing* mit einer probabilistischen Komponente. Dahinter steht die Metapher eines auskühlenden Metalls, das seine Energie schrittweise verringert.

Brachman/McGuinness 1988 stellen ein IR-System auf der Basis einer Boltzmann-Maschine vor. Ihr Ausgangspunkt ist Wissensrepräsentation in der Künstlichen Intelligenz in der Form von Frames. Information Retrieval wird als logische Deduktion

interpretiert, in der die Anfrage Ausgangspunkt einer Reihe von Folgerungen darstellt. Diese Sichtweise findet sich auch bei anderen Autoren (cf. z.B. van Rijsbergen 1986). Der Inferenz-Prozess bei Brachman/McGuinness 1988 wird insbesondere von Generalisierungs- und Spezialisierungsrelationen aus einer strukturierten Frames-Repräsentation des Gegenstandsbereichs geleitet. Brachman/McGuinness 1988 übertragen die Repräsentation von Frames auf eine Boltzmann-Maschine um auch nicht exakte Übereinstimmungen (*partial match*) besser ableiten zu können. Ihr System CRUCS (Conceptual Retrieval using Connectionist Style) bildet die Eigenschaften aller beteiligten Klassen und Objekte auf Neuronen ab. Diese *Micro-Features* dürfen jedoch nicht wie Brachman/McGuinness 1988 mit sub-symbolischen Repräsentationen in einem Backpropagation-Netzwerk verwechselt werden, da sie eine symbolische Entsprechung besitzen. Brachman/McGuinness 1988 beschreiben, wie ein Objekt, das in einer komplexen hierarchischen Struktur steht, auf ein neuronales Aktivierungsmuster übertragen wird. In diesem Prozess werden alle Neuronen, die in dem Pfad von dem zu aktivierenden Muster bis zum höchsten Konzept stehen, auf höchste Aktivierung gesetzt. Wie in Abbildung 4-1 deutlich wird, erhöht diese Art der Repräsentation die Verteiltheit der Muster. Ebenso erzeugt sie eine Ähnlichkeit zwischen benachbarten Neuronen. So ist der Aktivierungsvektor des Musters *Innendienst*, der in Abbildung 4-1 aktiviert ist, sehr ähnlich zu dem Aktivierungsvektor von *Angestellte*. In vielen Anwendungsfällen ist eine Erhaltung der Ähnlichkeit über mehrere Stufen hinweg nicht erwünscht. Um diese Anforderung mit den Vorteilen der obigen Repräsentation zu verbinden, könnten höhere Ebenen schwächere Aktivierung erhalten.

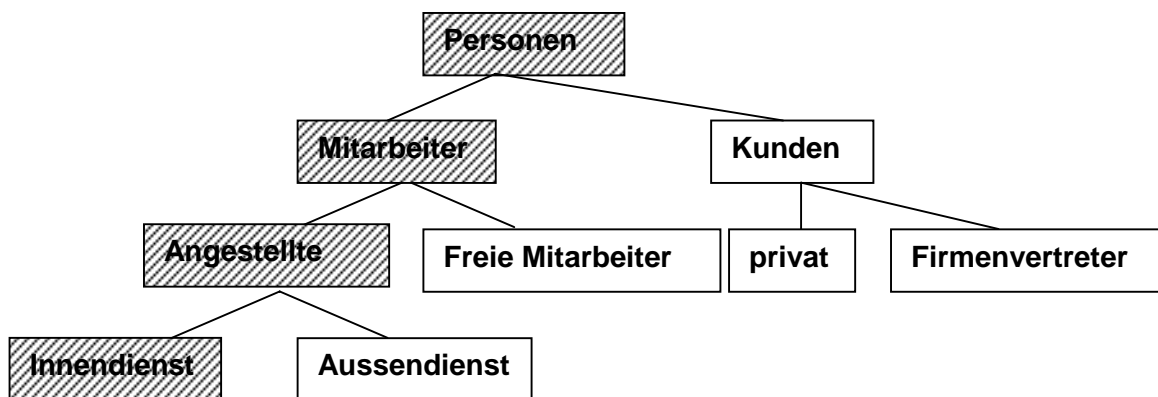


Abbildung 4-1: Beispielhafte Hierarchie. Schraffierte Muster sind aktiviert.

Das Netzwerk von Brachman/McGuinness 1988 ist in Schichten von Neuronen aufgeteilt. Eine Schicht vereint die als *individuals* bezeichneten Dokumente. Die Eigenschaften der Objekte bzw. Dokumente bilden eine weitere Schicht, die von Brachman/McGuinness 1988 als *Micro-Features* eingeführt wird. Dahinter stehen Konzepte wie *objekt-orientiert* oder *LISP-ähnlich*, während die Objekte konkrete Programmiersprachen wie LOOPS repräsentieren.

Die ursprünglichen Frames werden intellektuell erstellt. Brachman/McGuinness 1988 beschreiben eine Anwendung für die Beziehungen zwischen verschiedenen Programmiersprachen. Für größere Anwendungen wäre die Formalisierung umfassenden Wissens erforderlich, was sehr aufwendig ist und u.a. große Konsistenzprobleme mit sich bringt. Für ein IR-System, das nicht auf ein kleines

Anwendungsgebiet restringiert ist, kann dieses Verfahren kaum angewendet werden. Der Ansatz von Brachman/McGuinness 1988 bedient sich sowohl neuronaler Netze als auch symbolischer Verarbeitungsmechanismen und gehört so zu den hybriden Systemen. Auch einige Spreading-Activation-Netzwerke integrieren Elemente symbolischer Wissensverarbeitung (siehe Abschnitt 4.3.2.7).

Vor dem Retrieval-Prozess speichern Brachman/McGuinness 1988 die gewünschten Muster in der Boltzmann-Maschine. Dann werden abhängig von Anfrage einige Neuronen aktiviert und konstant auf dieser Aktivierung gehalten (*clamped*). Dabei ist unklar, ob alle *Micro-Features* als Input dienen oder nur die positiv aktivierten. Zählen auch die nicht aktivierten und damit auf Null gesetzten Units zum Input-Vektor, müssen auch sie konstant auf diesem Wert gehalten werden. Damit bliebe in der gesamten Schicht wenig Spielraum für die Aktivierungsausbreitung und im gesamten Netz sind nur relativ wenig Neuronen frei modifizierbar.

Beim Anwendungsbeispiel Programmiersprachen und ihren Eigenschaften bestehen die Anfragen aus sehr konkreten Faktenabfragen, die einen logischen Ableitungsprozess erfordern, wie ihn z.B. objekt-orientierten Datenbanken gut lösen. Der Vorteil von CRUCS soll insbesondere bei nicht exakten Abgleichen liegen. Der *simulated annealing* Prozess läuft dabei mehrfach ab und ein Neuron bildet das Ergebnis. Dieses Neuron wird im nächsten Durchlauf konstant auf Aktivierung Null festgehalten und damit unterdrückt. Das Netz muss nun einen anderen stabilen Zustand finden und liefert ein anderes Neuron als zusätzliches Ergebnis. Brachman/McGuinness 1988 zeigen an einem Beispiel, dass es so zu *partial match* Ergebnissen kommt.

Der Ansatz von Brachman/McGuinness 1988 bietet einen guten Ausgangspunkt für die Integration von vorhandenem symbolischen Wissen in ein neuronales Netz. Er steht jedoch dem symbolischen Ansatz näher als dem konnektionistischen Paradigma. Aufgrund der aufwendigen, intellektuellen Vorarbeit kommt dieses Vorgehen für ein generelles IR-System kaum in Frage. Deshalb baut das System CRUCS auch nur auf einer sehr kleinen Datenbasis auf. Ein weiterer großer Nachteil besteht darin, dass das System nicht aufgrund von Nutzerurteilen lernt.

Die Verteilung von Eigenschaften und Objekten auf verschiedenen Schichten ähnelt der Funktionsweise von Spreading-Activation-Netzwerken, die Abschnitt 4.3 diskutiert.

4.2.3 Hetero-assoziative Systeme

Die bisher vorgestellten Systeme sind auto-assoziative Speicher, bei denen Input- und Output-Vektor im gleichen Merkmals-Raum liegen und die gleichen Objekte oder Eigenschaften repräsentieren. Das folgende System ist hetero-assoziativ und schlägt bereits die Brücke zum nächsten Abschnitt. Wie die Spreading-Activation-Modelle implementiert es eine Abbildung zwischen unterschiedlichen Räumen.

Den Ansatz von Bentz et al. 1989 entwickeln u.a. Heitland 1994 und Hagström 1996 zu SpaCAM (Sparsely Coded Associative Memory) weiter. Ausgangspunkt ist die Palm-Matrix, ein hetero-assoziativer Speicher, der Paare von Input- und Output-Mustern lernt.

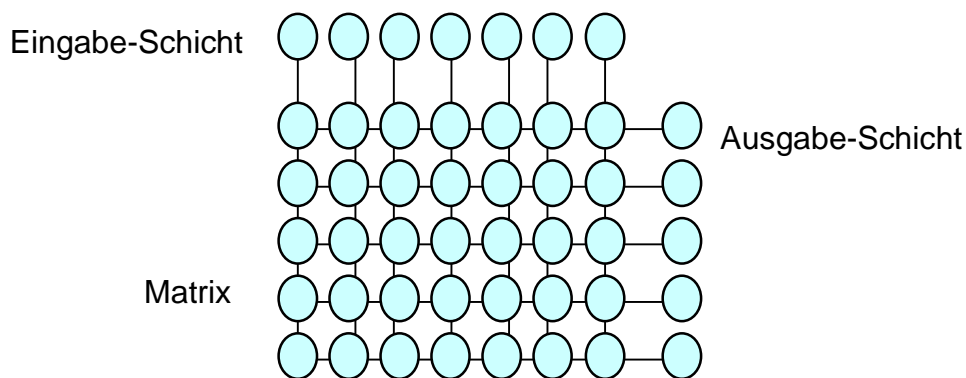


Abbildung 4-2: Schematische Darstellung einer Palm-Matrix (Hagström 1996:18)

Hagström 1996 zeigt, dass sich die Palm-Matrix als neuronales Netz mit einer Input- und einer Output-Schicht interpretieren lässt. In der Initialisierungsphase berechnet das Netz aus den zu speichernden Mustern die Verbindungsstärken. Durch Anlegen eines Input-Vektors ordnet das Verfahren alle Muster nach Ähnlichkeit zum Input.

Bentz et al. 1989 und Hagström 1996 untersuchen und entwickeln Verfahren zur effizienten Repräsentation der Matrix sowie zur effektiven Durchführung von Speicher- und Retrievalaktionen. Da die Matrizen nur sehr spärlich besetzt ist, berücksichtigt die Implementierung nur die mit Einsen gefüllten Zellen und verweist zwischen diesen mit einer Zeiger-Struktur. Dadurch wird SpaCAM sehr flexibel und kann dynamisch auch zur Laufzeit neue Muster aufnehmen. Bei Hopfield-Netzwerken ist dies problematisch, da die Architektur mit der Anzahl der Neuronen eine Obergrenze für die speicherbaren Muster setzt (siehe Abschnitt 3.5.3). SpaCAM ist gegenüber den Hopfield-Netzwerken nicht nur flexibel, sondern auch sehr effizient. Bentz et al. 1989 stellen fest, dass die Zugriffszeit von zwei Parametern abhängt, der typischen Anzahl von Termen pro Dokument und der Anzahl der Dokumente. Die Untersuchungen ergaben, dass die typische Anzahl von Termen oder Merkmalen der dominierende Parameter ist, und dass die Anzahl der Dokumente oder Muster nur eine sehr untergeordnete Rolle spielt. Variiert die Anzahl der Terme pro Dokument sehr stark, dann lohnt sich zu Effizienzsteigerung nach Bentz et al. 1989 sogar die Aufteilung der Kollektion in homogene Cluster.

Die Repräsentation der Muster bei Hagström 1996 analysiert ein Wort als Sequenz von Zeichen mit ASCII-Werten. Bei Wörtern der Länge n und m darzustellenden ASCII-Zeichen enthält der Repräsentationsvektor n mal m Stellen, wovon nur n besetzt sind. Dies erzeugt äußerst spärlich besetzte Vektoren, die sich auf die Effizienz der Verfahrens positiv auswirken (cf. Schwenker et al. 1996, Hagström 1996).

Hagström 1996 setzt SpaCAM für das Retrieval von Wörtern aus sehr großen Textmengen ein. Die hohe Fehlertoleranz liefert auch bei Tippfehlern im Input sinnvolle Ergebnisse und liefert in der Datenbank enthaltene falsche Schreibweisen. Dabei ist eine Obergrenze für die Wortlänge vorgesehen. Komplexere Anfragen aus mehreren Wörtern werden einzeln gestellt und die Ergebnislisten mit Fuzzy-Operatoren kombiniert. Diese Anwendung hat Eingang in mehrere kommerzielle Information Retrieval Anwendungen gefunden.

Ein weiteres assoziatives Speichermodell ist das Random Neural Network, das ähnlich wie das Hopfield-Netzwerk funktioniert. Ein besonderes Merkmal ist, dass der Zufall die Aktivierungsausbreitung steuert. Eine probabilistische Funktion bestimmt, welches Neuron als nächstes feuert. Stafylopatis/Likas 1992 setzen das Random Neural Network zum Speichern und Retrieval von Bildern ein, die von Merkmals-Vektoren repräsentiert werden. Ihr Netz ist hierarchisch organisiert. Zunächst versucht ein globales assoziatives Netz, den Input einem der gespeicherten Muster zuzuordnen. Treten dabei Fehler auf, versuchen weitere Netze aus Teilen der Repräsentation, für die sie das Training spezialisierte, das korrekte Muster zu finden.

4.3 Spreading-Activation-Modelle

In diesem Abschnitt wird das am weitesten verbreitete Information Retrieval Modell auf der Basis neuronaler Netze vorgestellt, das die IR-Literatur meist als Spreading-Activation-Modell bezeichnet. Dieser Name ist allerdings nicht sehr aussagekräftig, da alle künstlichen neuronalen Netze auf dem Prinzip der sich ausbreitenden Aktivierung beruhen. Der Begriff Spreading-Activation-Netzwerk identifiziert also keine besondere Klasse von Modellen.

Obwohl die Spreading-Activation-Netzwerke eindeutig neuronale Netze mit allen entscheidenden Merkmalen sind, lassen sie sich keinem der gängigen Typen zuordnen. Spreading-Activation-Netzwerke haben in der Regel bidirektionale Verbindungen, sind jedoch keine klassischen Hopfield-Netzwerke, da sie über keine Energie-Funktion verfügen. Manche der Netze sind lernfähig. Ihre Lernverfahren sind von der Perzeptron-Lernregel abgeleitet, jedoch haben sie im Gegensatz zum Perzeptron nicht nur vorwärtsgerichtete Verbindungen. Am nächsten stehen die Spreading-Activation-Netzwerke dem Pattern Associator Modell, das in McClelland/Rumelhart 1988 vorgestellt wird. Der Pattern Associator ist ein einfaches lineares Modell, das aus einer Schicht von Input-Units und einer Schicht von Output-Units besteht. Darin ist Lernen nach der Perzeptron-Lernregel vorgesehen. Allerdings verfügen Pattern Associator Systeme nur über Verbindungen in eine Richtung. Das etwas allgemeinere Modell in McClelland/Rumelhart 1988 ist das Interactive Activation and Competition Modell, das hemmende Verbindungen innerhalb der Schichten realisiert. Verbindungen innerhalb von Schichten kommen auch in manchen Spreading-Activation-Netzwerken vor, doch sind sie meistens verstärkend und implementieren so assoziative Beziehungen etwa zwischen Termen oder Dokumenten. Das Interactive Activation and Competition Modell ist jedoch nicht lernfähig.

Die Spreading-Activation-Netzwerke können also nicht eindeutig einer Klasse von Netzwerken zugeordnet werden. Da somit in der Neuroinformatik kein passender Begriff vorliegt, wird in der weiteren Darstellung der in der IR-Literatur übliche Begriff Spreading-Activation-Netzwerke beibehalten.

Eine Beschreibung der Spreading-Activation-Netzwerke erfolgt am besten über die Eigenschaften. Dazu eignet sich das in Abschnitt 3.3 vorgestellte Schema von Rumelhart et al. 1986. Demnach sind die Spreading Activation Netzwerke in Schichten aufgeteilte Netze mit bidirektionalen Verbindungen zwischen den Schichten. Bei lernenden Modellen kommen adaptierte Perzeptron-Regeln und damit einfache Delta-Regeln zum Einsatz.

In diesem Rahmen bewegen sich die Spreading-Activation-Netzwerke, wobei zwischen den einzelnen Ansätzen Unterschiede bestehen. So gibt es teilweise Verknüpfungen innerhalb der Schichten und unterschiedliche Aktivierungsfunktionen

kommen zum Einsatz. Diese Variationen werden bei den einzelnen Systemen vorgestellt. Da alle diese Systeme eine sehr einfache Struktur besitzen, die im wesentlichen nur das Grundprinzip der Aktivierungsausbreitung entlang gewichteter Verbindungen festlegt, ist der Name Spreading-Activation-Netzwerke sozusagen als kleinster gemeinsamer Nenner durchaus gerechtfertigt.

Da sich die Ansätze untereinander stark ähneln, beginnt dieses Kapitel mit einer Darstellung ihrer Funktionsweise. Im Anschluss (Abschnitt 4.3.2.1 bis 4.3.2.6) werden einige wichtige Systeme ausführlich diskutiert. Weitere Systeme werden nur kurz vorgestellt.

4.3.1 Funktionsweise eines Spreading-Activation-Netzwerks

Die Prinzipien des Grundmodells des Spreading-Activation-Netzwerks tauchen bei allen Systemen auf. Darin finden sich die typischen IR-Objekte Dokumente, Indexterme und Anfragen. Das Standard-Modell modelliert die Indexterme und die Dokumente als künstliche Neuronen.

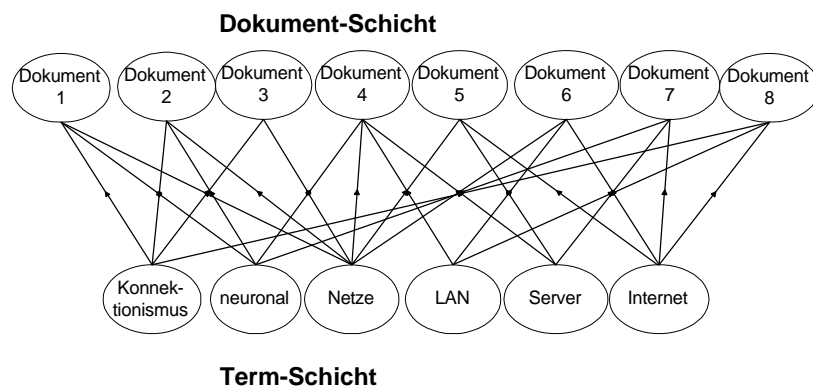


Abbildung 4-3: Zweischichtiges Spreading-Activation-Netzwerk mit beispielhaften Termen

Die Initialisierung der Gewichte der Verbindungen erfolgt anhand der Dokument-Term-Matrix aus der Indexierung. Der Benutzer formuliert die Anfrage als Liste von Termen. Das System aktiviert dann die gewählten Terme und die Aktivierung breitet sich im Netz aus. Zunächst werden die Dokument-Neuronen aktiviert, mit denen die Anfrage-Terme indexiert sind. Alle aktivierten Dokumente senden im zweiten Schritt Aktivierung an alle Terme, mit denen sie verknüpft sind. Nach einer bestimmten Anzahl von Schritten oder nachdem ein bestimmter Aktivierungswert erreicht ist, endet die Aktivierungs-Ausbreitung und das System präsentiert dem Benutzer die am stärksten aktivierten Dokumente als Ergebnis.

Spreading-Activation-Modelle besitzen im Rahmen der IR-Forschung eine hohe Plausibilität. Die Aktivierung eines Dokuments durch seine Indexterme ist eine einleuchtende Metapher. Der assoziative Charakter des IR-Prozesses tritt dabei deutlich hervor. Term-Expansion und Relevanz-Feedback ergeben sich in diesem Rahmen sehr natürlich als inhärente Eigenschaften des Modells. Im folgenden werden die einzelnen Phasen des Prozesses genauer analysiert.

4.3.1.1 Initialisierung

Die Spreading-Activation-Modelle bestehen aus zwei Schichten von Neuronen, die untereinander verbunden sind. Die Verbindungen im neuronalen Netz verlaufen also zwischen Dokumenten und Termen. Das Gewicht der Verbindungen muss demnach ein Maß für die Beziehung zwischen einem Dokument und einem Term sein. Diese Beziehung und ihr Wert spielt auch in den wichtigsten IR-Modellen die entscheidende Rolle für das Ranking. Das Gewicht stammt vom Gewichtungsalgorithmus der Indexierung. Die Dokument-Term-Matrix entspricht im Spreading-Activation-Modell der Verbindungsmatrix. Ein detaillierter Vergleich zwischen den beiden Matrizen, der weitere Verbindungsmöglichkeiten berücksichtigt, folgt in Abschnitt 4.3.4.

Die Netzwerkmodelle implementieren also kein neuartiges Inhaltsanalyse- und Repräsentationsverfahren, sondern übernehmen eine Dokument-Term-Matrix. Die Verbindungen übernehmen die Gewichte der entsprechenden Zelle der Dokument-Term-Matrix. Für das Netzwerk-Modell wie für das Vektorraum-Modell ist es unerheblich, ob die Dokument-Term-Matrix durch intellektuelle oder automatische Indexierung gewonnen wurde. Bei der Besprechung der einzelnen Systeme werden die jeweiligen Gewichtsformeln vorgestellt.

Tabelle 4-1 : Dokument-Term-Matrix für das Beispiel aus Abbildung 4-1

| Dokument | Dokument 1 | Dokument 2 | Dokument 3 | Dokument 4 | Dokument 5 | Dokument 6 | Dokument 7 | Dokument 8 |
|------------------------|------------|------------|------------|------------|------------|------------|------------|------------|
| Term | Dokument 1 | Dokument 2 | Dokument 3 | Dokument 4 | Dokument 5 | Dokument 6 | Dokument 7 | Dokument 8 |
| Konnektionismus | 0,8 | 0 | 0,8 | 0 | 0 | 0 | 0 | 0,6 |
| Neuronal | 0,6 | 0,8 | 0 | 0 | 0 | 0 | 0 | 0 |
| Netze | 0,6 | 0,8 | 0 | 0,8 | 0,8 | 0,8 | 0 | 0 |
| LAN | 0 | 0 | 0 | 0,6 | 0 | 0,6 | 0 | 0 |
| Server | 0 | 0 | 0 | 0,4 | 0,4 | 0 | 0,6 | 0 |
| Internet | 0 | 0 | 0 | 0 | 0,6 | 0,4 | 0,4 | 0,6 |

Die Verbindungen in den Spreading-Activation-Netzwerken sind bidirektional. Die Aktivierung läuft also sowohl von den Termen zu den Dokumenten als auch zurück. Die Gewichte der Verbindungen sind in fast allen Modellen symmetrisch. Sie besitzen also in beide Richtungen das gleiche Gewicht. Damit hat die Beziehung zwischen Term und Dokument den gleichen Wert wie die zwischen Dokument und Term. Die meisten IR-Modelle arbeiten mit dieser Annahme, die zwar sehr plausibel, aber nicht selbstverständlich ist. Die Term-Gewichte berechnen sich in der Regel aus Sicht der Terme, wie etwa die inverse Dokument-Frequenz zeigt (cf. z.B. Womser-Haccker 1997). Betrachtet man die Gewichte des neuronalen Netzes aus Richtung der Dokumente, ist die Berechnung einer analog definierten inversen Term-Frequenz denkbar, welche die Anzahl der Terme eines Dokuments berücksichtigt und zu völlig anderen Werten führt.

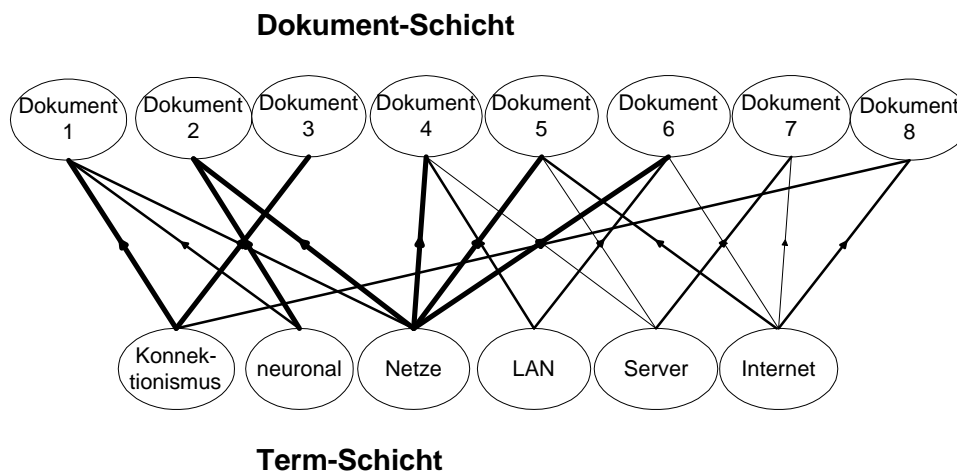


Abbildung 4-4: Initialisierung durch Setzen der Gewichte: Die Verbindungen des Netzes haben die Werte aus der Dokument-Term-Matrix in Tabelle 4-1 übernommen. Die drei unterschiedlichen Linienstärken repräsentieren die Werte in den Zellen (dickste Linie entspricht 0,8). Verbindungen mit dem Gewicht Null sind nicht eingezeichnet.

Bei der Initialisierung handelt es sich nicht um Lernen. Zwar werden die Parameter des Modells bei diesem Prozess verändert, jedoch fehlen grundlegende Eigenschaften eines Lernprozesses. Lernen in neuronalen Netzen geht in der Regel von einer zufälligen Initialisierung der Gewichte der Verbindungen aus. Von diesem zufällig gewählten Punkt im n-dimensionalen Raum ausgehend verändert ein Lernalgorithmus aufgrund einzelner externer Einflüsse der präsentierten Muster die Gewichte. Dadurch verändert sich das Verhalten des Modells zu den Mustern. Die reine Übernahme von Werten, die zudem ein eindeutig nicht-lernendes Verfahren ermittelt, ist somit kein Lernen im Sinne der Neuroinformatik. Allerdings kann auf dem so erreichten Stand ein Lernprozess einsetzen, der auf äußere Einflüsse reagiert und die Verbindungsgewichte verändert. Allgemein gesprochen ist Lernen ein Prozess, bei dem sich ein System durch Verändern von Parametern an Einflüsse aus der Umwelt anpasst und damit ein iterativer und interaktiver Vorgang. Auch aus dieser Perspektive wird deutlich, dass das einmalige Berechnen von Parametern keinesfalls Lernen darstellt.

4.3.1.2 Anfrage-Formulierung

Die Anfrage besteht im Information Retrieval für den Benutzer aus einer Liste von Begriffen. Diese wählt er entweder aus einer Liste der vorhandenen Terme aus oder gibt sie selbst ein. Die Neuronen, die diese Terme repräsentieren, werden aktiviert. In der Regel erhalten sie die maximal mögliche Aktivierung, die meist Eins beträgt. Aber auch eine Gewichtung der Terme der Anfrage ist denkbar.

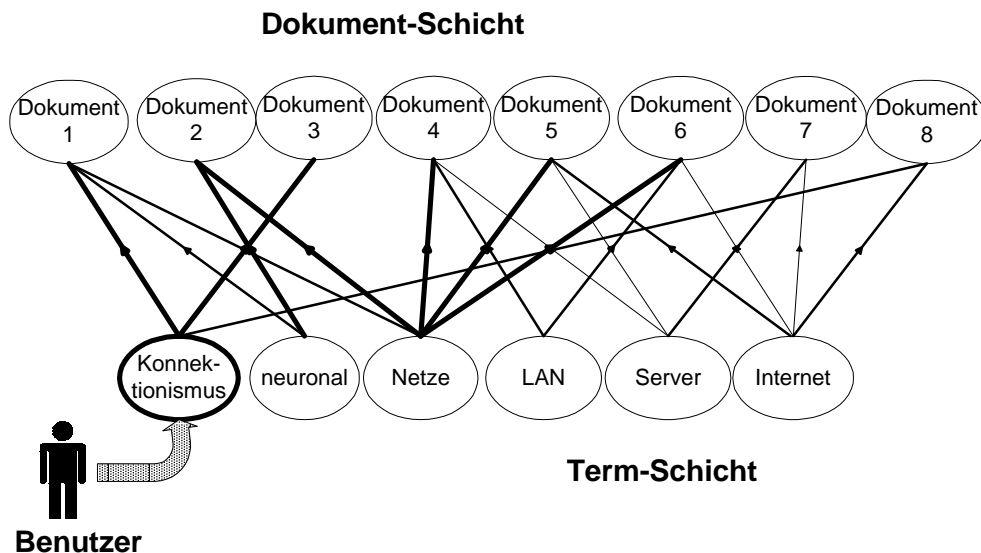


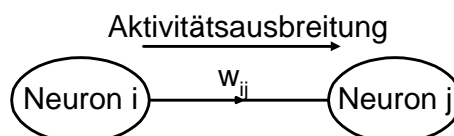
Abbildung 4-5: Anfrage als Setzen von Aktivierung: Die Neuronen der in der Anfrage vorkommenden Terme sind aktiviert. In diesem Beispiel besteht die Anfrage aus dem Begriff *Konnektionismus*. Die Aktivierung des Neurons ist hoch wie die Stärke der Linien andeutet.

Im Information Retrieval bilden grundsätzlich die Anfragen den Input und die Ergebnis-Dokumente den Output, was auch für die Spreading-Activation-Netzwerke zutrifft. Ebenso wie das Vektorraum-Modell sind auch die Spreading-Activation-Netzwerke in dieser Hinsicht flexibel. Das heißt, neben Termen können auch Dokumente als Input dienen. So kann ein System zu bereits als relevant bekannten Dokumenten weitere ähnliche Dokumente suchen. Dazu erhält im ersten Schritt nicht ein Term sondern ein Dokument Aktivierung. Der weitere Prozess verläuft dann völlig analog.

Die Spreading-Activation-Netzwerke sind noch flexibler und erlauben im Prinzip auch eine gemischte Eingabe, die sowohl aus Anfragen als auch aus Dokumenten besteht. Diese Möglichkeit bleibt zudem im laufenden Prozess erhalten. Sie taucht als Relevanz-Feedback in Abschnitt 4.3.1.4 noch einmal auf.

4.3.1.3 Retrieval

Der nächste Schritt im IR-Prozess besteht aus der Ausbreitung von Aktivierung im Netzwerk, also aus Spreading-Activation. Die Aktivierung verläuft entlang der Verbindungen und abhängig von deren Gewichten. Die Ausbreitung wird von der Ausbreitungsfunktion kontrolliert, die in den meisten Modellen sehr einfach ist:



Ausbreitungsfunktion: $Input_j = Output_i w_{ij}$

Abbildung 4-6: Aktivierungsfluss zwischen zwei Neuronen

Im ersten Schritt aktivieren die vom Benutzer gewählten Terme die Neuronen in der Dokument-Schicht. Je nach Stärke der Verbindung zwischen Dokument und Term und der Stärke des Gewichts des Anfrage-Terms aktiviert die Aktivierungsfunktion die Dokumente verschieden stark. Im nächsten Schritt läuft die Aktivierung auch von den Dokumenten zu den Termen. Ausgehend von allen aktivierten Dokumenten werden nun deren Terme aktiviert. Dabei können auch Terme aktiviert werden, welche die ursprüngliche Anfrage nicht enthielt. Damit ist im Modell die implizite Expansion der Anfrage um weitere Begriffe angelegt. Die neuen Terme kommen mit den vom Benutzer aktivierten häufig gemeinsam in Dokumenten vor, so dass die Term-Expansion auf Kookkurrenzen beruht (siehe Abbildung 4-7).

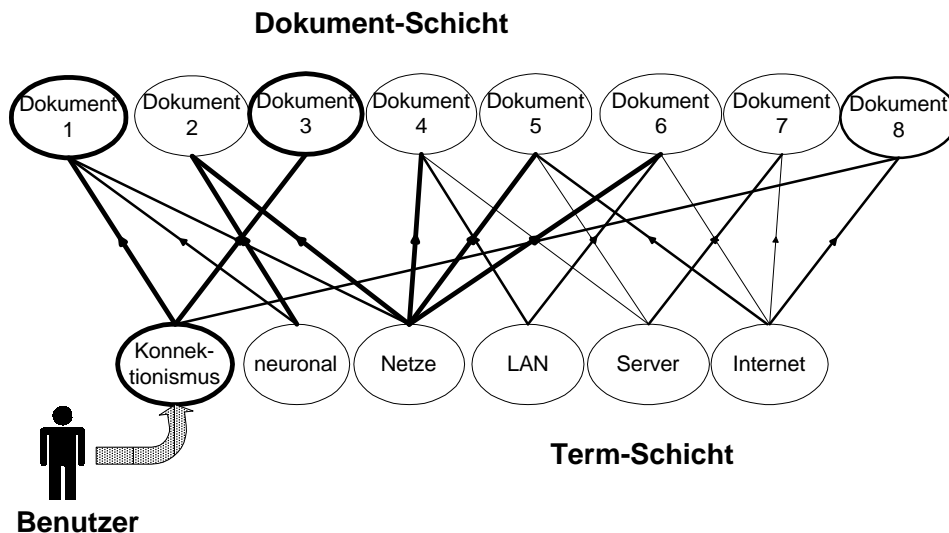


Abbildung 4-7: Aktivierung nach der ersten Phase: Die Aktivierung ist von den Term-Neuronen zur Dokument-Schicht geflossen. Dadurch wurden alle Dokumente aktiviert, die mit dem Term *Konnektionismus* verbunden sind. Je nach Stärke ihrer Verbindung mit dem Term erhielten sie mehr oder weniger Aktivierung, was die unterschiedlichen Stärken der Linien andeuten. Bei den aktivierten Dokumenten handelt es sich um diejenigen, die mit dem Term *Konnektionismus* indiziert sind, die also diesen Term enthalten.

Die Aktivierung der Dokument-Knoten ist eine Funktion von Input-Vektor und Verbindungsmatrix.

$$\text{Aktivierung}_i = f(\text{Input}_A, W)$$

Die Aktivierung eines einzelnen Neurons lässt sich unter Berücksichtigung der Output-Funktion der Term-Neuronen, der Propagierungsregel, der Input-Funktion als Summenfunktion und der Aktivierungsfunktion der Dokument-Neuronen berechnen:

$$\text{Aktivierung}_i = f_{\text{akt}}\left(\sum_j f_{\text{out}}(\text{Aktivierung}_j) w_{ij}\right)$$

Sind sowohl Output- als auch Aktivierungsfunktion die Identitätsfunktion, so ergibt sich nach dem ersten Schritt folgende Aktivierung:

$$\text{Aktivierung}_i = \sum_j \text{Aktivierung}_j w_{ij}$$

Je nach Modell setzt sich die Aktivierungsausbreitung in weiteren Schritten fort. Durch die fortschreitende indirekte Aktivierung immer neuer Knoten werden möglicherweise auch Terme aktiviert, die semantisch sehr weit von den ursprünglichen Anfrage-Termen entfernt sind. Je nach Verknüpfungsgrad des Netzes kann eventuell sehr schnell ein großer Teil des Netzes aktiviert werden. Das kann sich negativ auf die Retrievalqualität auswirken, da das System bei vielen hoch aktivierten Dokumenten nicht mehr gut diskriminiert. Zudem entspricht hohe Aktivierung in vielen Dokument-Neuronen nicht der Realität eines Information Retrieval Prozesses, bei dem ein System in der Regel nur relativ wenige Dokumente aus der Grundmenge als relevant einstuft.

Der Entwickler eines Netzes muss eine Balance zwischen zu geringer und zu starker Ausbreitung finden. Zu geringe Ausbreitung der Aktivierung nutzt die Vorteile des Spreading-Activation-Ansatzes nicht aus. Diese bestehen gerade darin, dass auch Terme aktiviert werden, die nicht in der Anfrage enthalten waren. Im Idealfall findet das System so semantisch ähnliche Begriffe.

Die starke Expansion der Anfrage durch zahlreiche Ausbreitungsschritte oder einen hohen Verknüpfungsgrad führt in der Regel zu mehr gefundenen Dokumenten. Damit erhöht es den Recall und vermindert die Precision, während sich eine geringe Ausbreitung umgekehrt auswirkt. Je nach Benutzersicht kann in einer Retrievalsituation der Recall oder die Precision wichtiger sein. Den Expansionsgrad könnte ein Benutzer über die Zahl der Ausbreitungsschritte steuern.

