

Hausarbeit zum Thema:

Das BSB Model:

*Ein nichtlineares autoassoziatives
neuronales Netzwerk*

Martin Pyka
martin.pyka@gmx.de
<http://www.martinpyka.de>
5. Fachsemester

Studiengang: Dipl. Informatik mit Text- und Sprachwissenschaft
Münster, den 04.04.05

Inhaltsverzeichnis

I	Einleitung.....	1
II	Autoassoziative Netze.....	2
	II.1 Assoziative Netze im Allgemeinen.....	2
	II.2 Assoziation der Eigenvektoren.....	4
	II.3 Binär- und Bipolarvektoren.....	5
	II.4 Hebbian Learning.....	6
III	Das BSB-Modell von Anderson.....	8
	III.1 Aufbau des BSB-Modells.....	8
	III.2 Die Power Methode.....	10
	III.3 Verschiebungen von Eigenwerten.....	12
	III.4 Training des BSB.....	13
	III.5 Eigenschaften.....	14
	III.6 Fähigkeiten des BSB-Modells.....	17
	III.7 Der Gamma-Wert.....	22
IV	Bedeutung für die Psychologie und Kognitionswissenschaft.....	24
V	Zusammenfassung und Ausblick.....	26

Abbildungsverzeichnis

Abb. II.1.1 Klassen von assoziativen Netzen	2	
Abb. II.3.1 Binärvektoren	5	
Abb. II.3.2 Bipolarvektoren	6	
Abb. II.4.1 Hebb-Regel	7	
Abb. III.1.1 Das BSB-Modell	8	
Abb. III.1.2 Assoziation eines Vektors	10	
Abb. III.5.1 Anzahl der Iterationen zur Assoziation von Vektoren	14	
Abb. III.5.2 Anzahl der nötigen Iterationen zur Zuordnung eines Eingabevektors		15
Abb. III.5.3 Attraktionsbecken zweier Vektoren mit den Eigenwerten 0,4 und 0,22		16
Abb. III.5.4 Komplexeres Assoziationsverhalten	17	
Abb. III.6.1 Raster mit vorgebenen Buchstabenmuster	18	
Abb. III.6.2 Musterergänzung durch das Brain-State-In-A-Box-Modell		18
Abb. III.6.3 Einordnung der Laute	19	
Abb. III.6.4 Das normierte System mit dem Raum der instabilen Zustände		20
Abb. III.6.5 Kategorisierung des BSB-Modell	20	
Abb. III.6.6 Beispiel für Mustervervollständigung	21	
Abb. III.7.1 Anzahl der Attraktoren für verschiedene Verminderungsraten		22

I Einleitung

Das Interesse an assoziativen Speichern ist so alt wie das Gebiet der neuronalen Netze selbst. Sie scheinen in ihrer emergenten Eigenschaft, Muster erkennen bzw. vervollständigen zu können, den kognitiven Fähigkeiten des menschlichen Gehirns am nächsten zu kommen. Eines der vielen Modelle, die diesem Teilgebiet der neuronalen Netze zuzuordnen ist, stellt dabei das Brain-State-In-A-Box Modell dar, eine metaphorische Anspielung an die Fähigkeiten des Gehirns, die im Rahmen dieses Modells durch die Ecken eines Würfels repräsentiert werden. Da das BSB-Modell sozusagen eine Untermenge der assoziativen Netze bildet, nähert sich diese Hausarbeit dem Thema von aussen nach innen. Zunächst werden in Kapitel II assoziative Netze im Allgemeinen vorgestellt. Das BSB-Modell zählt zu den autoassoziativen Netzen, also jenen, die in der Lage sind, vorgegebene und antrainierte Muster wieder zu erkennen und zu vervollständigen. In diesem Zusammenhang wird erklärt, welche Rolle die Eigenvektoren einer Verbindungsmatrix spielen, welchen Unterschied Binär- und Bipolarvektoren machen und wie sich durch das Hebbsche Lernverfahren eine Verbindungsmatrix erstellen lässt.

Im zweiten Teil der Hausarbeit wird auf das BSB-Modell im Detail eingegangen. Die Struktur dieses Netzes sowie die autoassoziativen Fähigkeiten finden dabei im Rahmen der Ausführungen eine besondere Gewichtung. In mehreren Unterkapiteln aufgeteilt finden sich ausserdem theoretische Erläuterungen zu den Optimierungs- und Veränderungsmöglichkeiten des Netzes. Praktische Anwendungsbeispiele des BSB-Modell sollen dabei die Assoziationsfähigkeiten des Netzes demonstrieren, aber auch auf seine Schwächen aufmerksam machen.

Das BSB-Modell stellt keinen Meilenstein in der Weiterentwicklung neuronaler Netze dar, doch hat es seinen festen Platz im Kanon der Netztypen gefunden. Es steht exemplarisch für eine ganze Reihe kognitiver Fähigkeiten, die sich modellhaft am Computer simulieren lassen. Aus diesem Grund sind die folgenden Darlegungen unter dem Aspekt geschrieben worden, die Parallelen zu natürlichen Phänomenen aufzuzeigen und deutlich zu machen, dass sich Erscheinungen oder Verhalten, die wir als komplex oder intelligent einstufen würden, auf typische Netzwerkeigenschaften zurückführen lassen, bei denen einzelne Individuen (in unserem Fall Neuronen) einfachen kommunikativen Regeln folgen und dabei nur einen geringen Teil der Informationsverarbeitung übernehmen.

II Autoassoziative Netze

Bevor das BSB-Modell vorgestellt wird, werden Assoziativspeicher im Allgemeinen näher erläutert.

II.1 Assoziative Netze im Allgemeinen

Vorwärtsgerichtete Neuronale Netze, wie das Perzeptron oder das ADALINE bilden einen Eingabevektor x auf einen Ausgabevektor y ab. Diese Netze heissen vorwärtsgerichtet, da ihre Eingabe ebenenweise weiter verarbeitet und durch sogenannte Ausgabeneuronen die Berechnung abgeschlossen wird. Bei Backpropagation-Netzen kommt als zusätzliche Eigenschaft hinzu, dass die Umgebung des Vektors x auf die Umgebung des Vektors y abbildet. Solche Netze nennt man deshalb auch *stetige Abbildungen*.

Assoziative Netze funktionieren anders. Durch sie soll zwar ein Vektor x auf einen Vektor y abgebildet bzw. assoziiert werden. Doch der Unterschied zu einem stetigen Abbildungsnetz besteht darin, dass auch die Umgebung von x auf y abgebildet wird. Der Vektor \tilde{x} soll also ebenfalls auf y abgebildet werden, wenn \tilde{x} nahe genug an x liegt. In der Praxis kann man also davon sprechen, dass verrauschte Werte einer Idealeingabe mit der korrekten Ausgabe assoziiert werden, was beispielsweise für den Bereich der Bilderkennung wichtig ist.

Assoziative Netze ähneln dabei ihrem biologischen Vorbild mehr, als andere Netzarten, da jedes Neuron nur lokale Informationen zur Problemlösung erhält. Auf Grundlage dieser Assoziativspeicher wurde auch das Hebbsche Lernverfahren (engl.: Hebbian learning) nach biologischen Vorbildern entwickelt, auf das in Kapitel **II.4 Hebbian Learning** näher eingegangen wird.

Es gibt insgesamt drei (nicht-disjunkte) Klassen von assoziativen Netzen¹:

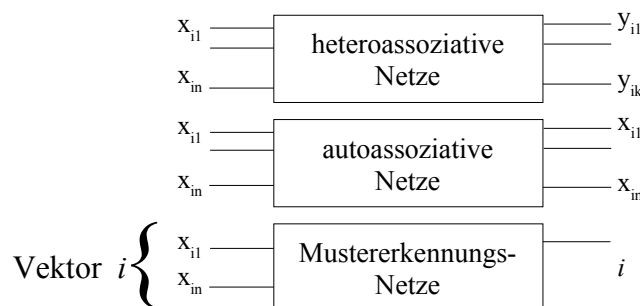


Abb. II.1.1 Klassen von assoziativen Netzen

Den allgemeinsten Typ bilden die *heteroassoziativen Netzen*, die m Eingabevektoren x^1, x^2, \dots, x^m mit m Ausgabevektoren y^1, y^2, \dots, y^m verknüpfen. Die Dimension der Eingabevektoren muss dabei nicht der Dimension der Ausgabevektoren entsprechen. Desweiteren soll ein Vektor \tilde{x} , der nahe genug an einem Vektor x^i liegt, ebenfalls mit dem Vektor y^i assoziiert werden. Dieses Verhalten wird durch ein Trainingsverfahren erreicht. Dabei ist die Zahl der zu lernenden Vektoren - vom Netz abhängig - begrenzt.

¹ Quelle: Theorie der neuronalen Netze, Rojas, S. 252, [ROJ01]

Mustererkennungsnetze sind eine besondere Untergruppe der heteroassoziativen Netze. Sie assoziieren die Eingabevektoren x^1, x^2, \dots, x^m mit den Skalaren $i = 1, 2, \dots, m$. Hier sollen Vektoren, die in der Umgebung eines Vektors x^i liegen, korrekt „benannt“ werden können.

Die *autoassoziativen Netze* sind ebenfalls ein Spezialfall der heteroassoziativen Netze, bei denen die Vektoren x^1, x^2, \dots, x^m mit sich selbst assoziiert werden sollen. Dies entspricht am deutlichsten der Herausforderung, verrauschte Eingaben zu identifizieren und korrekt zu vervollständigen. Sie bilden den Schwerpunkt dieser Hausarbeit und sind der allgemeine Fall des BSB-Modells.

Assoziative Netze werden in der Regel mit einer Schicht von Neuronen implementiert, die, je nach Typ, eine rekursive Rückkopplung erlauben oder nicht. Dabei sind die Leitungen zwischen den Eingabestellen zu den Neuronen gewichtet. Bei n Eingabewerten und k Ausgabewerten ergibt sich damit eine $n \times k$ -Matrix $W = \{w_{ij}\}$. Ist dabei die Aktivierungsfunktion die Identität, heißen die Neuronen *lineare Assoziatoren*. Der Ausgabevektor berechnet sich dann wie folgt:

$$1. \quad xW = y$$

Da man in aller Regel bei assoziativen Netzen m verschiedene n -dimensionale Eingabevektoren x^1, x^2, \dots, x^m mit m Ausgabevektoren y^1, y^2, \dots, y^m der gleichen Dimension assoziieren möchte, lässt sich die obere Gleichung für den allgemeinen Fall erweitern. Sei X die $m \times n$ -Matrix, deren Zeilen die Vektoren x^1, x^2, \dots, x^m bilden und Y die $m \times k$ -Matrix, deren Zeilen y^1, y^2, \dots, y^m entsprechen. Dann wird eine Matrix W gesucht, die folgende Gleichung erfüllt:

$$2. \quad XW = Y$$

Bei autoassoziativen Netzen wird dabei in Gleichung 2 das Y durch ein X ersetzt.

$$3. \quad XW = X$$

Bei Netzen mit Rückkopplung wird die Ausgabe des Netzes auch wieder als Eingabe verwendet. Dabei wird in den meisten Modellen, wie auch im BSB-Modell, vorausgesetzt, dass die Berechnung für jedes Neuron gleichzeitig geschieht. Derartige Netze werden *synchron* genannt. Dabei entsteht eine Folge von x_i Eingabevektoren. Soll das Netz also einen bestimmten, aber verrauschten, Eingabevektor mit seiner Idealform assoziieren, muss irgendwann gelten:

$$4. \quad xW = x$$

wobei x der Zeilenvektor ist. Damit ist x ein Eigenvektor der Matrix W mit dem Eigenwert 1. Derartige neuronale Netze werden *dynamische Systeme erster Ordnung* genannt.

II.2 Assoziation der Eigenvektoren

Am interessantesten für assoziative Speicher sind Gewichtungsmatrizen, die einen Vollsatz an Eigenvektoren besitzen, also bei einer $n \times n$ -Matrix n Eigenvektoren mit je n Eigenwerten haben. Daraus folgt, dass die Eigenvektoren linear unabhängig sind und damit den gesamten Raum linear darstellen können. Im folgenden wird gezeigt, dass ein reiner linearer Assoziator, wie wir ihn bisher kennengelernt haben, in aller Regel einen beliebigen Eingabevektor (unter gewissen Einschränkungen) mit dem Eigenvektor assoziiert, der den grössten Eigenwert hat.

Hierzu setzen wir folgendes voraus. Ohne Beschränkung der Allgemeinheit sei λ_1 der Eigenwert der Matrix W mit dem grössten absoluten Wert. Vorrausgesetzt wird dabei, dass $\lambda_1 > 0$ gilt. Ein n -dimensionaler Vektor a_0 kann als Linearkombination der n Eigenvektoren ausgedrückt werden:

$$5. \quad a_0 = \alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_n x_n$$

wobei angenommen wird, dass x_1, x_2, \dots, x_n die n Eigenvektoren sind und $\alpha_1, \alpha_2, \dots, \alpha_n$ ihre Vielfachheiten. Nach der ersten Iteration ergibt sich daraus:

$$6. \quad \begin{aligned} a_1 &= a_0 W \\ &= (\alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_n x_n) W \\ &= \alpha_1 \lambda_1 x_1 + \alpha_2 \lambda_2 x_2 + \dots + \alpha_n \lambda_n x_n \end{aligned}$$

Führt man weitere Iterationen durch, so potenzieren sich die Eigenwerte in der folgenden Form. Nach t Iterationen erhalten wir den Term:

$$7. \quad a_t = \alpha_1 \lambda_1^t x_1 + \alpha_2 \lambda_2^t x_2 + \dots + \alpha_n \lambda_n^t x_n$$

Das heisst also: Da λ_1 der grösste Eigenwert ist, dominiert er in der Potenzierung, was zur Folge hat, dass ein beliebiger Vektor a_0 in die Richtung des Eigenvektors x_1 gezogen wird, sofern α_1 nicht Null und der Eigenwert λ_1 positiv ist.

Stellt sich also die Frage, wie gross die sogenannten Attraktionsbecken der einzelnen Eigenvektoren sind, also wie die Vektormengen aussehen, die mit den jeweiligen Eigenvektoren assoziiert werden können. Betrachten wir dazu folgende Gewichtungsmatrix:

$$8. \quad W = \begin{Bmatrix} 2 & 0 \\ 0 & 1 \end{Bmatrix}$$

Die Eigenvektoren dieser Matrix lauten $(1, 0)$ und $(0, 1)$ mit den Eigenwerten 2 und 1. Das bedeutet also, dass jeder Anfangsvektor der Form (x_1, x_2) nach t Iterationen die Form $(2^t x_1, x_2)$ annimmt und damit mit dem ersten Eigenvektor assoziiert wird. Das

Attraktionsbecken für den ersten Eigenvektor ist also die Menge aller Vektoren (x_1, x_2) mit $x_1 > 0$. Der zweite Eigenvektor wird assoziiert mit der Menge aller Vektoren $(0, x_2)$ mit $x_2 > 0$. Es lässt sich also leicht sehen, dass die beiden Attraktionsbecken nicht gleichmässig über den Raum verteilt sind und überdies die Berechnung der Eigenvektoren sehr aufwendig ist. Wünschenswert wären also Netzeigenschaften, die es ermöglichen, eine gleichmässige Verteilung der Attraktionsbecken zu erzeugen und darüber hinaus eine schnellere Assoziation erlauben. Dies führt uns zu dem *nichtlinearen System*.

II.3 Binär- und Bipolarvektoren

Einige Assoziativspeichermodelle verwenden *binäre Vektoren*. Sie haben also in jeder Komponente entweder eine 1 oder eine 0. Damit entsprechen die gespeicherten Mustervektoren eines n -dimensionalen neuronalen Netzes jeweils den Ecken des Hyperwürfels, also $(0, 0)$, $(0, 1)$, $(1, 0)$ und $(1, 1)$. Abbildung II.3.1 veranschaulicht dies.

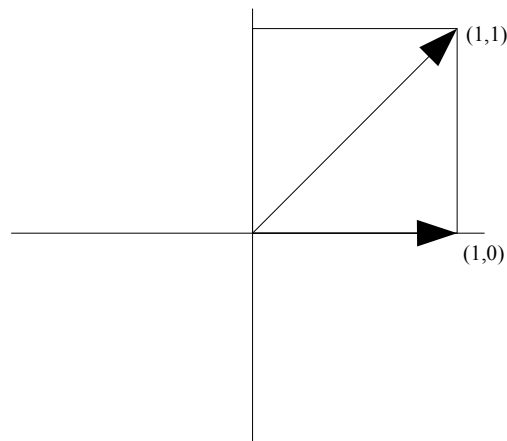


Abb. II.3.1 Binärvektoren

Neben den Binärvektoren gibt es noch eine weitere Möglichkeit, den vorhandenen Eingaberaum besser auszunutzen. Anstatt das Spektrum der Vektoren auf den Bereich 0 und 1 zu begrenzen, werden sogenannte *Bipolarvektoren* verwendet, deren einzelne Werte -1 oder 1 sein können. Beide Vektordarstellungen sind praktisch zueinander äquivalent, da sich durch eine Skalierung und Transformation des binären Eingaberaumes der bipolare erzeugen lässt. Die Ecken des bipolaren Würfels sind demnach $(-1, -1)$, $(1, -1)$, $(-1, 1)$ und $(1, 1)$, wie in Abbildung II.3.2 dargestellt.