In jedem Fall ist es sinnvoll, die Aktivierungsausbreitung auch zu hemmen. Folgende Strategien sind dabei denkbar:

- Die Anzahl der erlaubten Aktivierungsschritte wird beschränkt.
- Die gesamte Aktivierung des Netzes wird beschränkt:
 - Die Summe der Aktivierung im gesamten Netz muss immer konstant sein.
 - Die Summe der Aktivierung, die jedes Neuron aussendet, hat ein Maximum.
- Durch einen Verfallsfaktor (decay) verlieren alle Neuronen im Netz bei jedem Schritt bei bestimmtes Maß an Aktivierung. Dadurch ist sichergestellt, dass nur Neuronen, die bei vielen Schritten angesprochen werden, aktiv bleiben, während gleichzeitig das Rauschen von schwach und einmalig aktivierten Neuronen ohne Folgen bleibt.
- Die Neuronen einer Schicht sind untereinander vollständig mit hemmenden Verbindungen verknüpft. Dadurch wird ein Wettbewerb zwischen den Knoten ausgelöst, der in sogenannten Winner-Take-All-Netzwerken so weit ausgeprägt ist, dass immer nur ein Neuron in einer Schicht *feuert* (cf. Zell 1994:191). In Spreading-Activation-Netzwerken ist eine solche extreme Ausprägung nicht sinnvoll, da immer mehrere Dokumente und Terme aktiviert sein sollen. Ein einziger aktiver Term in der Term-Schicht führt kaum zur gewünschten zusätzlichen Aktivierung von semantisch ähnlichen Termen. Und wird nur ein Dokument-Neuron aktiviert, erhält der Benutzer eine Antwortmenge mit nur einem Dokument.

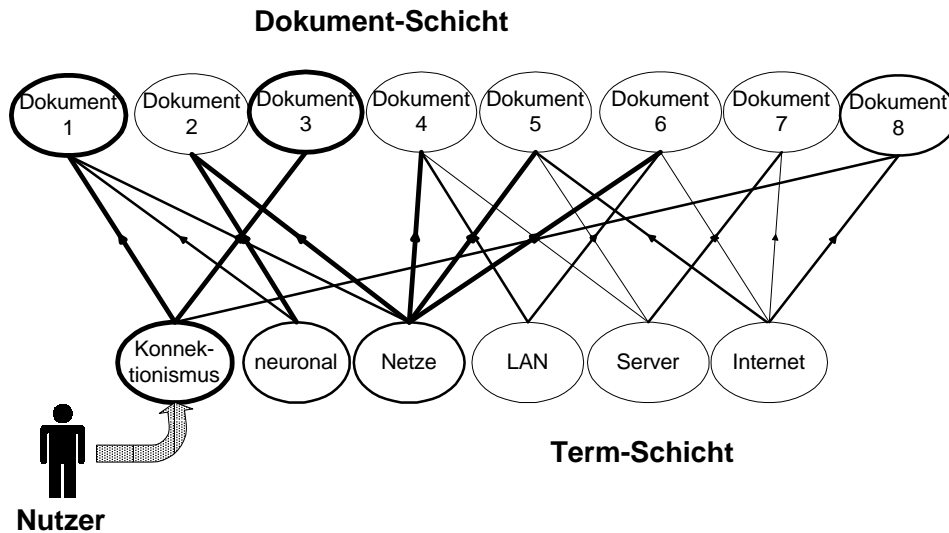


Abbildung 4-8: Automatische Termerweiterung nach mehreren Schritten: Beim zweiten Schritt senden die im ersten Schritt aktivierten Dokumente ihrerseits Aktivierung aus. Dadurch erhalten alle mit ihnen verbundenen Terme Impulse. So führt v.a. die hohe Aktivierung von Dokument 1 entlang der starken Verbindungen zu *neuronal* und *Netze* zu einer positiven Aktivierung in diesen beiden Knoten.

Die durch das Netz fließende Aktivierung kann semantisch als Relevanz oder Interesse interpretiert werden. Das Interessenspektrum, das hinter einer Anfrage steht, läuft durch das Netz und aktiviert relevante Dokumente und Terme. Ergebnis am Ende der Aktivierungsausbreitung sind die am stärksten aktivierten und damit interessantesten Dokumente. Die Aktivierung jedes Dokuments stellt dessen Retrieval Status Value (RSV) dar.

$$RSV_i = \text{Aktivierung}_i$$

Die Grundidee der Spreading-Activation-Netzwerke ist intuitiv einleuchtend. Jedoch sind wie bei der Arbeit mit neuronalen Netzen fast immer zahlreiche heuristische Entscheidungen notwendig.

- Der genaue Ablauf der Aktivierungsausbreitung
Durch den Aufbau des Netzes in Schichten ergibt sich die Möglichkeit, die Ausbreitung ebenfalls nur schichtweise zu erlauben. Dann sendet pro Phase eine Schicht und die andere empfängt. Das andere Szenario besteht darin, alle Neuronen in einer Phase gleichzeitig senden zu lassen, für alle das neue Input-Signal und anschließend die neuen Aktivierungslevel zu berechnen.
- Behandlung der Anfrage-Terme
Die Anfrage-Terme können nach der initialen Aktivierung den Aktivierungsregeln überlassen werden und somit auch abgeschwächt werden oder sie werden konstant auf dem Anfangswert gehalten (*clamped*).
- Wahl der Aktivierungs-, Input- und Outputfunktion und ihrer Parameter

4.3.1.4 Relevanz-Feedback

Bei Relevanz-Feedback bewertet ein Benutzer die Dokumente eines Zwischenergebnisses und das System optimiert davon ausgehend die Anfrage. Wie die Term-Erweiterung ergibt sich auch Relevanz-Feedback als inhärente Eigenschaft des Spreading-Activation-Modells, für das keine neuen Konzepte erforderlich sind. Die Veränderung von Aktivierung einzelner Neuronen, wie sie bei der Anfrage nötig ist, und die darauf folgende Ausbreitung reichen aus. Lediglich der Zeitpunkt des Eingriffs des Benutzers ändert sich. Während die Anfrage-Terme zu Beginn aktiviert werden, greift der Benutzer bei Relevanz-Feedback bei einem Zwischenstand ein und modifiziert die Aktivierung einzelner Neuronen. In der Regel betrachtet der Benutzer Dokumente und bewertet, ob sie für sein Informationsbedürfnis relevant sind oder nicht. Entsprechend wird die Aktivierung verändert. Relevante Knoten erhalten eine höhere oder maximale Aktivierung und nicht relevante Knoten verringern ihre Aktivierung oder setzen sie auf Null.

Relevanz-Feedback kann in Spreading-Activation-Netzen sehr flexibel ablaufen. Grundsätzlich ist jederzeit ein Eingriff möglich und neben Dokumenten kann ein Benutzer auch die bisher aktivierten Terme bewerten. Auch dieses Verfahren kann negative Auswirkungen von zu starker Aktivierungsausbreitung mindern. Prinzipiell könnte ein Benutzer nach jedem Schritt die aktivierten Terme und Dokumente betrachten und ihre Aktivierungswerte verändern. Das Funktionsprinzip des Systems bliebe dabei gleich.

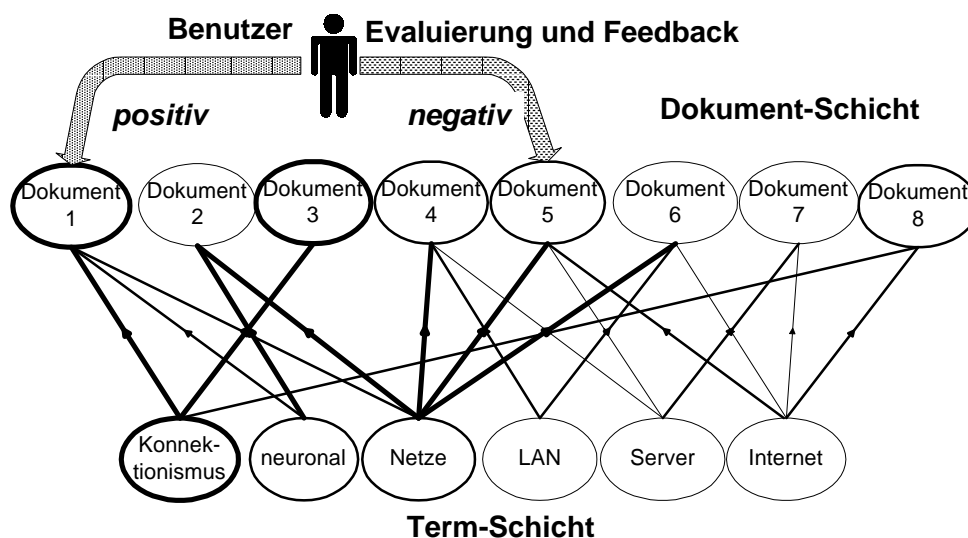


Abbildung 4-9: Relevanz-Feedback als Modifizierung der Aktivierungswerte von Neuronen: Zu einem bestimmten Zeitpunkt betrachtet der Benutzer die momentan am stärksten aktivierten Dokumente, bewertet sie, und das System nutzt diese Information. Im vorliegenden Beispiel erhielt der Benutzer die Dokumente 1, 3, 4, 5 und 8 und bewertet 1 als positiv und 5 als negativ. Vor den nächsten Ausbreitungsschritt erhöht das System die Aktivierung von Dokument 1 und verringert die von Dokument 5.

Im Detail werden die Informationen über Relevanz und Nicht-Relevanz des Benutzers verschieden eingesetzt. Vergibt der Benutzer eine positive Bewertung für ein

Dokument, so kann seine Aktivierung einmalig auf einen entsprechenden Wert gesetzt werden oder um einen bestimmten Wert erhöht werden. Die Höhe der Aktivierung in den nächsten Schritten kann dem Aktivierungsverlauf und der Aktivierungsregel überlassen werden. Alternativ dazu kann die Aktivierung auch auf dem gewünschten Level konstant gehalten werden. Dann behält das Neuron diese Aktivierung für den Rest des gesamten Prozesses. Das entsprechende positiv bewertete Dokument *belohnt* dann im weiteren Verlauf immer wieder die darin enthaltenen Terme. Daneben verwenden einige Systeme Relevanz Information der Benutzer zum Lernen und verändern damit das Modell durch Einstellen der Verbindungen.

4.3.2 Beispiele für Spreading-Activation-Netzwerke

Die oben beschriebene Funktionsweise der Spreading-Activation-Netzwerke ist in den folgenden Systemen realisiert. Die wichtigsten Systeme werden in eigenen Abschnitten diskutiert und die Abschnitte 4.3.2.5 bis 4.3.2.6 fassen weitere Systeme zusammen.

4.3.2.1 Probabilistic Indexing and Retrieval-Component-System (PIRCS)

Ein typisches Beispiel für ein Spreading-Activation-Netz bietet Kwok 1989, der eines der ersten Modelle dieser Art vorstellte. Lernfähigkeit ist vorgesehen und wird von Kwok 1991 zum Lernen aus Relevanz-Feedback ausgebaut. Das System nahm unter dem Namen PIRCS an mehreren Runden der TREC Evaluierungsstudie teil (cf. Kwok et al. 1993, Kwok/Grunfeld 1994, siehe Abschnitt 4.7).

Kwok versteht seinen Ansatz als „attempt to employ the NN [neural networks] paradigm to reformulate the probabilistic model of IR“ (Kwok 1989: 21). Darin zeigt sich bereits die Nähe der Spreading-Activation-Netzwerke zu den Standard-Verfahren im Information Retrieval.

Gegenüber dem oben dargestellten Standard-Ansatz zeichnet sich PIRCS durch einige Besonderheiten aus. Es besteht aus drei Schichten, wobei die mittlere Schicht Terme repräsentiert, die zur einen Seite mit den Anfragen und zur anderen mit den Dokumenten verbunden sind.

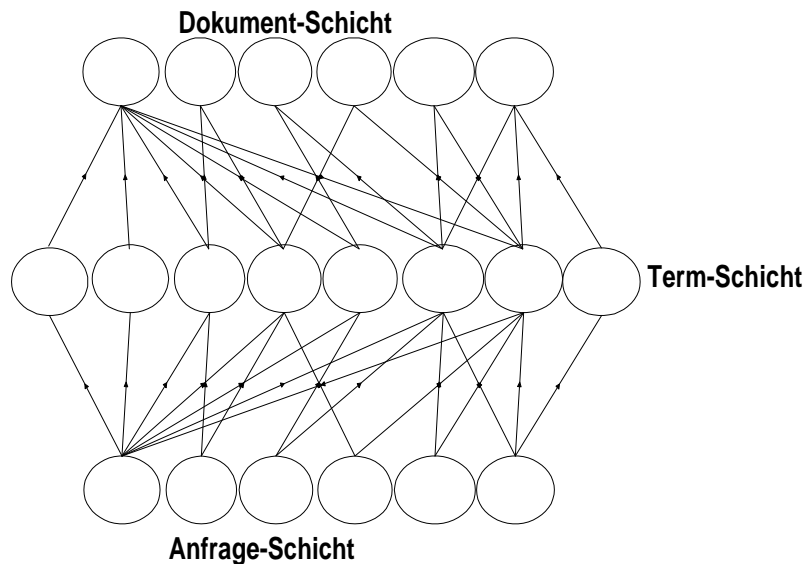


Abbildung 4-10: Dreischichtiges Netz nach Kwok 1989.

Sowohl Anfragen als auch Dokumente können als Input dienen, d.h. das Netz kann auch die relevanten Anfragen für ein Dokument zurückliefern. Dies ist der Fall bei Filter- oder Routing-Aufgaben. Dabei bleibt die Anfrage als Interessensprofil über einen längeren Zeitraum konstant und Dokumente, die neu zur Kollektion hinzukommen, werden diesen Profile zugeordnet. So identifizieren etwa Filtersysteme in einem Strom von aktuellen Nachrichten die, welche für ein Benutzerprofil besonders interessant sind. Routing-Aufgaben spielen auch in TREC eine Rolle (cf. Womser-Hacker 1997). Dies ist durchaus sinnvoll, da das Korpus hauptsächlich aus Zeitungstexten und Nachrichtenagentur-Meldungen besteht. PIRCS bearbeitete in TREC mehrfach sowohl Routing- als auch Ad-Hoc-Aufgaben (siehe Abschnitt 4.7). Auch für Routing-Aufgaben funktioniert das System völlig analog zu dem oben diskutierten Grundmodell für Spreading-Activation-Netzwerke. Lediglich die initiale Aktivierung erfolgt je nach Aufgabentyp in der Anfrage- oder Dokument-Schicht.

Die Übereinstimmung von PIRCS zum geschilderten Standard-Modell geht jedoch noch weiter. Betrachtet man das Ad-Hoc-Retrieval, also den Normalfall mit einer Anfrage als Eingabe und Dokumenten als Ausgabe, führt die Anfrage-Schicht zu keiner konzeptuellen Änderung und könnte auch weggelassen werden. Nach Kwok reduziert sich das dreischichtige Netz je nach Art des Retrievals zu einem zweischichtigen System. Folgende Gründe sprechen gegen die Einführung einer Anfrage-Schicht:

- Die Anfrage-Schicht leitet lediglich die Aktivierung zu den Anfrage-Termen in der Term-Schicht. Im zweischichtigen Modell erfolgt diese Aktivierung durch den Benutzer direkt in der Term-Schicht (siehe Abschnitt 4.3.1.1).
- Die Anfragen sind beim Ad-Hoc-Retrieval keine konstanten Objekte. Für jede Anfrage müsste vielmehr ein neues Neuron hinzugefügt werden. Zu Beginn des Systemeinsatzes wäre die Schicht damit leer und würde mit der Zeit anwachsen. Die Wiederverwendung einer Anfrage kommt selten vor. Damit dient die Anfragen-Schicht lediglich der Speicherung alter Anfragen. Im Falle von Routing verhält es sich genau umgekehrt, dann sind die Anfragen die konstanten Objekte und die Dokument-Schicht ist nicht notwendig.
- Dokumente und Anfragen können im IR als Objekte gleichen Typs betrachtet werden, die jeweils durch Terme beschrieben werden. Dies gilt nach Kwok auch

für sein Modell (cf. Kwok 1989:25). Betrachtet man die Eigenschaften eines Anfragen- und eines Dokument-Neurons, so sind beide durch Verbindungen zur Term-Schicht gekennzeichnet. Damit könnten sie auch in einer Schicht angeordnet sein und ihre Funktionalität würde sich nicht ändern.

Die Einführung von Anfrage-Knoten und ihre Zusammenfassung in einer eigenen Schicht dient also nur der Übersichtlichkeit und ist keine konzeptuelle Neuerung. Auch die Anpassung eines Netzwerks an Routing-Aufgaben erfordert nicht zwingend die Einführung einer dritten Schicht. Beim Routing ersetzen entweder die Anfragen die Dokumente oder beide Dokumente stehen gleichzeitig in einer Schicht. Alle diese Fälle erfasst man, wenn man bei dem zweischichtigen Standard-Modell von einer Objekt-Schicht und einer Eigenschaft-Schicht spricht.

Die Details der Initialisierung und Veränderung der Gewichte in PIRCS wird im folgenden am Beispiel von Kwok et al. 1993 erläutert, wobei die Unterschiede zu den anderen Aufsätzen minimal sind. Verbindungen innerhalb der Schichten existieren bei PIRCS nicht. Die Gewichte der Verbindungen sind asymmetrisch, haben also in beiden Richtung verschiedene Werte. Die Anfangswerte der Verbindungen werden aus den Häufigkeiten der Terme in den Dokumenten und Anfragen abgeleitet, wobei folgende Formeln gelten:

$$w_{ki} = \ln \frac{p}{1-p} + \ln \frac{1-s_{ik}}{s_{ik}}, \quad \text{wobei } s_{ik} = \frac{F_k - f_{ik}}{\text{Anzahl Terme} - L_i}$$

w_{ki} Verbindung von Term k zu Dokument i

p Konst. $p=1/50L_i$ Länge Dokument i

F_k, f_{ik} Frequenz von Term k in der Kollektion bzw. Dokument i

Kwok et al. 1993:155

$$w_{ik} = \frac{f_{ik}}{L_i} w_{ki} \text{ Verbindung von Dokument } i \text{ zu Term } k$$

L_i Länge Dokument i

f_{ik} Frequenz von Term k in Dokument i

Kwok et al. 1993:158

Die Formeln für die Verbindungsgewichte von den Termen zu den Dokumenten entsprechen im wesentlichen Gewichtungsmäßen, die Standard IR-Systemen einsetzen (cf. z.B. Womser-Hacker 1997). Darin wird sowohl die Term-Frequenz in der Kollektion als auch im Dokument berücksichtigt. In der Formel von Kwok et al. 1993:155 wirkt sich die Häufigkeit des Vorkommens eines Terms im Dokument nur minimal aus. Die Gewichte von den Dokumenten zu den Termen dagegen bestehen lediglich aus der längennormalisierten Dokument-Frequenz. Das Dokument aktiviert also die in ihm vorkommenden Terme ohne Rücksicht auf deren Vorkommen in der gesamten Kollektion. Vor dem Hintergrund der Betonung lokaler Informationsverarbeitung in neuronalen Netzen erscheint dies durchaus plausibel. Das Dokument initialisiert die Verbindung in seiner Richtung nur unter Verwendung von Wissen, das im Dokument

selbst steckt. Aus der anderen Richtung ist dieses Prinzip aber nicht realisiert. Die Terme initialisieren die Verbindungen zu den Dokumenten unter Zuhilfenahme globalen Wissens. Kwok et al. 1993 begründen diese Entscheidung nicht.

Lernen aus Relevanz-Feedback-Informationen verändert nur die Gewichte zu den Dokumenten, während die Verbindungen zu den Termen konstant bleiben. Auch diese Entscheidung wirkt willkürlich. Die Lernregel erinnert zum einen an das probabilistische IR-Modell und zum anderen an die Delta-Regel. Vor dem eigentlichen Lernen aus Relevanz-Feedback wird mit den gleichen Formeln sogenanntes *Self-Learning* durchgeführt, bei dem das Dokument als relevant zu sich selbst markiert wird. Dadurch gelangt keine neue Information in das Netz, so dass *Self-Learning* nur einen weiteren Schritt bei der Initialisierung darstellt. Dabei werden die Verbindungen von den Termen zu den Dokumenten durchlaufen, das entsprechende Dokument erhält die Aktivierung Eins zugewiesen und die Spreading Activation läuft einmal in die Term-Schicht. Die Aktivierung der Terme ist dann der entscheidende Faktor der Lernregel. Für diese Aktivierung spielen die Verbindungen von den Dokumenten zu den Termen eine große Rolle, bei deren Initialisierung die Kollektionsfrequenz keine Rolle spielt. Damit stellt das *Self-Learning* eine Stärkung des Einflusses der Termfrequenz in den Term-Dokument-Verbindungen dar. Die folgende Formel bestimmt den Lernprozess im Ad-Hoc-Betrieb:

$$\Delta w_{ki} = \frac{\eta_D (x_k - p_{ki}^{old})}{p_{ki}^{old} (1 - p_{ki}^{old})}$$

Δw_{ki} Änderung am Gewicht der Verbindung von Term k zu Dokument i

x_k Aktivierung Term k η_D Lernrate für Ad-Hoc

p_{ki}^{old} aktuelle Wahrscheinlichkeit, dass Dokument i relevant

Kwok et al. 1993:158

Im Gegensatz zur allgemeinen Delta-Regel (siehe Abschnitt 3.4.3) kommt der Teaching-Input in der Formel nicht vor. Beim Retrieval in PIRCS läuft die Aktivierung im ersten Schritt von den Termen zu den Dokumenten. Relevanz-Feedback setzt die Aktivierung der relevanten Dokumente auf 1 und die Aktivierung breitet sich wieder in Richtung Term-Schicht aus. Damit wirkt sich der Teaching-Input indirekt auf die Aktivierung der Term-Knoten aus, die sich dann als Faktor in der obigen Formel niederschlägt. Die Wahrscheinlichkeit p_{ki} , dass Dokument i relevant ist, ergibt sich im probabilistischen Modell aus dem Gewichtungsfaktor von Term k für Dokument i . Dieser Faktor ist in PIRCS bereits in das Verbindungsgewicht von Term zu Dokument geflossen. Damit ist die Lernregel eine Funktion der Lernrate, der aktuellen Aktivierung in Term k und der aktuellen Verbindungsstärke zwischen Term und Dokument. In den Experimenten in Kwok 1991a zeigte sich, dass Lernen von relevanten Dokumenten Verbesserungen bringt, nicht aber das Lernen von nicht-relevanten Dokumenten. Grundlage waren die kleinen Kollektionen CACM und CISI, die in der IR-Literatur zwar häufig zu Tests herangezogen werden, aufgrund ihres geringen Umfangs aber nur zu Ergebnissen mit bedingter Aussagekraft führen (für eine Beschreibung cf. Baeza-Yates/Ribeiro-Neto 1999:91ff.).

Neben der Veränderung von Verbindungsgewichten kreierte das Lernen auch neue Verbindungen. Dies kann als Veränderung der Verbindungsstärke von Null auf einen

Wert ungleich Null betrachtet werden. Die meisten neuronalen Netzwerkmodelle sehen vollständig verknüpfte Schichten vor, so dass die Schaffung der Verbindungen nicht gesondert behandelt wird.

Lernen durch die Schaffung von Verbindungen zwischen Dokumenten und Termen, die bisher nicht verbunden waren, stellen Kwok 1991a und 1991b vor. Später wurde diese Strategie auch in den TREC-Experimenten eingesetzt (siehe Abschnitt 4.7). Die gleichzeitige Aktivierung im Laufe einer Anfrage führt zu einer neuen Verbindung. Dabei werden nach dem Relevanz-Feedback die 15 bis 30 am stärksten aktivierten Terme, die nicht in der originalen Anfrage enthalten waren, auch mit den relevanten Dokumenten verbunden. Die Gewichte ergeben sich aus folgenden Formeln:

$$w_{ik} = \alpha x_k$$

w_{ik} Verbindung von Dokument i zu Term k

α Lernrate

x_k Aktivierung Term k

$$w_{ki} = \ln \frac{p_{ki}}{1 - p_{ki}} + \ln \frac{\text{Anzahl Terme}}{F_k}$$

$$p_{ki} = \beta \eta_D x_k$$

w_{ki} Verbindung von Term k zu Dokument i

β, η_D Lernraten

F_k Frequenz von Term k in der Kollektion

Kwok et al. 1993:159

Diese Implementation der Term-Expansion entspricht nicht der im Standard-Modell, bei dem sich die Expansion im Laufe der Aktivierungsausbreitung automatisch durch die Aktivierung weiterer Terme ergibt. PIRCS nutzt diese Fähigkeit des Spreading-Activation-Ansatzes nicht voll aus. Beim Retrieval fließt die Aktivierung nur einmal von den Anfrage-Termen zur Dokument-Schicht. Deshalb muss die Term-Expansion sich in konkreten Verbindungen niederschlagen. Nur beim Lernen erfolgt wie gezeigt ein weiterer Schritt und Aktivierung fließt wieder in die Term-Schicht zurück.

PIRCS orientiert sich stark am Vektorraum-Modell. Dies zeigen auch die Weiterentwicklungen. Kwok 1996 und Kwok/Chan 1998 z.B. benutzen kaum die Terminologie neuronaler Netze. Kwok 1996 befasst sich mit den Problemen kurzer Anfragen. Die Verbesserungen in Kwok/Chan 1998, die auf Pseudo-Relevanz-Feedback beruhen, das die ersten n Teffer als relevant betrachtet, sind auf jedes Vektorraum-Modell übertragbar. Dabei ließe sich gerade diese Technik mit einem Spreading-Activation-Ansatz mit mehreren Aktivierungsschritten elegant realisieren.

Unklar ist bei PIRCS wie bei vielen Modellen, in welchen Umfang Lernen das Modell verändert. Wie stark ändern sich die Verbindungen und wie hoch ist der Anteil der Verbindungen, die betroffen sind? Hat das Relevanz-Feedback entscheidenden Einfluss auf die Performanz des Netzes oder hängt die Qualität lediglich von der Initialisierung ab? Um dies zu messen, müsste die Retrieval-Qualität anhand von

Test-Anfragen mit einem nur initialisierten Netz und mit einem durch Lernen modifizierten Netz gemessen werden.

Im Rahmen der TREC Konferenzen erprobten Kwok et al. 1993 und Kwok/Grunfeld 1994 das Modell an großen Datenmengen, wobei sie auch das Lernen durch Relevanz-Feedback einsetzten. In TREC 5 wurde PIRCS für Retrieval in Chinesisch adaptiert (Kwok/Grunfeld 1995, cf. auch Kwok 1997). Insgesamt erreichte PIRCS bei TREC sehr gute Ergebnisse und zählte mehrfach zu den acht besten Systemen für einzelne Kategorien, die der Überblicksartikel erwähnt (siehe Abschnitt 4.7). So gehörte PIRCS bei den TREC-Konferenzen 5 bis 7 zu den acht besten Systemen beim Standard Retrieval.

Zusätzlich sind in PIRCS *soft boolean* Anfragen implementiert. Dabei wird eine boolesche Formel in Gewichte umgesetzt, die das Netz durch zusätzliche Neuronen auf der Seite der Anfragen realisiert. Dies ist die einzige bekannte Umsetzung von Booleschen Retrieval in ein neuronales Netz.

4.3.2.2 Adaptive Information Retrieval (AIR)

Belew 1986 und 1989 stellen das System Adaptive Information Retrieval (AIR) vor, das in drei Schichten Terme, Dokumente und Autoren repräsentiert. Wie im Grundmodell der Spreading-Activation-Systeme für Information Retrieval (siehe Abschnitt 4.3.1) vorgesehen, breitet sich die Aktivierung in AIR in mehreren Phasen aus.

Für die Initialisierung der Gewichte zwischen Dokumenten und Termen werden typische IR-Indexierungstechniken verwendet. Belew 1989 verwendet ein Gewichtungsschema auf Basis der inversen Dokument-Frequenz der Terme. Die Verbindungen sind bidirektional und symmetrisch. Zusätzlich sind Autoren mit ihren Dokumenten verbunden. Die Summe aller Gewichte im System bleibt konstant. Beim Hinzufügen neuer Dokumente zur Laufzeit berücksichtigt dies der Gewichtungsalgorithmus.

Die Anfrage kann aus Elementen verschiedenen Typs bestehen und potenziell Terme, Dokumente und Autoren enthalten. Ebenso liefert das Ergebnis verschiedene Typen von Elementen. Auch Relevanz-Feedback kann sich auf alle Knoten beziehen. Damit nutzt AIR die Flexibilität des Spreading-Activation-Ansatzes in dieser Hinsicht voll aus.

Lernen durch Relevanz-Feedback verändert die Gewichte der Verbindungen und beeinflusst dadurch das Modell. Das Feedback kann vier Werte annehmen, darunter auch negative: *very relevant*, *relevant*, *irrelevant*, *very irrelevant*. Die vom Benutzer vorgegebene Feedback-Information läuft im Netz entlang der Verbindungen. Die Verbindungsgewichte werden nach einer der Delta-Regel ähnlichen Lernregel modifiziert (siehe Abschnitt 3.4.3). Die Aktivierungsstärke des Ausgangsneurons und das Relevanz-Feedback-Signal sind die entscheidenden Faktoren:

$$\Delta w_{ij} = \eta$$

w_{ij} Verbindung von Knoten i zu Knoten j

η Lernrate

Belew 1989:14

AIR enthält auch Verbindungen innerhalb von Schichten. Diese entstehen durch Lernen. Belew 1986 berichtet vom Entstehen bemerkenswerter Assoziationen innerhalb der Schichten. So wurden in einem Fall Wörter mit gleichem Stamm sehr stark assoziiert und ein irrtümlich falsch geschriebener Autor hatte einen starken Bezug zu dem Neuron mit der richtigen Schreibweise (Belew 1986:186).

Die Tests mit AIR beruhen auf einem Korpus von 1500 bibliografischen Angaben. Für die Indexierung stehen nur die Titel zur Verfügung, so dass eine Generalisierung der gewonnenen positiven Ergebnisse nicht möglich scheint. Das Netz für 1500 bibliographische Angaben besitzt insgesamt 5000 Neuronen, so dass 10 Millionen Verbindungen möglich sind. Durch die Initialisierung werden nur 0,2% davon besetzt (cf. Belew 1986). Die Entscheidung, ob die Aktivierung eines Anfrage-Terms konstant bleibt, oder vom Netz festgelegt wird, löst AIR mit einem Kompromis. Die Anfrage-Terme werden zwischen ein und zehn Epochen konstant gehalten. Dies führte zu den besten Ergebnissen. Bei der Anfrage waren auch negative Terme erlaubt, die dann als negative Aktivierung ins Netz eingingen.

Auch das Lernen durch Relevanz-Feedback führte bei Belew 1986 zu Verbesserungen. Allerdings ergaben sich negative Folgen, die wohl auf die geringe Anzahl von Dokumenten zurückzuführen ist. Innerhalb des Netzes entstand ein untereinander stark assoziiertes Cluster von Dokumenten, die häufig eine negative Beurteilung erhielten. Sobald eines davon im Verlauf der Spreading-Activation aktiv wurde, aktivierte es das gesamte Cluster. Trotz einiger weiterer kleiner Schwächen der Lernregel glaubt Belew 1986, dass das System insgesamt überzeugt und sich noch optimieren lässt.

AIR verfügt über eine Benutzungsoberfläche, die den Fluss von Aktivierung und die beteiligten Verbindungen und Neuronen transparent macht. Die Oberfläche von Belew benutzt sehr kleine Knoten und die Beschriftung verläuft teilweise schräg. Damit wirkt die Benutzungsoberfläche eher verwirrend und wenig benutzungsfreundlich. Allerdings ist die Idee, den Spreading-Activation-Prozess zu visualisieren, durchaus interessant und verfolgenswert. Dies gilt insbesondere, da Information Retrieval Systeme in der Regel kaum transparent sind und für den Benutzer eine *black box* darstellen.

Rose/Belew 1991 bauen AIR durch die Integration semantischen Wissens zu dem hybriden System SCALIR aus, das in Abschnitt 4.3.2.7 besprochen wird.

4.3.2.3 SYRENE

Mothe 1994 präsentiert mit SYRENE ein weiteres zweischichtiges Netz. Die Schichten repräsentieren Terme und Dokumente. Die Verbindungen zwischen den Termen und Dokumenten werden durch die Indexierung initialisiert und mit der inversen Dokument Frequenz gewichtet. Daneben entstehen bei der Initialisierung bereits Verbindungen innerhalb der Term-Schicht. Diese initialisiert SYRENE aufgrund von Kookkurrenzen oder semantischem Wissen. Mothe 1994 nennt vier Arten von Verbindungen:

- Aufgrund von Thesaurusbeziehungen manuell vergebene Verbindungen:
 - Synonyme
 - Spezifische Begriffe
 - Allgemeine Begriffe
- Automatisch erstellte Verbindungen:

- Kookkurrenz-Analyse

Die Werte der Kookkurrenz-Verbindungen ergeben sich wie im Standard-Modell für Spreading-Activation-Netzwerke aus den Gewichtungen der Terme für die berechnet. Die semantischen Beziehungen werden ebenfalls in numerische Werte umgesetzt. Jede Klasse von Beziehungen erhält heuristisch einen Wert, z.B. w_{sy} für Verbindungen zwischen Synonymen. Alle Verbindungen von dieser Klasse besitzen diesen gesetzt. Die semantischen Beziehungen führen so zu zusätzlichen Verbindungen mit numerischen Gewichten, die das Netzwerk wie normale Verbindungen behandelt. Das Netz verarbeitet somit Gewichtungsinformationen und zusätzliche semantische Informationen einheitlich.

Bei der Aktivierungsausbreitung ist denkbar, dass der Benutzer nur bestimmte semantische Verbindungsarten auswählt und die übrigen ausblendet.

Mothe 1994 glaubt, dass die entstehenden assoziativen Beziehungen in verschiedenen Kollektionen und für verschiedene Benutzer unterschiedlich sind, dass jedoch ein Kern der semantischen und statistischen Links zwischen den Termen über mehrere Text-Kollektionen hinweg konstant bleibt. Diese könnten einen weithin gültigen Bestand assoziativer Beziehungen bilden, der ohne weitere intellektuelle Überprüfung auf weitere Text-Bestände übertragen wird. Die Beschreibung der Realisierung zeigt aber nicht deutlich, wie dieser Kern identifiziert werden soll. Sollte dieses Verfahren in Tests die Qualität des Retrievals verbessern, dann ist die Identifizierung eines Kerns zur Übertragung auf neue Datenbestände nur für eine Anfangsphase sinnvoll. Das Ausnutzen der spezifischen Eigenschaften eines Korpus lässt eine höhere Qualität erwarten. So erreichen verschiedene Information Retrieval Systemen bei verschiedenen Korpora oft unterschiedliche Qualität.

Die Aktivierungsausbreitung erfolgt zunächst innerhalb der Term-Schicht und setzt sich nach eventuell mehreren Phasen in Richtung der Dokument-Schicht fort. Von den Dokumenten fließt nur beim Relevanz-Feedback Aktivierung in die Term-Schicht. In diesem Fall bewertet der Benutzer die stark aktivierten Dokumente. Deren Aktivierung wird dann je nach Grad des Benutzerurteils verändert. Die Aktivierung fließt wiederum in die Term-Schicht und die nun aktivierten Terme senden ihre Signale erneut in die Dokument-Schicht. Auch bei Relevanz-Feedback sind damit maximal drei Schritte oder Phasen der Aktivierungsausbreitung erlaubt. Lernen in Form von Gewichtsänderungen findet durch die Relevanz-Feedback-Informationen nicht statt. Die Informationen des Benutzers über Relevanz der vorgelegten Dokumente besteht nur flüchtig und geht für spätere Anfragen verloren.

Aufgrund der restriktiven Ausbreitung von Aktivierung ist es sinnvoll, die Kookkurrenzen von Termen zu berechnen und im Netz zu verankern, da eine indirekte Aktivierung wie im Standard-Modell für Spreading-Activation-Netzwerke (siehe Abschnitt 4.3.1.3) nicht möglich ist. Die Stärken des Spreading-Activation-Ansatzes wie die inhärente Term-Expansion und die Lernfähigkeit neuronaler Netze schöpft SYRENE nicht aus.

Mothe 1994 testet SYRENE mit einer kleinen Kollektion von 67 Dokumenten und zwölf Anfragen. Damit optimiert sie einige Parameter des Systems. Aufgrund des geringen Umfangs dieser Testmenge lassen sich die Ergebnisse aber nicht auf andere Daten übertragen.

Interessant ist der Ansatz von Mothe 1994 vor allem deshalb, weil die Autorin die Beziehung zwischen Spreading-Activation-Netzwerken und Standard IR-Modellen formal untersucht und ihre Ergebnisse empirisch belegt. Zunächst beweist die Autorin, dass Spreading-Activation nach Initialisierung und einem einzigen

Aktivierungsschritt bei entsprechender Wahl der Aktivierungsfunktion zu den gleichen Ergebnissen führt wie ein Vektorraum-Modell mit äquivalenter Ähnlichkeitsfunktion. Mothe 1994 stellt drei Ähnlichkeitsfunktionen des Vektorraum-Modells den entsprechenden Aktivierungsfunktionen der Spreading-Activation-Modelle gegenüber:

| | Ähnlichkeitsfunktion im Vektorraum-Modell | Aktivierungsfunktion im Spreading-Activation-Netzwerk |
|-----------------|---|---|
| Inneres Produkt | $\vec{D}_j \vec{Q} = \sum_i d_{ij} q_i$ | $Akt_j^D(\tau = 1) = \sum_i Akt_i(\tau = 0) TW(i) w_{ij}$ |
| Kosinus-Maß | $Cos(\vec{D}_j \vec{Q}) = \frac{\sum_i d_{ij} q_i}{\sqrt{\sum_i q_i^2} \sqrt{\sum_i d_{ij}^2}}$ | $Akt_j^D(\tau = 1) = \frac{\sum_i Akt_i(\tau = 0) TW(i) w_{ij}}{\sqrt{Akt_i(\tau = 0)^2} \sqrt{(TW(i) w_{ij})^2}}$ |
| Jaccard-Maß | $Jac(\vec{D}_j \vec{Q}) = \frac{\sum_i d_{ji} q_i}{\sum_i q_i^2 + \sum_i d_{ji}^2 - \sum_i q_i d_{ji}}$ | $Akt_j^D(\tau = 1) = \frac{S}{\sum_i (Akt_i(\tau = 0))^2 + \sum_i (TW(i) w_{ij})^2 - S}$ $S = \sum_i Akt_i(\tau = 0) TW(i) w_{ij}$ |

Mothe 1994:283

Mothe 1994 demonstriert die enge Beziehung zwischen den etablierten Information Retrieval Modellen und dem Spreading-Activation-Ansatz sehr eingängig. Nach einem Aktivierungsschritt sind die beiden Modelle äquivalent und führen so auch zu identischer Retrieval-Qualität. Damit demonstriert Mothe 1994 auch, dass im Spreading-Activation-Ansatz kein völlig neuartiges Modell darstellt.