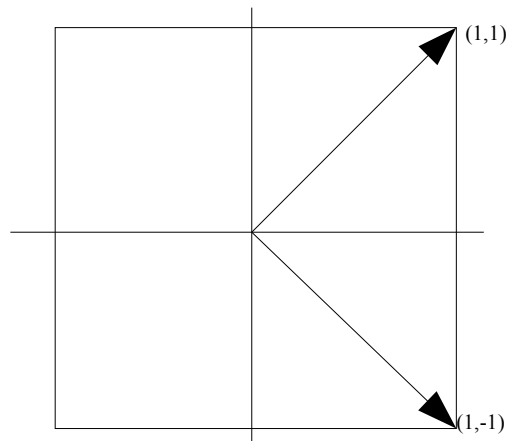


Abb. II.3.2 Bipolarvektoren

Eine bipolare Darstellung hat den Vorteil, dass die Wahrscheinlichkeit, dass zwei Vektoren orthogonal zueinander sind, grösser ist, als in der binären Darstellung, was sich anhand der beiden Abbildungen leicht nachvollziehen lässt. Orthogonale Vektoren können somit leichter gespeichert werden. Darüber hinaus kann als Aktivierungsfunktion für das neuronale Netz die Vorzeichenfunktion $\text{sgn}(x)$ mit dem Schwellenwert 0 verwendet werden.

$$9. \quad \text{sgn}(x) = \begin{cases} 1 & \text{falls } x \geq 0 \\ -1 & \text{falls } x < 0 \end{cases}$$

Damit die Vektoren durch die einzelnen Iterationsschritte die Grenzen des Eingaberaumes nicht verlassen können, werden Limitierungsfunktionen benutzt. In Kapitel III.1 **Aufbau des BSB-Modells** wird die Limitierungsfunktion für das BSB-Modell vorgestellt, weshalb an dieser Stelle eine genauere Behandlung des Themas entfällt.

II.4 Hebbian Learning

Die Hebbsche Lernregel besagt, dass in einem neuronalen Netz die Verbindungen verstärkt werden sollen, die in einem Durchlauf besonders stark genutzt wurden. Sie ist eine der einfachsten Lernregeln für neuronale Netze und wurde von dem Psychologen Donald Olding Hebb eingeführt, der als Vater der kognitiven Psychobiologie angesehen wird. Wir werden den Hebbschen Lernalgorithmus für Assoziativspeicher mit bipolaren Vektoren besprechen, also für Netze, die einen n -dimensionalen bipolaren Vektor x in den n -dimensionalen bipolaren Vektor y abbilden. Nach der Regel sollen also die Gewichte w_{ij} um einen Wert Δw_{ij} erhöht werden, wenn die Eingabe an der Stelle i die Zahl x_i und im selben Iterationsschritt die Ausgabe von Neuron j die Zahl y_j ist. Daher berechnet sich die Gewichtsmatrix wie folgt:

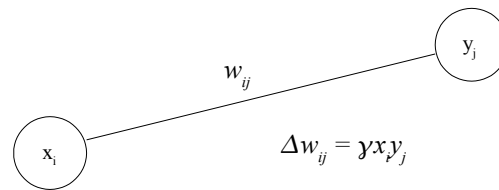


Abb. II.4.1 Hebb-Regel

Die Variable γ ist dabei eine Lernkonstante, die beim Training eines Assoziativspeichers auf 1 gesetzt wird. Als Anfangsmatrix W wird die Nullmatrix verwendet. Wird die Hebb-Regel auf alle Gewichte angewendet, entsteht die sogenannte *Korrelationsmatrix*, die einen Vektor x_1 mit einem Vektor y_1 assoziiert. Sollen nun m n -dimensionale Vektoren x_1, x_2, \dots, x_m auf m n -dimensionale Vektoren y_1, y_2, \dots, y_m abgebildet werden, wendet man die Hebb-Regel auf alle Vektorenpaare an und summiert die daraus entstehenden Korrelationsmatrizen auf.

10.
$$W = W^1 + W^2 + \dots + W^m$$

Dies ist die allgemeine Variante der Hebb-Regel für assoziative Speicher. Das BSB-Modell wandelt diesen Algorithmus leicht ab. Aber die Grundidee bleibt die selbe: man verstärkt die Gewichte zwischen zwei Neuronen, wenn diese untereinander Impulse austauschen sollen.

III Das BSB-Modell von Anderson

Kommen wir nun zum Schwerpunkt dieser Hausarbeit, dem BSB-Modell. In den folgenden Unterkapiteln, wird der Aufbau des BSB-Modells erklärt und in verschiedenen Testreihen untersucht. Des weiteren sind einzelne Unterkapitel wichtigen Details des BSB-Modells gewidmet.

III.1 Aufbau des BSB-Modells

Das Brain-State-In-A-Box (BSB) Modell wurde 1977 von Anderson, Silverstein, Ritz und Jones) eingeführt und ist ein einfaches nichtlineares neuronales Netzwerk, das einen Modellierungsversuch der Assoziationsfähigkeit des menschlichen Gehirns darstellt. Es formalisiert die natürlichen Vorgänge kognitiver Funktionen und mentaler Berechnungen, dessen natürliche Entsprechungen Vorbild sind. So schreibt Anderson:

„In any case, focusing on models for human cognitive computation would seem to be more important than designing yet another network capable of solving X-OR or parity.“ James A. Anderson²

Im Gegensatz zu den meisten klassischen autoassoziativen Netzen, wie zum Beispiel dem Hopfield-Netz, ist das BSB vollständig vernetzt. Das bedeutet:

- zwei Knoten können für beide Übertragungsrichtungen unterschiedlich gewichtete Verbindungen haben
- ein Knoten kann mit sich selbst verbunden sein (direkte Rückkopplung)

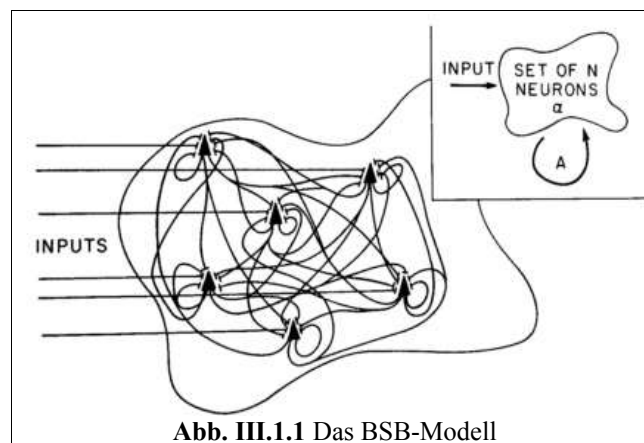


Abb. III.1.1 Das BSB-Modell

Abbildung III.1.1 zeigt eine grafische Darstellung so eines Netzes, deren Verbindungen sich auch in Matrixschreibweise zusammen fassen lassen. Übergibt man diesem Netz eine Eingabe, kommt dies einer Multiplikation mit einem Vektor gleich (siehe dazu auch **II.1 Assoziative Netze im Allgemeinen**, S. 2). Diese rein lineare Berechnung der Ausgabe ist jedoch für den praktischen Gebrauch nicht nutzbar, da neben den gewünschten Endaktivierungen auch fingierte Endzustände eintreten können. Aus diesem Grund führt man eine Limitierungsfunktion ein, die in jeder Dimension den Vektor zwischen den Werten -1 und +1 eingrenzt. Die Eckpunkte des Hyper-Würfels repräsentieren dabei die

2 Vgl. An Introduction to Neuronal Network, S. 494, [AND01]

möglichen Zielzustände, in die ein Eingabevektor durch genügend viele Iterationen überführt werden kann. Die möglichen Zielzustände im dreidimensionalen Fall wären demnach:

11. $(-1, -1, -1), (-1, -1, +1), (-1, +1, -1), (-1, +1, +1),$
 $(+1, -1, -1), (+1, -1, +1), (+1, +1, -1), (+1, +1, +1)$

Allerdings hängt es von der Gewichtsstruktur des Netzes ab, welche Eckpunkte tatsächlich als Zielzustände verwendet werden. Daraus ergeben sich folgende Eigenschaften für das BSB-Modell³:

- Die Menge der zulässigen Eingaben liegt in einem limitierten Kontinuum von $-1 \leq a_i \leq +1$, so dass der durch die zulässigen Eingabevektoren aufgespannte Raum durch -1 und +1 begrenzt ist.
- Die Aktivierung des BSB-Netzes wird über mehrere Aktivierungszyklen kumuliert, so dass unabhängig von der Gewichtung der Verbindungen eine der Ecken des Hyperkubus erreicht wird.
- In jedem Zyklus wird die aktuelle Eingabe limitiert, so dass die Aktivierung nicht über die gesuchte Ecke hinauswachsen kann.
- Der Eingabevektor kann so gewählt werden, dass in jeder Dimension ein Mass für die Bekanntheit und Sicherheit hinsichtlich der Struktur des Zielzustands eingebracht werden kann; so können unter anderem folgende Aktivierungen (und natürlich weitere Abstufungen) gewählt werden:

a_i als Eingabemuster	Initialaktivierung a_i
-1	-1
unbekannt	0
+1	+1

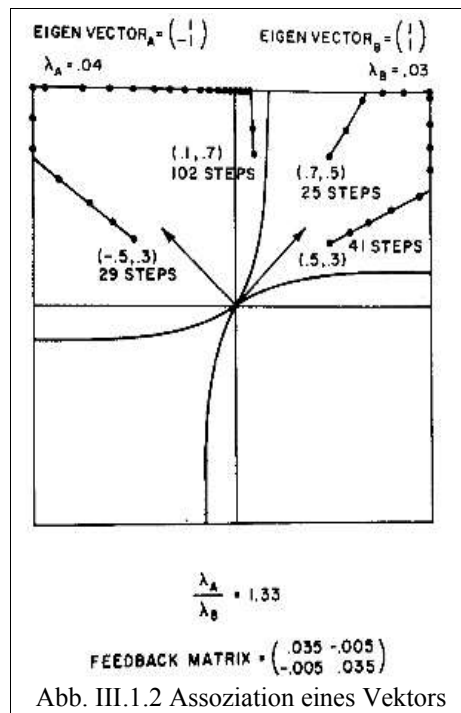
Die bereits angesprochene Limitierungsfunktion hat zum Ziel den Vektor x für den nächsten Zeitschritt $t+1$ zu berechnen und gegebenenfalls auf die Grenzen -1 und +1 zu kürzen. Sie lautet:

12.
$$x(t+1) = \text{LIMIT}(\gamma x(t) + \alpha Ax(t) + \delta f(0))$$

Der Term besteht aus folgenden Komponenten. Das Feedback auf die Eingabe $x(t)$ wird durch $Ax(t)$ berechnet. t ist dabei der Zeitschritt und bei der ersten Eingabe 0. Durch die Konstante α kann dabei die Stärke des Feedbacks gesteuert werden. Der Einfluss des vorhergehenden Vektors wird durch $\gamma x(t)$ berechnet, wobei γ einen konstanten Faktor repräsentiert. In einigen Fällen kann es nützlich sein, den Startvektor $f(0)$ in die Berechnung mit einfließen zu lassen, der mit einer Konstanten δ (meistens 0 oder 1) multipliziert wird.

3 Vgl. Neuronale Netze, S. 129, im Wortlaut ähnlich

Die LIMIT-Funktion kürzt dabei, wenn nötig den Ausgabevektor gemäss der unteren und oberen Grenze. Durch diese Gleichung entsteht ein einfaches dynamisches System, durch das nach einer bestimmten Anzahl an Iterationen der Eingabevektor eine Ecke des Hyperkubus anstrebt. Erreicht wird das, in dem die Feedback-Konstante zunächst positiv ist und damit eine Ausdehnung des Vektors erlaubt. Gelangt dieser an die Grenzen des Hyperkubus, wird die Konstante verringert, bis der Vektor in jeder Dimension einen Grenzwert erreicht hat. Diese möglichen finalen Zustände werden *Attraktoren* genannt.



Der Name Brain-State-In-A-Box leitet sich also aus der konzeptionellen Vorstellung heraus ab, dass die Assoziation einer Eingabe mit einem Eckpunkt einer Form der Wissensrepräsentation gleichkommt, die mit der Funktionsweise des menschlichen Gehirns vergleichbar ist.

III.2 Die Power Methode

Ein Pendant zum englischen Ausdruck Power Method liess sich in der benutzten Literatur und im Internet nicht ausfindig machen. In der deutschen Literatur sowie im deutschsprachigen Internet liess sich der Begriff gar nicht erst finden. Zum leichteren Verständnis wird daher im Rahmen dieser Hausarbeit die direkte Übersetzung *Power Methode* verwendet.

Im Grunde haben wir das Prinzip der Power Methode bereits im Kapitel **II.2 Assoziation der Eigenvektoren** kennen gelernt. Eine Matrix A wird mit einem Eingabevektor $v(0)$ multipliziert. Die Ausgabe $v(1)$ wird wiederum mit A multipliziert. Durch die ständige Wiederholung dieses Vorgangs wird erkennbar, dass sich der anfangs eingegebene Vektor in eine Richtung ganz besonders ausrichtet, nämlich nach dem Eigenvektor mit dem grössten Eigenwert. Wenn e_1, \dots, e_n die Eigenvektoren einer Matrix sind, dann lässt sich v

(0) auch schreiben als $\sum a_i e_i$, wobei a_i die gewichteten Koeffizienten sind. Daraus folgt

$$13. \quad v(1) = Av(0) = A \sum a_i e_i = \sum \lambda_i a_i e_i$$

wobei λ_i der Eigenwert für den Eigenvektor e_i ist. Das Verhältnis zwischen zwei Komponenten i und j des Vektors beträgt demnach

$$14. \quad a_i \lambda_i / a_j \lambda_j$$

Nach n Iterationen wird ersichtlich, dass sich der Faktor λ potenziert:

$$15. \quad (a_i / a_j) (\lambda_i / \lambda_j)^n$$

Da sich in der Praxis unter allen Eigenwerten zwei Eigenwerte λ_i und λ_j finden lassen, für die gilt $|\lambda_i| > |\lambda_j|$, wächst das Verhältnis zwischen diesen beiden Eigenwerten über alle Grenzen. Das heisst also, der Eigenvektor mit dem grössten Eigenwert wird sich bei entsprechend vielen Iterationen am stärksten durchsetzen. Er wird der *dominante* Eigenvektor genannt. Lediglich die Eingabevektoren, deren Koeffizient zum dominanten Eigenvektor Null ist, werden in die Richtung des zweitstärksten Eigenvektors gezogen.

Durch dieses Verfahren lassen sich Eigenwerte für eine beliebige Genauigkeit berechnen. Dazu iteriert man so oft, bis gilt:

$$16. \quad v(t+1) \approx Av(t)$$

Das Verhältnis zwischen $v(t+1)$ und $v(t)$ ist dabei der gesuchte Eigenwert λ . Je mehr Iterationen man vor der Berechnung durchführt, um so genauer lässt sich λ berechnen.

Kommen wir noch einmal auf die Power Methode zurück, mit der ein Eingabevektor mit einem Eigenvektor assoziiert wird. In der klassischen linearen Variante wird dabei immer der Eigenvektor mit dem grössten Eigenwert assoziiert. Die Assoziationsfähigkeiten des Brain-State-In-A-Box Modell lassen es aber zu, dass auch *schwächere* Eigenvektoren assoziiert werden können, sofern der Eingabevektor ausreichend stark in diese Richtung zeigt. Erreicht werden kann dies auf Grund der Tatsache, dass das BSB-Modell nichtlinear ist und damit eine beliebige Ausdehnung eines Vektors nicht stattfinden kann. Der Eingabevektor wird vielmehr in eine Ecke eines Hyperkubus gedrängt. Mit welcher Ecke er dabei genau assoziiert wird, hängt dabei nicht nur vom Feedback der Netzmatrix ab, sondern auch von der immer wieder präsenten Eingabe des Ursprungsvektors und der Addition des Vektors aus dem letzten Assoziationsschritt (siehe auch dazu die Limitierungsfunktion auf Seite 9). Die Erscheinung, dass eine Eingabe mit zuvor antrainierten *Idealbildern* über dieses Verfahren in Verbindung gebracht wird, lässt sich mit den kognitiven Fähigkeiten des Gehirns vergleichen, über die Anderson schreibt:

„We once suggested that such a system when combined with simple Hebbian learning acted like analysis in terms of what are called distinctive features in psychology, criterial

features that are particularly useful in making discriminations“ Anderson ⁴

Die Haupteigenschaft, die sich durch das BSB-Modell also simulieren lässt, ist die emergente Fähigkeit, in einer Eingabe (z.B. Bildinformationen) die wesentlichen Entscheidungsmerkmale zu erkennen und nach diesen das Bild zu vervollständigen oder zu bereinigen. Nur, wenn diese unverkennbaren Eigenschaften („*distinctive features*“) fehlen oder nicht stark genug ausgeprägt sind, werden untergeordnete Attribute entsprechend berücksichtigt und zur Assoziation verwendet.

III.3 Verschiebungen von Eigenwerten

Wie wir bereits in Kapitel II.2 **Assoziation der Eigenvektoren** gesehen haben, lassen sich die Vektoren mit den grössten Eigenwerten am besten assoziieren. Dabei ist es einleuchtend, dass ein Vektor besonders dann gut assoziiert werden kann, wenn sein Eigenwert verhältnismässig grösser ist, als andere Eigenwerte, da er sich in der Potenzierung mit sich selbst am schnellsten absetzt. Der Assoziationsprozess lässt sich also beschleunigen, wenn es möglich ist die Eigenwerte so zu verändern, dass sie in einem starken Verhältnis zu den Eigenwerten stehen, dessen Eigenvektoren nicht assoziiert werden sollen. Eine Möglichkeit, um das zu erreichen, ist, das *Eigenwertespektrum*, also die Menge aller Eigenwerte einer Matrix, zu verschieben. Dies lässt sich ganz einfach erreichen, in dem das Vielfache der Einheitsmatrix I zu der Netzmatrix A addiert oder subtrahiert wird. Die neue Matrix A' berechnet sich deshalb wie folgt:

$$17. \quad A' = A + \gamma I$$

Der Grund, warum dadurch alle Eigenvektoren um γ verschoben werden, ist leicht zu erklären. Für jeden Vektor x gilt $Ix = x$. Wenn nun e ein Eigenvektor zur Matrix A ist mit $Ae = \lambda e$, dann folgt daraus für $A'e$.

$$18. \quad \begin{aligned} A'e &= Ae + \gamma Ie \\ &= \lambda e + \gamma e \\ &= (\lambda + \gamma)e \end{aligned}$$

Das gesamte Eigenwertespektrum wurde also um γ verschoben. Auf diese Weise kann das Verhältnis zwischen den einzelnen Eigenwerten verändert und dadurch ein schnelleres Konvergenzverhalten erreicht werden.

Ein weiteres Problem sind negative Eigenwerte, da sie bewirken, dass mit jeder Iteration der Eigenvektor seine Richtung wechselt (denn die Komponenten des Vektors wechseln mit jeder Iteration ihr Vorzeichen). In dem BSB-Modell, sowie auch in allen anderen neuronalen Netzen, die Feedback zulassen, bleibt der vorhergehende Vektor darum bestehen, was in isolierter Form so aussieht:

$$19. \quad x(t+1) = \gamma x(t) + Ax(t)$$

Diese Formel verschiebt den Eigenwert um $+\gamma$ und vergrössert damit die positiven

⁴ Anderson, S. 503

Eigenwerte bzw. verkleinert die negativen Eigenwerte.

III.4 Training des BSB

In der von mir recherchierten Literatur zum Brain-State-In-A-Box Modell fanden sich nur wenige, überwiegend in Prosa gehaltene, Erläuterungen der Lernverfahren.⁵ Aus diesem Grund werden die verschiedenen Verfahren im Rahmen dieser Hausarbeit kurz und in ihren wichtigsten Eigenschaften vorgestellt.

Wie bereits in Kapitel II erwähnt, lässt sich die Gewichtungsmatrix über die Hebbsche Lernregel erzeugen, wobei dabei ein paar Einschränkungen zu machen sind. Ist $\{x_i\}$ die Menge aller Vektoren, die gelernt werden müssen, so wird die resultierende Netzmatrix bei initialen Anfangsgewichten $w_{ij}(0)=0$ von dem äusseren Produkt der beiden Vektoren aufgespannt.

$$20. \quad W = W^T = \gamma \sum_i x_i (x_i)^T$$

Die Netzmatrix ist symmetrisch, woraus folgt, dass die Eigenvektoren orthogonal zueinander und die Eigenwerte aus \mathbb{R} und Null oder positiv sind. Nach diesem *Lernverfahren* müssen also die zu lernenden Muster nicht nur linear unabhängig, sondern auch orthogonal kodiert sein, um einen stabilen Zustand darstellen zu können.

Ein weiteres Lernverfahren, das für das BSB-Modell verwendet werden kann, ist der Widrow-Hoff-Algorithmus für autoassoziative Netze. Das idealisierte Ziel, das bei diesem Verfahren verfolgt wird, ist, dass alle zu lernenden Eigenvektoren den Eigenwert Eins oder zumindest dicht beieinander liegende Eigenwerte haben. Das hat zur Folge, dass sich das allgemeine Eigenwertespektrum verkleinert, also die Differenz zwischen dem grössten und dem kleinsten Eigenwert geringer wird (gemeint sind dabei auch die Eigenwerte, dessen Eigenvektoren nicht gelernt werden sollen). Die Assoziationsfähigkeit des Brain-State-In-A-Box Modells wird dabei auf Grund der Nichtlinearität kaum beeinträchtigt. Der Widrow-Hoff-Algorithmus sieht im einzelnen wie folgt aus:

$$21. \quad \vec{w}_i^{neu} = \vec{w}_i^{alt} + \eta \cdot \frac{1}{n} \cdot \sum_{k=1}^n \delta_i^{(k)} \cdot \vec{x}^{(k)}$$

$$\delta = |t_i^{(k)} - o_i^{(k)}|$$

Für jedes Neuron innerhalb des Netzes wird die gewünschte Ausgabe folgendermassen berechnet. Die betragsmässige Differenz zwischen der tatsächlichen und der gewünschten Ausgabe ($|t_i^{(k)} - o_i^{(k)}|$) im k -ten Lernschritt wird mit dem anzutrainierenden Eingabevektor $\vec{x}^{(k)}$ multipliziert. Aus der Summe aller Produkte wird der Durchschnitt berechnet und mit einer Lernrate η modifiziert (die im Online-Training zugleich auch die Gewichtung des jeweiligen Lernschrittes repräsentieren kann). Der so erhaltene Verschiebungsvektor wird zum alten Gewichtungsvektor des jeweiligen Neuron hinzuaddiert. Auf diese Weise können wir nach genügend vielen Lernschritten ebenfalls eine geeignete Matrix für das Assoziationsverfahren erhalten.

⁵ Unter anderem in [AND01] und [PAT01]

III.5 Eigenschaften

Kommen wir zu ein paar Eigenschaften und Phänomene, die sich anhand des BSB-Modell zeigen lassen.

Zunächst ist es untersuchenswert, wie gut Vektoren, vor allen Dingen in Grenzbereichen, assoziiert werden und wie gleichmässig die Attraktionsbecken im Raum verteilt sind. Dazu dokumentiert Anderson⁶ einen Versuchsaufbau, der im folgenden wiedergegeben werden soll. Verwendet wird ein 1000-dimensionales Netz, dass durch das Hebbsche Lernverfahren auf 10 Vektoren trainiert wurde, die zunächst aus bipolaren Komponenten (also +1 oder -1) zufällig erstellt und anschliessend normalisiert wurden. Die Eigenwerte reichen von 0,22 bis 0,4 und liegen jeweils um 0,02 von einander entfernt. Bei diesem Lernverfahren wurde gemessen, wie viele Iterationen nötig waren, bis der Vektor ein Attraktionsbecken des BSB-Modells erreichte. Wie man bereits vermutet, konnten natürlich die Vektoren mit grösseren Gewichtungen, also grösseren Eigenwerten, schneller mit einem Attraktor assoziiert werden, als die Vektoren mit kleineren Eigenwerten.

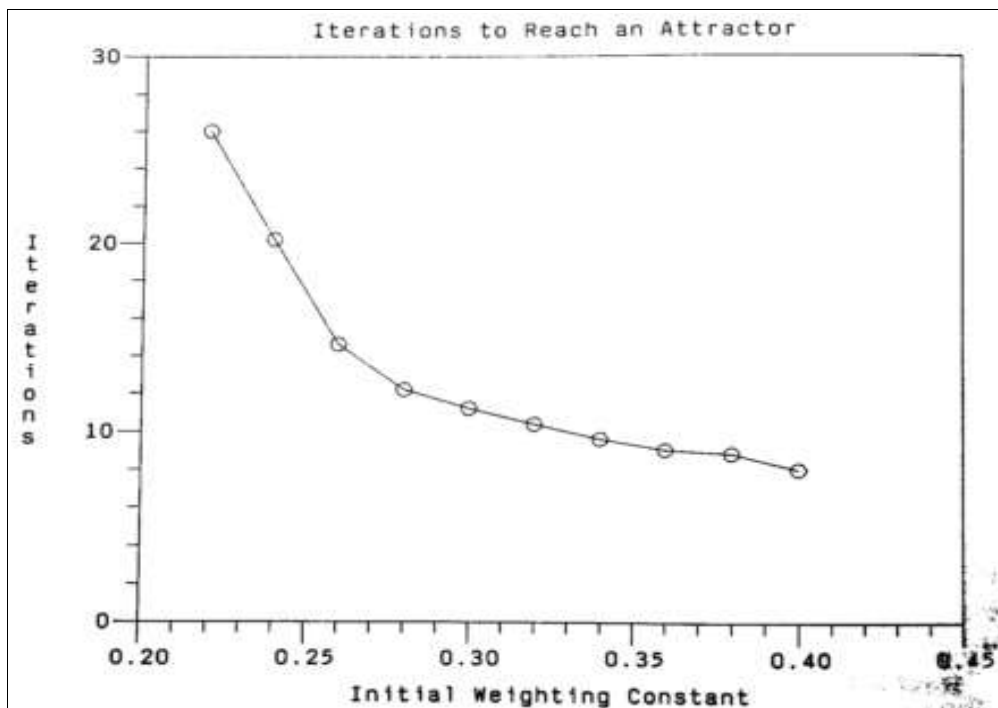


Abb. III.5.1 Anzahl der Iterationen zur Assoziation von Vektoren

Wie man in Abbildung III.5.1 erkennt, sind für die Assoziation eines Vektors mit einer Gewichtung von 0,22 mehr als doppelt so viele Iterationen nötig, wie für einen Vektor mit dem Eigenwert 0,4, wobei die Anzahl der Iterationen exponentiell ansteigt, je kleiner die Eigenwerte werden. In dieser Grafik können wir also generell beobachten, wie gut die Assoziationseigenschaften jedes Vektors im Verhältnis zu den anderen ist. Doch wie gut kann ein beliebiger Vektor in ein bestimmtes Attraktionsbecken geführt werden? Diese Frage lässt sich am besten untersuchen, wenn wir die Komplexität der Simulation auf ein Minimum reduzieren und von möglichst vielen unsicheren Parametern befreien. Um also zu testen, wie gut ein Vektor assoziiert wird, verwenden wir folgenden Versuchsaufbau.