4.3.2.4 Mercure

Mercure (cf. Boughanem et al. 1999, Boughanem/Soule-Dupuy 1994/1997) ist ein weiteres Spreading-Activation-Netzwerk im Rahmen der TREC Evaluierungsstudie getestet wurde (cf. Boughanem/Soule-Dupuy 1997/1998). Abschnitt 4.7 vergleicht die Ergebnisse mit anderen Systemen.

Mercure verfügt über eine Input- und Output-Schicht, die jedoch nur als Interface dienen und keine weitere entscheidende Bedeutung haben. Von der Anfrage-Schicht werden die Ter

m-Neuronen nach einem Gewichtungsschema aktiviert. Im Kern enthält Mercure Verbindungen zwischen einer Term-Schicht und einer Dokument-Schicht und innerhalb der Term-Schicht. Die Gewichte werden aufgrund von Vorkommenshäufigkeiten und Kookkurrenzen initialisiert. Nachdem Boughanem/Soule-Dupuy 1994 noch ein einfacheres Gewichtungsschema einsetzen, berechnen sich die Gewichte in Boughanem/Soule-Dupuy 1997 aus folgender Formel:

$$w_{ij} = \frac{freq_{ij}}{\sqrt{\sum_{k=1}^T (freq_{ik}^2 \log(\frac{M}{m_k})^2)}} \log \frac{M}{m_i}$$

w_{ki} Gewicht der Verbindung von Term i zu Dokument j

$freq_{ij}$ Frequenz von Term i in Dokument j

M, T Anzahl Dokumente und Anzahl Terme in der Kollektion

m_i Anzahl Dokumente mit Term i in der Kollektion

Boughanem/Soule-Dupuy 1997:2f.

Unter Einbeziehung der Dokumentlänge entwickeln Boughanem/Soule-Dupuy 1998 ein elaboriertes Gewichtungsschema, das an das Okapi-Verfahren angelehnt ist (cf. Robertson et al. 1997). Okapi ist ein Information Retrieval System, das mehrfach mit Erfolg an TREC teilgenommen hat und auf dem Vektorraum-Modell beruht.

$$w_{ij} = \frac{\frac{(1 + \log(freq_{ij}))}{1 + \log(average_j(freq_{ij}))} (h_1 + h_2 \log(\frac{N}{n_i}))}{h_3 + h_4 \frac{doc_length}{average_doc_length}}$$

w_{ij} Gewicht der Verbindung von Term i zu Dokument j

$freq_{ij}$ Frequenz von Term i in Dokument j

N Anzahl Dokumente in der Kollektion

n_i Anzahl Dokumente mit Term i in der Kollektion

h_x verschiedene Parameter

Boughanem/Soule-Dupuy 1998:3

Mit dieser Formel erzielte Mercure bessere Ergebnisse. Analog veränderten Boughanem/Soule-Dupuy 1998 auch die Gewichtung der Anfrage-Terme.

Die Verbindungen innerhalb der Term-Schicht ergeben sich aus folgender Gleichung:

$$c_{ij} = \alpha \frac{\sum_{k=1}^T (w_{ik} w_{jk})}{\sqrt{\sum_{k=1}^T w_{ik}^2 + \sum_{k=1}^T w_{jk}^2 - \sum_{k=1}^T (w_{ik} w_{jk})}}$$

w_{ij} Gewicht der Verbindung von Term i zu Dokument j

T Anzahl der Terme in der Kollektion

α Parameter

Boughanem/Soule-Dupuy 1997:2f.

In Boughanem/Soule-Dupuy 1994 lernt Mercure aus Relevanz-Feedback und verändert die Verbindungen innerhalb der Term-Schicht. Die Lernregel ist an die Delta-Regel angelehnt und berücksichtigt die Anzahl der als relevant und nicht relevant eingestuft Dokumente:

$$\Delta w_{ij} = \alpha \delta \text{Aktivierung}_i \cdot \text{Aktivierung}_j$$

α Lernparameter

δ Parameter, abh. von der Anzahl der relevanten Dokumente

Boughanem/Soule-Dupuy 1994:525

Das Lernverfahren wurde mit einer kleinen Menge von 350 bibliographischen Angaben getestet. Die Ergebnisse dürfen bei dieser kleinen Grundlage nur als erster Hinweis gewertet werden. Die bei TREC eingesetzten Varianten von Mercure beinhalten kein Lernen. Dort dient Relevanz-Feedback lediglich für eine Term-Expansion. Die Aktivierungsausbreitung ist dabei stark begrenzt und umfasst pro Anfrage nur drei Schritte, so dass die Aktivierung nur einmal aus der Dokument-Schicht in die Term-Schicht zurück fließt. Dies wird als Backpropagation bezeichnet, was missverständlich ist, da sich dieses Verfahren stark vom Backpropagation-Algorithmus unterscheidet (siehe 3.5.4). Mercure erreichte in TREC 6 in zwei Kategorien des Standard Retrieval einen Platz unter den besten acht Systemen (siehe Abschnitt 4.7).

4.3.2.5 Weitere nicht lernende Spreading-Activation-Netze

Ein frühes Beispiel für ein Spreading-Activation-Netz bieten Salton/Buckley 1988. Ihr System orientiert sich stark an semantischen Netzen. Semantische Netze sind ein Wissensrepräsentations- und -verarbeitungsmechanismus, in dem eine Netzstruktur die Beziehungen zwischen Objekten widerspiegelt. Die Verbindungen haben semantische Labels und erlauben Ableitungsprozesse. Gewichte wie in neuronalen Netzen sind für die Verbindungen nicht vorgesehen (cf. Rich/Knight 1991:251 ff.). Ein Überblick über Spreading-Activation-Netzwerke mit Elementen semantischer Netze bietet Abschnitt 4.3.2.7.

Da in einem IR-System normalerweise wenig semantische Beziehungen aus dem Anwendungsfall bekannt sind und maschinenlesbar zur Verfügung stehen, greifen auch Salton/Buckley 1988 nach einer Diskussion der Chancen semantischer Netze im IR auf ein neuronales Spreading-Activation-Netz zurück, das im wesentlichen dem

Standard-Modell ohne Lernen entspricht (cf. Abschnitt 4.3). Die semantischen Beziehungen werden dabei durch die Gewichte aus der Indexierung ersetzt.

Eine Möglichkeit für semantische Beziehungen zwischen Dokumenten, die Salton/Buckley 1988 anführen, sind Zitate. Dies wären Verbindungen innerhalb von Schichten, die in diesem Modell auch vorgesehen sind. Die Stärke der Spreading-Activation-Modelle besteht jedoch gerade darin, dass solche direkten Verbindungen aufgrund der indirekten, assoziativen Beziehungen nicht nötig sind. Ähnliche Dokumente sollten auch ohne die explizite Angabe von Synonymen gefunden werden, da ein Term sein Synonym indirekt über die Dokument-Schicht aktiviert. Ein Term aktiviert Dokumente, die dann wiederum Terme aktivieren. Da die Terme in den gleichen Dokumenten vorkommen, sind sie ähnlich und aktivieren wiederum ähnliche Dokumente. Die Netzwerkfunktionen sind so gewählt, dass die Aktivierungsausbreitung folgende Ähnlichkeitsfunktion implementiert:

$$similarity(D, Q) = A \sum_{t=1}^m \frac{w_{qt} w_{dt}}{\left(\sum_{k=1}^m w_{qk} \right) \left(\sum_{j=1}^n w_{jt} \right)}$$

Salton/Buckley 1988:151

Diese Formel weist darauf hin, dass nur ein Aktivierungsschritt vorgesehen ist und bestätigt die Ergebnisse von Mothe 1994, die auf die Äquivalenz von Vektorraum-Modell und Spreading-Activation-Netzwerken hinweist. Damit unterliegt das Modell den gleichen Schwächen wie die meisten der vorgestellten Systeme, es nutzt die Möglichkeit der Spreading-Activation-Netzwerke nicht voll aus.

Salton/Buckley 1988 vergleichen für sechs kleine IR Testkollektionen die Performanz ihres einfachen Spreading-Activation-Ansatz mit der eines Vektorraum-Modells. Dabei evaluieren sie verschiedene Verfahren für die Termgewichtung. Längennormalisierung verbessert die Resultate beider Verfahren. Das Vektorraum-Modell erreichte mit Berücksichtigung der inversen Dokument-Frequenz und Normalisierung der Dokumentlänge die besten Resultate. Dieses Vektorraum-Modell setzte ein ausgefeilteres Gewichtungsschema ein als die Spreading-Activation-Netzwerke, die in keinem Fall die inverse Dokument-Frequenz berücksichtigen. Das Vektorraum-Modell ohne inverse Dokument-Frequenz schnitt dagegen sehr viel schlechter ab als das einfachste Spreading-Activation-Netz. Salton/Buckley 1988 zeigen somit, dass Spreading-Activation durchaus vergleichbare Ergebnisse erbringt.

Aus dem gleichen Jahr datiert der Ansatz von Cochet/Paget 1988, der eine Kollektion von Bildern, die durch intellektuell vergebene Terme repräsentiert wird, in einem Spreading-Activation-Ansatz modelliert. Cochet/Paget 1988 führen einige Besonderheiten ein, die bei anderen Netzen nicht zu finden sind. Die Aktivierungsfunktion berücksichtigt die Verbindungsstärken des gesamten Netzes und realisiert damit keine streng lokale Berechnung.

$$\Delta w_{ij} = \frac{\sum w_{ij}}{\sum_{J=1}^{Akt(v_j) \neq 0} w_{ij}} netinput$$

Cochet/Paget 1988:666

Die Summe aller bestehenden Gewichte wirkt sich negativ auf den Betrag der Gewichtsveränderung aus, während die Summe der aktiven Gewichte positiv wirkt. Letztere besitzen damit doppelten Einfluss, da sie im Netinput, der Summe der gewichteten Eingänge, bereits enthalten sind.

Weiterhin führen Cochet/Paget 1988 einen Mechanismus zur Veränderung der Topologie der Verbindungen ein, den sie Absorption nennen. Dieser dient dazu, die Zahl der Verbindungen zu begrenzen. Die Absorption analysiert die gemeinsamen Bilder von Termen. Bildet die Menge der Bild-Dokumente eines Terms eine Untermenge zu einem anderen Term, so entsteht eine neue Verbindung zwischen den Termen, die die Verbindungen des ersten Terms zu seinen Bilder ersetzt. Eine starke Beziehung oder eine hohe Kookkurrenz führt zu einer Verbindung innerhalb der Schicht, wie dies auch bei manchen anderen Modellen der Fall ist. Zusätzlich werden aber einige der originalen Verbindungen getilgt.

Lernfähigkeit durch Relevanz-Feedback ist lediglich geplant, wobei Cochet/Paget 1988 zu Recht feststellen, dass keine Grenze zwischen Retrieval und Lernphase besteht, sondern beide gehen ineinander über und verlaufen parallel.

Boyd et al. 1994 setzen ein zweischichtiges Netz im TREC Kontext ein (siehe Abschnitt 4.7). Sie wollen es als Maßstab für ihre eigentlichen semantischen Experimente nutzen. Eine Schicht repräsentiert Terme und die zweite Topics. Da die Autoren damit ein Routing-Experiment durchführen, handelt es sich dabei um die Dokumenten-Schicht. Die eigentlichen Dokumente werden als Aktivierung an die Term-Schicht angelegt und aktivieren so die für sie relevanten Topics.

Wilkinson/Hingston 1991 und 1992 stellen ebenfalls ein Netz vor, das eine Term- und eine Dokument-Schicht benutzt. Die von den Autoren beschriebene Anfrage-Term-Schicht kann vernachlässigt werden, da sie nur die Aufgabe hat, Anfrage-Terme über Verbindungen mit einem festem Gewicht mit dem Wert Eins in die eigentliche Term-Schicht, die hier Dokument-Term-Schicht genannt wird, zu propagieren. Durch geschickte Wahl der Aktivierungsfunktion und Propagierungsregel implementiert das Netz von Wilkinson/Hingston 1991 die Kosinus-Ähnlichkeitsfunktion. Nach einem Aktivierungsschritt ohne implizite Expansion erhalten sie auch empirisch die gleichen Ergebnisse wie in einem Test mit dem Vektorraum-Modell und der Kosinus-Ähnlichkeitsfunktion. Wilkinson/Hingston 1991 integrieren Relevanz-Feedback in ihr System. Die im ersten Ergebnis vom Benutzer als relevant erkannten Dokumente werden stark aktiviert und behalten diese Aktivierung für den weiteren Verlauf. Sie aktivieren wiederum bestimmte Terme und diese neue Dokumente. Relevanz-Feedback verändert nur Aktivierung, die Gewichte des Netzes bleiben konstant.

Wilkinson/Hingston 1991 und 1992 experimentieren mit traditionellen IR-Kollektionen, wie CACM (cf. Baeza-Yates/Ribeiro-Neto 1999:92f.). Sie zeigen, dass das Netz nach dem ersten Aktivierungsschritt die gleiche Qualität erreicht wie der Kosinus. Messungen nach einzelnen Schritten zeigen, dass sich die durchschnittliche Precision schnell erhöht und dann langsam sinkt. Das beste Ergebnis erzielte das System nach nur zwei Aktivierungsschritten. Nach zwanzig Schritten war der durchschnittliche Recall nur leicht gesunken und lag immer noch über dem Vektorraum-Modell mit Kosinus-Ähnlichkeitsmaß. Auch die Ergebnisse mit Relevanz-Feedback verbesserten sich bei einer geringen Anzahl von Aktivierungsschritten. Wilkinson/Hingston 1991 zeigen für alle Test-Kollektionen, dass zumindest eine geringe Anzahl von Aktivierungsschritten positiv auf das Retrievalergebnis wirkt.

Das Modell von Mothe/Soule-Dupuy 1992 besteht ebenfalls aus zwei Schichten, eine Schicht repräsentiert die Dokumente und eine die Terme. Die Anfrage besteht aus einer Untermenge von Termen, die der Benutzer auswählt. Mothe/Soule-Dupuy 1992 lassen auch Verbindungen innerhalb der Schichten zu. Relevanz-Feedback verändert auch in diesem Modell lediglich die Aktivierung des Netzes.

Crestani 1997a schlägt ein Spreading-Activation-Modell in einem Hypertext vor. Das Spreading-Activation-Netz besteht dabei aus den semantischen Beziehungen innerhalb des Hypertexts. Damit präsentiert Crestani 1997a ein hybrides Modell, gibt jedoch keinen Algorithmus für die Kombination beider Wissensarten an. Crestani 1997a betont, dass Spreading-Activation eingeschränkt werden muss, da ansonsten schnell das gesamte Netz hoch aktiviert ist. Die semantischen Beziehungen sieht er als eine Möglichkeit, die Aktivierung zu hemmen. Da semantisch motivierte Verbindungen aber zusätzliche Links sind, erhöhen sie eher die Gesamtaktivierung. Weitere Netze mit semantischen Elementen diskutiert Abschnitt 4.3.2.7.

Einen sehr interessanten Ansatz und fundierte experimentelle Ergebnisse bieten Syu et al. 1996. Sie kombinieren die Dimensionalitätsreduktion von Latent Semantic Indexing mit dem Spreading-Activation-Netzwerk. LSI reduziert die Dokument-Term-Matrix auf zwischen zwanzig und 200 LSI-Terme. Dadurch ist die Term-Schicht bei Syu et al. 1996 wesentlich kleiner als in anderen Systemen. Vor der Komprimierung erstellen sie die Dokument-Term-Matrix mit folgender Gewichtungsfunktion:

$$w_{ij} = \frac{tf_{ij}idf_j}{\max_i tf_i \log(\text{Anzahl_Dokumente})}$$

Syu et al. 1996:148

Im Gegensatz zum Standard-Modell breitet sich die Aktivierung zunächst in der Term-Schicht aus, wo ein winners-take-all-Verfahren wirkt. Dadurch begrenzen die Autoren die Anzahl der aktivierten Terme. In der nächsten Phase läuft die Aktivierung mehrfach zur Dokument-Schicht und zurück. In der Dokument-Schicht wirkt ebenfalls ein winners-take-all-Verfahren. Zusätzlich hält eine Funktion die Aktivierung der Knoten im Bereich [0; 1]. Stopp-Kriterium ist wie in einem Hopfield-Netzwerk (siehe Abschnitt 3.5.3 und 4.2.1) ein stabiler Zustand (Equilibrium), bei dem weitere Aktivierungsschritte keine wesentliche Veränderung des Netzwerkzustands bewirken. Die aktivierten Dokumente haben dann Werte nahe Eins erreicht.

Im Gegensatz zu vielen anderen Systemen durchläuft die Aktivierung mehrfach die Verbindungen zwischen Termen und Dokumenten und nutzt so die Vorteile des Spreading-Activation-Modells besser aus. Die potenziellen Nachteile werden durch mehrere hemmende Faktoren kontrolliert (Constrained Spreading Activation).

Für die Experimente greifen Syu et al. 1996 auf vier klassische Test-Kollektionen zurück. Sie vergleichen die Performanz ihres Systems sowohl zu einem Netzwerk ohne Komprimierung durch LSI als auch zu einem Vektorraum-Modell mit Kosinus-Ähnlichkeitsfunktion auf Basis der reduzierten Repräsentation. Das Spreading-Activation-Netzwerk mit LSI-Vektoren brachte die besten Resultate. Zusätzlich experimentieren sie für jede Kollektion mit verschiedenen Niveaus der Komprimierung zwischen zehn und 400 LSI-Dimensionen und finden so für jede Kollektion die optimale Komprimierung.

Die Ergebnisse deuten insgesamt darauf hin, dass Systeme mit Aktivierungsausbreitung in mehreren Schritten durchaus positive Ergebnisse erzielen und die

Beschränkung vieler Modelle auf ein oder zwei Schritte nicht unbedingt erforderlich ist.

4.3.2.6 Weitere lernende Spreading-Activation-Netzwerke

Layaida/Caron 1994 benutzen in ihrem Modell ebenfalls eine Schicht für Autoren, die mit der Dokument-Schicht verbunden ist. Autoren werden mit ihren Dokumenten verknüpft und erhalten zudem Verbindungen für gemeinsame Publikationen. Auch in diesem Netz steht vor der Term-Schicht eine Anfrage-Schicht, die nur die Aufgabe hat, die Terme in der Term-Schicht zu aktivieren. Wie in Abschnitt 4.3.2.1 diskutiert, sind solche Schichten nicht erforderlich. Die Initialisierung der Term-Dokument-Verbindungen erfolgt mit einer Form der inversen Dokument Frequenz. Die Terme sind untereinander verknüpft und die Gewichte der Verbindungen werden mit einem Assoziationsfaktor initialisiert.

Suchbedingung in Anfragen können sowohl Terme als auch Autoren sein. Die Autoren-Schicht wird auch bei Anfragen benutzt, die ausschließlich aus Termen bestehen. Das System lernt langfristig aus Relevanz-Feedback-Information und verändert die Verbindungsgewichte innerhalb der Term-Schicht und zwischen Term- und Dokument-Schicht. Die Lernregel ähnelt der Delta-Regel des Backpropagation-Algorithmus, führt aber nicht zu subsymbolischen Repräsentationen, da keine versteckte Schicht beteiligt ist. Innerhalb der Autoren-Schicht modifiziert eine Form Hebb'schen Lernens (siehe Abschnitt 3.4.3) die Verbindungsstärken. Layaida/Caron 1994 berichten von Experimenten mit 250 Dokumenten und 25 Anfragen. Das Netz ähnelt dem System AIR (cf. Belew 1989, siehe Abschnitt 4.3.2.2).

Wong et al. 1993 konstruieren ein Netz, dessen Architektur sich vom Standard-Modell unterscheidet. Darin gibt es zwei Term-Schichten, eine für Dokument-Terme und eine für Anfrage-Terme. Die Verbindungsmatrix zwischen diesen beiden Schichten ist somit eine Assoziationsmatrix, die Termerweiterungen realisiert. Grundsätzlich können in diesem Modell das Anfrage- und Dokument-Vokabular unterschiedlich sein, so dass auch heterogene Datenbestände erschließbar sind. Zusätzlich fasst eine Output-Schicht mit nur einem Neuron die Aktivierung der Anfrage-Term-Schicht zusammen, wie Abbildung 4-11 zeigt.

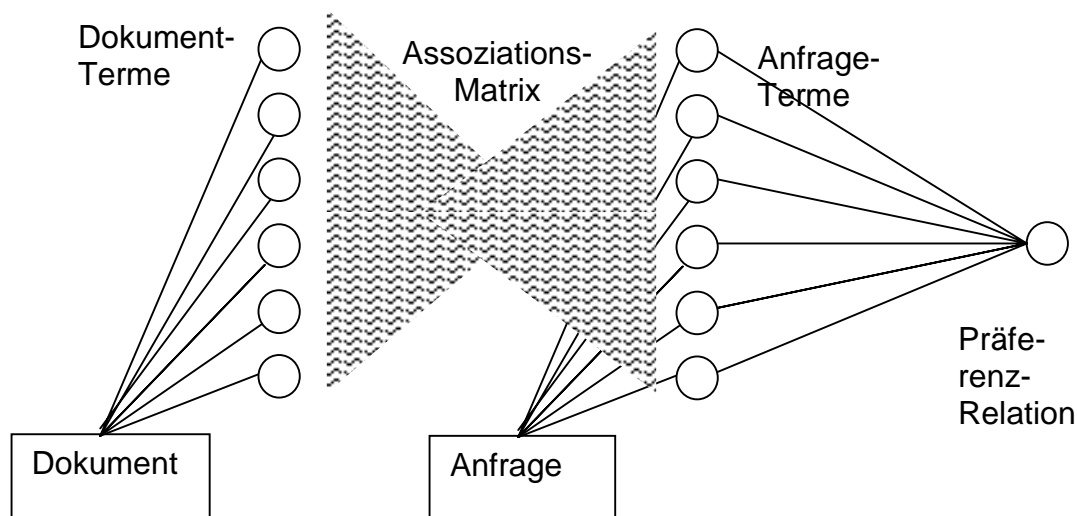


Abbildung 4-11: Struktur des Modells von Wong et al. 1993

An beiden Schichten werden die in Dokument und Anfrage vorkommenden Terme aktiviert. Dies geschieht durch eigene Schichten für Dokumente und Anfragen, die aber weiter keine Rolle spielen. An der Dokument-Schicht wird der Differenz-Vektor zweier Dokumente angelegt, d.h. das System vergleicht bei jedem Schritt zwei Dokumente in bezug auf eine Anfrage. Die Aktivierung läuft durch das Netz und eine Output-Schicht mit nur einer Unit, die mit der Anfrage-Term-Schicht voll vernetzt ist, sammelt ein Fehlersignal. Es misst, ob die Matrix die Präferenz des richtigen Dokuments bei dieser Anfrage erkennt. Bei einem Output größer Null hat das System ein korrektes Ranking ermittelt, bei negativem Output wird die Matrix adaptiert. Bei linearer Aktivierungsfunktion und dem Verzicht auf Gewichtungparameter bei der Spreading-Activation ergibt sich folgender Output:

$$g(d, q) = \sum_j \sum_i d_i a_{ij} q_j = dAq$$

A Assoziationsmatrix

d Dokumentvektor (Differenzvektor)

q Anfragevektor

Wong et al. 1993:109

Damit ist der Output ein einfaches Produkt, das sich auch ohne neuronales Netz implementieren ließe.

Die Assoziationsmatrix wird durch gradient-descent-Lernen also durch Minimierung eines Fehlers mit Relevanz-Feedback optimiert. Ausgangspunkt ist die vereinfachende Annahme, dass die Rankingfunktion sich zu den Dokumenteigenschaften linear verhält. Die Komplexität dieser Abbildung ist jedoch nicht bekannt, so dass diese Annahme problematisch ist. Mit dieser Vereinfachung beweisen Wong et al. 1993, dass ihr Lernverfahren eine Lösung in endlich vielen Schritten erreicht, falls eine Lösung existiert.

Wong et al. 1993 experimentieren mit einer Testkollektion, die aus 82 Dokumenten und 35 Anfragen besteht, von denen 31 für das Training und nur vier für den Test benutzt wurden. Die Kollektion enthält 1217 Terme, wobei nur 200 in allen Anfragen vorkommen. Gegenüber einer Ähnlichkeitsberechnung mit dem Kosinus-Maß verbessert sich der durchschnittliche Recall um über 60%.

Interessant an diesem Modell ist der Einsatz zweier Term-Schichten. Dadurch wird das Indexierungsvokabular transparent auf das Anfragevokabular abgebildet. Desweiteren besitzt das System zwei Eigenschaften, in denen es sich vom Standard-Spreading-Activation-Modellen unterscheidet und die auch das COSIMIR-Modell auszeichnen (siehe Abschnitt 4.6.2).

- Es gibt ein einziges Output-Neuron, das das Ranking bestimmt.
- Der Retrievalprozess wird in Dokument-Anfrage-Paare zerlegt, die das Netz einzeln verarbeitet.

4.3.2.7 Semantische Spreading-Activation-Netzwerke

Hybride Systeme verwenden mindestens zwei verschiedene Arten von Wissensmodellierung. Typischerweise ist eine davon sub-symbolisch und eine symbolisch. Das Ziel hybrider Systeme ist es, die Nachteile einer Technik durch die Vorteile einer anderen auszugleichen. Damit bilden sie eines der interessantesten Themen sowohl in der Kognitionswissenschaft als auch in der praktischen Anwendung, wobei gerade bei hybriden Systemen heuristische Faktoren eine große Rolle spielen.

Ein Beispiel für hybride Systeme sind konnektionistische Expertensysteme. Dies sind Systeme in denen sowohl neuronale Netze als auch konventionelle Expertensysteme aus der Künstlichen Intelligenz, die aufgrund von Regel Inferenzen bilden, eine Rolle spielen. Einen Überblick über die Koppelung von neuronalen Netzen mit Fuzzy Logik bieten Nauck et al. 1994. Auch die semantischen Spreading Activation Netzwerke können als hybride Systeme betrachtet werden.

Semantische Netze sind keine neuronalen Spreading-Activation-Netze im Sinne der in Kapitel 3 vorgestellten Modelle. Semantische Ansätze benutzen inhaltliche Verbindungen wie *Synonym* oder *Is-a*, während Verbindungen in neuronalen Netzen nicht inhaltlich belgt sind und ausschließlich numerische Gewichtung besitzen. Gemeinsam ist den Systemen das Prinzip der sich ausbreitenden Aktivierung. Automatische semantische Verfahren haben sich im Information Retrieval bisher nicht durchgesetzt, da sie für Massendaten nicht geeignet sind. Manuelles Setzen der Verbindungen ist für einen realistischen großes Korpus kaum möglich. Trotzdem greifen einige wenige Systeme auf sicheres semantisches Wissen zurück oder kombinieren Elemente semantischer Netze mit neuronalen Spreading-Activation-Netzen. Einen Überblick über semantische Spreading-Activation-Netze im Information Retrieval bietet Crestani 1997.

Ein erster und durchaus häufig vorkommender Schritt in die Richtung semantischer Netze ist die Integration semantischer Beziehungen in neuronale Verbindungen. Semantische Verbindungen werden nach Heuristiken in numerische Beziehungen umgewandelt wie etwa in SYRENE (siehe Abschnitt 4.3.2.3). Die Integration einer Autoren-Schicht ermöglicht die Realisierung einiger sicherer semantischer Beziehungen. Die Beziehung zwischen einem Autor und seinem Dokument ist eindeutig, die Beziehung zwischen Term und Dokument dagegen nur vage. Eine Autoren-Schicht enthalten die Systeme von Belew 1989 (AIR, siehe Abschnitt 4.3.2.2) und Layaida/Caron 1994 (siehe Abschnitt 4.3.2.6).

Neben Autorenschaft ergeben sich weitere Ansatzpunkte für die Integration semantischen Wissens, die auch in vielen kommerziellen Literaturdatenbanken erfasst sind, wie etwa Zitationsbeziehungen oder Ko-Autorenschaft. Die Arten von Verbindungen veranschaulicht ein Überblick im Abschnitt 4.3.4.

Das System von Layaida/Caron 1994 (siehe Abschnitt 4.3.2.6) ist ein Beispiel für die Integration von Ko-Autorenschaft. Neben den üblichen Schichten für Dokumente, Terme und Anfragen beinhaltet ihr Spreading-Activation-Modell eine Schicht für Autoren. Autoren, die gemeinsam publizieren, erhalten Verbindungen, die das Lernen nicht beeinflusst. Zwischen Autoren und ihren Dokumenten fügen Layaida/Caron 1994 Verbindungen mit der Stärke eins (= maximal) ein. Diese Links haben eine klare Semantik und bilden den ersten Schritt in Richtung hybride Systeme.

Mothe 1994 (siehe auch Abschnitt 4.3.2.3) verfolgt einen ähnlichen Ansatz. In dem zweischichtigen Spreading-Activation-Netzwerk gibt es in der Term-Schicht Verbindungen zwischen den Termen. Diese werden entweder durch Kookkurrenzen oder durch semantisches Wissen initialisiert. Als mögliche Instanzen von solchen intellek-

tuell vergebenen Verbindungen nennt Mothe 1994 Synonyme, spezifischere und generellere Begriffe. Mothe 1994 glaubt aber, dass viele der semantischen und statistischen Links über mehrere Datenbanken hinweg konstant bleiben.

In bestimmten Anwendungsfällen liegen semantische Beziehungen zwischen den beteiligten Klassen von Objekten vor und sollten dann ausgenutzt werden. Ein System, das semantische und numerische Verbindungen kombiniert, ist SCALIR von Rose/Belew 1991. SCALIR (Symbolic and Connectionist Approach to Legal Information Retrieval) integriert konnektionistische und symbolische Elemente und ist ein vielversprechender hybrider Ansatz.

Rose/Belew 1991 gehen von der Beobachtung aus, dass Experten in juristischen Fragen zwar die üblichen Probleme haben, ihr intuitives Wissen in Regeln zu formulieren. Daneben existieren eindeutige und sichere Sachverhalte wie etwa die Beziehungen zwischen verschiedenen Instanzen eines Rechtsstreits oder der Bezug einer Urteilsbegründung zu einem bestimmten Gesetz. SCALIR basiert auf früheren Arbeiten von Belew im Rahmen des Spreading-Activation-Ansatzes und der Implementierung seines Systems AIR (cf. Abschnitt 4.3.2.2). Das Netzwerk besteht aus Schichten für Gesetzestexte, Urteile und Terme. Zwischen den einzelnen Neuronen können verschiedene Verbindungen bestehen.

S-Links (symbolic) bilden die symbolische Komponente. Sie verfügen über ein Label wie z.B. *widerrufen* oder *Verweis*, sind fest zugewiesen nur innerhalb von Schichten zugelassen. Die C-Links (connectionist) werden wie bei Kwok 1989 mit Termhäufigkeiten initialisiert und können durch Relevanz-Feedback verändert werden. Sie bestehen von Anfang an zwischen Termen und Urteilen und zwischen Termen und Gesetzen. Während des Lernens können sie an allen Stellen entstehen. Durch sie soll das Netz im laufenden Betrieb die komplexen und subtilen Zusammenhänge lernen, welche die intuitive Komponente des informationellen Prozesses bilden.

SCALIR verfügt weiterhin über sogenannte H-Links (hybrid), die nur zwischen Neuronen für Urteile vorkommen. Diese H-Links verändern ihr Gewicht durch Lernen, besitzen aber eine feste semantische Bedeutung. Die Bedeutungen stammen aus einer Liste, die die Autoren zwei Achsen zuordnen. Die Position auf einer Achse spiegelt die Ähnlichkeit der Fälle wider. Sie reicht von *follows* (höchste Ähnlichkeit) bis zu *overruled* (niedrigste Ähnlichkeit).

Die Neuronen in SCALIR besitzen einen zweistelligen Aktivierungsvektor mit Komponenten für C- und S-Aktivierung, die entlang den entsprechenden Verbindungen läuft. Während die S-Links nur die Aktivierungskomponente der mit ihnen verknüpften Beziehung weiterleiten, leiten C-Links das gesamte Spektrum der Aktivierung weiter. Da die C-Links also auch semantische Aktivierung weiterleiten, besteht SCALIR nicht nur aus zwei parallelen Netzen, die unabhängig voneinander Information verarbeiten und kann zu Recht als hybrides System bezeichnet werden.

SCALIR erhält das semantische Wissen aus bestehenden Wissensbasen wie etwa einem Verzeichnis von Gerichtsurteilen und ihren Beziehungen. Dann lernt es anhand von Relevanz-Feedback nach einem heuristischen Algorithmus, den Rose/Belew 1991 als *localized reinforcement* bezeichnen. Verbindungen, die viel zu einem positiven Resultat beigetragen haben, werden verstärkt und umgekehrt. Ähnliche Verfahren setzen manche der lernenden Spreading-Activation-Ansätze ein (siehe Abschnitt 4.3.2). Sub-symbolische Informationsverarbeitung ist auch in SCALIR nicht realisiert.

SCALIR verfügt über eine grafische Benutzungsoberfläche, in der Einschränkungen des Dokumenttyps auf z.B. Urteile möglich sind. Die nach der Anfrage am stärksten

aktivierten Knoten werden in ihrem Beziehungsgeflecht visualisiert. SCALIR ist ein interessanter Ansatz, der das Spreading-Activation-Modell erweitert und heuristische, aus dem Anwendungsfall gewonnene Strukturen integriert. Das System ist für den Anwendungsfall Gerichtsurteile optimiert. Damit ist die Generalisierbarkeit allerdings sehr eingeschränkt. Insbesondere liegt in vielen anderen Domänen nicht so gut strukturiertes und erschlossenes Wissen vor.

Die beiden im folgenden vorgestellten Systeme basieren auf Wissen, das in noch stärkerem Maße strukturiert ist.

Cohen/Kjeldsen 1987 präsentieren das System GRANT, das Forscher bei der Suche nach geeigneten Förderprogrammen unterstützen soll. GRANT besteht aus einem Netzwerk der Forschungslandschaft, mit semantischen Verbindungen wie *component of*, *causes* oder *effected by*. Der Benutzer aktiviert für ihn interessante Forschungsthemen und die Aktivierung breitet sich entlang der Verbindungen aus. Die Ausbreitung wird von drei Mechanismen beschränkt:

- *Distance-constraint*: Die Aktivierung darf maximal über vier Knoten laufen.
- *Fan-out-constraint*: Knoten für sehr generelle Konzepte wie *Person* leiten die Aktivierung nicht weiter.
- Sicherheitswerte für die Regeln: Die Aktivierungsausbreitung in einem semantischen Netz gleicht der Anwendung von Regeln in einem Produktionssystem. Die Inferenz, ob eine Fördereinrichtung ein Thema fördert, basiert auf zwei Prämissen. Die Fördereinrichtung fördert ein weiteres Thema, das zu dem gesuchten in Beziehung steht. Je nach Beziehung und Art der Themen erhält jede Regel einen Sicherheitswert, so dass sich bei mehreren Ergebnissen, aufgrund der Sicherheitswerte ein Ranking entsteht. Die Regeln entsprechen den Verbindungen des Netzes, der Sicherheitswert erinnert somit an die Verbindungsgewichte.

Cohen/Kjeldsen 1987 bewerten GRANT mit den im Information Retrieval üblichen Qualitätsmaßen. Wie fast alle semantischen Ansätze hat auch GRANT den Nachteil, dass das Wissen intellektuell erfasst werden muss und das Gesamtsystem nicht auf andere Domänen übertragbar ist.

Das bereits in Abschnitt 4.2 diskutierte System CRUCS (Brachman/McGuinness 1988) benutzt semantische Verbindungen innerhalb einer Boltzmann-Maschine. Das semantische Netz repräsentiert die Beziehungen zwischen logikbasierten Programmiersprachen, führt aber aufgrund der Implementierung als Boltzmann-Maschine vages Retrieval in Form eines *partial match* durch.

Die Kombination von zwei verschiedenen Wissenstypen führt auch zu Problemen. Bei den Spreading-Activation-Netzen kann das Nebeneinander unterschiedlicher Verbindungen zu Widersprüchen führen. Es sind Fälle denkbar, in denen objektives semantisches Wissen dem momentanen Nutzeranforderungen widerspricht. So kann beispielsweise die Aktivierung von weiteren Dokumenten eines Autors unerwünscht sein. Lernende Systeme müssen dann das semantische Wissen mit anderen Arten von Verbindungen ausgleichen. Damit wird der Lernprozess natürlich erschwert.