6 An Introduction to Neuronal Networks, S. 507

Die beiden Vektoren mit den Eigenwerten 0,4 und 0,38 bilden eine gedachte Fläche, auf der 100 Positionen bestimmt werden, die zwischen diesen beiden Vektoren liegen und an die ein normalisierter Vektor gelegt wird. Diese 100 Positionen unterscheiden sich also ungefähr um 1 Grad von einander. Für jede dieser Positionen wird nun gemessen, wieviele Iterationen nötig sind, um den Eingabevektor mit einem der beiden Attraktoren zu assoziieren. Das Ergebnis ist in Abbildung III.5.2 veranschaulicht.

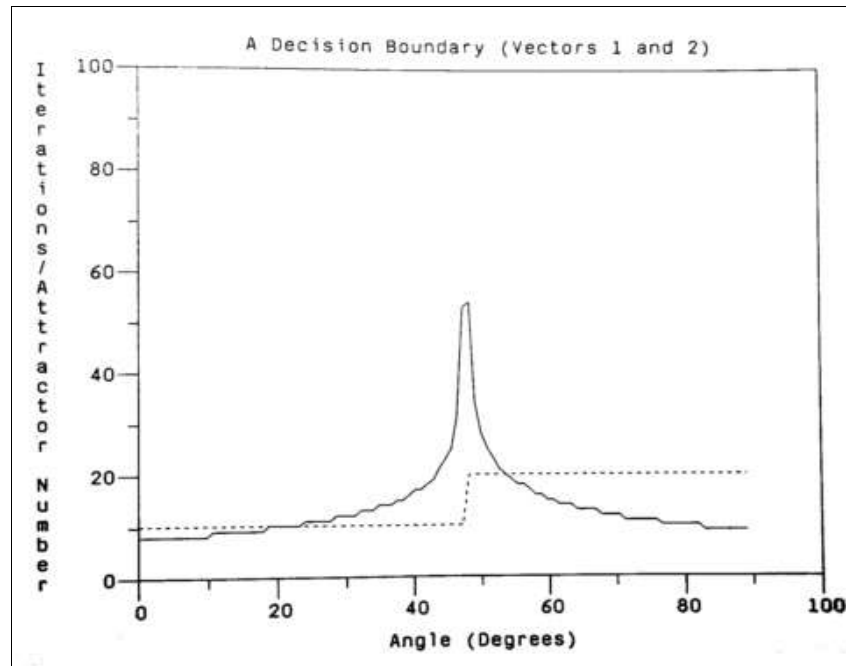


Abb. III.5.2 Anzahl der nötigen Iterationen zur Zuordnung eines Eingabevektors

Die beiden zu assoziierenden Vektoren lagen 90 Grad auseinander. Die durchgezogene Linie zeigt, wieviele Iterationen für die Zuordnung nötig waren. Auffällig ist dabei, dass selbst bei einer Abweichung von 40 Grad der Testvektor noch relativ schnell einem der beiden Vektoren zugeordnet werden konnte. Nur in der Mitte, bei etwa 45 Grad, dauerte die Zuordnung unverhältnismässig lange. Die gestrichelte Linie zeigt dabei, mit welchem Attraktor der Vektor assoziiert wurde⁷. Wenn man genau hinschaut, kann man feststellen, dass der Verlauf der Kurve nicht ganz symmetrisch und vor allen Dingen nicht ganz mittig ist. So liegt das Maximum der Kurve nicht bei 45 Grad, sondern bei etwa 48 Grad. Hier zeigt sich also die Auswirkung der Eigenwerte der Vektoren auf die Assoziation. Je grösser ein Eigenwert im Verhältnis zu einem anderen ist, desto grösser ist auch das Attraktionsbecken des jeweiligen Vektors. Führt man also das selbe Experiment mit zwei Vektoren aus, die die Eigenwerte 0,4 und 0,22 haben, ergibt sich eine wesentlich deutlichere Asymmetrie, wie es in Abbildung III.5.3 zu sehen ist.

⁷ Der Wert auf der y-Achse geteilt durch 10

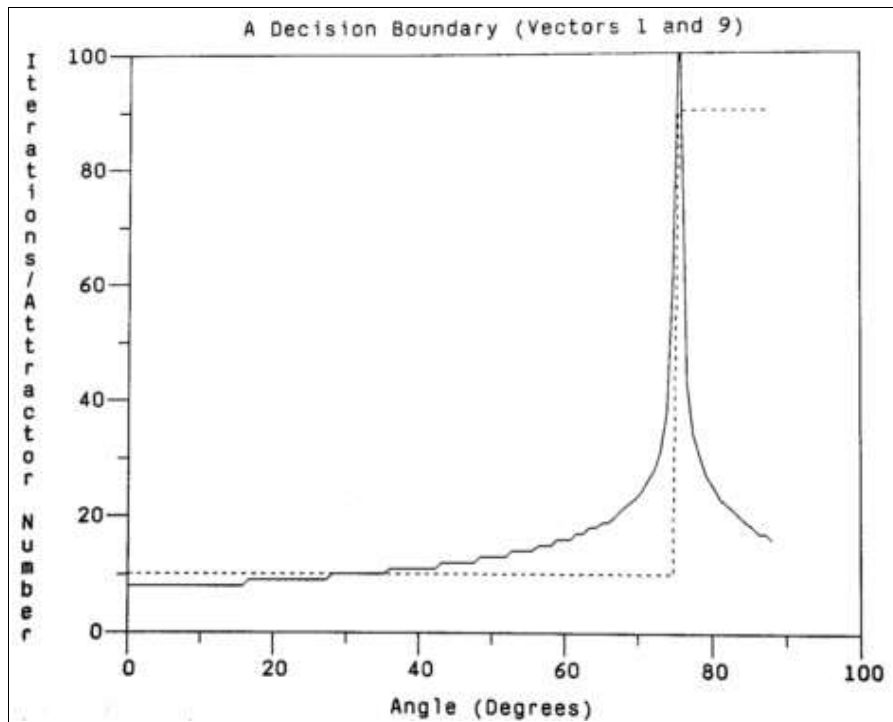


Abb. III.5.3 Attraktionsbecken zweier Vektoren mit den Eigenwerten 0,4 und 0,22

Obwohl Vektor 1 mit einem Eigenwert von 0,4 eine Gewichtung von nur 64,5% hat ($0,4 / (0,4 + 0,22)$) nimmt das dazugehörige Attraktionsbecken einen Raum von etwa 83% ein. Hier zeigt sich deshalb, dass selbst verhältnismässig kleine Differenzen zwischen den Eigenwerten grosse Verschiebungen der Attraktionsbecken mit sich ziehen. So ein Kurvenverlauf muss allerdings nicht immer so gleichmässig sein, wie es in den letzten beiden Abbildungen demonstriert wurde. Abbildung III.5.4 zeigt das etwas komplexere Verhalten des BSB-Modells, wenn es Eingaben zwischen den Vektoren 5 und 9 assoziieren soll, die die Eigenwerte 0,32 und 0,24 haben.

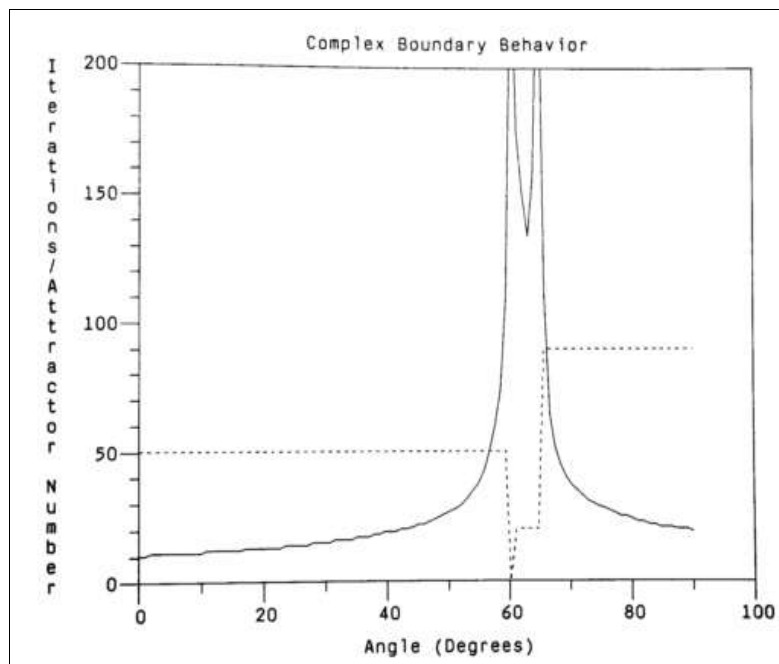


Abb. III.5.4 Komplexeres Assoziationsverhalten

Der Verlauf der Kurve in der Nähe von Vektor 5 und 9 gleicht den Ergebnissen der vorherigen Experimente. Im Bereich der Grenze lässt sich jedoch beobachten, dass die Anzahl der Iterationen plötzlich kleiner wird. Die gestrichelte Linie zeigt dabei an, dass eine Assoziation mit Vektor 3 in den Bereich 61-65 Grad statt fand und bei 60 Grad das BSB-Modell den Vektor mit einem Attraktor assoziierte, der gar nicht antrainiert wurde (Attraktor 0). Anderson schreibt zu diesem Phänomen:

„The BSB system has spurious attractors, although the word spurious is misleading. These attractors are what can give the system some ability to generalize in cognitively oriented simulations.“ Anderson⁸

III.6 Fähigkeiten des BSB-Modells

Im folgenden sollen die Fähigkeiten des BSB-Modells anhand einiger Beispiele demonstriert werden. Insbesondere im Bereich der Bilderkennung bzw. *Bildergänzung* liegen die besonderen Stärken des Modells. Im letzteren Fall dokumentiert Klaus Peter Kratzer in dem Buch „*Neuronale Netze – Grundlagen und Anwendungen*“⁹, wie das BSB-Modell die Struktur von vier gelernten Buchstaben auf einem 6x6-Raster lernt und in der Lage ist, Teile des Bildes zu vervollständigen. Als Ausgangsbasis wurde dabei das Netz auf die Buchstaben A, B, C und D konditioniert.

⁸ „An Introduction“, Anderson, S. 510

⁹ Neuronale Netze, Kratzer, S. 131, [KRA01]

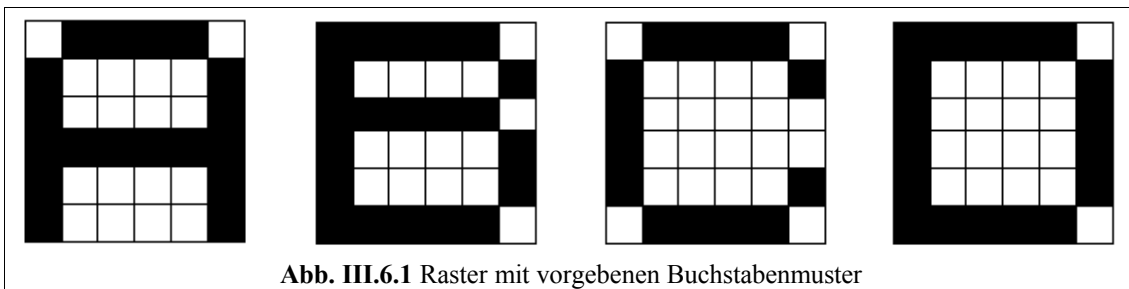
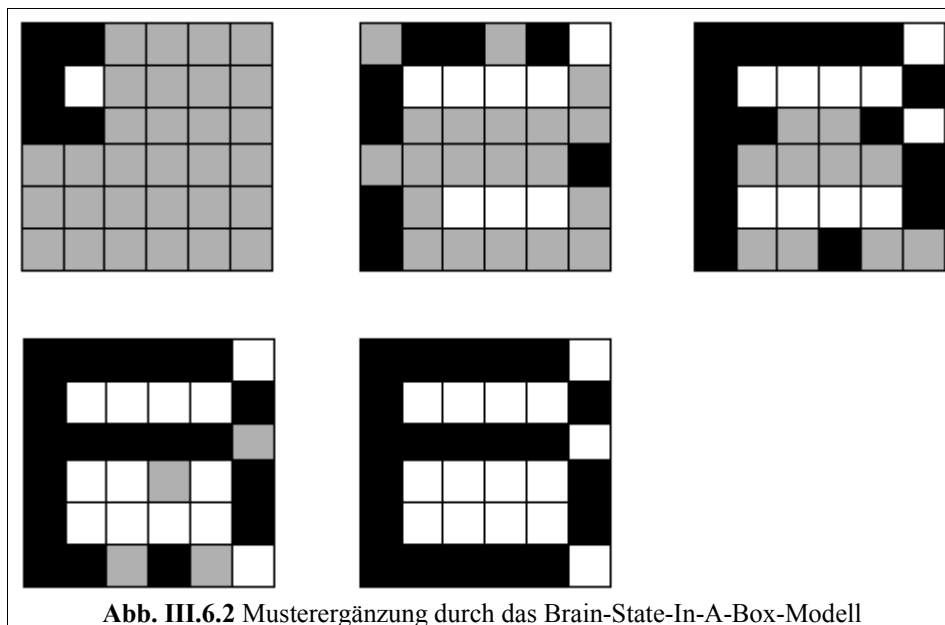


Abbildung III.6.1 zeigt die trainierten Muster. Im Test wurde dem BSB-Modell 1/6 des Musters des Buchstaben B vorgegeben, nämlich die linke obere Ecke. Dass das Netz in der Lage war, innerhalb von nur wenigen Iterationen, den Ausschnitt zur Vervollständigen und das korrekte Bild des Buchstaben B anzuzeigen, ist sicherlich auch darauf zurück zu führen, dass der vorgegebene Ausschnitt ein signifikanter Bereich der trainierten Musters war. In Abbildung III.6.2 sind die einzelnen Iterationsschritte nochmal grafisch verdeutlicht. Indifferente Aktivierungen im Bereich der Null sind in den Grafiken dabei grau hinterlegt. Es ist dabei deutlich zu erkennen, wie schnell das Netz das vorgegebene Muster korrekt vervollständigt, auch wenn es sich partiell manchmal von dem Idealbild entfernen kann. So lässt sich eine minimale Abweichung in der zweiten Grafik erkennen, in der das erste Feld (oben, links) grau hinterlegt ist, obwohl es sowohl bei der Anfangsinitialisierung, als auch beim späteren Endbild wieder schwarz dargestellt wird.



Die Parameter für die Limitierungsfunktion in (III.1) auf Seite 9 waren dabei $\gamma=2$ $\alpha=0,5$ $\delta=2,5$

Interessant bei diesen Parametern ist die Tatsache, dass insbesondere γ und δ stark gewichtet wurden, was bedeutet, dass der Eingabevektor und der jeweils aktuelle Iterationsvektor zur Berechnung des nächsten Netzzustandes eine wichtige Rolle spielen. Dies führt zu der Annahme, dass eine Art „Erinnerungsvermögen“ für die Effizienz des

Assoziationsverhalten des Netzes von hohem Stellenwert ist.

Ein anderer Anwendungsbereich des Brain-State-In-A-Box Modell wird in dem Buch „Neuronale Netze“¹⁰ von Rüdiger Brause angeführt. Hierbei gelang es, eine experimentalpsychologische Beobachtung der „kategorischen Sprachwahrnehmung“ mit Hilfe des BSB-Netzes zu simulieren. Es geht dabei um folgendes Phänomen: Wenn ein Mensch vor die Aufgabe gestellt wird, /b/- und /p/-Laute nach stimmhaft oder stimmlos zu kategorisieren, lässt sich beobachten, dass diese Zuordnung sehr eindeutig, mit nur geringen Übergangszuständen erfolgt. So beschreibt der Artikel ein Experiment, bei dem Versuchspersonen synthetisch hergestellte Laute, die sich langsam und kontinuierlich vom stimmhaften /b/ zum stimmlosen /p/ veränderten, diesen beiden Kategorien zuordnen mussten, wobei Zwischenstufen erlaubt waren. Interessanterweise fiel die Bewertung der Laute in grossen Teilen sehr eindeutig aus. Also entweder wurde ein Laut eindeutig als stimmhaft oder als stimmlos klassifiziert. Nur ganz wenige Laute wurden einer Zwischenregion zugeordnet.