Ein generell einsetzbares Information Retrieval System sollte aus diesen Gründen und aufgrund des hohen intellektuellen Aufwands für die Definition semantischer Beziehungen ohne diese auskommen. Eine zusätzliche Berücksichtigung erscheint jedoch in gut strukturierten Anwendungsfällen sinnvoll.

Die bisherigen Überlegungen beziehen sich auf die Ausnutzung semantischer Beziehungen zwischen Dokumenten, Termen und Anfragen. Dieses Wissen kann im

Retrieval-Prozess eine Rolle spielen und etwa zur Erhöhung der Interaktivität eingesetzt werden. Das Verfolgen von semantisch klar interpretierbaren Verbindungen zum Beispiel zu anderen Dokumenten eines Autors oder zu zitierten Dokumenten stellt einen Mehrwert für den Anwender eines Information Retrieval Systems dar, der sich auch leicht vermitteln lässt. Allerdings ergeben sich auch daraus einige Probleme. Zum einen verweisen Zitationen, falls sie überhaupt maschinell verarbeitbar vorliegen, nicht immer auf Dokumente innerhalb des Korpus. Eine etwas andere Situation ergibt sich bei der Betrachtung von Internet-Dokumenten. Hier bestehen Hypertext-Verbindungen, hinter denen semantisch sehr verschiedene Beziehungen stehen. Sie stehen prinzipiell vollständig maschinell lesbar zur Verfügung. Die zur Zeit populären Internet-Suchmaschinen nutzen die Verbindungen zumindest dahingehend, dass häufige Verweise die Relevanz einer Seite erhöhen (cf. Mönnich 1999). Bekavac 1999 stellt ein Modell vor, das die Einbettung einer Internet-Seite in die Struktur des Servers in den IR-Prozess einbezieht. Die Einbeziehung der Links stößt jedoch aufgrund der hohen Anzahl von Seiten an Grenzen.

4.3.3 Hopfield- und Spreading-Activation-Netzwerke für Transformationen

Auch konnektionistische Modelle wurden bereits für Transformationsaufgaben getestet. Hopfield Netzwerke sind assoziative Speicher, die Retrieval aus unvollständigen Mustern erlauben (siehe Abschnitt 3.5.3). Als Information Retrieval System speichern Hopfield-Netze die Dokumente als Energie-Minima und die Anfrage wird als unvollständiges Muster eingegeben, von dem aus das Netz zum nächstliegenden Dokument konvergiert (siehe Abschnitt 4.2).

Chen 1995 speichert Muster aus verschiedenen Thesauri in einem Hopfield-Netzwerk. Benutzer definieren Cluster von zusammengehörenden Begriffen, die aus verschiedenen Thesauri stammen und definieren so Transformationen. Im Retrievalfall liefert das Netz nach Eingabe von Termen ähnliche Terme aus mehreren Thesauri. Ein weiteres Hopfield-Netzwerk nach diesem Prinzip stellen Lin/Chen 1994 im Bereich multilinguales Retrieval vor. Es leistet einen Umstieg zwischen chinesischen und englischen Termen.

Allerdings weicht diese Implementierung stark vom Konzept der Hopfield-Netzwerke ab. Anstelle der Hopfield-Lernregel, die garantiert, dass jedes Muster als Energie-Minimum abgespeichert wird, werden die Verbindungsstärken aus der Indexierung übernommen. Damit ähnelt das Netz eher einem Spreading-Activation-Netzwerk mit nur einer Schicht als einem Hopfield-Netzwerk.

Auch die in Abschnitt 4.3 beschriebenen Spreading-Activation-Netzwerke können für Transformationen adaptiert werden. Diese Systeme bestehen meist aus zwei Schichten, zwischen denen Aktivierung fließt. Eine davon repräsentiert die Anfrage-Terme und die zweite die Dokumente. Die Stärke der Verbindungen beschreibt die Wichtigkeit des Terms für das jeweilige Dokument. Beim Einsatz für Transformationen repräsentiert jede Schicht eine Begriffssystematik und die Verbindungsstärken geben die Gewichte der Assoziationen zwischen den Termen in den unterschiedlichen Systematiken an. In einem solchen System verschwimmt die Grenze zwischen einem auto- und einem hetero-assoziativen Netzwerk. Da auch innerhalb der Term-Schicht eines Spreading-Activation-Netzwerks Verbindungen möglich sind, lässt sich das oben besprochene zweischichtige Netzwerk formal kaum davon abgrenzen. Vielmehr kann es als Sonderfall einer Term-Schicht mit Verbindungen innerhalb der

Schicht betrachtet werden, bei dem nur zwischen bestimmten Gruppen Verbindungen zugelassen sind. Die Gruppen sind die unterschiedlichen Begriffs-Systematiken. Die Abgrenzung der Begriffs-Systematiken kann durch zwei Verfahren verdeutlicht werden. Zum einen kann zwischen zwei Indexierungsvokabularen eine Schicht mit Objekten eingezeichnet werden, die von beiden Vokabularen beschrieben wird. Ein solches Netz, wie es Abbildung 4-12 skizziert, schlagen Lee/Dubin 1999 vor.

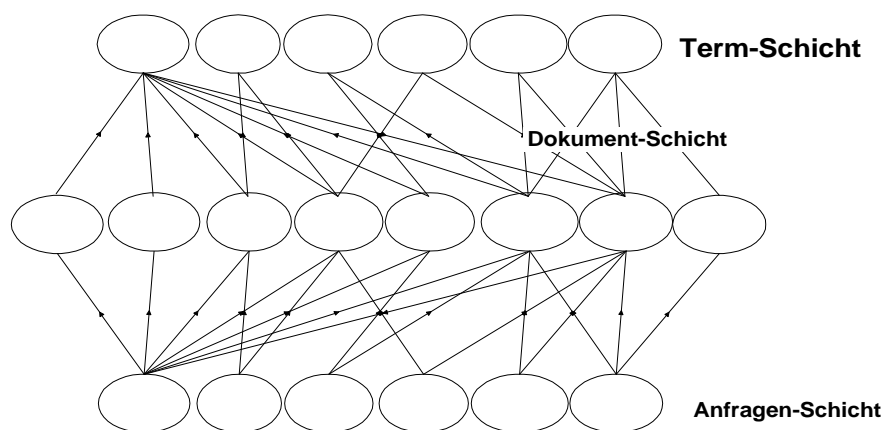


Abbildung 4-12: Mögliche Architektur eines Spreading-Activation-Netztes für Transformationen

Diese Architektur zeichnet sich durch hohe Flexibilität aus, da sowohl Terme aus beiden Vokabularen als auch ein Objekt in der mittleren Schicht als Input dienen können. Ein Schritt in diese Richtung erfolgt bereits durch die dreischichtigen Spreading-Activation-Netzwerke, die zusätzlich zur Dokument- und Term-Schicht eine Autoren-Schicht besitzen. Formal bilden die Autoren auch Eigenschaften der Dokumente. Damit bilden Terme und Autoren die Begriffs-Systematiken, die mit den Dokumenten verbunden sind und die sich formal sogar teilweise überlappen könnten (vgl. Abschnitt 4.3.4).

4.3.4 Vergleich der Spreading-Activation-Netzwerke mit dem Vektorraum-Modell

Die Dokument-Term-Matrix aus dem Vektorraum-Modell für Information Retrieval wird beim Spreading-Activation-Netzwerk als Verbindungsmatrix übernommen. Damit wird sie zur Dokument-Neuron-Term-Neuron-Matrix. Eine vollständige Verbindungsmatrix für ein neuronales Netz sollte sich jedoch immer über alle Neuronen erstrecken, so dass sowohl Zeilen als auch Spalten für alle Neuronen vorhanden sind. Damit werden Verbindungen zwischen allen Paaren von Neuronen möglich. Die folgende schematische Übersicht zeigt diese Matrizen im Überblick:

| | Dokumente | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------|---|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| Terme | <table border="1"> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | Dokument-Neuronen | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---------------|---|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| Term-Neuronen | <table border="1"> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | Dokument-Neuronen | Term-Neuronen | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------------|---|---------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| Dokument-Neuronen | <table border="1"> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table> | | | | | | | | | | | | | | | | | | | <table border="1"> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Term-Neuronen | <table border="1"> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table> | | | | | | | | | | | | | | | | | | | | | | | | | <table border="1"> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table> | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Abbildung 4-13: Schematische Gegenüberstellung von Dokument-Term-Matrix und vollständiger Verbindungsmatrix, bei der die entsprechenden Verbindungen grau unterlegt sind.

Neben den Verbindungen zwischen Termen und Dokumenten können damit auch Verbindungen innerhalb von Schichten definiert werden. Die von einigen Modellen vorgenommene Erweiterung des Standard-Modells für Spreading-Activation-Systeme um Verbindungen innerhalb von Schichten lässt sich demnach als Erweiterung der Matrix interpretieren. Auch die Frage nach der Symmetrie der Verbindungen stellt sich bei Betrachtung der Matrix erneut. Die folgende Abbildung fasst die Möglichkeiten zusammen:

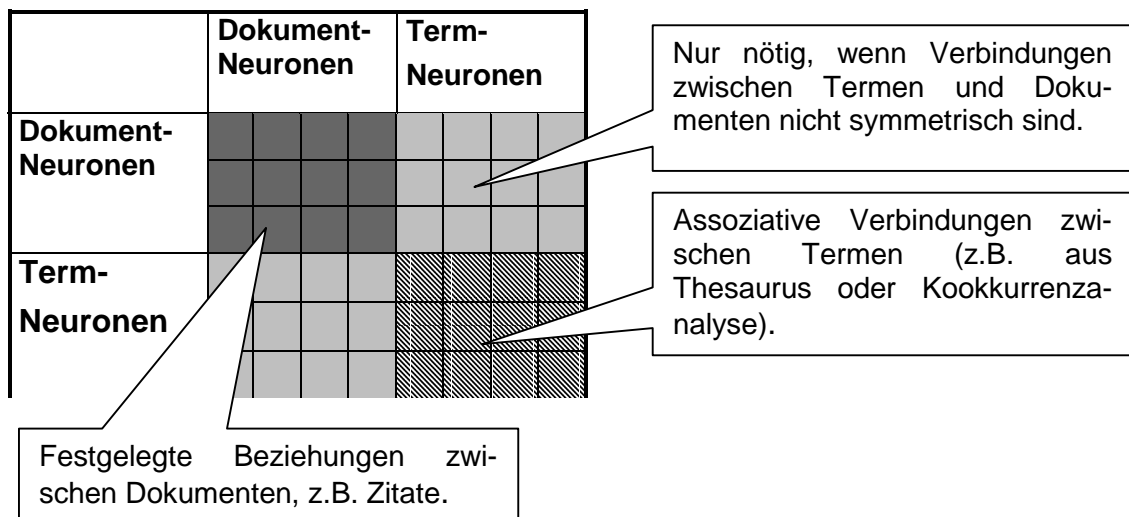


Abbildung 4-14: Die vollständige Verbindungsmatrix ermöglicht zum einen assoziative Verbindungen innerhalb von Schichten. Daneben veranschaulicht diese Darstellung, dass die Symmetrie der Verbindungen zwischen Dokumenten und Termen nicht selbstverständlich ist.

Die Einführung einer zusätzlichen Schicht für Autoren, wie sie manche Systeme vornehmen, eröffnet weitere Möglichkeiten, wie Abbildung 4-15 zeigt.

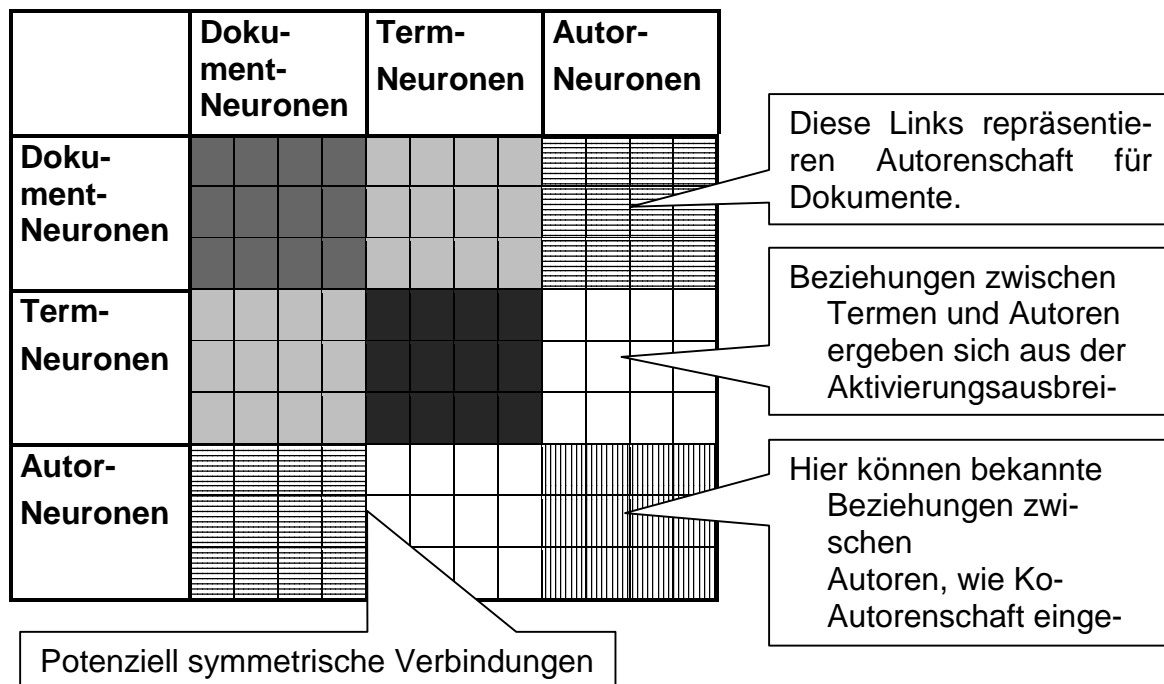


Abbildung 4-15: Die Verbindungsmatrix nach der Einführung einer Autoren-Schicht. Die nicht beschrifteten Verbindungstypen sind bereits in den obigen Abbildungen enthalten.

Dieser Überblick unterstreicht erneut die Nähe zwischen Spreading-Activation-Modellen und dem Vektorraum-Modell.

4.3.5 Fazit: Spreading-Activation-Modelle

Die Spreading-Activation-Modelle haben sich im Information Retrieval als mögliche Alternative zum probabilistischen Modell und dem Vektorraum-Modell etabliert. Die abschließende Bewertung berücksichtigt folgende Vor- und Nachteile:

Vorteile der Spreading-Activation-Netze:

- Erfolg einiger implementierter Systeme
- Tragfähige Metapher des IR-Prozesses
- Flexibilität innerhalb des Modells

Nachteile der Spreading-Activation-Netze:

- Ansätze zur Flexibilität werden allgemein wenig benutzt
- Erfolgreiche Systeme schöpfen die Flexibilität nicht aus
- Spreading-Activation-Netze sind formal dem Vektorraum-Modell sehr ähnlich
- Diese Ähnlichkeit ist in den implementierten Systemen besonders hoch
- Die Lernfähigkeit ist niedrig ausgeprägt
- Spreading-Activation-Netze nutzen nicht alle Stärken neuronaler Netze

Die Spreading-Activation-Modelle bieten einige erhebliche Vorteile:

- Die Leistungsfähigkeit der Spreading-Activation-Modelle ist mit anderen Information Retrieval Techniken vergleichbar, wie die guten Ergebnisse von PIRCS und Mercure bei den TREC-Studien gezeigt haben (siehe Abschnitt 4.7, cf. Voorhees/Harman 1998a, 1999a). Für eine endgültige Entscheidung, welches Modell besser ist, reichen diese Ergebnisse allerdings nicht aus. Ein solches Ergebnis ist auch auf lange Sicht nicht zu erwarten, da die Eigenschaften der Anwendungsgebiete sehr unterschiedlich sind und Information Retrieval Systeme bei Vergleichen mit unterschiedlichen Kollektionen sehr unterschiedliche Ergebnisse erreichen.
- Die Ausbreitung von Aktivierung als Metapher für Relevanz und Interesse des Benutzers ist sehr plausibel. Die Aktivierung eines Dokuments durch seine Index-Terme ist anschaulich und Term-Expansion und Relevanz-Feedback ergeben sich in diesem Rahmen sehr natürlich. Wie die Diskussion der Funktionsweise in Abschnitt 4.3.1 unterstreicht, bieten Spreading-Activation-Netze eine gute Metapher des Information Retrieval Prozesses.
- Die Spreading-Activation-Modelle sind äußerst flexibel. Prinzipiell können sowohl Dokumente als auch Index-Terme gleichzeitig als Input und Output dienen. Weiterhin sind nach jedem Aktivierungsschritt Eingriffe des Benutzers möglich. So kann als zusätzlicher Input zu jeder Zeit die Auswahl und die Aktivierung zusätzlicher Terme oder Dokumente erfolgen. Die Möglichkeiten des Benutzers und somit die Interaktivität des Retrievalprozesses können im Rahmen des Modells auf einfache Weise erhöht werden.

Den Vorteil der erhöhten Flexibilität schränken die folgenden Beobachtungen ein:

- Trotz der Plausibilität einer Erhöhung der Interaktivität durch die Möglichkeit zahlreicher Eingriffe durch den Benutzer lässt sich diese Vorgehen wohl kaum in die Praxis übertragen. Zahlreiche Untersuchungen zeigen, dass Relevanz-Feedback als einfache Art der Interaktion nachweislich eine der besten Strategien zur Verbesserung des Retrievalergebnisses ist. Trotzdem wird es von Benutzern relativ selten eingesetzt (cf. Womser-Hacker 1997). Sie scheinen bereits diesen Aufwand zu scheuen, so dass kaum davon auszugehen ist, dass eine verstärkte Interaktion überhaupt genutzt würde. Die große Flexibilität der Spreading-Activation-Netze hinsichtlich Eingabe und Möglichkeiten der Interaktion kann somit nicht als entscheidender Vorteil gewertet werden.
- Die Flexibilität u.a. durch häufiges Feedback ist insbesondere bei den in TREC getesteten Systemen nicht realisiert und nicht vorgesehen. Sie setzt die mehrfache Aktivierungsausbreitung zwischen den Schichten voraus, die nur wenige Systeme erlauben.

Die letzte Beobachtung führt zur Frage, was die Spreading-Activation-Modelle als IR-Modell auszeichnet. Die von vielen Autoren betonte Nähe zu den klassischen Modellen deutet bereits an, dass die Spreading-Activation-Modelle keine völlig neuartige Modellklasse darstellen.

- Die Standard Spreading-Activation-Netzwerke für Information Retrieval ähneln stark der Vektorraum-Repräsentation. Doszkocs et al. 1990:231 drücken dies folgendermaßen aus: „A connectionist network representation can be compared with

the well-known vector space model of information retrieval“. Auch Chen 1995:199 teilt diese Ansicht: „Neural networks computing, in particular, seems to fit well with conventional retrieval models such as the vector space model and the probabilistic model.“

- Die Ähnlichkeit geht jedoch über eine formale Ähnlichkeit der Repräsentation weit hinaus. So weist Mothe 1994 (cf. Abschnitt 4.3.2.4) theoretisch und empirisch nach, dass ein Spreading-Activation-Modell nach einem Aktivierungsschritt äquivalent zum Vektorraum-Modell ist. Erst wenn weitere Aktivierungsschritte folgen, unterscheidet sich das Spreading-Activation-Netzwerk von etablierten IR-Modellen. Dies ist jedoch nur in wenigen Systemen der Fall und dazu gehören nicht die in TREC überprüften Systeme PIRCS und Mercure (siehe Abschnitt 4.7). Damit wird auch dieser potenzielle Vorteil der Spreading-Activation Modelle nicht ausgenutzt. Möglichweise induziert der Aufbau der Standard-TREC Tests diese Einschränkung, da Relevanz-Feedback darin eine untergeordnete Rolle spielt.
- Wie in vielen anderen Information Retrieval Modellen kann auch bei den Spreading-Activation-Modellen die Bedeutung heuristischer Faktoren nicht unterschätzt werden. Dazu zählen die Wahl der Aktivierungsfunktion, die Lernregel, die Anzahl der Aktivierungsschritte und die Wahl zahlreicher anderer Parameter. Besonders wichtig sind in diesem Zusammenhang hemmende Mechanismen, die ein schnelle und vollständige Aktivierung des Netzes nach einigen wenigen Schritten verhindern.

Aus der Sicht der neuronalen Netze bestehen Schwächen der Spreading-Activation-Modelle darin, dass die Lernfähigkeit niedrig ist und keine versteckte Schicht die Mächtigkeit der Netze erhöht. Eine versteckte Schicht mit symbolisch nicht interpretierbaren Neuronen erhöht die Leistungsfähigkeit eines Netzes wesentlich (siehe Abschnitt 3.5.4.1, cf. Zell 1994).

- Bei den Vorstellungen der Spreading-Activation-Netze wird die Initialisierung des Netzes anhand der Termhäufigkeiten meist als Lernen bezeichnet. Dabei handelt es sich jedoch nicht um Lernen, sondern lediglich um das Einstellen der Verbindungen auf die Ausgangssituation. Die Möglichkeit, ohne die initialen Verbindungsstärken ein Netz allein aufgrund von Benutzerurteilen zu trainieren, wird teilweise pessimistisch eingeschätzt. Dabei gilt vor allem die notwendige Größe der Netze für realistische Anwendungen als wichtiges Argument:

„The possibility of developing learned connection strengths based on a large number of queries and the desired document weight is infeasible given the size of the network. The connection weights are determined using techniques developed for information retrieval and are fixed.“ (Wilkinson/Hingston 1992:72)

- Die größte Schwäche der Spreading-Activation-Modelle wird aus der kognitions-wissenschaftlichen Warte deutlich. Die große Stärke neuronaler Netze besteht in ihrer sub-symbolischen Leistungsfähigkeit, die besonders der Backpropagation-Ansatz repräsentiert. Dieses Potenzial wird nicht genutzt, keines der Modelle verfügt über versteckte Schichten, die nicht symbolisch interpretiert werden können. Diese Einschätzung beruht jedoch nicht nur auf der kognitions-wissenschaftlichen Theorie. Das Backpropagation-Netzwerk ist das mit Abstand am häufigsten

eingesetzte Modell im Rahmen der neuronalen Netze. Es kann umfassende Klassen von Funktionen implementieren, von denen viele von zweischichtigen Netzen nicht erreicht werden können. Da nicht bekannt ist, von welcher Art und Komplexität eine optimale Ähnlichkeits- oder Match-Funktion ist, sollte ein möglichst mächtiges Modell gewählt werden.

Trotz der genannten Schwächen bilden die Spreading-Activation-Modelle ein plausibles Modell, das den verbreiteten Information Retrieval Modellen formal sehr ähnlich ist und bei TREC teilweise vergleichbare Ergebnisse erzielen konnte (siehe Abschnitt 4.7). Bereits modellintern ergeben sich Verbesserungsmöglichkeiten. Unter verschiedenen Blickwinkeln fallen Schwächen des Modells ins Auge. Die semantischen Netze bilden für den wenig strukturierten Bereich des Text-Retrieval keine Alternative. Im folgenden werden daher weitere IR-Modelle mit neuronalen Netzen analysiert.

4.4 Kohonen-Netze im Information Retrieval

Kohonen-Netze oder selbstorganisierende Karten (Self-Organizing-Maps, SOM) erfreuen sich in den letzten Jahren im Information Retrieval steigender Beliebtheit. Ihre Funktionsweise erläutert Abschnitt 3.5.1. Kohonen-Netze dienen als unüberwachtes Cluster-Verfahren, das ähnliche Muster auf topologisch nahe Neuronen in der Kohonen-Schicht abbildet. Die Abbildung von Dokumenten aus dem vieldimensionalen Termraum auf eine zweidimensionale dargestellte Karte nutzen viele Ansätze als Möglichkeit zur Visualisierung und Interaktion. Dahinter steht die Hoffnung, dass semantisch ähnliche Dokumente oder Terme in der Kohonen-Schicht nahe beieinander liegen. Gloor 1997 sieht Ähnlichkeit neben z.B. Verlinkung und Sequentialisierung als eines von sieben Konzepten für die Gestaltung von Navigation in Hypertext-Systemen wie dem Internet.

4.4.1 Grundprinzip

Clustering mit dem Kohonen-Netz stellt wie Latent Semantic Indexing (siehe Abschnitt 2.5) eine Form der Dimensionsreduktion dar.

Bei der Reduktion gehen natürlich immer Aspekte des Ausgangsraums verloren wie Abbildung 4-16 illustriert. Bei LSI werden diese Verluste durch Weglassen kleiner Singular Values gesteuert, die bei der Rekonstruktion zu Verlusten führen. Bei SOM ist die Rekonstruktion nicht möglich und es gibt keine Abschätzung über die Qualität der Komprimierung. Während LSI und ähnliche Verfahren nach der Reduktion eine Ähnlichkeitsberechnung anschließen, dienen Self Organizing Maps vorwiegend der Visualisierung.

Bei explorativen und assoziativen Suchen sollen Benutzer ausgehend von bekannten Objekten entlang der Cluster interessante Dokumente oder Terme finden. Die Verwendung des Begriffs Dimension für die Kohonen-Schicht in der Literatur ist zweideutig. Ein Kohonen-Netz bildet grundsätzlich einen n -dimensionalen Raum in einen m -dimensionalen ab. Während die Dimensionen der Eingabe-Schicht durch die Daten vorgegeben sind, ist die Anzahl der Klassen in der Kohonen-Schicht beliebig. Die Anzahl wird heuristisch festgelegt und sollte der Anzahl der gewünschten Klassen entsprechen. Die Kohonen-Schicht dient meist der Visualisierung und ordnet die Neuronen zweidimensional an. Die m Dimensionen oder Cluster werden zu Koordi-

naten von Punkten im zweidimensionalen Darstellungsraum. Grundsätzlich können die Ausgangs- oder Kohonen-Neuronen jedoch beliebig und in mehr als drei-dimensionalen Räumen angeordnet werden.

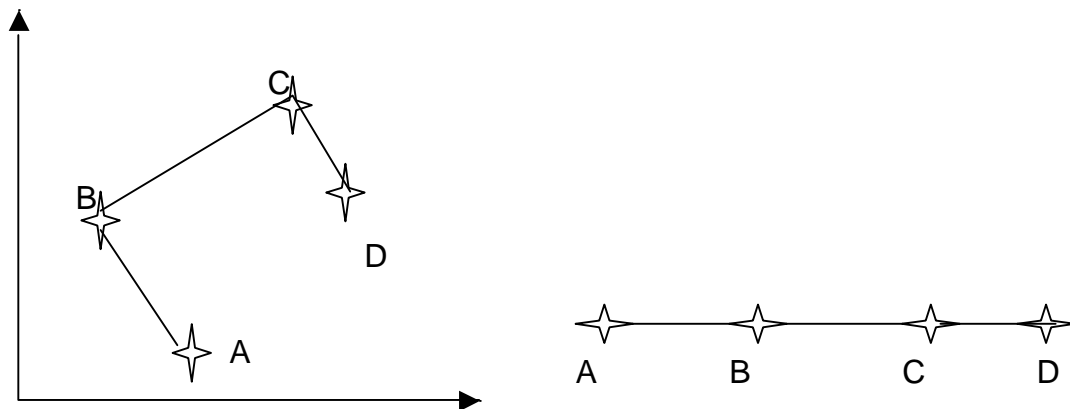


Abbildung 4-16: Reduktion durch Abbildung von einem zweidimensionalen auf einen eindimensionalen Raum. Bereits dieses einfache Beispiel zeigt, dass die Abbildung nicht völlig ähnlichkeiterhaltend ist. In der zweidimensionalen Darstellung sind sich die Objekte A und D ähnlicher als B und C. Auf der Geraden haben A und D dagegen minimale Ähnlichkeit.

Die Mächtigkeit von Kohonen-Netzen bestätigen z.B. Graupe/Kordylewski 1998, die ein System für medizinische Diagnose mit realen Daten vorstellen. Das System klassifiziert Patientendaten besser als statistische Verfahren und andere neuronale Netze. Es implementiert eine Datenbankabfrage mit unvollständigen Mustern. Die im folgenden beschriebenen Systeme bearbeiten Probleme des Text-Retrieval mit dem Kohonen-Netz.

4.4.2 Systeme

Lin et al. 1991 nutzen ein Kohonen-Netz für das Erstellen einer zweidimensionalen semantischen Karte. Eine Kollektion von 140 Artikeln zur Künstlichen Intelligenz wurde anhand der Titel indexiert und mit einem Kohonen-Netzwerk verarbeitet. Die Dokumentvektoren mit der Länge 25 dienten als Input für das Training. Das heißt, es wurden nur 25 Terme analysiert. Die Punkte in der Karte repräsentieren Dokumente, die Regionen sind aber nach Termen benannt. Dazu erstellten Lin et al. 1991 Term-Vektoren, in diesem Fall Einheitsvektoren, die ein Dokument mit nur einem Term nachbilden. Der Term-Vektor mit der höchsten Ähnlichkeit zu einem Punkt oder Neuron in der Karte bzw. dessen Gewichtsvektor bestimmt die Benennung. Die Region eines Terms erstreckt sich nur über benachbarte Neuronen, da das Lernverfahren der Kohonen-Karte Ähnlichkeit als Nähe im zweidimensionalen Raum darstellt. Lin et al. 1991 beobachten, dass die Dokument-Vektoren neben den Termen, mit denen sie indexiert sind, weitere Terme mit Gewichten ungleich Null enthalten. Diese bilden assoziative Erweiterungen der Indexterme, da sie in der Kollektion häufig gemeinsam mit ihnen vorkommen. Aufgrund der starken Dimensionsreduktion eignet sich die Karte als Grundlage einer grafischen Benut-

zungsoberfläche für assoziative Navigation im Korpus. Die Autoren schlagen vor, die Karte zur Visualisierung des Ergebnisses eines IR-Systems einzusetzen. Eine Diskussion der Kohonen-Netze als Benutzungsoberfläche folgt am Ende dieses Abschnitts.

Rozmus 1995 kritisiert an dem Original-Algorithmus von Kohonen, dass die Muster nicht homogen in der Output-Schicht verteilt sind. In seiner Implementierung, die 50.000 bibliographische Angaben in Cluster einteilt, verwendet er einen veränderten Algorithmus, der die Muster besser im Zielraum verteilt. Dabei stellt sich die Frage, inwieweit die Daten nicht homogene Verteilungen erfordern. Da ein Cluster-Algorithmus aber ohnehin nicht die gesamte, komplexe Struktur widerspiegelt und eine bessere Verteilung die Benutzerfreundlichkeit erhöht, können derartige Überlegungen durchaus sinnvoll sein. Zavrel 1996 wendet eine Erweiterung des Kohonen-Netzes u.a. auf die Cranfield-Kollektion an. Nach einer Einordnung in die Forschungskontexte Visualisierung, Kohonen-SOM und neuronale Netze im IR präsentiert Zavrel 1996 seine Erweiterung des Kohonen-Algorithmus, bei der während des Trainings neue Neuronen in die Kohonen-Schicht eingefügt werden. Dazu misst jedes Kohonen-Neuron die Größe des von ihm abgedeckten Eingaberaums. Das Neuron, das den größten Bereich von potenziellen Eingabemustern vertritt, erhält einen Nachbarn, der diesen Raum mit ihm teilt und so eine bessere Differenzierung zulässt. Damit verfolgt Zavrel 1996 ein ähnliches Ziel wie Rozmus 1995, der auch eine homogene Verteilung der Muster anstrebt.

Zavrel 1996 evaluiert die Qualität des Kohonen-Netzwerks als IR-System im Vergleich zu anderen Clustering-Methoden und dem Vektorraum-Modell. Dazu wird das Clustering-Verfahren als IR-Modell interpretiert und die Anfrage dient als Eingabe im Kohonen-Netz. Der Cluster, den das aktivierte Ausgabe-Neuron repräsentiert, ist die Ergebnismenge. Die Evaluierung erfolgt also auf der Basis einer booleschen Menge und kommt damit dem Clustering-Verfahren entgegen. Das Ergebnis des Vektorraum-Modells, eine gerankte Liste aller Dokumente, wird ebenfalls als boolesche Menge interpretiert. Das Vektorraum-Modell verwendet in diesem Fall das Kosinus-Ähnlichkeitsmaß. Die Anzahl der Dokumente bestimmt Zavrel 1996 aus der Anzahl der durchschnittlich in den Clustern enthaltenen Dokumenten. Die vom Vektorraum-Modell gelieferte Liste von Dokumenten wird bei einem Schwellenwert in zwei boolesche Mengen geteilt. Die Clustering-Verfahren geben damit das Retrieval-Verfahren vor und die im Vektorraum-Modell enthaltene Reihenfolge der Dokumente geht für die Bewertung verloren. Zudem wird die Größe der zu berücksichtigenden Menge aus dem Durchschnitt der Clustering-Verfahren berechnet. Trotzdem schneidet das Vektorraum-Modell in dem Vergleich von Zavrel 1996 besser ab als alle Clustering-Verfahren, unter denen das modifizierte Kohonen-Netz die beste Qualität erreicht. Als Maßstab zieht Zavrel 1996 das E-Maß von van Rijsbergen heran, das Recall und Precision kombiniert.

Chen et al. 1996 übertragen den Kohonen-Ansatz auf Internet-Dokumente. Trotz der Beschränkung auf eine Kategorie des Internet-Katalogs Yahoo (entertainment) führen sie eine Vielzahl semantischer Karten ein. Sobald ein Cluster mehr als eine bestimmte Menge von Seiten enthält, wird er auf einer eigenen Karte dargestellt. Die auf Basis einer Trainingsmenge von 10.000 Seiten entstehenden Karten wurden in Benutzertests empirisch getestet. Dabei zeigte sich, dass viele Versuchspersonen im Umgang mit der Karte ein Gefühl der Desorientierung befiel, das auch in assoziativ verknüpften Hypertexten vorkommt und als „lost in hyperspace“-Syndrom bekannt ist (cf. Kuhlen 1991). Die Orientierung fiel oft schwer und die Benutzer fragten nach geordneten Listen der enthaltenen Suchwörter. Die Karte eignet sich nach

Einschätzung der Autoren vor allem für assoziative Suchen, bei denen Benutzer einen Überblick über die vorhandenen Suchbegriffe wünschen.

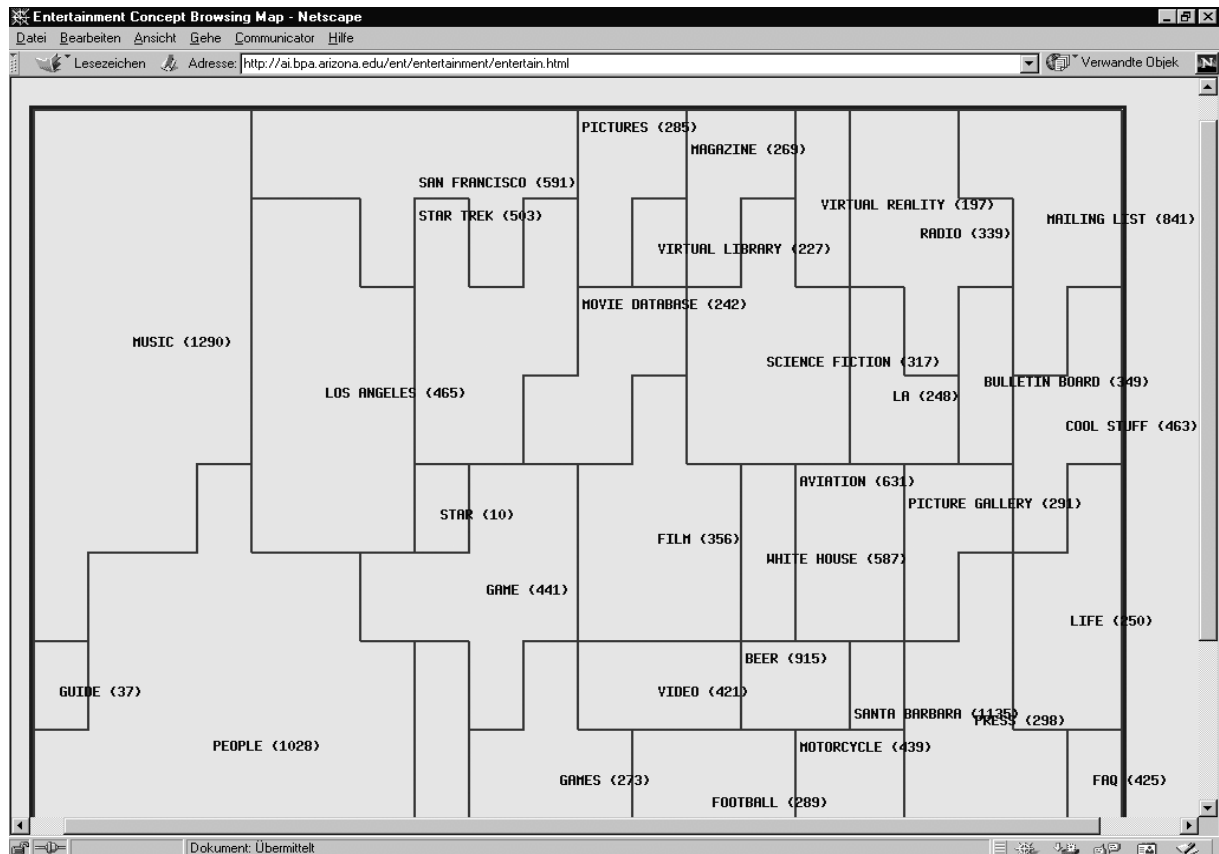


Abbildung 4-17: Ein Ausschnitt aus der Visualisierung der Kohonen-Karte von Chen et al. 1996

Eine kleinere Implementierung dieses Systems organisiert Äußerungen aus elektronischen Online-Besprechungen (cf. Orwig et al.1997). Die Testmenge umfasst 202 Dokumente mit 190 Termen und damit Eingabedimensionen. Die Dokumente werden auf eine 10x20 Neuronen große Kohonen-Schicht abgebildet. Die Qualität der Cluster wird mit intellektuell erstellten Cluster verglichen, wobei die maschinell erstellten etwas schlechter waren. Die Kosten für die Erstellung sind jedoch erheblich niedriger. Falls in der Domäne solche Cluster für Analysen von elektronischen Treffen regelmäßig erstellt werden, bietet sich die Kohonen-Karte als Alternative an. Das Problem der Überladung einer Karte tritt in diesem Anwendungsfall kaum auf, da jede Besprechung nur eine begrenzte Anzahl von Redebeiträgen umfasst.

Eine weitere Implementierung einer Kohonen-Karte, die als Interface im Internet zur Verfügung steht, ist WEBSOM¹ (cf. Kohonen 1997/1998, Honelka et al. 1997). WEBSOM stellt Karten mit großen Mengen von Newsgroup-Artikeln zur Verfügung. Aufgrund der großen Anzahl von Dokumenten ordnen die Autoren wie im Ansatz von Chen et al. 1996 die Karten in mehreren Schichten an. Die Cluster in der ersten

¹ <http://websom.hut.fi/websom>

Karte führen also zu weiteren Karten und erst beim dritten oder vierten Schritt gelangt der Benutzer zu einem Cluster von Newsgroup-Artikeln. Die Interaktion beschränkt sich bei mehrschichtigen Systemen nicht auf das Klicken in einer Karte. Zusätzlich ist in der Seite ein Kontrollelement integriert, um zu einer anderen Karte zu wechseln. Die Beschriftung der Cluster ist stark verkürzt, was die Interaktion erschwert. Der Benutzer muss die Langform der Einträge rechts außerhalb der Visualisierung lesen. Dies stört die angestrebte Direktheit der Interaktion.

Kaski 1998 schlägt die Methode Random-Mapping für Dimensionalitätsreduktion als Vorverarbeitung für Kohonen-SOM vor. Dabei wird die originale Dokument-Term-Matrix mit einer Matrix zufällig gewählter Werte multipliziert. Die Größe der Zufallsmatrix bestimmt die Reduktion der Dimensionalität in der Ergebnismatrix. Kaski 1998 führt das Random-Mapping ein und experimentiert mit einer Kollektion von 1800 Artikeln aus 20 Newsgroups. Die ursprünglich 5781 Dimensionen werden mit Random-Mapping auf 90 Dimensionen und mit Principal-Component-Analysis (PCA, eine Art Faktorenanalyse) auf 50 Dimensionen reduziert. Dann klassifiziert eine Kohonen-Karte mit 768 Neuronen die Dokumente. Jede Klasse wird nach der Newsgroup benannt, von der sie die meisten Artikel enthält. Die Qualität misst Kaski 1998 über der Trennfähigkeit, indem er die Anzahl der Artikel bestimmt, die nicht zur in dem Cluster vorherrschenden Newsgroup gehören. Dies erscheint fragwürdig, da eine Zuordnung unabhängig von der Zugehörigkeit zu den Diskussionsforen semantisch sehr sinnvoll sein kann. Häufig senden Benutzer identische Nachrichten an mehrere Newsgroups. Nach Kaski 1998 erzielte Random-Mapping eine etwas bessere Trennfähigkeit als PCA. Beide waren wiederum etwa 1% schlechter als die Originalmatrix.

Kohonen 1998 wendet das Verfahren auf größere Datenmengen an. Eine Million Newsgroup-Artikel führen zu 63.773 Termen oder Dimensionen, die durch Random-Mapping auf 315 Dimensionen reduziert wurden. Das Kohonen-Netz ordnete die Dokumente in über 100.000 Cluster. Mehrfach geschachtelte Karten erlauben den Zugriff. Inwieweit ein Browsing-Ansatz den Zugriff auf eine derart große Menge von Dokumenten erleichtert, ist eher fragwürdig.

Schatz 1998 überträgt die Input-Muster auf einen dreidimensionalen Raum, in dem der Benutzer navigiert (*spaceflight*). Die dritte Dimension ist aber dabei nicht homogen besetzt. Nur die am Rand der zweidimensionalen Oberfläche liegenden Bereiche verlaufen nennenswert in der dritten Dimension. Dadurch entsteht eine Art Terrain, das sicher leichter zu verstehen ist, als ein homogen mit Dokumenten besetzter dreidimensionaler Raum. Allerdings ist fraglich, ob jedoch ein Vorteil gegenüber den obigen zweidimensionalen Darstellungen besteht.

Hui/Goh 1996 implementierten ein komplettes IR-System, dessen Kern ein Kohonen-Netz bildet. Ihr System lernt aus Relevanz-Feedback, indem es seine Gewichte verändert, hat jedoch einige Schwächen. In der Trainingsphase extrahiert es Terme, die dann in eine kodierte Repräsentation umgeformt werden. Das Trainingsverfahren präsentiert sie dem Netz einzeln und der Kohonen-Algorithmus ordnet sie in Cluster. Beim Retrieval werden ebenfalls die Terme der Anfrage einzeln an das Netz gelegt und einem Cluster zugeordnet. Die Terme aller Cluster, die die Anfrage-Terme aktivieren, bilden dann die eigentliche Anfrage. Das Kohonen-Netz dient also der Termexpansion. Alle Dokumente, in denen die Terme in den aktivierten Clustern vorkommen, bilden die Ergebnismenge. Wie bei anderen Kohonen-Modellen entsteht dadurch ein unflexibles Retrieval-System.

Zudem schlagen Hui/Goh 1996 ungünstig kodierte Repräsentationen vor. Die Terme werden entweder auf Basis ihrer Schreibweise oder einer phonetischen Transkribierung in hexadezimale Zahlen umgewandelt. Dadurch werden zwar ähnlich

geschriebene oder gesprochene Deskriptoren gefunden, eine semantische Ähnlichkeit zwischen ähnlichen Repräsentationen ist jedoch rein zufällig. Bei der Standard-Repräsentation repräsentiert jedes Vektorelement einen Term. Jedes Dokument und jede Anfrage besteht aus einem Vektor, in dem die Werte die Wichtigkeit des jeweiligen Terms für das Dokument ausdrücken. Dadurch haben ähnliche Dokumente ähnliche Repräsentations-Vektoren.

Hui/Goh 1996 integrieren als einzige unter den vorgestellten Kohonen-IR-Systemen Relevanz-Feedback. Für die Benutzerurteile geben sie eine Skala von eins bis zehn vor. Die Skalenwerte sind in einem Fuzzy-Modell den Konzepten *most important*, *relatively important* und *least important* zugeordnet. Ein Fuzzy-Algorithmus verändert ausgehend von den Benutzerurteilen die Verbindungsgewichte. Als Vergleichsmaßstab für die Performanz des Systems bei der Wiedererkennung von Termen dient ein ART-Netzwerk. Die Performanz der SOM liegt minimal über der von ART, wobei die Auswirkungen von Relevanz-Feedback nicht untersucht werden. Die Tests basieren auf einer Kollektion von nur 100 Dokumenten, so dass sie nicht sehr aussagekräftig sind.

Der hohe Zeitbedarf der Kohonen Self-Organizing Map gilt als einer ihrer größten Nachteile. Dem begegnen z.B. Kaski 1998 und Kohonen 1998 mit einer Komprimierung der Eingangsdaten. Merkl 1995 komprimiert mit einem Backpropagation-Netzwerk. Sein experimentelles IR-System sucht nach den Beschreibungstexten einer C++ Klassenbibliothek, aus der 489 Terme extrahiert wurden. In Merkl et al. 1994 werden für einen ähnlichen Anwendungsfall 39 Eigenschaften extrahiert. Das System von Merkl 1995 organisiert die Software-Komponenten auf einer zweidimensionalen Kohonen-Karte. Merkl 1995 verringert die langen Trainingszeiten des Kohonen-Netzes, indem er den Merkmalsraum von 489 auf 75 und einmal 30 Merkmale komprimiert. Ein Backpropagation-Netzwerk, bei dem Input und Output identisch sind, leistet die Komprimierung. Das Netz lernt, Inputmuster zu kopieren. Die dazwischenliegende versteckte Schicht enthält wesentlich weniger Neuronen als die Input- und Output-Schicht und damit der Merkmalsraum. Nach erfolgreichem Training enthält die versteckte Schicht eine reduzierte Repräsentation der Muster, die durch Beschreiten der Verbindungen von der versteckten Schicht in die Output-Schicht wieder expandiert werden kann.

Durch die Komprimierung konnte Merkl 1995 die Trainingszeit seines Netzes auf unter 20% der ursprünglichen Zeit drücken. Allerdings vergleicht er nicht die Qualität der komprimierten mit der ursprünglichen Repräsentation. Die Analyse des Netzes beschränkt sich auf eine intuitive Überprüfung durch den Autor. Die Eignung für reale *software-reuse*-Szenarien müsste in empirischen Untersuchungen überprüft werden.

4.4.3 Fazit: Kohonen-Netze im Information Retrieval

Dieser Überblick über implementierte Information Retrieval Systeme auf der Basis von Kohonen-Netzen zeigt vor allem, dass in realen Umgebungen der Umfang der zu verarbeitenden Daten für eine zweidimensionale Karte zu groß ist. Die meisten Karten erlauben weniger als 1000 Eingangsdimensionen und Muster. Chen et al. 1996 und WEBSOM (cf. Kohonen 1997/1998) testen mit realen Daten und lösen das Problem durch die Einführung mehrerer Schichten von Kohonen-Netzen, was weitere Interaktionsmechanismen zum Wechseln von einer Karte zur anderen erfordert. Das ursprüngliche Grundprinzip einer einfachen und scheinbar natürlichen Visualisierung

wird dadurch überlagert. Eine weitere Schwäche besteht darin, dass das Kohonen-Netz nicht den Kern eines IR-Systems implementiert.

Die meisten Systeme liefern ein vollständiges Cluster von Dokumenten als Ergebnismenge. Die statische Zuordnung der Dokumente in die Cluster bildet damit die Grundlage für das eigentliche Retrieval. In diesem Modell ist der dynamische Aspekt von Retrieval-Prozessen und die Integration von Relevanz-Feedback schwierig zu modellieren. Selbst wenn ein Benutzer einige Dokumente des Clusters als relevant und andere als nicht relevant einordnet, findet eine automatisch neu formulierte Anfrage in der Regel nur wieder das gleiche Cluster. Gelingt es einem Verfahren, dann ein zweites Cluster zu treffen, sind die relevanten Dokumente aus dem ersten gefundenen Cluster nicht mehr in der Treffermenge.

Grundsätzlich erscheint die Kohonen-Karte als Repräsentation einer Dokumentenkollektion problematisch. Um eine einfache und benutzerfreundliche Visualisierung zu erhalten, werden die zahlreichen Dimensionen des Term-Raums auf zwei reduziert. Dass eine solche Reduktion die Struktur der Kollektion noch adäquat widerspiegelt, ist eher unwahrscheinlich. Trotz der visuellen Adäquatheit ist die kognitive Repräsentation des Term-Raums wohl komplexer als eine zweidimensionale Karte. Die Clusterbildung ist stark kontextabhängig. Die zweidimensionale Darstellung erlaubt in einer größeren Kollektion insgesamt nur eine begrenzte Anzahl von Assoziationen in Form räumlicher Nachbarschaft. Zwar bietet die Kohonen-Karte für kleinere Mengen von Dokumenten einen möglichen Einstieg für assoziative und explorative Informationsbedürfnisse, als einziger Zugang ist sie in jedem Fall ungeeignet.

Einen interessanten Anwendungsfall aus Lin et al. 1991 greift Lin 1995 in seinen Tests mit verschiedenen kartenorientierten Darstellungen auf. Die Tests von Lin 1995 basieren auf einer sehr kleinen Menge von Daten. Die Aufgabe für die Benutzer bestand darin, aus 133 dargestellten Dokumenten zehn zufällig ausgewählte zu finden, was keine reale Retrievalsituation widerspiegelt. Die Grundidee der Beschränkung auf kleine Mengen ist jedoch verfolgenswert. Dient die Karte nur der Ergebnisanzeige, ist das Problem der großen Datenmengen gelöst. Durch die Beschränkung auf eine kleine Aufgabe innerhalb des Retrievalprozesses könnten sich die Vorteile der Visualisierung in einem ausgewogenen Gesamtsystem besser entfalten. Optimierte Benutzungsoberflächen für Textretrieval versuchen, einzelne Komponenten so zu integrieren, dass ihre Nachteile durch andere Interaktionsmöglichkeiten aufgehoben werden, gleichzeitig aber ihre Vorteile voll ausgenutzt werden (cf. Krause/Schaefer 1998).

4.5 Adaptive Resonance Theory-Modelle

Information Retrieval Systeme auf Basis der Adaptive Resonance Theory (ART) sind bisher selten. Die Funktionsweise der ART-Netzwerke beschreibt Abschnitt 3.5.2. ART ist ein unüberwachtes Clusterverfahren, das im Gegensatz zu Kohonen-SOM selbständig neue Cluster erzeugt. Für Anwendungen von ART im IR gelten ebenfalls die meisten der unter den Kohonen-Netzen (siehe Abschnitt 3.5.1) diskutierten Probleme von Clustermethoden.

Im letzten Abschnitt erschienen bereits zwei Ansätze, die ART oder ähnliche Verfahren einsetzen. Der Algorithmus von MacLeod/Robertson 1991 berechnet die Ähnlichkeit wie im ART-Algorithmus vorgesehen ebenfalls zweimal. Die Autoren betrachten ihre System als sehr nahe verwandt zu ART. Hui/Goh 1996 implementieren

ein ART-Netzwerk als Vergleichsmaßstab zu ihrem Fuzzy-Kohonen-Netz. Die Fuzzy-Variante des Kohonen-Netzwerks erlaubt abgestufte Zugehörigkeitsgrade von Mustern zu Neuronen in der Kohonen-Schicht. Dagegen ordnet das Standard-Kohonen-Verfahren ein Muster immer nur einem Neuron in der Kohonen-Schicht zu (siehe Abschnitte 3.5.1 und 4.4).

Das prominenteste System in diesem Kontext ist NIRS (Neural Information Retrieval System, cf. Escobedo et al. 1993, Caudell 1994, Smith et al. 1997). Die Firma Boeing setzt NIRS zum Retrieval ähnlicher Bauteile bei der Konstruktion von Flugzeugen ein. Damit ist es dem Fakten-Retrival zuzuordnen. Die Vagheit bei den Anfragen und die Unsicherheit bei der Repräsentation der Objekte führen in dem Anwendungsfall zu typischen Information Retrieval Problemen. In der aktuellsten Beschreibung von Smith et al. 1997 umfasst die Datenbasis über 90.000 Teile und mehrere Tausend Benutzer greifen darauf zu. Die Autoren realisierten Zugriffsmöglichkeiten sowohl auf PC-Basis, auf Workstation und per Browser.

Smith et al. 1997 beschreiben den Anwendungsfall. Untersuchungen hatten gezeigt, dass der Entwurf neuer Bauteile häufig nicht erforderlich ist, da entsprechende Bauteile bereits existieren oder durch kleine Modifikationen aus bestehenden Teilen entwickelt werden können. Effizientes Retrieval ähnlicher Bauteile vermeidet zahlreiche Neuentwürfe, die erhebliche Kosten verursachen. Ziel ist es, bei Eingabe eines neu zu entwickelnden Bauteils, ein Cluster ähnlicher, bereits existierender Teile zu finden. Bei der Entwicklung eines solchen Systems fanden die Konstrukteure bei Boeing keine für alle Anforderungen gültige Definition von Ähnlichkeit, sondern nur sehr allgemeine Kriterien, die eine Rolle spielen. Dies ist durchaus typisch beim Entwurf von Systemen, die intuitive Expertenaufgaben simulieren. Escobedo et al. 1993 wählten in dieser Situation ein nicht überwachtes, selbstorganisierendes Netz, das in der Lernphase keine Zielvorgaben benötigt.

Die Bauteile werden per Computer Aided Design (CAD) digital entworfen, so dass alle Daten maschinell lesbar zur Verfügung stehen. Die Autoren leiten daraus die Repräsentation der Bauteile in Form von binären Vektoren ab. Die Komponenten werden zunächst in eine definierte standardisierte Lage gebracht. Dann extrahiert das System für jedes Teil drei geometrische Repräsentationen, die Umriss, Lage der Befestigungslöcher und Lage der Kanten abbilden. Diese dreifache Repräsentation erlaubt es dem Benutzer, das für seine Situation passende Kriterium zu wählen und die Anzahl der gefundenen Teile in gewissen Grenzen zu modifizieren. Damit hat der Benutzer Möglichkeiten, die ein unüberwachter Klassifikations-Algorithmus in der Regel nicht bietet.

Escobedo et al. 1993 ermöglichen dies durch einen modularen und hierarchischen Aufbau aus zahlreichen ART-1-Modulen. Das erste Modul sortiert alle Bauteile nach seinem Umriss in Cluster. Diese Cluster unterteilt eine zweite Schicht mit zwei ART-1-Netzen in feinere Cluster. Davon greift jeweils eines auf die Repräsentation der Befestigungslöcher und das andere auf die Kanten zu. In einer dritten Schicht werden die Ergebnisse der Cluster für Löcher und Kanten kombiniert. Damit kann ein Benutzer ausgehend von einem Input-Muster mit mehreren Kriterien nach ähnlichen Bauteilen suchen. Will er ähnliche Formen finden, durchläuft das System das erste Modul und liefert eine relativ große Ergebnismenge. Weitere mögliche Kriterien sind Löcher, Kanten oder beides zusammen. Das gesamte System umfasst ca. 8000 ART-1 Netzwerke (Caudell et al. 1994). Auffällig ist, dass Löcher und Kanten nicht erstes oder einziges Kriterium sein können. Offensichtlich spielt dies im Anwendungsfall keine Rolle.

Die Schachtelung mehrerer Schichten von Klassifikatoren ähnelt dem Vorgehen von Chen et al. 1996 und dem WEBSOM-System, die große Bestände von Internet-Seiten mit Schichten von Kohonen-Netzen organisieren (siehe Abschnitt 4.4.2). Bei Escobedo et al. 1993 motivieren diese Schichten jedoch semantisch und schaffen daraus keine komplexe Benutzungsoberfläche, welche die Karte mit hierarchischen Steuerungselementen mischt. NIRS nutzt die Klassifikation nicht zur Interaktion, sondern liefert je nach Kriterium die Cluster auf einem bestimmten Level. Die Komplexität von NIRS erhöht nicht die Komplexität für den Benutzer.

Caudell et al. 1994 adaptieren NIRS für dreidimensionale Repräsentation. Die Verwendung von Pixeln erzeugt sehr große Mustervektoren. Das Lernverfahren von ART ist komplexer als der Kohonen-Algorithmus und erfordert viel Zeit. Caudell et al. 1994 komprimieren die Muster-Vektoren mit einem spezifischen Verfahren für binäre Vektoren und verringern so die Lernzeit erheblich.

NIRS ist ein vielversprechender Ansatz. Es erreicht durch seine Architektur eine Erweiterung der Standard-Klassifikationsverfahren, die dem Benutzer Interaktionsmöglichkeiten bietet. Das System hat sich in der Praxis an realen Daten bewährt, gerade weil es sich an den spezifischen Anwendungsfall anpasst. Deshalb lässt es sich nicht ohne weiteres auf andere Bereiche übertragen. Eine Anpassung an Textretrieval fällt wie für alle Clustering-Verfahren schwierig. Trotzdem ist das Faktenretrievalsystem NIRS aufgrund der vagen Natur der Repräsentation und der Vagheit des Anwendungsfalls Ähnlichkeit ein typisches IR-System.

4.6 Backpropagation-Architekturen für Information Retrieval

Modelle für Information Retrieval auf der Basis neuronaler Netze sollten die bisher diskutierten Schwächen vermeiden. Damit gelten vor allem die folgenden Zielvorgaben, die sich aus einer stärkeren Berücksichtigung der Stärken neuronaler Netze ergeben:

- Die sub-symbolischen Fähigkeiten des Backpropagation Ansatzes erlauben die Implementation einer großen Anzahl Klassen von Funktionen.
- Die Lernfähigkeit sollte stark ausgeprägt sein.
- Die im IR bewährten üblichen Indexierungsverfahren, Repräsentationen und Gewichtungsschemata sollten benutzt werden können.
- Der Kern des IR-Prozesses sollte auch den Schwerpunkt der Modellierung bilden.

Dafür bietet das Backpropagation-Netzwerk gute Ansatzpunkte. Dieser Abschnitt fasst Möglichkeiten zusammen, den Backpropagation-Algorithmus im Information Retrieval einzusetzen.

4.6.1 Lernen als Gradientenabstieg

Der Backpropagation-Algorithmus gehört zu den Gradientenverfahren, die eine Fehlerfunktion orthogonal zur Gradienten minimieren (cf. Zell 1994:106). Auch andere lernende Systeme, die keine neuronalen Netze nutzen, setzen teilweise Gradientenverfahren ein.

Lewis et al. 1996 und Papka et al. 1996 befassen sich mit Lernen aus Relevanz-Feedback und vergleichen ein Gradientenverfahren mit den im Information Retrieval weit verbreiteten Algorithmen von Rocchio und Widrow-Hoff (cf. Papka 1996). Diese Relevanz-Feedback Algorithmen versuchen, den berechneten Fehler zwischen Anfrage und den als relevant eingestuften Dokumenten durch neue Gewichtung der Anfrage-Terme zu minimieren.

Die Lernverfahren bestimmen die Gewichtungen der Terme in der neuen Anfrage nach Relevanz-Feedback des Benutzers. Während der Ansatz von Rocchio die Gewichte in einem Schritt berechnet, nähert sich das Gradientenverfahren von Lewis et al. 1996 und Papka et al. 1996 wie der Backpropagation-Algorithmus in kleinen Schritten einer besseren Lösung an.

Lewis et al. 1996 und Papka et al. 1996 testen den neuen Ansatz mit Teilen der TREC-Kollektion (siehe Abschnitt 4.7) und erreichen bei den meisten Experimenten bessere Ergebnissen als mit dem verbreiteten Rocchio-Algorithmus für Relevanz-Feedback (cf. Baeza-Yates/Ribeiro-Neto 1999:119). Diese positiven Ergebnisse sprechen dafür, den Backpropagation-Algorithmus für Information Retrieval weiter zu testen.

4.6.2 COSIMIR-Modell

Das COSIMIR-Modell (für eine ausführliche Beschreibung cf. Mandl 2000) realisiert den zentralen Prozeß im Information Retrieval, den Abgleich zwischen Anfrage- und Dokument-Repräsentation in einem Backpropagation-Netzwerk. Input für das COSIMIR-Modell sind gleichzeitig eine Anfrage und ein Dokument. Über eine oder mehrere versteckte Schichten wird die Aktivierung bis zur Ausgangsschicht propagiert, die nur aus einem Neuron besteht und die Relevanz bzw. Ähnlichkeit zwischen den beiden Input-Objekten repräsentiert. Im Training wird die Relevanz von verschiedenen Kombinationen von Dokumenten und Anfragen gelernt.

Das COSIMIR-Modell greift auf eine durch Indexierung gewonnene Dokument-Term-Matrix zurück. Es basiert also wie die anderen in Kapitel 4 besprochenen Modelle auf einem Indexierungsverfahren aus dem Information Retrieval. Das COSIMIR-Modell setzt ein, sobald die Objekt-Repräsentationen vorhanden sind.

Der Kern jedes Retrievalprozesses besteht darin, auf Basis der in der Indexierung gewonnenen Repräsentationen ein Dokument und eine Anfrage zu vergleichen und die Ähnlichkeit zwischen beiden als Maß für die Relevanz zu berechnen. Für diesen Schritt wählt in der Regel ein Entwickler eines Information Retrieval Systems eine Ähnlichkeitsfunktion. Dafür kommen die mathematischen Ähnlichkeitsfunktionen wie Kosinus oder Dice in Frage. Die Entscheidung für eine bestimmte Ähnlichkeitsfunktion wird fast nie aus den Eigenschaften dieser Funktion begründet. Vielmehr fällt diese Entscheidung meist aufgrund heuristischer Überlegungen oder einer empirischen Überprüfung der Ergebnisse. An dieser Stelle im Information Retrieval Prozess setzt das COSIMIR-Modell an, in dem diese heuristische Entscheidung nicht mehr nötig ist.

Wie Abbildung 4-18 zeigt, werden an der Input-Schicht von COSIMIR gleichzeitig eine Dokument- und eine Anfrage-Repräsentation angelegt. Jedes Neuron repräsentiert dabei einen Term. Über eine oder mehrere Zwischen-Schichten breitet sich die Aktivierung aus und das Netz berechnet den Output. Die Output-Schicht besteht aus nur einem Neuron, das die Relevanz des Input-Dokuments für die Input-Anfrage repräsentiert.

Wie jedes Backpropagation-Netzwerk wird auch das COSIMIR-Modell zunächst trainiert und dann für neue Daten eingesetzt. In der Trainingsphase lernt COSIMIR die Relevanz von verschiedenen Kombinationen von Dokumenten und Anfragen. Die von Benutzern vorgegebene Relevanz dient dabei als Maß für die Ähnlichkeit zwischen Anfrage und Dokument. Dafür benötigt das COSIMIR-Modell Trainingsdaten, die aus Urteilen von Benutzern über die Relevanz von Dokumenten zu Anfragen bestehen.

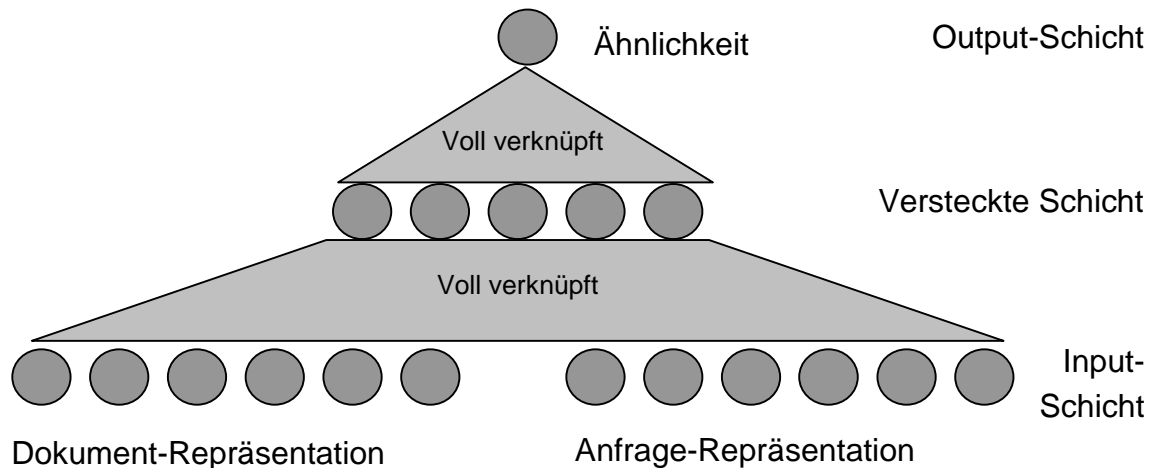


Abbildung 4-18: Das COSIMIR-Modell

In der Trainingsphase destilliert COSIMIR das Wissen aus den Ähnlichkeitsurteilen und speichert es in den Verbindungsstärken. Darin steckt das Wissen, wie das Vorkommen oder Nicht-Vorkommen von Termen die Ähnlichkeitsbewertung beeinflusst. Um in der Einsatzphase die Ähnlichkeit möglichst genau berechnen zu können, muss COSIMIR generalisieren und die Ähnlichkeitsfunktion von den Trainingsbeispielen auf die Beispiele im Einsatz übertragen.

In der Einsatzphase formuliert der Benutzer eine neue Anfrage. Nach Erstellung des Repräsentationsvektors erhält das COSIMIR-Modell die Anfrage und je ein Dokument als Input. Für jedes in der Datenbasis enthaltene Dokument berechnet COSIMIR die Ähnlichkeit zur Anfrage. Die ähnlichsten Dokumente werden dem Benutzer als Ergebnis präsentiert.

Die Neuronen der meisten Modelle bilden in verschiedenen Varianten die am Information Retrieval Prozess beteiligten Objekte wie Anfragen, Dokumente und Terme ab. Der entscheidende Vorteil der Architektur von COSIMIR liegt in der Berücksichtigung und Repräsentation der Relevanz in einem eigenen Neuron.

Die Größe der Merkmals-Räume stellt ein Problem für alle Information Retrieval Verfahren mit neuronalen Netzen dar. Die Komprimierung der Eingangsdaten ist für COSIMIR der wichtigste Anwendungsfall von LSI. Komprimierung allgemein und mit LSI ist eine erfolgreich angewandte Technik im Information Retrieval. Das größte Problem von COSIMIR ist die Größe der entstehenden Matrizen. Die Kombination von COSIMIR und Komprimierung liegt also auf der Hand.

4.6.3 COSIMIR für heterogene Repräsentationen

Das COSIMIR-Modell kann, wie im letzten Abschnitt gezeigt, in einem heterogenen Kontext mit dem Transformations-Netzwerk kombiniert werden. Daneben kann COSIMIR auch mit jedem anderen Transformations-Verfahren eingesetzt werden. Ursachen der Heterogenität und weitere Verfahren zu ihrer Behandlung bietet das Kapitel 2. Wie im letzten Abschnitt gezeigt, wird der Transformation entweder das Dokument oder die Anfrage vor dem Vergleich im COSIMIR-Netzwerk in einen anderen Term-Raum abgebildet. Der Output eines Transformations-Netzwerks dient als Input eines COSIMIR-Modells.

COSIMIR ermöglicht aber noch eine weit flexiblere Reaktion auf heterogene Begriffsschemata. COSIMIR macht keine Annahmen über die mathematische Form der kognitiven Ähnlichkeitsfunktion, die es implementiert. Auch über die Input-Daten macht COSIMIR keine Annahmen und so müssen auch beide Seiten des Inputs nicht homogen sein und aus identischen Term-Räumen stammen. Somit müssen Dokument und Anfrage nicht im gleichen Repräsentationsschema vorliegen, um mit dem COSIMIR-Modell die Ähnlichkeit zu bestimmen. Voraussetzung sind lediglich genügend Trainingsdaten. Die Benutzerurteile müssen Objekte beschreiben, die von Vektoren unterschiedlicher Länge repräsentiert werden. In diesem Fall greift der Input direkt auf die heterogenen Vektoren zu und berechnet daraus in einem Netzwerk die Ähnlichkeit wie Abbildung 4-19 zeigt.

Der explizite Transformationsschritt entfällt in diesem Modell. Statt dessen lernt COSIMIR anhand der Benutzerurteile direkt, wie die Ähnlichkeit aus den Gewichten der Terme aus den unterschiedlichen Indexierungsverfahren abzuleiten ist. Diese Terme können auch weitgehend verschieden sein.

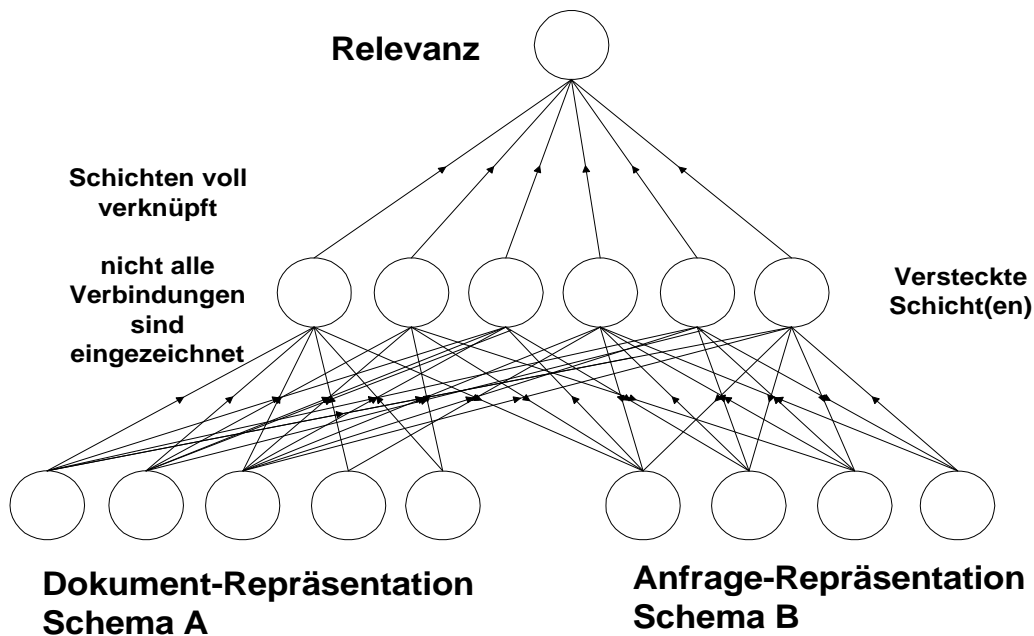


Abbildung 4-19: COSIMIR-Modell für Heterogenitätsbehandlung

4.6.4 Transformations-Netzwerk

Das Transformations-Netzwerk ist ein neuronales Netz, das aufgrund von Trainingsdaten den Übergang zwischen verschiedenen Repräsentationen lernt. Unter den Systemen mit neuronalen Netzen, die im Information Retrieval eingesetzt werden, nutzt es die Fähigkeiten insbesondere zur sub-symbolischen und toleranten Informationsverarbeitung mit am besten aus. Der folgende Abschnitt stellt einen Ansatz vor, in dem ein Transformations-Netzwerk eine Abbildung zwischen homogenen Repräsentationen lernt. Die Erweiterung auf heterogene Repräsentationen führt zu keinen wesentlichen Problemen.

4.6.4.1 Der Ansatz von Crestani/van Rijsbergen 1997

Crestani/van Rijsbergen 1997 stellen ein Backpropagation-Netzwerk vor, das Abbildungen zwischen identischen Repräsentationen ausführt und so eine initiale Benutzeranfrage in eine für das Informationsproblem optimierte Anfrage zu transformieren. Die optimale Anfrage ermitteln sie aus einer Modifikation der originalen Anfrage durch Relevanz-Feedback.

Crestani/van Rijsbergen 1997 erkennen als Schwäche vieler Information-Retrieval-Systeme auf der Basis neuronaler Netze das Fehlen von sub-symbolischen Repräsentationen, wie sie v.a. der Backpropagation-Algorithmus erzeugt. Somit ist das konnektionistische Paradigma nicht ausgeschöpft. Das Modell von Crestani/van Rijsbergen 1997 besteht aus einem typischen Backpropagation-Netz mit einer versteckten Schicht, das sub-symbolische Repräsentationen ermöglicht. Die Input-Schicht repräsentiert die Anfrage-Terme und die Output-Schicht die Dokument-Terme.

Dieses Modell lernt optimale Kombinationen von Anfragen und Dokumenten und wendet das daraus gewonnene Wissen auf nicht bekannte Anfragen an. Ein Problem dieses Ansatzes als Retrieval-System besteht darin, dass die Ausgabe-Schicht immer nur ein Dokument repräsentiert. Beim Training werden aus einer Anfrage mit n relevanten Dokumenten daher n Trainingbeispiele mit der gleichen Anfrage. Das Netz erhält so allerdings widersprüchliche Informationen, da der gleiche Input mit verschiedenen Output-Werten kombiniert und trainiert wird. Im Retrieval-Fall findet das Netz auch immer nur eine Dokument-Repräsentation, was auf keinen Fall befriedigend ist. Dieses optimale Dokument ist in der Kollektion so auch kaum vorhanden.

Crestani/van Rijsbergen 1997 versuchen beide Probleme zu lösen. Als alternatives Lernverfahren berechnen sie aus allen relevanten Dokumenten zu einer Anfrage einen Cluster-Repräsentanten und benutzen diesen zum Lernen. Sie nennen dies *horizontal learning*, während das Lernen mit allen Paaren von relevanten Dokumenten und Anfragen als *total learning* bezeichnet wird. Daneben testen die Autoren auch eine Zwischenstufe, das *vertical learning*, bei dem nur eine Untermenge der relevanten Dokumente als gewünschter Output für eine Anfrage eingesetzt werden. Letzteres Verfahren führte zu den besten Ergebnissen.

Beim Retrieval benutzen Crestani/van Rijsbergen 1997 die vom Netz gefundene Dokument-Repräsentation wiederum als Anfrage und schicken sie an ein herkömmliches IR-System, das die Ähnlichkeit aller Dokumente mit dieser Anfrage berechnet und die ähnlichsten als Ergebnis zurückgibt. Die Output-Schicht repräsentiert während des Trainings also Dokumente, beim Einsatz liefert sie eine Anfrage. Ein weiterer Schwachpunkt besteht in der binären Repräsentation der Dokumente. Die bisherige IR-Forschung hat gezeigt, dass gewichtete Repräsentationen in der

Regel bessere Ergebnisse bringen und diese lassen sich auch problemlos integrieren.

Crestani/van Rijsbergen 1997 testen ihr System mit einem Ausschnitt der Cranfield-Kollektion¹. Sie benutzen die kleinere Cranfield-I Variante mit 200 Dokumente und 42 Anfragen mit Relevanzurteilen. Da die Autoren nur die in den Anfragen und Dokumenten vorkommenden Deskriptoren benutzen, können sie ihr Netz auf 195 Input-Neuronen (Anfrage-Terme) und 1142 Output-Neuronen (Dokument-Terme) beschränken. Die versteckte Schicht besteht aus 100 Units. Dies führt zu einem Netz mit über 100.000 Verbindungen. Demgegenüber stehen bei *horizontal learning*, das die bessere Ergebnisse erbrachte, nur 42 Muster für Training und Test zur Verfügung. Selbst wenn alle Dokumente für alle Anfragen relevant wären, stünden bei auch *total learning* nur 800 Muster zur Verfügung. Von den Mustern nutzen die Autoren maximal 30% für das Training. Nach der Faustregel von Bigus 1996, sollte die Trainingsmenge für jede Verbindung im Netz mindestens zwei Beispiele enthalten. Bei Crestani/van Rijsbergen 1997 ist das Verhältnis Trainingsbeispiels zu Verbindungen eins zu 1000. Das Netzwerk ist also sehr stark unterspezifiziert. Die Zahl der Trainingszyklen beträgt konstant 300 festgelegt. Üblich ist es, das Training zu beenden, wenn der Fehler in der Testmenge nach einer Verringerung wieder zu steigen beginnt.

Die Ergebnisse sind besser als man bei dieser Situation erwarten könnte und sollten mit Vorsicht betrachtet werden. Eine Übertragung der Ergebnisse auf andere Datenmengen ist bei dieser Grundlage nicht möglich. Die Experimente für *vertical learning* führten zu den besten Resultaten. Dabei bilden ein oder zwei Drittel der relevanten Dokumente zu einer Anfrage die Muster. Input ist die Anfrage und Output die verschiedenen Dokumente. *Vertical learning* übertraf den Standard-Ansatz. Dabei muss bedacht werden, dass dem lernenden Verfahren mit den Relevanzurteilen mehr Wissen zur Verfügung steht als dem Standard-IR-Verfahren. Dagegen ergab sich weder eine Verbesserung für *horizontal learning*, wobei ein Cluster-Repräsentant als Output dient, und *total learning*, bei dem alle relevanten Dokumente ins Training einfließen.

Interessant an den Experimenten ist, dass die adaptierten Anfragen sich teilweise sehr deutlich von den ursprünglichen Formulierungen unterschieden. In vielen Fällen waren Terme ganz verschwunden. Weiterhin überschneiden sich die Treffermengen kaum. D.h. beide Anfragen finden relevante Dokumente, aber jedes findet andere. Dies deckt sich mit den Ergebnissen der TREC-Konferenz² (cf. Voorhees/Harman 1999), nach denen verschiedene Information Retrieval Systeme bei vergleichbarer Gesamtqualität unterschiedliche relevante Dokumente in der Ergebnismenge liefern.

Auch wenn die Ergebnisse nicht ohne weitere Tests auf größere Textkollektionen übertragbar sind, so ist sehr vielversprechend, dass die modifizierten Anfrage relevante Dokumente finden, die das Vergleichssystem nicht findet. Kurzfassungen des Ansatzes und der Ergebnisse finden sich in Crestani 1993 und 1995.

Cortez et al. 1995 schlagen ein Transformations-Netzwerk mit fast identischer Funktionalität vor. Ein Backpropagation-Netzwerk bildet die Anfrage-Terme auf die Index-Terme ab, wobei die Schichten allerdings unterschiedlich groß sind. Ein induktiver Lernalgorithmus analysiert vor dem Training die Dokumente und extrahiert die signifikantesten und am stärksten diskriminierenden Terme, die dann den Output bilden. Ein kleines Experiment mit diesem Verfahren nutzt die ADI-Kollektion mit 35 Anfra-

¹ <ftp://ftp.cs.cornell.edu/pub/smart/cran/>

² <http://trec.nist.gov>

gen und 82 Dokumenten, von denen die Titel vorliegen. Am Input wurden die 116 in den Anfragen vorkommenden Terme und in der Output-Schicht die 66 signifikanten Dokument-Terme angelegt. Die gesamte Kollektion wird zum Training des Backpropagation-Netztes und eine Teilmenge zum Test benutzt, wobei aus den Original-Anfragen durch das zufällige Entfernen von Termen unvollständige Fragen gebildet wurden. Die unvollständigen Anfragen aktivieren im Output 83% der Dokument-Terme, die sie mit den vollständigen Anfragen gelernt hatten. Aussagen zu der Qualität des Retrievals erfolgen nicht.

4.6.4.2 Adaption für heterogene Repräsentationen

Die Einschränkung von Crestani/van Rijsbergen 1997 auf zwei identische Term-Räume ist nicht notwendig. Das Transformations-Netzwerk ist flexibel und lässt sich für verschiedenste Anwendungsfälle adaptieren, sobald genügend Trainingsdaten in der Form eines Doppelkorpus vorliegen.

Das Transformations-Netzwerk kann im Bereich der Heterogenitätsbehandlung eingesetzt werden und zwischen heterogenen Term-Räumen abbilden. Es erlaubt z.B. die Abbildung automatisch indexierter Dokumente auf ein intellektuelles Indexierungsschema. Die intellektuelle Indexierung wird dadurch maschinell nachgebildet. Ein derartiges Modell erhält als Input eine automatisch erstellte Repräsentation eines Dokuments. Der Input-Vektor verfügt über eine Dimension für jeden in der Kollektion vorkommenden Term. Während des Trainings und zur Laufzeit wird an der Input-Schicht je ein Dokument-Vektor angelegt. An jedes Neuron wird der Wert der jeweiligen Dimension in dem Dokument abgetragen. Dabei kann es sich z.B. je nach Vorkommen des Begriffs um Null oder Eins handeln, aber auch um reelle Werte aus dem Intervall zwischen Null und Eins, die z.B. die inverse Dokument-Frequenz (cf. Baeza-Yates/Ribeiro-Neto 1999:29) repräsentieren.

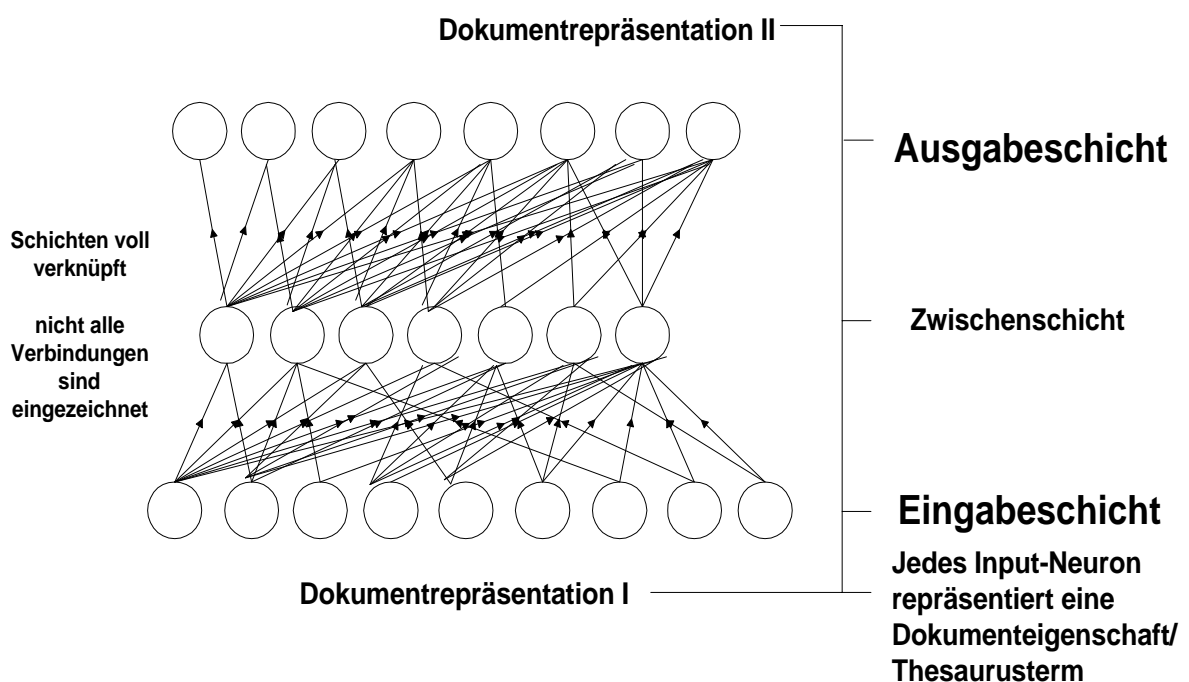


Abbildung 4-20: Das Transformations-Netzwerk

Der Output des Netzes besteht in einer Repräsentation des gleichen Objekt innerhalb eines anderen Indexierungsschemas und damit eines anderen Term- oder Merkmalsraum. Der Output-Vektor besitzt eine Dimension für jeden Term im zweiten Indexierungs-Schema. Dies kann z.B. ein Thesaurus mit kontrollierten Termen sein. In der Trainingsphase lernt das Netz die Transformations-Funktion zwischen den beiden Term-Räumen anhand von Beispielen, für die beide Repräsentationen vorhanden sind. Das Korpus muss also mit zwei Verfahren indexiert worden sein und alle Trainings-Dokumente müssen in beiden Merkmalsräumen vorliegen. Die Erstellung eines solchen Doppelkorpus erfordert einen hohen intellektuellen Aufwand. Nach Möglichkeit sollten bei der Planung eines Transformation-Netzwerks vorhandene Daten erschlossen werden.

Die aufwendige und teure Erstellung der Daten kann teilweise durch heuristische Verfahren umgangen werden. Diese Daten besitzen dann nicht die Qualität eines Doppelkorpus, sie sollten aber je nach Anwendungsfall in Betracht gezogen werden. So ersetzt Mandl 1999 die intellektuelle Indexierung einer Dokumentenmenge mit einem kontrollierten Vokabular durch die Volltextsuche mit den Termen des Vokabulars. So gelingt es, eine Transformation zu dem Vokabular herzustellen, mit dem die Dokumente bereits indexiert sind.

Anhand der zweifach indexierten Dokumente des Doppelkorpus lernt das Transformations-Netzwerk die komplexen Zusammenhänge zwischen dem Vorkommen von Termen in beiden Term-Mengen. Beim erfolgreichen Training erwirbt das Netz die Fähigkeit zu generalisieren und die Funktion auch auf bisher unbekannte Dokumente anzuwenden. Damit kann es nach dem Training in der Einsatzphase ausgehend von einer Repräsentation die andere erstellen.

Ein weiterer wichtiger Anwendungsfall für das Transformations-Netzwerk ist die Abbildung zwischen zwei intellektuell erstellten Repräsentationen mit unterschiedlichen Thesauri. Bibliotheken oder Informationsservicestellen schließen sich heute mehr und mehr zu digitalen Bibliotheken zusammen, um so ihren Benutzern den Zugriff auf größere Datenmengen zu erleichtern. Oft benutzt ein Informationsanbieter einen eigenen Thesaurus, an den sich die Benutzer gewöhnt haben. Durch automatische Transformationen werden die Dokumente anderer Bibliotheken oder Informationsservicestellen mit dem eigenen Thesaurus zugänglich gemacht. Voraussetzung ist ein Doppelkorpus, den die Informationsanbieter bei einer Überlappung der Korpora eventuell in den eigenen Daten vorfinden. Falls dies nicht der Fall ist, kann eine Menge gezielt zusätzlich erschlossen werden.

Weitere Experimente mit dem Transformations-Netzwerk waren erforderlich, da die bisher durchgeführten Experimente von Crestani/van Rijsbergen 1997 nur auf sehr kleinen Datenmengen basierten. Zudem findet die Transformation bei Crestani/van Rijsbergen 1997 und Crestani 1995 zwischen identischen Repräsentationen oder Term-Räumen statt.

4.7 Neuronale Netze bei TREC

Eine Messlatte für experimentelle und kommerzielle Information Retrieval Systeme ist die Text Retrieval Conference (TREC). Da Forscher für ihre IR-Tests verschiedenste Textkollektionen benutzten, waren die Ergebnisse bisher oft nicht vergleichbar. Die TREC-Initiative hat diese Situation verbessert. Sie bietet eine einheitliche Testumgebung und sind als gemeinsame Plattform zum Leistungsvergleich organisiert. Die Beiträge werden jährlich publiziert (cf. Harman 1993/94/95/96, Voor-

hees/Harman 1997/98/99), einen Überblick gibt Womser-Hacker 1997. Die Initiative findet großen Anklang; so beteiligen sich an TREC 6 bereits 58 Retrieval-Systeme (cf. Voorhees/Harman 1998).

TREC bietet den Teilnehmern eine große Dokument-Kollektion, Anfragen und übernimmt für eingereichte Ergebnisse die Auswertung. Als Trainingsdaten stehen die Kollektionen der Vorjahre zur Verfügung, die intellektuelle Relevanzurteile enthalten. Die Ergebnisse der Systeme werden untereinander verglichen. Als Datengrundlage bietet TREC hauptsächlich Zeitungs- und Nachrichtentexte. Viele Experimente außerhalb von TREC haben jedoch gezeigt, dass ein Verfahren bei anderen Daten zu anderen Ergebnissen führen kann. Bereits aus diesem Grund erlaubt auch TREC keine endgültige Entscheidung über das *beste* Retrievalsystem.

Der wichtigste Bestandteil der TREC-Experimente ist das Ad-hoc Retrieval, bei dem die Standard-Situation im Information Retrieval der Ausgangspunkt ist. Ein Benutzer stellt eine Anfrage, die Ergebnisse aus einer großen Menge von Text-Dokumenten liefert. Daneben gibt es Routing-Aufgaben, die einem automatisierten Filter entsprechen. Feststehende Routing-Aufgaben treffen dabei auf ein Strom von Dokumenten, aus dem die relevanten gefiltert werden. Die Veranstalter erstellen die Anfragen (im TREC-Jargon Topics) in drei Detaillierungsebenen. Neben Überschrift und Kurzbeschreibung in einem Satz gibt es eine sog. Langbeschreibung. Die Teilnehmer entscheiden sich für eine Fassung und arbeiten damit. Getrennte Bewertungen erfolgen für Retrieval-Systeme, die mit intellektuell aus den Topics erstellten Anfragen arbeiten. Neben dem Standard-Retrieval-Experimenten und den Routing-Experimenten haben sich weitere Teile, sogenannte Tracks etabliert, darunter ein Cross-Language-Track, ein Interactive Track, bei dem Relevanz-Feedback wichtig ist und ein Track mit sehr großer Datenmenge.

Um die Anzahl der relevanten Dokumente für die Bestimmung des Recall abzuschätzen wird die Pooling-Methode angewandt, bei der Juroren alle Ergebnis-Dokumente der verschiedenen Systeme bewerten, aber nicht die gesamte Kollektion durchsuchen.

Tabelle 4-2: Systeme mit neuronalen Netzen bei TREC

| Konferenz | Ad-hoc Experimente | Routing/Filtering Experimente | Weitere Experimente |
|-----------|--------------------|-------------------------------|--|
| TREC 1 | PIRCS, MatchPlus | PIRCS | |
| TREC 2 | PIRCS, MatchPlus | PIRCS, Boyd et al., MatchPlus | |
| TREC 3 | PIRCS | PIRCS | |
| TREC 4 | PIRCS | PIRCS | |
| TREC 5 | PIRCS, Mercure | PIRCS, Mercure | PIRCS (Chinese) |
| TREC 6 | PIRCS, Mercure | PIRCS, Mercure | Mercure (Cross-lingual), PIRCS (Chinese, High Precision) |
| TREC 7 | PIRCS, Mercure | PIRCS, Mercure | PIRCS (High Precision) |

An TREC nahmen von Anfang an einige Systeme teil, die auf neuronalen Netzen beruhen (z.B. in TREC I: Gallant et al. 1993, Kwok et al. 1993). Insgesamt nahmen vier Systeme teil, davon gehören drei zum Spreading-Activation-Ansatz. PIRCS (sie-

he Abschnitt 4.3.2.1, cf. Kwok/Grunfeld 1994/6, Kwok et al. 1999) und Mercure (siehe Abschnitt 4.3.2.4, cf. Boughamen/Soule-Dupuy 1997/8, Boughamen et al. 1999) haben beide bei ihrer ersten Teilnahme die kleinere Textmenge (Kategorie B) bearbeitet und haben dann innerhalb eines Jahres den Umstieg zur vollen Textmenge (Kategorie A) geschafft. Dies zeigt, dass diese Modelle ausgereift sind und auch größere Textmengen bewältigen. Tabelle 4-2 gibt einen Überblick über die eingesetzten neuronalen Netzwerk Systeme.

Ein Vorteil der Spreading-Activation-Netze PIRCS und Mercure besteht in der gleichberechtigten Behandlung von Dokumenten und Anfragen. Die TREC Routing-Experimente ergeben sich daher ganz natürlich. Beim Routing liegen Interessensprofile vor, zu denen aus einem Strom von Dokumenten die relevanten Texte gefiltert werden. Das Modell von Kwok/Grunfeld 1994 und 1996 benutzt die neuen Dokumente als Input. Sie aktivieren die für sie relevanten Anfragen. PIRCS (Kwok/Grunfeld 1994/6) lernt darüber hinaus anhand von Relevanz-Feedback durch die Veränderung von Gewichten. Mercure erweitert die Anfrage durch Spreading-Activation (Boughamen/Soule-Dupuy 1997/8).

Boyd et al. 1994 (siehe auch Syu/Lang 1994) versuchen, semantisches Wissen aus einem Thesaurus in ihr Spreading-Activation-Netz zu integrieren. Der Thesaurus enthält thematische Rollen für Wörter wie z.B. *Beneficiary* oder *Cause*. Für die Category B Routing konnten die semantischen Experimente allerdings nicht durchgeführt werden. Das als Vergleichsmaßstab vorgesehene Standard-Spreading-Activation-Netz wurde zur TREC-Konferenz eingereicht. Die Ergebnisse sind nach Angaben der Autoren schlecht.

Der in MatchPlus eingesetzte Ansatz von Gallant et al. 1993 und 1994 ist nicht vollständig offengelegt, da die Firma HNC ihn kommerziell einsetzt. Daran wirken zwei neuronale Netze, wobei es sich höchstwahrscheinlich um Spreading-Activation-Ansatz und/oder Self-Organizing Maps handelt, da offensichtlich keine Daten für überwachtetes Lernen benötigt werden. Das Modell verwendet sogenannte Kontext-Vektoren. Das Vektorraum-Modell im Information Retrieval verwendet Vektoren, in denen jede Komponente einen Deskriptor repräsentiert. Dadurch entstehen sehr lange Vektoren, von denen viele den Wert Null besitzen, die also spärlich besetzt sind. Je länger die Vektoren, desto höher ist die Rechenzeit für das Retrieval. Kontext-Vektoren sind ein semantischer Ansatz zur Dimensionalitätsreduktion. Ziel von Reduktionen ist nicht nur eine schnellere Verarbeitung der Muster, sondern auch bessere Verteiltheit der Muster und damit bessere Ähnlichkeitswerte. Die Komponenten in Kontext-Vektoren repräsentieren semantische Eigenschaften wie *human*, *day* oder *heavy*. Diese Eigenschaften werden entweder intellektuell oder aus hochfrequenten Begriffen gewonnen. Im Vektor für einen Deskriptor nehmen die Komponenten Werte an, die den Deskriptor semantisch beschreiben. Der Deskriptor *astronomer* hätte z.B. in der Komponente *human* den höchsten Wert und in der Komponente *day* einen niedrigen Wert. Dies kann intellektuell geschehen, Gallant et al. 1993 und 1994 berichten von Algorithmen, die Kontext-Vektoren halb- und vollautomatisch erstellen. Kontext-Vektoren repräsentieren also zunächst Deskriptoren. Dokumente und Anfragen ergeben sich als die gewichtete Summe der enthaltenen Einzelterme.

Einschränkungen von TREC haben sich wahrscheinlich auf die Gestaltung der teilnehmenden Systeme ausgewirkt. Sowohl PIRCS als auch Mercure haben an den Basistests teilgenommen. Relevanz-Feedback spielt lediglich in einem anderen Teil von TREC, dem Interactive Track, eine Rolle. Möglicherweise verzichten die beiden

Systeme in ihren TREC Experimenten deshalb auf das den Spreading-Activation-Modelle inhärente Relevanz-Feedback.

Die Bewertung in TREC erfolgt mit den Maßen Recall und Precision. Die Precision wird für Recall-Niveaus zwischen 0,1 bis 0,9 berechnet und aus diesen Werten die durchschnittliche Precision bestimmt. Diese Zahl führt zu den fünf besten Systemen, die jedes in einem Überblicksartikel erwähnt werden. Eine Grafik zeigt die Recall-Precision-Kurve dieser Systeme in Recall-Schritten von 0,1 zwischen 0,1 bis 0,9.

In TREC 5 zählte PIRCS sowohl bei Ad-hoc mit der Kurzbeschreibung und automatischer Anfragen-Generierung, bei Ad-hoc mit manueller Anfrage-Generierung als auch bei den Routing-Experimenten zu den besten acht Systemen (cf. Voorhees/Harman 1997a). Beim Chinese-Track erzielte PIRCS das beste Ergebnis (cf. Wilkinson 1998) unter zehn Teilnehmern (cf. Smeaton/Wilkinson 1997).

In TREC 6 liegen PIRCS und Mercure in der Spitzengruppe der ad-hoc Systeme. Unter den 57 eingereichten Ergebnissen mit automatischer Anfragen-Generierung nutzten sechzehn die vollständige Topic-Beschreibung, 29 die Kurzbeschreibung und zwölf nur den Titel. Bei den Titeln und Langbeschreibungen gehörten PIRCS und Mercure jeweils zu den acht besten Systemen (cf. Voorhees/Harman 1998a). Dabei ist PIRCS besser als Mercure, allerdings liegen die acht ersten Systeme sehr eng zusammen und der Unterschied ist somit gering. Bei den Routing-Experimenten konstruierten 28 Systeme die Anfrage automatisch. PIRCS und Mercure zählen auch hier wieder zu den besten acht Systemen. Beim Chinese-Track erreichte PIRCS wie im Jahr vorher ein sehr gutes Ergebnis (cf. Wilkinson 1998). Im High Precision Track hatte PIRCS mit die schlechtesten Ergebnisse, erstellte aber als einziges System die Anfrage automatisch (cf. Buckley 1998).

In TREC 7 findet sich PIRCS nach wie unter den acht besten Systemen der ad-hoc-Systeme mit automatischer Anfragengenerierung, Mercure jedoch nicht mehr (cf. Voorhees/Harman 1999a).

Bei den Information Retrieval Systemen aus der Familie der neuronalen Netze bei TREC handelt es sich somit um Spreading-Activation-Netzwerke. Insbesondere PIRCS zeigt die Leistungsfähigkeit dieser Systeme. Es nahm an allen bisherigen TREC-Konferenzen teil und erreichte häufig einen der Spitzenplätze. Mercure gelang bei drei Teilnahmen einmal ein Vorstoss in die Gruppe der acht besten Systeme einer Kategorie. Boyd et al. 1994 versuchen, ein semantisch angereichertes Spreading-Activation-Netzwerk zu implementieren. Die Ergebnisse fielen allerdings schlecht aus. Andere Modelle als Spreading-Activation-Netzwerke spielen praktisch keine Rolle. Der Algorithmus von MatchPlus ist nicht offengelegt und nahm nur an den beiden ersten Konferenzen teil.

4.8 Fazit: Neuronale Netze im Information Retrieval

Von allen analysierten Klassen von Modellen werden drei für große Mengen von realen Daten eingesetzt:

- Assoziativspeicher wie z.B. Hopfield-Netzwerke (siehe Abschnitt 4.2) sind eine sehr interessante Klasse von neuronalen Netzen, deren Grundfunktionalität im Retrieval besteht. Auto-Assoziativspeicher liefern gespeicherte Muster zu unvollständigen Input-Mustern.
- Spreading-Activation-Netzwerke (siehe Abschnitt 4.3) wurden bisher am häufigsten eingesetzt und mit großen Dokumentmengen evaluiert. Sie stellen nur

bedingt eine eigene konzeptuelle Klasse von IR-Verfahren dar, sondern lehnen sich eng an die bekannten Modelle an.

- Kohonen-SOM und ART-Netze (siehe Abschnitte 4.4 und 4.5) sind die Grundlage von Systemen mit großen Datenmengen. Allerdings handelt es sich um Clustering-Verfahren, die für wichtige Schritte im Information Retrieval nicht direkt anwendbar sind.

Alle diese Systeme haben folgende Nachteile:

- Fehlen von sub-symbolischer Verarbeitung:
Diese Systeme nutzen die Mächtigkeit neuronaler Netze nicht aus. Der sehr häufig benutzte und mächtige Backpropagation-Algorithmus wird nicht eingesetzt.
- Lernfähigkeit bleibt weitgehend eingeschränkt:
Wenn Lernen implementiert ist, beschränkt es sich meist auf Relevanz-Feedback. Die Auswirkungen für das gesamte Modell werden nicht quantifiziert und sind wohl minimal.
- Das am intensivsten erforschte Modell der Spreading-Activation-Netzwerke stellt keine konzeptuell neuartige Klasse von IR-Modellen dar.

Information Retrieval Modelle auf der Basis des Backpropagation-Algorithmus erscheinen sehr erfolgversprechend. Von den bestehenden Modellen ist besonders das Transformations-Netzwerk interessant. Es eignet sich für die Heterogenitätsbehandlung (siehe Abschnitt 4.6.4) und zur Vorverarbeitung der Anfrage, erfordert aber noch ein vollständiges Retrievalsystem. Das Transformations-Netzwerk wird im folgenden Kapitel mit einer realen Datenbasis evaluiert.

5 Evaluierung des Transformations-Netzwerks

Dieses Kapitel beschreibt Experimente mit dem Transformations-Netzwerk auf der Basis von realen und großen Datenmengen.

Das Transformations-Netzwerk ist relativ einfach zu handhaben und ist in der praktischen Anwendung mit statistischen Verfahren vergleichbar. Kommerzielle Software steht für die Plattform Windows zur Verfügung. Aus technischer Sicht steht dem Einsatz des Transformations-Netzwerks nichts im Wege.

Bei der Erstellung der Daten ist zu beachten, dass jeder Term auch mehrmals in der Trainingsmenge vorkommt. Wichtig ist für den praktischen Einsatz in einem Informationssystem, dass die Transformation für ein Doppelkorpus in beide Richtungen trainiert werden muss.

5.1 Transformations-Netzwerk: Thesaurus zu Klassifikation

Die Evaluierung der Heterogenitätsbehandlung mit dem Transformations-Netzwerk erfordert Daten, die mit zwei Indexierungsschemata erschlossen sind. Dafür wurden Daten aus dem Bereich der Sozialwissenschaften benutzt.

5.1.1 Datenbanken des Informationszentrum Sozialwissenschaften

Das Informationszentrum Sozialwissenschaften in Bonn (IZ) betreibt die Datenbanken SOLIS und FORIS, die über verschiedene online-Hosts und als CD-ROM zugänglich sind (cf. Zimmer 1998, 1998a). SOLIS ist eine Literaturdatenbank für die Sozialwissenschaft, die Nachweise zu selbständigen und unselbständigen Publikationen enthält. Am IZ verschlagworten Indexierer die Dokumente intellektuell und erstellen Abstracts, soweit diese nicht vorhanden sind. Daneben pflegt das IZ die Projektdatenbank FORIS, die mit den Ergebnissen einer jährlichen Umfrage bei einschlägigen Forschungseinrichtungen in den Sozialwissenschaften und benachbarten Disziplinen gefüllt wird. Auch die Dokumente in FORIS werden intellektuell nach dem IZ-Thesaurus (cf. Informationszentrum Sozialwissenschaften 1997) indexiert.

Der Thesaurus umfasst ca. 10.500 Terme, wovon 6900 als Deskriptoren vergeben werden können. Die übrigen Terme werden im Sinne der Einheitlichkeit des Indexierungsvokabulars nicht benutzt, bei ihnen gibt der Thesaurus den Begriff an, den der Indexierer benutzen soll. Darüber hinaus enthält der Thesaurus für viele Einträge Über-, Unter- und verwandte Begriffe. Daneben werden alle Dokumente zusätzlich in einer Klassifikation von Fachgebieten eingeordnet, die 159 Einträge enthält. Die Disziplinen sind dabei aus dem Blickwinkel der Sozialwissenschaften geordnet.

Im Rahmen der crosslingualen TREC-Experimente werden IZ-Daten bei TREC und der neuen europäischen Initiative Cross-Language Evaluation Forum (CLEF)¹ als Experimentierdaten angeboten. Dazu wurde ein Ausschnitt aus SOLIS und FORIS gebildet, Anfragen formuliert und relevante Dokumente gesucht. Bei den TREC-Experimenten bilden die IZ-Daten gemeinsam mit Texten der Neuen Züricher Zeitung aus der gleichen Zeit einen Pool, aus dem mit einer Anfrage sowohl sozialwissenschaftliche Fachtexte als auch Zeitungstexte recherchiert werden.

Für das hier besprochene Experiment wurden 12.965 Dokumente aus SOLIS und FORIS benutzt, die den Grundstock der CLEF-Kollektion bilden. Von dieser Grundmenge bilden 12.000 Dokumente die Trainingsmenge und der Rest die Testmenge. Alle Ergebnisse in diesem Abschnitt gehen von dieser Datenmenge aus und beziehen sich auf die Testmenge.

Laut Aussagen der Indexierer am IZ sind Thesaurus und Klassifikation unabhängig voneinander und formalisierbare Beziehungen zwischen ihnen existieren nicht. Das Transformations-Netzwerk versucht in diesem Experiment, eine Abbildung zwischen diesen zwei unterschiedlichen intellektuellen Indexierungs-Schemata zu leisten.

Ein automatisiertes System, das diese Transformation vornimmt, könnte innerhalb des IZ und für bestimmte Nutzergruppen gewinnbringend eingesetzt werden. Auch beim Retrieval kann die Transformation für Benutzer sinnvoll sein, die immer mit der Klassifikation arbeiten und mit ihr vertraut sind. Würde nun aus Kostengründen die Indexierung nach der Klassifikation eingestellt, könnte ein automatisches System diesen Zugang durch die Transformation aufrecht erhalten. Ein weiterer Anwendungsfall ergibt sich aus dem Schalenmodell, in dem Objekte von einer Form der Inhaltserschließung in eine andere übertragen werden (siehe Abschnitt 2.1). Sollten externe Informationsanbieter ihre Dokumente nur mit dem IZ-Thesaurus erschließen, so werden diese Dokumente auch für Benutzergruppen zugänglich, die mit der Klassifikation vertraut sind.

Unabhängig vom Retrieval könnte eine solches System die Indexierer unterstützen und nach Vergabe der Terme aus dem Thesaurus Terme aus der Klassifikation vorschlagen. Einen solchen Vorschlagmodus realisieren auch andere Transformations-Systeme (siehe Abschnitt 2.4).

5.1.2 Transformations-Netzwerk und LSI

Für jedes Dokument sind nach dieser Kumulierung durchschnittlich dreizehn Terme und 2,3 Klassifikations-Einträgen vergeben. Besonders für die Thesaurus-Terme schwankt die Zahl jedoch stark, nämlich zwischen zwei und 39.

¹ <http://www.iei.pi.cnr.it/DELOS/CLEF>

Für die ca. 13.000 Dokumente liegen sowohl die Thesaurus-Einträge und die Klassifikation vor. Darin sind 5555 Thesaurus-Begriffe und 142 der 159 Klassifikations-Einträge belegt. Für den Test werden auf der Input Seite nur 3800 Terme ausgewählt, die mindestens in vier Dokumenten vorkommen Ansonsten kann kaum sichergestellt werden, dass jeweils alle Terme, die in der Testmenge enthalten sind, auch trainiert werden. Als gewünschter Ziel-Output gibt das System nicht alle 142 Klassen auf der untersten Hierarchieebene der Klassifikation vor, sondern 70 intellektuell kumulierte Klassen, was zu einer günstigeren Anzahl von Beispielen pro Klasse führt. Auch dadurch wird versucht, mit der Testmenge nichts zu prüfen, was vorher nicht mit Trainingsmenge gelernt wurde.

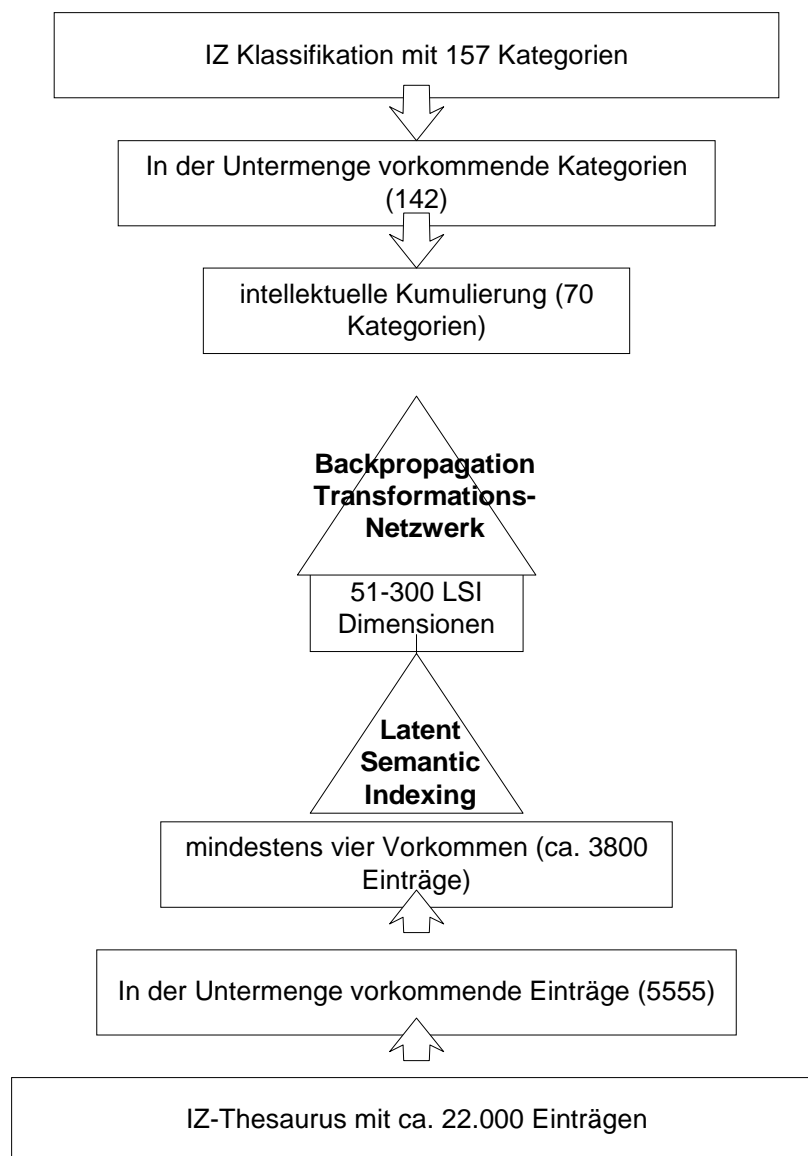


Abbildung 5-1: Schema der Transformation vom IZ-Thesaurus zur IZ-Klassifikation

Die Berücksichtigung von ca. 3800 Thesaurus-Termen, die mindestens viermal vorkommen, führt zu sehr langen Vektoren und damit zu sehr großen Netzen. Diese sind zum einen mit gängiger Hardware nicht trainierbar und die Anzahl der Trainingsbeispiele sind im Vergleich zu den Verbindungen zwischen den Neuronen zu gering. Die Faustregel von Bigus 1996 fordert zwei Muster pro Verbindung. Es stehen jedoch nur 12.000 Muster zur Verfügung und bei nur zehn versteckten Neuronen ergibt sich bereits ein Netz von 38.700 Verbindungen.

Auch hier wurde diese Problematik durch eine Komprimierung mit Latent Semantic Indexing (LSI) gelöst. Die 3800 Dimensionen wurden nach Tests mit Werten zwischen 50 und 300 auf 107 LSI-Faktoren reduziert.

Durch diese Reduktion der Dimensionen mit LSI bleibt das Netz klein. Die Experimente erfolgten mit der Software DataEngine (siehe Abschnitt 3.6). Die Netze konvergieren dabei sehr schnell. Nach etwa 30 - 40 Epochen änderten sich der Trainings- und Testfehler kaum mehr. Die Größe der versteckten Schicht wurde durch den Test mehrerer Konfigurationen optimiert. Das beste Netz besitzt zwanzig versteckte Neuronen.

Als Vergleichsmaßstab (Baseline) dient ein statistisches Verfahren, das aufgrund gemeinsamen Vorkommens in den Termen Assoziationen zwischen Thesaurus und Klassifikation berechnet. Dazu wurden die Kookkurrenzen berechnet und in einer Assoziationsmatrix festgehalten. Diese Matrix dient als Grundlage der Transformation wie in Abbildung 5-2 angedeutet.

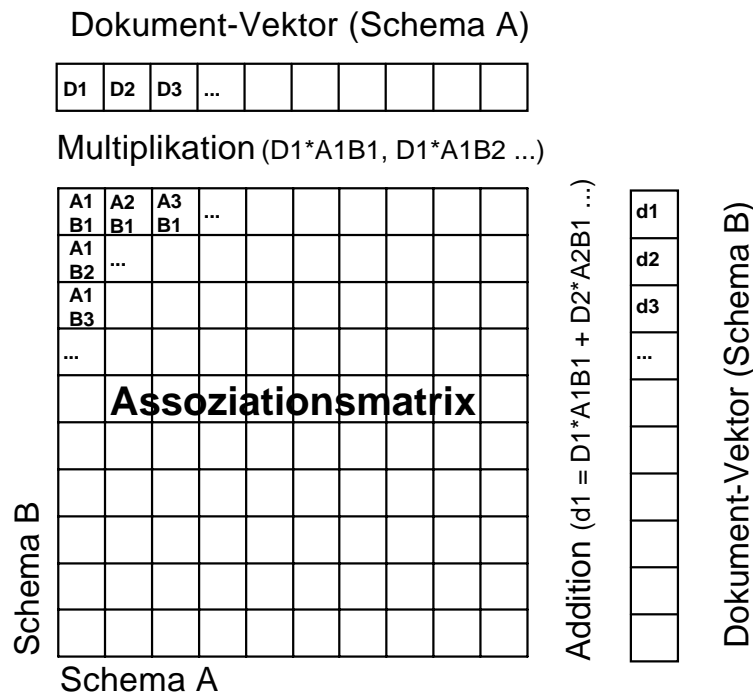


Abbildung 5-2: Statistische Transformation als Baseline

5.1.3 Ergebnisse

Der Testfehler des Netzes entspricht nicht dem endgültigen Qualitätsmaß für die Transformation. Das Netz misst für jedes Gewicht eines Terms die Übereinstimmung des Trainingswertes mit dem Ergebnis des Netzes. Die möglichst exakte Übereinstimmung der Gewichte gibt aber nicht den Ausschlag für eine

erfolgreiche Transformation. So wie beim normalen Retrieval nicht der Absolutwert der Retrieval Status Value entscheidend ist, sondern die sich ergebende Reihenfolge der Dokumente, kommt es auch bei der Bewertung der Transformation in der Regel auf die Reihenfolge der Gewichte an. Das Training des neuronalen Netzes verringert primär den Fehler in der Testmenge. Im vorliegenden Fall ergeben sich durchschnittliche, quadrierte Fehler von ca. 0,1. Abbildung 5-3 zeigt beispielhaft die Parameter und Ergebnisse eines Netzes während des Trainings.

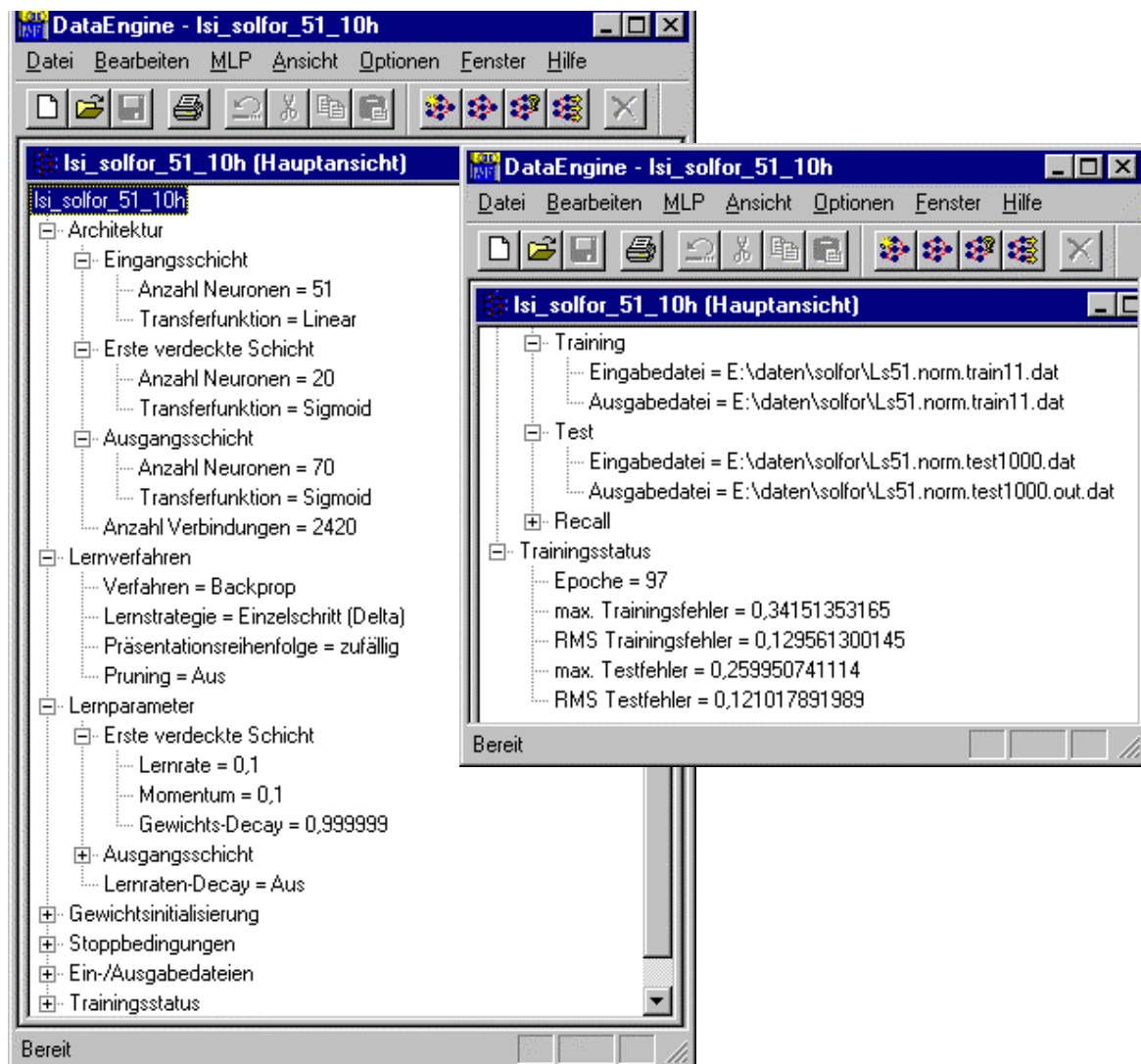
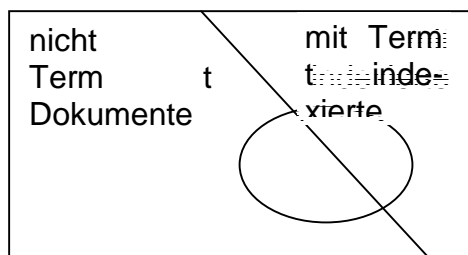


Abbildung 5-3: Netzwerk, Experimentparameter und Ergebnisse in DataEngine

Die üblichen Maße für die Qualität von Transformationen sind Recall und Precision bezogen auf die Terme des Zielvokabulars (cf. z.B. Yang 1995, Apté et al. 1994). Dabei ersetzen die Terme des Zielvokabulars die Anfragen, auf die sich Recall und Precision beim Retrieval beziehen. Ausgangspunkt ist beim Retrieval eine Anfrage und bei den Transformationen ein Term. Ergebnis ist in beiden Fällen eine Menge von Dokumenten. Eine Transformation bestimmt die Menge von Dokumenten, der sie den Term aus dem Zielvokabular zuordnet. Diese Menge wird dann mit den tatsächlichen Zuordnungen verglichen oder, falls diese nicht vorhanden ist, nachträglich von

Experten beurteilt (siehe Abbildung 5-4). Im zweiten Fall entsteht das analoge Problem zur Recall beim Retrieval, dass die Zielmenge nicht genau bekannt ist und nur durch sehr hohen intellektuellen Aufwand bestimmt werden kann. Bei der Verwendung vager Verfahren zur Transformation entsteht in der Regel keine Menge von zugeordneten Dokumenten, sondern jedes Dokument erhält ein Gewicht für jeden Term. Damit muss auch bei Term-Recall und Term-Precision ein Recall-Precision-Graph erstellt werden. Abbildung 5-5 zeigt das Ergebnis der Transformation für das Transformations-Netzwerk und für das Vergleichsexperiment mit einem statistischen Verfahren.



Der Term t aus dem Indexierungsschema B ist einigen der Dokumente zugeordnet (rechts der schrägen Linie). Das Ergebnis der Transformation ist eine Menge von Dokumenten (Oval), für die Term t vorgeschlagen wird.

| | Von Transformation vorgeschlagen | Von Transformation nicht vorgeschlagen |
|-----------------|----------------------------------|--|
| Indexiert | R_E | R_N |
| Nicht indexiert | N_E | N_N |

$$\text{Term-Recall} = \frac{R_E}{R_E + R_N}$$

$$\text{Term-Precision} = \frac{R_E}{R_E + N_E}$$

Abbildung 5-4: Term-Recall und Term-Precision

Diese Maße sind problematisch. Im Falle der Transformationen gilt dies um so mehr. Zwar ergibt sich formal eine völlig analoge Formel (siehe Abbildung 5-4), sie misst aber nicht die Retrieval-Qualität, wie die Begriffe Recall und Precision andeuten. Während beim Retrieval der Erfolg vom Urteil des Benutzers abhängt und daraus berechnet wird, spielt dieser Gesichtspunkt bei den Transformationen zunächst keine Rolle. Hier bestimmen Recall und Precision die Qualität einer Zuordnung von Termen, also einer automatischen Indexierung auf Basis einer bereits vorliegenden Indexierung. Erst der Einsatz dieser Indexierung im Retrieval zeigt ihren Wert für den Benutzer. Unabhängig davon kann eine Transformation als Unterstützungssystem für die Indexierung verwendet werden und z.B. einem Indexierer Vorschläge machen (vgl. Abschnitt 2.4). In diesem Fall ist die beschriebene Form der Evaluierung adäquater. Natürlich ist eine gute Indexierung die Basis für erfolgreiche Retrieval und somit hängen die beiden Formen des Recall zusammen.

Um die angeführten Bedenken zu verdeutlichen und einer Verwechslungsgefahr zu begegnen, werden im folgenden im Gegensatz zur Literatur die Qualitätsmaße für Transformationen als Term-Recall und Term-Precision bezeichnet. Diese Benennung verdeutlicht, dass die Grundlage ihrer Berechnung in der Vergabe von Termen und nicht in einer Anfrage besteht. Weiterhin erfolgt eine zusätzliche Bewertung der Ergebnisse anhand einer anschaulicheren Methode.

Abbildung 5-5 stellt die Ergebnisse des Transformations-Netzwerks und der Baseline gegenüber. Die Kurven verlaufen fast gleichförmig, so dass die Qualität der beiden Verfahren für diesen Anwendungsfall praktisch identisch ist. Insgesamt liegen die

Werte eher niedrig, die durchschnittliche Precision beträgt 0,19. In anderen Experimenten wird von höheren Werten berichtet. Yang 1995 z.B. erzielte durchschnittliche Precision-Werte zwischen 0,32 und 0,88 bei der Anwendung statistischer Verfahren für verschiedene Kollektionen. Dieses relativ niedrige Ergebnis lässt vermuten, dass in diesem Anwendungsfall die Transformationen sehr schwierig sind.

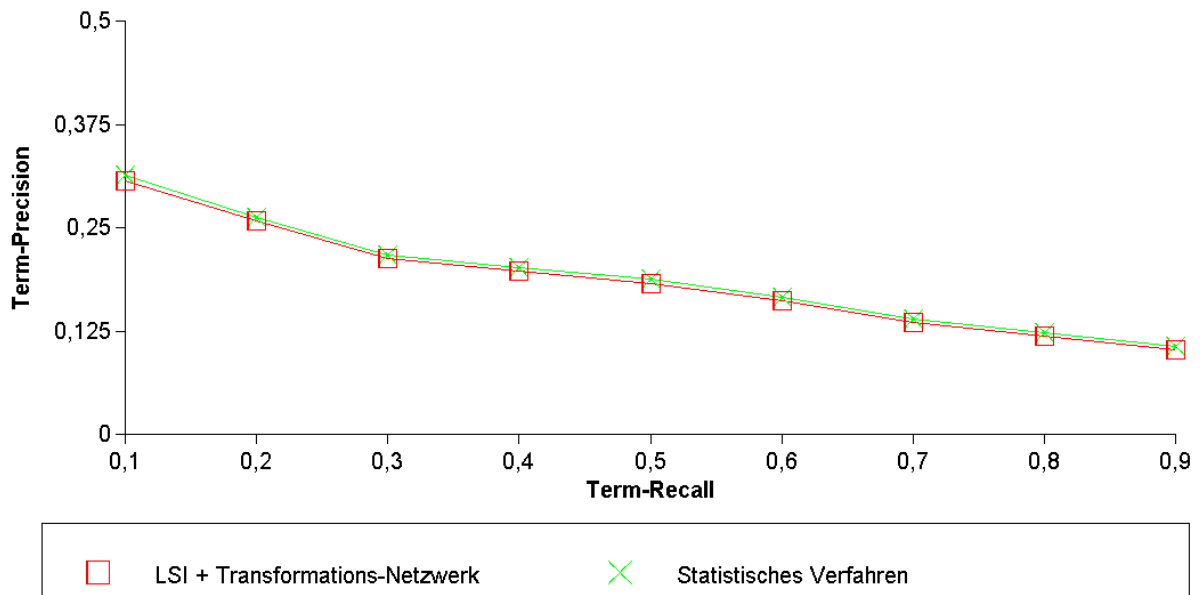


Abbildung 5-5: Ergebnis als Recall-Precision-Grafik

Ein erster Vergleich legt zunächst die Interpretation nahe, die Ergebnisse seien identisch. Eine nähere Betrachtung zeigt aber, dass nur die Qualität sehr ähnlich ist, die Ergebnisse dagegen weitgehend unterschiedlich. Dazu werden die Rangfolgen verglichen, die sich ergeben, wenn beide Verfahren ausgehend von einem Term den Dokumenten ein Gewicht zu diesem Term zuordnen. Um diese Rangfolgen vergleichen zu können, eignet sich der häufig in der Literatur genannte Spearmansche Rangfolgenkoeffizient:

$$\text{Spearman: } r = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)}$$

Hartung 1984:191, Clauß/Ebner 1979:126

Berechnet man den Spearman-Koeffizienten für alle 70 Zielklassen für die 1000 Dokumente in der Testmenge und den Durchschnitt daraus, so ergibt sich ein Wert von -0,05. Dies weist darauf hin, dass praktisch keine Korrelation zwischen den Rangfolgen besteht und die zwei Verfahren die Dokumente völlig unterschiedlich sortieren.

Eine weitere Analyse veranschaulicht die Unterschiedlichkeit der Ergebnisse noch besser. Sie untersucht die Schnittmengen der Ergebnisse der beiden Verfahren. Dazu wurden für jeden Term die ersten 20 Dokumente aus der Rangfolge gewählt und mit den 20 ersten Dokumenten des anderen Verfahrens verglichen. Im Durchschnitt

ergeben sich für die 70 Terme in der Klassifikation nur 0,27 gemeinsame Dokumente von 20. Betrachtet man nur die relevanten Dokumente unter den ersten 20, also die mit einer korrekten Zuordnung, sinkt dieser Wert auf 0,14. Dies zeigt sehr deutlich, dass jedes Verfahren andere Treffer bringt. Die gleiche Analyse für die ersten 100 Dokumente führt zu einer durchschnittlichen Größe der Schnittmenge von 4,92 Dokumenten und zu 0,83 gemeinsamen relevanten Dokumenten. Tabelle 5-1 fasst diese Ergebnisse zusammen.

Tabelle 5-1: Schnittmengen aus den besten Dokumenten

| | Größe der Schnittmenge | in % | Größe der Schnittmenge der relevanten Dokumente | in % |
|----------------------|-------------------------------|-------------|--|-------------|
| Ersten 20 Dokumente | 0,27 | 5,43 | 0,14 | 0,71 |
| Ersten 100 Dokumente | 4,92 | 4,92 | 0,83 | 0,83 |

Diese Analyse erinnert an Ergebnisse der TREC-Konferenz. Dort hat sich ebenfalls gezeigt, dass die besten IR-Systeme sehr ähnliche Qualität erreichen, dass sich ihre Ergebnisse jedoch stark unterscheiden. Dies führte zur Entwicklung von Fusionsverfahren, die dies ausnutzen. Derartige Ansätze der Mehrfachindexierung versuchen, mehrere Verfahren zu kombinieren, um so insgesamt mehr relevante Dokumente finden. Das Gesamtergebnis besteht aus einer Kombination mehrerer Ergebnismengen unterschiedlicher Verfahren (cf. Womser-Hacker 1997).

Dies bedeutet auch, dass die Optimierung nicht nur die Suche nach dem besten Information Retrieval Verfahren bedeutet. Vielmehr muss ein neuartiges Verfahren nicht notwendigerweise alle anderen übertreffen. Ein neues IR-Verfahren dessen Qualität mit bestehenden Verfahren vergleichbar ist, das aber eine weitgehend andere Ergebnismenge und damit auch andere Treffer bringt, kann bei der Fusion einen positiven Beitrag leisten. Diese Überlegungen gelten auch für die Heterogenitätsbehandlung.

Die Werte von Term-Recall und Term-Precision sind zwar die üblichen Bewertungsmaßstäbe für Transformationen, sie sind jedoch nicht sehr anschaulich. Die folgende zusätzliche Bewertung des Ergebnisses überträgt den Erfolg des Verfahrens auf einzelne Dokumente und zeigt so, inwieweit sich die Transformation als Vorschlagsmodus eignet. In der Testmenge von 1000 Dokumenten sind 69% aller Zuordnungen und 97% aller Nicht-Zuordnungen richtig. Dies bedeutet, dass pro Dokument 1,4 von 2,0 Einträge und 32,4 von 35,9 Nicht-Zuordnungen richtig erkannt werden, bzw. dass pro Dokument 3,8 Einträge falsch hinzukommen. Dies scheint als erste Annäherung befriedigend. Bei einer Anwendung als Vorschlagsmodus müssen vom menschlichen Indexierer v.a. nicht passende Einträge gelöscht werden, was einfacher ist als neue passende zu finden. Bei dieser Interpretation der Ergebnisse wird deutlich, dass eine stark unterschiedliche Anzahl von Trainingsbeispielen für die einzelnen Klassen die Aufgabe für das Netz erschwert. Die Anzahl der Beispiele für eine Zielklasse schwankt in der Trainingsmenge zwischen vier und 3300, der Durchschnitt in der Gesamtmenge beträgt 206. Erwartungsgemäß ergibt sich eine starke Korrelation zwischen Anzahl von Trainingsbeispielen und Treffern (siehe Abbildung 5-6).

digitalen Bibliothek, die dem Benutzer flexiblen Zugriff auf alle diese Daten ermöglicht, ohne dass die heterogenen Erschließungsverfahren zu semantischen Problemen führen.

Das in den beiden folgenden Abschnitten beschriebenen Experimente benutzen doppelt indexierte Text-Dokumente vom IZ und dem SSG der USB Köln.

5.2.1 USB-Thesaurus zu IZ-Klassifikation

In einem ersten Schritt konnten mit Hilfe des konnten 1979 Dokumente identifiziert werden, die sowohl mit den Termen der USB und mit der Klassifikation des IZ indexiert sind. Davon wurden 1779 für das Training und 200 für den Test benutzt. Für die Dokumente sind im Durchschnitt 2,2 Terme und maximal sieben Terme aus dem USB-Thesaurus vergeben. Die Aufgabe ist damit sehr schwierig, da zum einen wenig doppelt indexierte Dokumente zur Verfügung stehen und weiterhin wenig USB-Terme vergeben sind, so dass automatische Verfahren wenig Evidenz für die Transformation erhalten.

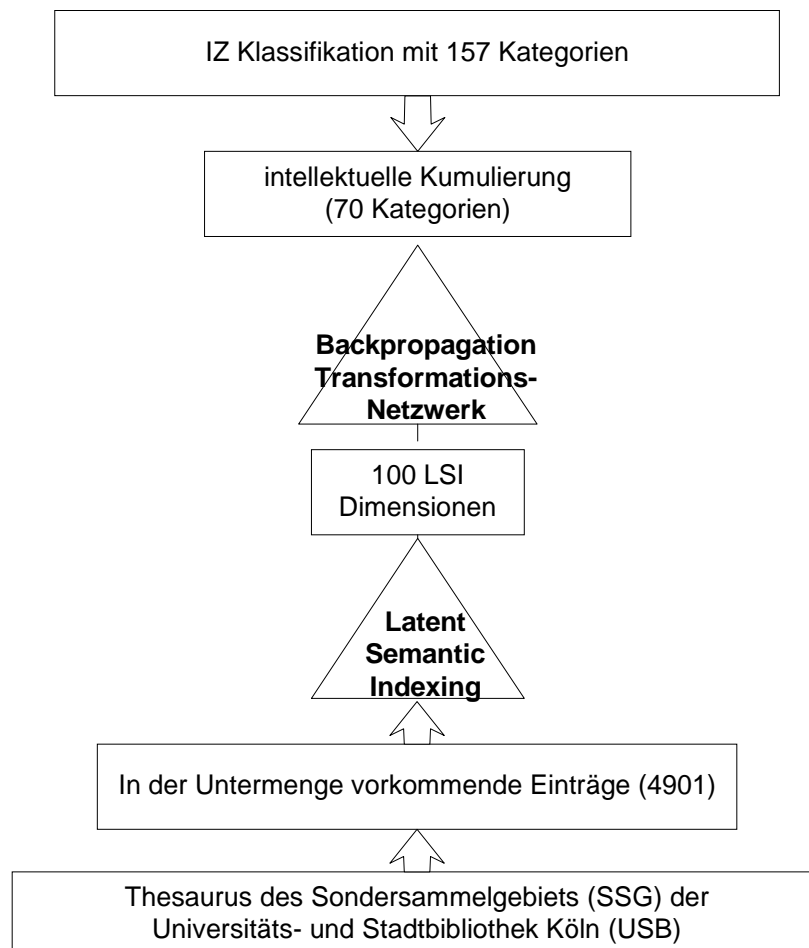


Abbildung 5-7: Schema der Transformation vom USB-Thesaurus zur IZ-Klassifikation

Für die IZ-Klassifikation werden wie im Experiment im vorigen Abschnitt einige inhaltlich sehr nahe verwandte Klassen zusammengelegt, so dass insgesamt 70 Klassen vorliegen. Die verwendeten Dokumente umfassen ca. 4900 Terme des USB-

Thesaurus. Um diese mit einem Transformations-Netzwerk bearbeiten zu können erfolgt eine Reduktion mit Latent Semantic Indexing. Den Input für das Transformations-Netzwerk liefern 100 der errechneten LSI-Dimensionen. Das neuronale Netz besteht also aus 100 Input-Neuronen und 70 Output-Neuronen für die IZ-Klassifikation. Dazwischen liegt eine versteckte Schicht mit 20 Neuronen. Den Ablauf der Transformation skizziert Abbildung 5-7.

Als Vergleichsmaßstab dient eine Transformation der gleichen Daten mit einem statistischen Verfahren auf Basis einer Assoziationsberechnung. Mit der für das Training verwendeten Menge wird eine statistische Transformation erstellt, die für die Dokumente der Test-Menge durchgeführt und bewertet wird. Die Bewertung anhand der Term-Precision und des Term-Recalls erfolgt analog zu dem in Abschnitt 5.1 beschriebenen Experiment.

Die Resultate zeigen eine höhere Term-Precision für das Transformations-Netzwerk als die auf LSI basierende Transformation durch eine statistische Assoziation. Die durchschnittliche Term-Precision liegt für das Transformations-Netzwerk bei 0,11 und für die statistische Transformation bei 0,064. Bei niedrigem Term-Recall ist das Verfahren sogar fast 100% besser, während die beiden Kurven sich bei hohen Recall-Werten stärker annähern wie Abbildung 5-8 zeigt.

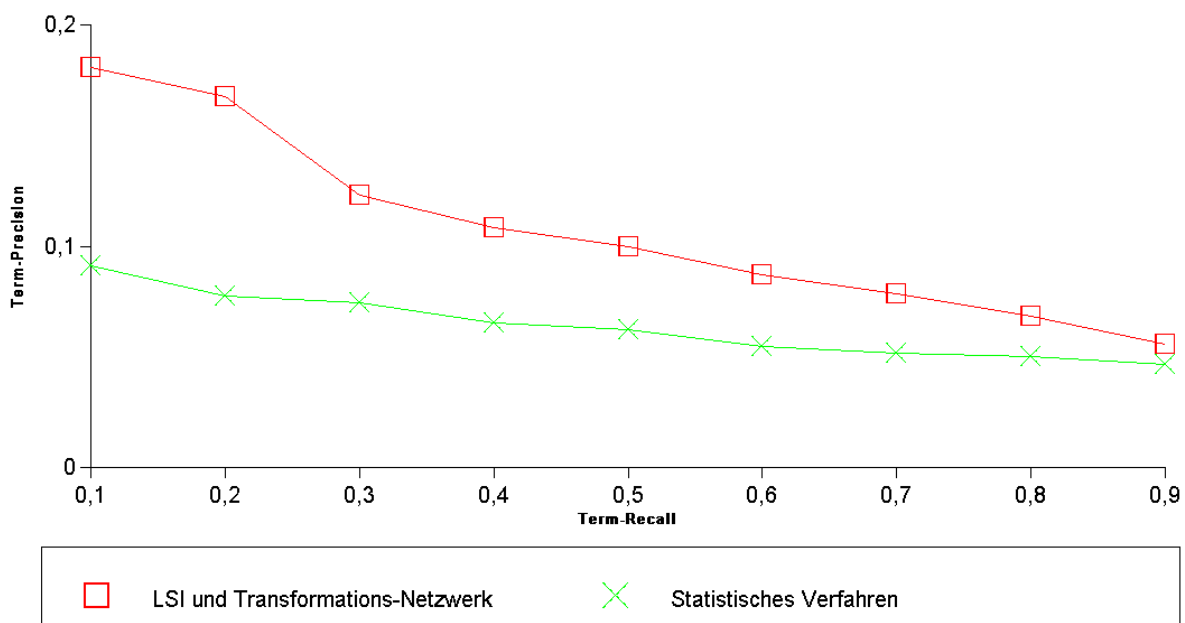


Abbildung 5-8: Ergebnis der Transformation vom USB-Thesaurus zur IZ-Klassifikation als Recall-Precision-Grafik

Allerdings darf dieses Ergebnis nicht überbewertet werden. Die Anzahl der benutzten Dokumente ist insgesamt niedrig und die Term-Precision Werte liegen insgesamt eher niedrig. Dieses insgesamt zufriedenstellende Resultat rechtfertigte die Erstellung eines größeren Korpus für ein weiteres Experiment mit größerer Aussagekraft.

5.2.2 USB-Thesaurus zu IZ-Thesaurus

Im zweiten Schritt wurden ca. 15.000 Dokumente identifiziert, die sowohl vom IZ als auch vom SSG inhaltlich erschlossen wurden. Damit ist eine Abbildung auch auf den detaillierten IZ-Thesaurus möglich. Die bisherigen Transformationen hatten als Ziel

immer die IZ-Klassifikation mit relativ wenig Einträgen und Zuordnungen pro Dokument. Da die Anzahl der Dokumente pro IZ-Thesaurus-Term sehr unterschiedlich ist, fokussiert das Experiment auf die 100 häufigsten Terme im Korpus.

Das IZ hat für die doppelt indexierten Dokument durchschnittlich 11,2 Terme vergeben, die USB 2,9. Die maximale Anzahl von Termen pro Dokument beträgt 56 beim IZ bzw. 19 bei der USB. Die USB hat 10% der Terme nur einmal in dem Korpus vergeben und weitere 21% nur zweimal. Beim IZ liegen diese Werte etwas höher und erreichen 26% bzw. 36%. Insgesamt wird in dieser Menge ein IZ-Term durchschnittlich 38 und ein USB-Term durchschnittlich 15 Dokumenten zugeteilt.

Wie in dem vorhergehenden Experiment komprimiert LSI die USB-Repräsentation der Objekte und zwar in diesem Fall auf 160 Dimensionen. Somit ergibt sich der in Abbildung 5-9 skizzierte Ablauf der Transformation.

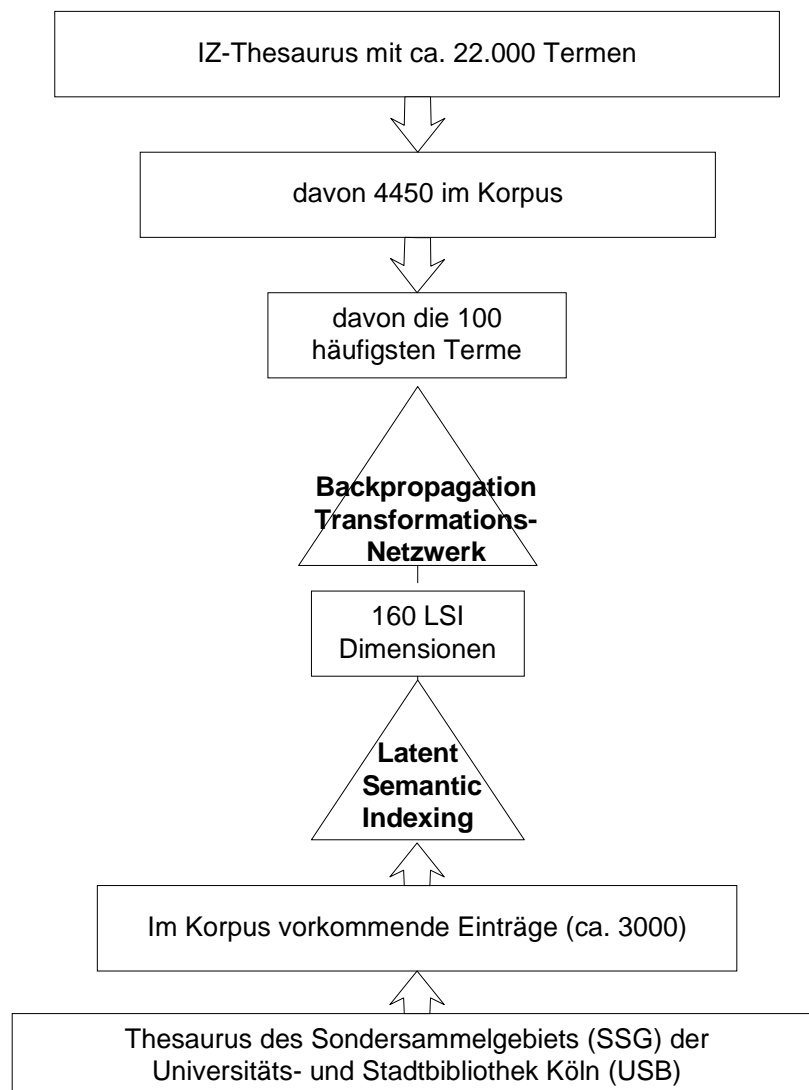


Abbildung 5-9: Schema der Transformation vom USB-Thesaurus zum IZ-Thesaurus

Die Qualität der Transformation für die Dokumente in der Test-Menge messen wieder die Größen Term-Precision und Term-Recall (siehe Abschnitt 5.1.3). Im Ergebnis zeigt sich eine wesentlich höhere Term-Precision für die Kombination aus LSI und neuronalen Transformations-Netzwerk als für die auf LSI basierende statistische Transformation. Die durchschnittliche Term-Precision liegt für das Transformations-Netzwerk bei 0,24 und für die statistische Transformation bei 0,044. Die Recall-Precision-Kurve verläuft für das neuronale Netz erheblich besser für das statistische Verfahren, wie Abbildung 5-10 zeigt.

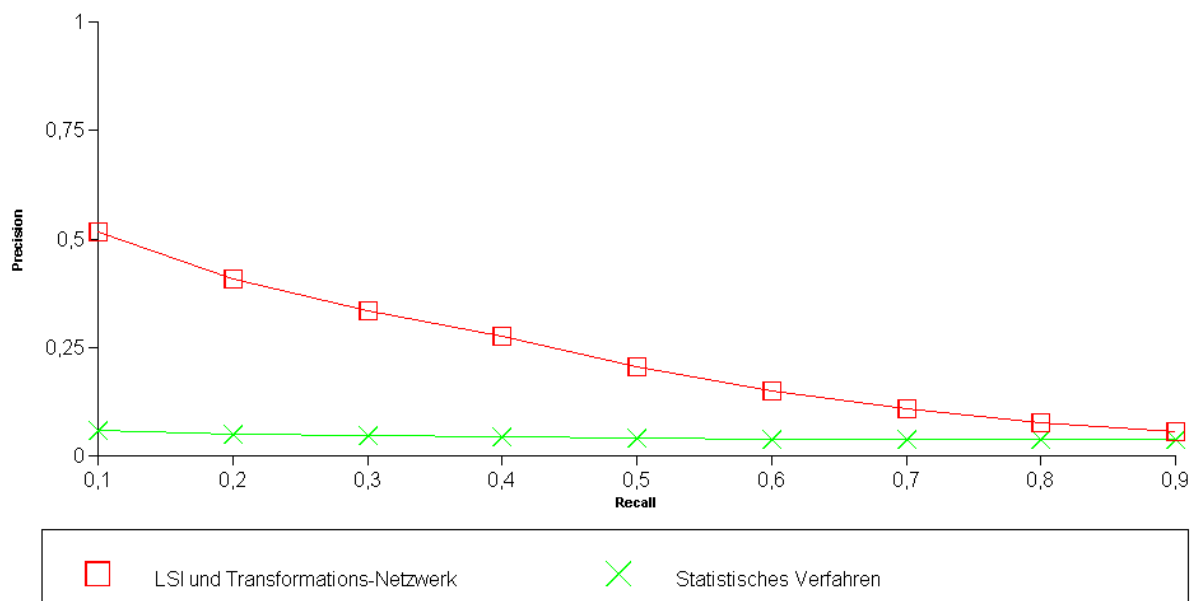


Abbildung 5-10: Ergebnis der Transformation USB-Thesaurus zu IZ-Thesaurus als Recall-Precision-Grafik

Das gute Abschneiden des neuronalen Netzes gegenüber dem statistischen Verfahren bei diesem Experiment weist darauf hin, dass die Ergebnisse sehr stark von den Daten abhängen. Welches das optimale Verfahren ist, muss für jeden Anwendungsfall erneut geprüft werden.

6 Fazit

Semantisch heterogene Datenbestände erfordern Transferkomponenten, die mit unsicherem Wissen operieren können. Dafür bieten sich neuronale Netze an, die aufgrund ihrer Lernfähigkeit, Fehlertoleranz und „weichen“ Informationsverarbeitung auf bei unsicherem Wissen zu einem leistungsfähigen Modell gelangen. Zahlreiche Information Retrieval Systeme basieren bereits auf neuronalen Netzen.

Assoziative Speicher und Kohonen-Netzwerke werden zwar in Information Retrieval Systemen für große Datenmengen eingesetzt, für den Vergleich zwischen Anfrage und Dokument eignen sie sich aber aufgrund ihrer spezifischen Architektur nicht unmittelbar. Die meisten und erfolgreichsten Systeme beruhen auf dem einfachen neuronalen Spreading-Activation-Netzwerk, das nur in geringem Umfang lernfähig ist. Das mächtige Backpropagation-Netzwerk ist nur im experimentellen COSIMIR-Modell und im Transformations-Netzwerk realisiert, das bisher kaum empirisch untersucht wurde. Auch erfolgversprechende Ansätze zur Dimensionalitätsreduktion wie Latent Semantic Indexing wurden im Information Retrieval bisher kaum mit dem Backpropagation-Netzwerk kombiniert. Für Heterogenitätsbehandlung erscheint das Transformations-Netzwerk als das vielversprechendste neuronale Netz.

Experimente mit dem Transformations-Netzwerk in Kombination mit Latent Semantic Indexing und der Vergleich mit einem statistischen Verfahren führten zu interessanten Ergebnissen. Je nach Datengrundlage war Qualität der Transformation sehr unterschiedlich. In einem Fall war das Transformations-Netzwerk wesentlich besser, in einem anderen Experiment ergab sich annähernd gleiche Qualität. Welches Verfahren zum besten Ergebnis führt, hängt also stark von den Daten ab.

Damit zeigt sich eine Analogie zwischen Transformationen und Retrieval. Die TREC-Studien haben für das Retrieval gezeigt, dass sich die Qualität der besten, aktuellen IR-Systeme nur unwesentlich unterscheidet, während ihre Ergebnisse sehr stark unterscheiden. Das heisst, jedes System findet ungefähr gleich viele relevante Dokumente, jedes findet aber andere. Dies führte zur Entwicklung von Fusions-Ansätzen, die versuchen, die Ergebnisse einzelner Verfahren so zu kombinieren, dass ein besseres Gesamtergebnis entsteht.

Die Experimente mit dem Transformations-Netzwerk haben gezeigt, dass bei Transformationen ähnliche Phänomene auftreten können. Verschiedene Verfahren wie das Transformations-Netzwerk und ein statistischer Ansatz führen teilweise zu gleicher Qualität, die Schnittmenge der Ergebnisse ist aber relativ klein. Jedes Verfahren findet also andere Treffer, in diesem Fall Terme aus dem Ziel-Term-Raum. Als Konsequenz sollte auch die Heterogenitätsbehandlung verstärkt auf Fusions-Verfahren setzen. Das *beste* Transformations-Verfahren muss also von Fall zu Fall gefunden oder von einem lernenden Verfahren ermittelt werden. Mandl/Womser-Hacker 2000 stellen ein adaptives System vor, das den Einfluss einzelner Verfahren auf das Fusionsergebnis auf Relevanz-Fedback-Entscheidungen des Benutzers lernt.

Literaturverzeichnis

Alle hier und in Fußnoten aufgeführten Internetquellen (URLs) wurden am 2.6.2000 verifiziert.

- Alonso-Betanzos, Amporo; Fontenla-Romero, Oscar; Guijarro-Berdiñas, Bertha; Principe, Juan Carlos (1999): A Multi-Resolution Principal Component Analysis Neural Network for the Detection of Foetal Heart Rate Patterns. In: Zimmermann 1999.
- Apté, Chidanand; Damerou, Fred; Weiss, Sholom (1994): Towards Language Independent Automated Learning of Text Categorization. In: Bartell et al. 1994. S. 23-30
- Baeza-Yates, Ricardo; Ribeiro-Neto, Berthier (Hrsg.)(1999): Modern Information Retrieval.
- Bartell, Brian; Cottrell, Garrison; Belew, Richard (1994): Automatic Combination of Multiple Retrieval Systems. In: Croft, Bruce (Hrsg.): Proc. of the 17th Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR '94). Dublin, 3.-6.7.1994. London et al.
- Bartlmae, Kai (1998): A Countryrisk Assessment Framework Using Neural Multitask Learning. In: Zimmermann 1998. S. 1019-1023.
- Beasley, David; Bull, David; Martin, Ralph (1993a): An Overview of Genetic Algorithms: Part 1, Fundamentals. In: University Computing 15 (2). S. 58-69.
- Beasley, David; Bull, David; Martin, Ralph (1993b): An Overview of Genetic Algorithms: Part 2, Research Topics. In: University Computing 15 (4). S.170-181.
- Bekavac, Bernard (1999): Suche und Orientierung im WWW. Verbesserung bisheriger Verfahren durch Einbindung hypertextspezifischer Informationen. Konstanz.
- Belew, Richard (1986): Adaptive Information Retrieval: Machine Learning in Associative Networks. PhD Dissertation. University of Michigan, Ann Harbor.
- Belew, Richard (1989): Adaptive Information Retrieval: Using a Connectionist Representation to Retrieve and Learn about Documents. In: Belkin/van Rijsbergen 1989. S. 11-20.
- Belkin, Nicholas; Ingwersen, Peter; Pejtersen, Annelise (Hrsg.)(1992): Proc. of the 15th Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR '92). Kopenhagen, Dänemark. 21-24.6.1992. London et al.
- Belkin, Nicholas; Rijsbergen, C.J. van (Hrsg.)(1989): Proc. of the 12th Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR '89) Cambridge, MA, USA 25.-28.6.89. London et al.
- Bentz, Hans-Joachim; Hagström, Michael; Palm, Guenther (1998): Information Storage and Effective Data Retrieval in Sparse Matrices. In: Neural Networks 2 (4). S. 289-293.
- Berry, Michael (1992): Large Scale Sparse Singular Value Computations. In: Intl. Journal of Supercomputer Applications. S.13-49.
- Berry, Michael; Do, Theresa; O'Brien, Gavin; Krishna, Vijay; Varadhan, Sowmini (1993): SVDPACKC (Version 1.0) User's Guide. Arbeitspapier. Computer Science Department. University of Tennessee in Knoxville, USA.
<http://www.netlib.org/svdpack/svdpackc.tgz>
- Berry, Michael; Dumais, Susan; Letsche, Todd (1995): Computational Methods for Intelligent Information Access. In: Proc. of ACM Supercomputing '95. San Diego, CA. S. 1-38.
- Biebricher, B.; Fuhr, Norbert; Lustig, G.; Schwantner, M.; Knorz, Gerhard (1988): The Automatic Indexing System AIR/PHYS - from Research to Application. In: Chiaramella, Yves (Hrsg.): Proc. of the 11th Intl. Conf. on Research/Development in Information Retrieval. ACM. New York. S. 333-342.
- Bigus, Joseph (1996): Data Mining with Neural Networks. Solving Buisness Problems from Application Development to Decision Support. New York et al.

- Bookstein, Abraham; Chiaramella, Yves; Salton, Gerard; Raghavan, Vijay V. (Hrsg.)(1991): Proc. of the 14th Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR '91). Chicago, IL, USA 13.-16.10.91. New York.
- Bordogan, Gloria; Pasi, Gabriella; Petrosino, Alfredo (1996): Relevance Feedback Based on a Neural Network. In: Zimmermann 1996. S. 846-850.
- Boughanem, M.; Chrismont, C.; Soulé-Dupuy, C. (1999): Query Modification based on Relevance Back-Propagation in an ad hoc Environment. In: Information Processing and Management 35. S. 121-139.
- Boughanem, M.; Soulé-Dupuy, C. (1994): Query Expansion and Neural Network. In: Intelligent Multimedia Information Retrieval Systems and Management. Proc. of the RIAO 94 (Recherche d'Information assistée par Ordinateur). Rockefeller University. New York. S. 519-532.
- Boughanem, M.; Soulé-Dupuy, C. (1997): MercureO2: adhoc and routing tasks. In: Voorhees/Harman 1997.
- Boughanem, M.; Soulé-Dupuy, C. (1998): Mercure at trec6. In: Voorhees/Harman 1998
- Boyd, Richard; Driscoll, James; Syu, Inien (1994): Incorporating Semantics within a Connectionist Model and a Vector Processing Model. In: Harman 1994. S. 291-302.
- Buckley, Chris (1998): TREC 6 High-Precision Track. In: Harman/Voorhees 1998. S. 69-71.
- Caid, William R.; Dumais, Susan; Gallant, Stephen (1995): Learned Vector-Space Models for Document Retrieval. In: Information Processing & Management 31(3). S. 419-429.
- Caruana, Rich (1994): Learning Many Related Tasks at the Same Time With Backpropagation. In: Advances in Neural Information Processing Systems 7 (Proc. of NIPS '94) S. 657-664. <http://www.cs.cmu.edu/afs/cs.cmu.edu/user/caruana/pub/papers/nips94.ps>
- Caruana, Rich (1997): Multitask Learning. In: Machine Learning 28. S. 41-75. <http://www.cs.cmu.edu/afs/cs.cmu.edu/user/caruana/pub/papers/mlj97.ps>
- Caudell, Thomas; Smith, Scott; Escobedo, Richard; Anderson, Michael (1994): NIRS: Large Scale ART-1 Neural Architectures for Engineering Design Retrieval. In: Neural Networks 7(9). S. 1339-1350.
- Chen, Hsinchun (1994): A Machine Learning Approach to Document Retrieval: An Overview and an Experiment. Technical Report. MIS Department. University of Arizona, Tucson.
- Chen, Hsinchun (1995): Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning, and Genetic Algorithms. In: Journal of the American Society for Information Science. JASIS 46(3). S. 194-216.
- Chen, Hsinchun; Schuffels, Chris; Orwig, Richard (1996): Internet Categorization and Search: A Self-Organizing Approach. In: Journal of Visual Communication and Image Representation. 7(1). S. 88-101.
- Chung, Yi-Ming; Pottenger, William; Schatz, Bruce (1998): Automatic Subject Indexing Using an Associative Neural Network. In: Proc. of the Third ACM Conf. on Digital Libraries. 23.-26.6.1998, Pittsburgh, PA, USA. S. 59-68
- Cichocki, Andrzej; Unbehauen, Rolf (1993): Neural Networks for Optimization and Signal Processing. Stuttgart et al.
- Clauß, G.; Ebner, H. (1979): Grundlagen der Statistik. Frankfurt a.M.
- Cochet, Yves; Paget, Gerard (1988): ZZENN: ZIG ZAG Epigenetic Neural Networks and Their Use in Information Systems. In: Personnaz, L.; Dreyfus, G. (Hrsg.): Neural Networks: From Models To Applications. Proc. of Neuro'88. S. 663-672.
- Cortez, Edwin; Park, Sang; Kim, Seonghee (1995): The Hybrid Application of an Inductive Learning Method and a Neural Network for Intelligent Information Retrieval. In: Information Processing and Management 31(6). S. 789-813.
- Creput, Jean-Charles; Caron, Armand (1997): An Information Retrieval System Using a New Neural Network Model. In: Cybernetica XL(2). S. 127-139.
- Crestani Fabio (1993): Learning Strategies for an Adaptive Information Retrieval System using Neural Networks. In: Proc. of the IEEE Intl. Conf. on Neural Networks. San Francisco, California, USA. S. 244-249. <http://ir.dcs.gla.ac.uk/publications/papers/Postscript/crestani93a.ps.gz>

- Crestani, Fabio (1995): Implementation and Evaluation of a Relevance Feedback Device Based on Neural Networks. In: Mira, José; Cabestany, Joan (Hrsg.): From Natural to Artificial neural Computation: Intl. Workshop on Artificial Neural Networks. Spain. S. 1-8.
- Crestani, Fabio (1997): Application of Spreading Activation Techniques in Information Retrieval. In: Artificial Intelligence Review 11 (6). S. 453-482.
<http://ir.dcs.gla.ac.uk/publications/papers/Postscript/crestani96a.ps.gz>
- Crestani, Fabio (1997a): Retrieving Documents by Constrained Spreading Activation on Automatically Constructed Hypertexts. In: Zimmermann 1997. S. 1210-1214.
- Crestani, Fabio; Rijsbergen, Keith van (1997): A Model for Adaptive Information Retrieval. In: Journal of Intelligent Information Systems 8 (1). S. 29-56.
<http://ir.dcs.gla.ac.uk/publications/papers/Postscript/crestani97a.ps.gz>
- Croft, Bruce; Moffat, Alistair; Rijsbergen, Keith van; Wilkinson, Ross; Zobel, Justin (1998): Proc. of the 21st Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR '98). Melbourne 24.-28.8.1998. New York.
- Deerwester, Scott; Dumais, Susan T.; Harshman, Richard (1990): Indexing by Latent Semantic Analysis. In: Journal of The American Society For Information Science (JASIS) 41(6). S. 391-407.
- Dorffner, Georg (1991). Konnektionismus. Von neuronalen Netzwerken zu einer natürlichen KI. Stuttgart.
- Doszkocs, T.E.; Reggia, J.; Lin, Xia (1990): Connectionist Models and Information Retrieval. In: Annual Review of Information Science and Technology (ARIST) 25. S. 209-260.
- Dumais, Susan (1994): Latent Semantic Indexing (LSI) and TREC-2. In: Harman 1994. S. 105-115.
- Dumais, Susan; Letsche, Todd; Littman, Michael; Landauer, Thomas (1997): Automatic Cross-Language Retrieval Using Latent Semantic Indexing. In: Hull/Oard 1997. S. 15-21.
- Escobedo, Richard; Smith, Scott; Caudell, Thomas (1993): A Neural Information Retrieval System. In: Intl. Journal of Advanced Manufacturing Technology 8(4). S. 269-274.
- Fayyad, Usama (1997): Editorial. In: Data Mining and Knowledge Discovery 1(1). S. 5-10.
- Ferber, Reginald (1997): Automated Indexing with Thesaurus Descriptors: A Cooccurrence Base Approach to Multilingual Retrieval. In: Peters, Carol; Thanos, Constantino (Hrsg.): Research and Advanced Technology for Digital Libraries. 1st European Conf. ECDL'97. Pisa, 1.-3.9.97. Berlin et al. S. 233-252.
- Fox, Edward (Hrsg.)(1995): Proc. of the 18th Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR 95). Seattle, USA 9.-13.7.95. New York.
- Frei, Hans-Peter; Harman, Donna; Schäuble, Peter; Wilkinson, Ross (Hrsg.)(1996): Proc. of the 19th Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR '96). Zürich, 18.-22.8.96. New York.
- Gallant, Stephen; Caid, William; Carleton, Joel; Hecht-Nielsen, Robert; Qing, Kent; Sudback, David (1993): HNC's MatchPlus System. In: Harman 1993. S. 107-111.
- Gallant, Stephen; Caid, William; Carleton, Joel; Gutschow, Todd; Hecht-Nielsen, Robert; Qing, Kent; Sudback, David (1994): Feedback and Mixing Experiments with Match Plus. In: Harman 1994. S. 101-104.
- Gauch, Susan; Wang, Jianying (1997): Corpus Analysis for TREC 5 Query Expansion. In: Voorhees/Harman 1997.
- Gloor, Peter (1997): Elements of Hypermedia Design: Techniques for Navigation & Visualization in Cyberspace. Birkhäuser: Boston.
- Graupe, D.; Kordylewski, H. (1998): A Large Memory Storage and Retrieval Neural Network for Adaptive Retrieval and Diagnosis. In: Intl. Journal of Software Engineering and Knowledge Engineering 8(1). S. 115-138.
- Grael, A.; Ludwig, L.; Renners, I. (1999): Comparison of Neuro/Fuzzy Methods for Toxicity Evaluation. In: Zimmermann 1999.
- Hagström, Michael (1996): Textrecherche in großen Datenmengen auf der Basis spärlich codierter Assoziativmatrizen. Dissertation. Universität Hildesheim.
- Harman, Donna (1992): Relevance feedback revisited. In: Belkin et al. 1992. S. 1-10.

- Harman, Donna (Hrsg.)(1993): The First Text Retrieval Conference (TREC-1). NIST Special Publication 500-207. National Institute of Standards and Technology. Gaithersburg, Maryland, 4.-6.11.1992. http://trec.nist.gov/pubs/trec1/t1_proceedings.html
- Harman, Donna (Hrsg.)(1994): The Second Text REtrieval Conference (TREC-2). Publication 500-215. National Institute of Standards and Technology. Gaithersburg, Maryland, 31.8.-2.9.1993. http://trec.nist.gov/pubs/trec2/t2_proceedings.html
- Harman, Donna (Hrsg.)(1995): The Third Text REtrieval Conference (TREC-3). NIST Special Publication 500-225. National Institute of Standards and Technology. Gaithersburg, Maryland, 2.-4.11.1994. http://trec.nist.gov/pubs/trec3/t3_proceedings.html
- Harman, Donna (Hrsg.)(1996): The Fourth Text Retrieval Conference (TREC-4). NIST Special Publication 500-236. National Institute of Standards and Technology. Gaithersburg, Maryland, 1.-3.11.1995. http://trec.nist.gov/pubs/trec4/t4_proceedings.html
- Hartung, Joachim (1984): Lehr- und Handbuch der angewandten Statistik. München, Wien.
- Heitland, Michael (1994): Einsatz der SpaCAM-Technik für ausgewählte Grundaufgaben der Informatik. Dissertation. Universität Hildesheim.
- Heuer, Andreas; Saake, Gunter (1997): Datenbanken : Konzepte und Sprachen. Bonn et al.
- Honkela, Timo; Kaski, Samuel; Lagus, Krista; Kohonen, Teuvo (1997): WEBSOM-Self-Organizing Maps of Document Collections. In Proc. of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4-6, Helsinki University of Technology, Neural Research Centre, Espoo, Finland. S. 310-315.
- Hui, Siu; Goh, Angela (1996): Incorporating Fuzzy Logic with Neural Networks for Document Retrieval. In: Engineering Applications of Artificial Intelligence 9(5). S. 551-560.
- Hull, David; Oard, Doug (1997): Cross-Language Text and Speech Retrieval Papers from the 1997 AAAI Spring Symposium. Technical Report SS-97-05.
- Informationzentrum Sozialwissenschaften (1997): Thesaurus Sozialwissenschaften. Deutsch-Englisch. Bearbeitet von Hannelore Schott. Bonn.
- Kaski, Samuel (1998): Dimensionality Reduction by Random Mapping: Fast Similarity Computation for Clustering. In: Proc. IJCNN'98, Intl. Joint Conf. on Neural Networks, Piscataway, NJ. vol.1. S.413-418.
- Kaski, Samuel; Lagus, Krista; Honkela, Timo; Kohonen, Teuvo (1998): Statistical Aspects of the WEBSOM System in Organizing Document Collections. In: Computing Science and Statistics 29. S. 281-290.
- Kinnebrock, W. (1992): Neuronale Netze. Grundlagen, Anwendungen, Beispiele. München; Wien.
- Kluck, Michael; Krause, Jürgen; Müller, Matthias; Schmiede, R.; Wenzel, H.; Winkler, S.; Meier, W. (2000): Virtuelle Fachbibliothek Sozialwissenschaften: IZ-Arbeitsbericht 19, Informationszentrum Sozialwissenschaften, Bonn. <http://www.bonn.iz-soz.de/publications/series/working-papers/index.htm#Virtuelle>
- Kohonen, Teuvo (1984). Self-Organization and Associative Memory. Berlin et al.
- Kohonen, Teuvo (1997): Exploration of very large databases by self-organizing maps. In Proc. of ICNN'97, Intl. Conf. on Neural Networks. IEEE Service Center, Piscataway, NJ. S. PL1-PL6.
- Kohonen, Teuvo (1997²a): Self-Organizing Maps. Springer: Berlin et al.
- Kohonen, Teuvo (1998): Self-organization of Very Large Document Collections: State of the art. In Niklasson, L.; Bodén, M.; Ziemke, T. (Hrsg.): Proc. of ICANN '98, 8th Intl. Conf. on Artificial Neural Networks, Springer: London. vol. 1, S. 65-74.
- Krause, Jürgen (1996): Principles of Content Analysis for Information Retrieval Systems: An Overview. In: Zuell, C.; Harkness, J.; Hoffmeyer-Zlotnik, J. (Hrsg.): Text Analysis and Computer. ZUMA-Nachrichten Spezial. Mai. Mannheim. S.77-100. ftp://ftp.zuma-mannheim.de/pub/zuma/zuma-nachrichten_spezial/znspezial1.pdf
- Krause, Jürgen (1996a): Informationserschließung und -bereitstellung zwischen Deregulation, Kommerzialisierung und weltweiter Vernetzung. („Schalenmodell"). IZ-Arbeitsbericht 6, Informationszentrum Sozialwissenschaften, Bonn. <http://www.bonn.iz-soz.de/publications/series/working-papers/index.htm#Informationserschließung>

- Krause, Jürgen (1998): Innovative Current Research Information Systems in the Information Society. In: CRIS '98 Current Research Information Systems. Luxemburg, 12-14.3.1998.
ftp://ftp.cordis.lu/pub/cybercafe/docs/krause.zip
- Krause, Jürgen; Schaefer, André (1998): Textrecherche-Oberfläche für ELVIRA II. ELVIRA-Arbeitsbericht 16, IZ Sozialwissenschaften, Bonn.
- Kuhlen, Rainer (1991): Hypertext: ein nicht-lineares Medium zwischen Buch und Wissensbank. Berlin.
- Kuhlen, Rainer (1999): Die Konsequenzen von Informationsassistenten: Was bedeutet informationelle Autonomie oder wie kann Vertrauen in elektronische Dienste in offenen Informationsmärkten gesichert werden? Frankfurt a.M.
- Kwok, K. L. (1989): A Neural Network for Probabilistic Information Retrieval. In: Belkin/Rijsbergen 1989. S. 21-30.
- Kwok, K. L. (1991a): Query Modification and Expansion in a Network with Adaptive Architecture. In: Bookstein et al. 1991. S. 192-201.
- Kwok, K. L. (1991b): Query Learning Using an ANN with Adaptive Architecture. In: Machine Learning '91. S. 260-264.
- Kwok, K. L. (1996): A New Method of Weighting Query Terms for Ad-Hoc Retrieval. In: Frei et al. 1996 S. 187-195.
- Kwok, K. L.; Chan, M. (1998): Improving Two-Stage Ad-Hoc Retrieval for Short Queries. In: Croft et al. 1998. S. 250-256.
- Kwok, K.L.; Grunfeld, L. (1994): TREC2 Document Retrieval Experiments using PIRCS. In: Harman 1994. S. 233-242.
- Kwok, K.L.; Grunfeld, L. (1996): TREC-4 Ad-Hoc, Routing Retrieval and Filtering Experiments Using PIRCS. In: Harman 1996.
- Kwok, K.L.; Grunfeld, L.; Chan, M; Dinstl, N.; Cool, C. (1999): TREC-7 Ad-Hoc, Routing Retrieval and Filtering Experiments Using PIRCS. In: Voorhees/Harman 1999. S. 343-352.
- Kwok, K.L.; Papadopoulos, L.; Kwan, Y.Y. (1993): Retrieval Experiments with a Large Collection Using PIRCS. In: Harman 1993. S. 153-172.
- Layaida, Redouane; Caron, Armand (1994): Applications of the Backpropagation Algorithm to an Information Retrieval System. In: Intelligent Multimedia Information Retrieval Systems and Management. Proc. of the RIAO 94 (Recherche d' Information assistée par Ordinateur). Rockefeller University. New York. S. 161-171.
- Lee, Joon Ho (1995): Combining Multiple Evidence from Different Properties of Weighting Schemes. In: Fox 1995. S. 180-188.
- Lee, Jonghoon; Dubin, David (1999): Context-Sensitive Mapping with a Spreading Activation Network. In: Hearst, Marti; Frederic, Gey; Richard, Tong (Hrsg.): Proc. of the 22nd Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR '99). Berkeley, CA 15-19.8.99. New York.
- Lewis, David; Schapire, Robert; Callan, James; Papka (1996): Training algorithms for linear text classifiers In: Frei et al. 1996. S. 298-306.
- Lin, C.; Chen, Hsinchun (1994): An Automatic Indexing and Neural Network Approach to Concept Retrieval and Classification of Multilingual (Chinese-English) Documents. Technical Report. MIS Department, University of Arizona, Tucson, USA. Juli 1994.
- Lin, Xia (1995). Searching and Browsing on Map Displays. In: Kinney, Thomas (Hrsg.): Proc. of the 58th Annual Meeting American Association for Information Science (Vol. 32) ASIS'95. Chicago, Oktober. S. 13-18.
- Lin, Xia; Soergel, Dagobert; Marchionini, Gary (1991): A Self-Organizing Semantic Map for Information Retrieval. In: Bookstein et al. 1991. S. 262-269.
- MacLeod, Kevin J.; Robertson, W. (1991): A Neural Algorithm for Document Clustering. In: Information Processing & Management 27 (4). S. 337-346.
- Mandl, Thomas (1998): Vague Transformations in Information Retrieval. In: Zimmermann, H.; Schramm, V. (Hrsg.): Knowledge Management und Kommunikationssysteme: Workflow Management, Multimedia, Knowledge Transfer. Proc. 6. Int. Symposium für Informationswissenschaft. (ISI '98). 3. – 7. 11. 1998, Prag, Konstanz. S. 312 – 325.

- Mandl, Thomas (1999): Effiziente Implementierung von statistischen Assoziationen im Text-Retrieval. In: Ockenfeld, Marlies; Mantwill, Gerhard (Hrsg.): Information und Region. 51. Jahrestagung der Deutschen Gesellschaft für Informationswissenschaft und Informationspraxis e.V. (DGI). Hamburg, 21.-23.9.1999. Frankfurt a.M. S. 159-172.
- Mandl, Thomas (2000): Einsatz neuronaler Netze im Information Retrieval. Arbeitsbericht. Informationswissenschaft, Universität Hildesheim.
- Mandl, Thomas; Womser-Hacker, Christa (2000): Ein adaptives Information-Retrieval-Modell für Digitale Bibliotheken. In: Proc. 7. Intl. Symposium für Informationswissenschaft. (ISI '2000). Darmstadt. 8-10.11.2000.
- McClelland, James; David Rumelhart (1988): Explorations in Parallel Distributed Processing. A Handbook of Models, Programs, and Exercises. Cambridge (MA.) et al.
- Merkel, Dieter (1995): Content-Based Document Classification with Highly Compressed Input Data. In: Proc. Intl. Conf. on Artificial Neural Networks ICANN '95. Paris. Oktober 9-13. Bd. 2, S. 239-244.
- Merkel, Dieter; Tjoa, A Min; Kappel, Gerti (1994): Learning the Semantic Similarity of Reusable Software Components. In: Frakes, William B. (Hrsg.): Proc. of the Third Intl. Conf. on Software Reuse: Advances in Software Reusability. Rio de Janeiro, 1.-4.11. 1994. Washington et al. S. 33-41.
- Moody, John (1992): The Effective Number of Parameters: An Analysis of Generalization and Regularization in Nonlinear Learning Systems. In: Moody, John E.; Hanson, S.J.; Lippmann, R.P. (Hrsg.): Advances in Neural Information Processing Systems (NIPS) 4. San Mateo, CA.
- Mothe, Josiane (1992): SYRENE: An Information Retrieval System Based on Neural Approaches: Experimental Results. In: Fifth Intl. Conference. Neural Networks and their Applications. NEURO NIMES '92. Nimes, France, Nov. 1992. EC 2, S. 81-91.
- Mothe, Josiane (1994): Search Mechanisms Using a Neural Network Model. In: Intelligent Multimedia Information Retrieval Systems and Management. Proc. of the RIAO '94 (Recherche d'Information assistée par Ordinateur). Rockefeller University. New York, 11.-13.10.94. S. 275-294.
- Mothe, Josiane; Soule-Dupuy, C (1992): Integration of a Connectionist Model in Information Retrieval Systems. In: Aleksander, I.; Taylor, J. (Hrsg.): Artificial Neural Networks, Proc. of the 1992 Intl. Conf. (ICANN-92). Brighton, Sept. 1992. S. 1611-14, vol 2 of 2.
- Mönnich, Michael (1999): Kriterien zur Bewertung und Auswahl von Internetsuchmaschinen. In: Schmidt, Ralph (Hrsg.): Proc. der 21. Online Tagung der DGI. Aufbruch ins Wissensmanagement. Deutsche Gesellschaft für Informationswissenschaft und -praxis. Frankfurt, 18.-20.5. S. 141-151.
- Mori, Hirohiko; Chung, Cheng Long; Kinoe, Yousuke; Hayashi, Yoshio (1990): An Adaptive Document Retrieval System Using a Neural Network. In: Intl. Journal of Human-Computer Interaction 2(3). S. 267-280..
- Nauck, Detlef; Klawon, Frank; Kruse, Rudolf (1994): Neuronale Netze und Fuzzy-Systeme. Grundlagen des Konnektionismus, Neuronaler Fuzzy-Systeme und der Kopplung mit wissensbasierten Methoden.
- Newell, Allen; Simon, H.A. (1976): Computer Science as Empirical Inquiry: Symbols and Search. In: Communications of the ACM 19(3). S. 113-126.
- Orwig, Richard; Chen, Hsinchun; Nunamaker, Jay (1997): A Graphical, Self-Organizing Approach to Classifying Electronic Meeting Output. In: Journal of the American Society for Information Science. JASIS 48(2). S. 157-170.
- Papka, Ron; Callan, James; Barto, Andrew (1996): Text-Based Information Retrieval Using Exponentiated Gradient Descent. In: Touretzky, David (Hrsg.): Advances in Neural Information Processing. San Mateo, California. S. 3-9.
- Personnaz, L; Guyon, I.; Dreyfus, G. (1986): Neural Network Design for Efficient Information Retrieval. In: Bienenstock, E.; Fogelman Soulie, F.; Weisbuch, G. (Hrsg.): Disordered Systems and Biological Organization. Berlin: Springer. S. 227-231.
- Refenes, A.N.; Azema-Barac, M. (1994): Neural Network Applications in Financial Asset Management. In: Neural Computing & Applications 2(1). S.13-39.
- Rich, Elaine; Knight, Kevin (1991): Artificial Intelligence. McGraw Hill: New York et al.
- Rijsbergen, Keith van (1979): Information Retrieval. London et al.
- Rittberger, Marc (1995): Auswahl von Online-Datenbanken - Eine Recherveschnittstelle für das Online-Retrieval integriert in das Konstanzer Hypertext System. Dissertation, Universität Konstanz.

- Rodeghier, Mark (1997): Marktforschung mit SPSS: Analyse, Datenerhebung und Auswertung.
- Rojas, Raúl (1993): Theorie der neuronalen Netze. Eine systematische Einführung. Berlin et al.
- Rose, D.; Belew, Richard (1991): A Connectionist and Symbolic Hybrid for Improving Legal Research. In: Intl. Journal of Man-Machine-Studies 31(1). S. 1-33.
- Rozmus, J. Michael (1995): Information Retrieval by Self-Organizing Maps. In: Williams, Martha (Hrsg.): Proc. 16th National Online Meeting. New York, 2.-4.5.1998. Medford, NJ, 1995. S. 349-354.
- Rumelhart, David; McClelland, James; PDP Research Group (1986): Parallel Distributed Processing. Explorations in the Microstructure of Cognition. vol.1: Foundations. vol.2: Psychological and Biological models. Cambridge, MA et al.
- Rumelhart, David; Hinton, Geoffrey; McClelland, James (1986): A General Framework for Parallel Distributed Processing. In: Rumelhart/McClelland (1986) vol. 1. S. 45-76.
- Salton, Gerard; Buckley, Chris (1988): On the Use of Spreading Activation Methods in Automatic Information Retrieval. In: Chiaramella, Yves (Hrsg.): Proc. of the 11th Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR '88). Grenoble 13-15.7.88. New York. S. 147-160.
- Schatz, Bruce (1998): High-Performance Distributed Digital Libraries: Building the Interspace on the Grid. In: 7th IEEE Int Symp High-Performance Distributed Computing (July). S. 224-234. <http://www.canis.uiuc.edu/archive/papers/hpdc.pdf>
- Schatz, Bruce; Johnson, Eric; Cochrane, Pauline (1996): Interactive Term Suggestion for Users of Digital Libraries: Using Subject Thesauri and Co-occurrence Lists for Information Retrieval. In: Fox, Edward; Marchionini, Gary (Hrsg.): Proc. of the 1st ACM Intl. Conf. on Digital Libraries. Bethesda, Maryland 20.-23.3.96. New York. S. 126-133.
- Scherer, Andreas (1997): Neuronale Netze: Grundlagen und Anwendungen. Braunschweig, Wiesbaden.
- Scholtes, J.C. (1992): Neural Nets in Information Retrieval: A Case Study of the 1987 Pravda. In: Ruck, Dennis (Hrsg.): Science of Artificial Neural Networks: Proc. of the Intl. Society for Optical Engineering, 1710 (SPIE Conf., Orlando, FL.) S. 631-641.
- Schütze, Hinrich; Pedersen, Jan (1997): A Cooccurrence-Based Thesaurus and two Applications to Information Retrieval. In: Information Processing & Management 33(3). S. 307-318.
- Schwenker, F.; Sommer, F.; Palm, G. (1996): Iterative Retrieval of Sparsely Coded Associative Memory Patterns. In: Neural Networks 9(3). S. 445-455.
- Searle, Warren (2000): Neural Networks: FAQ (Frequently Asked Questions and Answers), News-group comp.ai.neural-nets. <ftp://ftp.sas.com/pub/neural/FAQ.html>
- Smeaton, Alan; Wilkinson, Ross (1997): Spanish and Chinese Document Retrieval in TREC-5. In: Voorhees/Harman 1997. S. 57-64.
- Smith, Scott; Escobedo, Richard; Anderson, Michael; Caudell, Thomas (1997): A Deployed Engineering Design Retrieval System Using Neural Networks. In: IEEE Transactions on Neural Networks 8(4). S. 847-851.
- Smolensky, Paul (1988): On the Proper Treatment of Connectionism. In: Behavioral and Brain Sciences 11. S. 1-74.
- Stafylopatis, Andreas; Likas, Aristidis (1992): Pictorial Information Retrieval Using the Random Neural Network. In: IEEE Transactions on Software Engeneering 18 (7). 1992. S. 590-600.
- Syu, Inien; Lang, S. D. (1994): Heuristic Information Retrieval: A Competition-Based Connectionist Model. In: Intelligent Multimedia Information Retrieval Systems and Management. Proc. RIAO '94 (Recherche d'Information assistée par Ordinateur). Rockefeller Univ.. New York. S. 248-265.
- Syu, Inien; Lang, S.; Deo, Narsingh (1996): Incorporating Latent Semantic Indexing into a Neural Network Model for Information Retrieval. In: Barker, Ken; Oezsu, Tamer (Hrsg.): ACM Conf. on Information and Knowledge Management (CIKM.'96). Rockville MD. 12.-16-11.96. S. 145-153.
- Tversky, Amor (1977): Features of Similarity. In: Psychological Review 84(4) Juli 1977. S. 327-351.
- Voorhees, Ellen; Gupta, Narendra, Johnon-Laird, Ben (1995): Learning Collection Fusion Strategies. In: Fox 1995. S. 172-179.
- Voorhees, Ellen; Harman, Donna (Hrsg.)(1997): The Fifth Text Retrieval Conference (TREC-5). NIST Special Publication 500-238. National Institute of Standards and Technology. Gaithersburg, Maryland, 20.-22.11.1996. http://trec.nist.gov/pubs/trec5/t5_proceedings.html

- Voorhees, Ellen; Harman, Donna (1997a): Overview of the Fifth Text REtrieval Conference (TREC-5). In: Voorhees/Harman 1997. S. 1-28.
- Voorhees, Ellen; Harman, Donna (Hrsg.)(1998): The Sixth Text Retrieval Conference (TREC-6). NIST Special Publication 500-240. National Institute of Standards and Technology. Gaithersburg, Maryland, 19.-21.11.1996. http://trec.nist.gov/pubs/trec6/t6_proceedings.html
- Voorhees, Ellen; Harman, Donna (1998a): Overview of the Sixth Text REtrieval Conference (TREC-6). In: Voorhees/Harman 1998. S. 1-24.
- Voorhees, Ellen; Harman, Donna (Hrsg.)(1999): The Seventh Text Retrieval Conference (TREC-7). NIST Special Publication 500-242. National Institute of Standards and Technology. Gaithersburg, Maryland, 9.-11.11.1999. http://trec.nist.gov/pubs/trec7/t7_proceedings.html
- Voorhees, Ellen; Harman, Donna (1999a): Overview of the Seventh Text REtrieval Conference (TREC-7). In: Voorhees/Harman 1999. S. 1-24.
- Waard, W. P. de (1994): Neural techniques and postal code detection. In: Pattern Recognition Letters 15 (2). S.199-205.
- Wilkinson, Ross; Hingston, P. (1992): Incorporating the vector space model in a neural network used for document retrieval. In: Library HiTech News 10 (1-2), S. 69-75.
- Wilkinson, Ross; Hingston, Philip (1991): Using the Cosine Measure in a Neural Network for Document Retrieval. In: Bookstein et al. 1991. S. 202-210.
- Wilkinson, Ross (1998): Chinese Document Retrieval at TREC-6. In: Harman/Voorhees 1998. S. 25-29.
- Womser-Hacker, Christa (1997): Das MIMOR-Modell. Mehrfachindexierung zur dynamischen Methoden-Objekt-Relationierung im Information Retrieval. Habilitationsschrift. Universität Regensburg, Informationswissenschaft.
- Wong, S. K. M.; Cai, Y. J.; Yao, Y. Y. (1993): Computation of Term Associations by a Neural Network. In: Korfhage, Robert; Rasmussen, Edie; Willett, Peter (Hrsg.)(1993): Proc. of the 16th Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval. (SIGIR '93). Pittsburgh 27.6.-1.7.93. New York. S. 107-115.
- Yang, Yiming (1995): Noise Reduction in a Statistical Approach to Text Categorization. In: Fox 1995. S. 256 - 263.
- Young, Paul (1994): Cross-Language Information Retrieval Using Latent Semantic Indexing. Tech. Report. Univ. of Tennessee. <http://www.cs.utk.edu/~library/TechReports/1994/ut-cs-94-259.ps.z>
- Zadeh, Lofti (1994): What is BISC? http://http.cs.berkeley.edu/projects/Bisc/bisc.memo.html#what_is_sc
- Zavrel, Jakub (1996): Neural Navigation Interfaces for Information Retrieval: Are They More than an Appealing Idea? In: Artificial Intelligence Review 10. S. 477-504.
- Zell, Andreas (1994): Simulation neuronaler Netze. Bonn et al.
- Zimmer, Monika (1998): SOLIS – Sozialwissenschaftliches Literaturinformationssystem. <http://www.bonn.iz-soz.de/information/databases/solis/index.htm>
- Zimmer, Monika (1998a): FORIS - Forschungsinformationssystem Sozialwissenschaften. <http://www.bonn.iz-soz.de/information/databases/foris/index.htm>
- Zimmermann, Hans-Jürgen (Hrsg.)(1995): Datenanalyse: Anwendung von DataEngine mit Fuzzy Technologien und Neuronalen Netzen. Düsseldorf.
- Zimmermann, Hans-Jürgen (Hrsg.)(1996): EUFIT '96. 4th European Congress on Intelligent Techniques and Soft Computing. Aachen, 1996.
- Zimmermann, Hans-Jürgen (Hrsg.)(1997): EUFIT '97. 5th European Congress on Intelligent Techniques and Soft Computing. Aachen, 8.-11.9.1997.
- Zimmermann, Hans-Jürgen (Hrsg.)(1998): EUFIT '98. 6th European Congress on Intelligent Techniques and Soft Computing. Aachen, 8.-10.9.1998.
- Zimmermann, Hans-Jürgen (Hrsg.)(1999): EUFIT '99. 7th European Congress on Intelligent Techniques and Soft Computing. Aachen, 13.-16.9.1999.
- Zimmermann, Harald; Schramm, Volker (Hrsg.)(1998): Knowledge Management und Kommunikationssysteme: Workflow Management, Multimedia, Knowledge Transfer. Proc. 6. Intl. Symposium für Informationswissenschaft. (ISI '98). 3.-7.11.98, Karlsuniversität Prag. Konstanz.