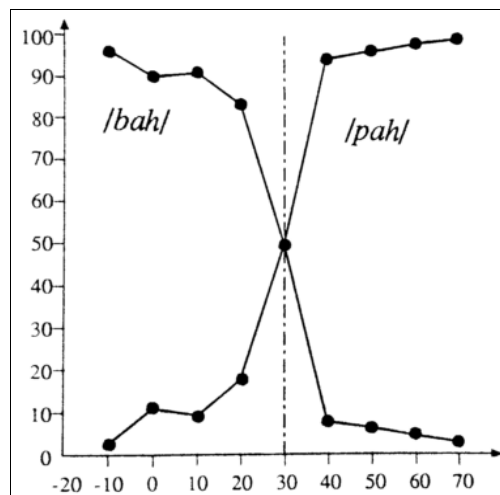


Abb. III.6.3 Einordnung der Laute

Abbildung III.6.3 zeigt eine Auswertung des Versuchs. Die x-Achse gibt dabei die Stimm-Einsatzzeit in Millisekunden an, die y-Achse die prozentuale Zuordnung. Bei der künstlichen Produktion der Laute wurde die Länge des /b/- bzw. /p/-Lautes kontinuierlich linear verändert, was die Vermutung nahe gelegt hätte, dass in der Grafik ebenfalls lineare prozentuale Zuordnungen ersichtlich werden. Statt dessen sind die bereits angesprochenen eindeutigen Zuordnungen sichtbar. Lediglich bei dem Schwellenwert t_0 von 30ms viel den Versuchspersonen eine eindeutige Zuordnung schwer. Rüdiger Brause erklärt dazu folgendes:

„Es wird vermutet, daß in der Sprachverarbeitung zuerst ein vor-kategorischer Speicher der Sprachlaute und danach eine Kategorienbildung (Klassifikation) vorhanden ist. Bei Konsonanten fällt durch die kurze Dauer der Sprachlaute die Kategorisierung besonders klar auf, da die Klassifizierung über typische Frequenzen (Formante) durch die unregelmäßigen Lautformen hier schwer möglich ist“ Rüdiger Brause¹¹

In der Simulation mit dem BSB-Modell wurden zwei 8-dimensionale Vektoren a und b als

¹⁰ Neuronale Netze, Rüdiger Brause, S. 177, [BRA01]

¹¹ „Neuronale Netze“, Rüdiger Brause, S. 177

Sprachmerkmale ausgewählt, und das neuronale Netz so aufgebaut, dass diese beiden Vektoren als Eigenvektoren zwei stabile Ecken im normierten System darstellten. Danach wurde das Netz in einen instabilen Zustand x gebracht, der auf einem Einheitskreis zwischen diesen beiden Vektoren lag, und durch ein Gaußverteiltes Rauschen um eine Zufallsabweichung zusätzlich modifiziert, so dass das Konvergenzziel nicht deterministisch war. Abbildung III.6.4 zeigt einen 2-dimensionalen Schnitt dieses Vektorraumes.

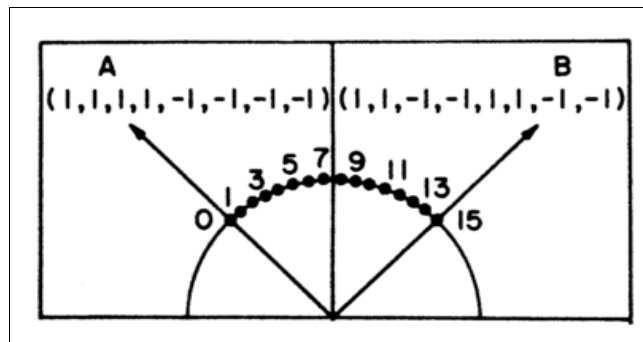


Abb. III.6.4 Das normierte System mit dem Raum der instabilen Zustände

Für die Standardabweichungen von $\sigma=0.1$ bis $\sigma=0.4$ wurden jeweils 100 Iterationsläufe für die Kategorie A durchgeführt, die ja repräsentativ für die Lautzuordnung „stimmhaft“ stand. Wie man in Abbildung III.6.5 sehen kann, sieht das Ergebnis ganz ähnlich zu den Versuchsergebnissen aus, die mit Testpersonen gemacht wurden.

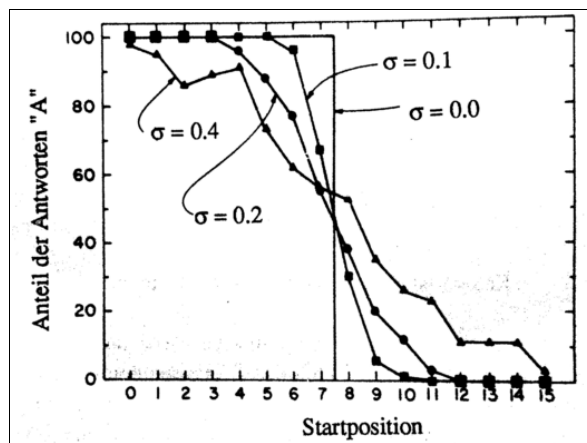


Abb. III.6.5 Kategorisierung des BSB-Modell

Erst bei einer deutlichen Abweichung von 0.4 von dem Idealvektor A gerät das System in Schwanken und hat schon bei den Vektoren 5-9 Schwierigkeiten mit einer eindeutigen Zuordnung. Der Versuch zeigt also, dass die Anwendung des Brain-State-In-A-Box Modell ähnliche emergente Verhaltensweisen zeigen kann, wie die kognitiven Prozesse des Menschen.

Ein weiteres Beispiel für das Assoziationsverhalten eines BSB-Netzes, mit dem antrainierte Muster vervollständigt werden können, stellt Anderson in „An Introduction

to *Neuronal Networks*¹² dar. Einem Netz, das auch 200 Neuronen besteht, wurden mit dem Widrow-Hoff Lernalgorithmus in 150 Durchläufen 15 Vektor-Daten beigebracht. Es handelt sich dabei um US-Staaten und ihre Hauptstädte. Für das Assoziationsexperiment wurden dem Netz lediglich die Staaten *genannt* und gemessen, wie lange es dauerte, bis der korrekte Name der Hauptstadt assoziiert wurde. Abbildung III.6.6 zeigt die Ergebnisse für zwei Vorgaben:

0.	_____	Oklahoma	Limited:	0
3.	_____	Oklahoma	Limited:	64
3.	_____	Oklahoma	Limited:	64
3.	_____	Oklahoma	Limited:	71
4.	_k_____ity__	Oklahoma	Limited:	105
5.	Ok ah___ City__	Oklahoma	Limited:	145
6.	Oklahoma City__	Oklahoma	Limited:	159
7.	Oklahoma City__	Oklahoma	Limited:	180
8.	Oklahoma City	Oklahoma	Limited:	194
9.	Oklahoma City	Oklahoma	Limited:	200
0.	_____	Louisiana	Limited:	0
1.	_____	Louisiana	Limited:	72
2.	_____	Louisiana	Limited:	72
3.	_____ge__	Louisiana	Limited:	76
4.	B__o__o__ge__	Louisiana	Limited:	114
5.	B__on Ro ge ___	Louisiana	Limited:	148
6.	B__ton Rouge ___	Louisiana	Limited:	178
7.	Baton Rouge ___	Louisiana	Limited:	195
8.	Baton Rouge	Louisiana	Limited:	200

Abb. III.6.6 Beispiel für Mustervervollständigung

Der Aktivierungswert eines Neurons liegt bei 0,5. Das heisst also, dass ein Buchstabe genau dann angezeigt wird, wenn alle zugehörigen Neuronen 0,5 überschreiten. Die Zahl hinter „Limited:“ gibt an, wieviele Neuronen bereits ein oberes oder unteres Limit erreicht haben. Sobald also das BSB-Netz die Ecke eines Attraktionsbecken erreicht hat, besitzen alle 200 Neuronen einen eindeutigen Zustand. In dem Test konnten alle 15 Datensätze korrekt vervollständigt werden. Anderson dokumentiert dabei, dass dies bei einem klassischen Neuronalen Netz, das durch den Widrow-Hoff-Algorithmus trainiert wurde bei 10 Datensätzen nur mit einigen fehlerhaften Abweichungen gelang. Dies legt also den Schluss nahe, dass grade die Assoziation und in diesem Zusammenhang die Mustervervollständigung eine Stärke des Brain-State-In-A-Box Modell ist.

Die Lernparameter zeigen, dass auch in diesem Beispiel die *Erinnerung* an die Ausgangsdaten und an den vorhergehenden Zustand eine grosse Rolle bei der Assoziation spielen. Sie lauten $\gamma=0.9, \alpha=0.5, \delta=1.0$. Zur Erinnerung: die Parameter für die Vervollständigung der Rasterbuchstaben (Seite 17), für die allerdings nur 36 Neuronen verwendet wurden, waren $\gamma=2, \alpha=0.5, \delta=2.5$.

¹² „An Introduction to Neuronal Networks“, Anderson

III.7 Der Gamma-Wert

In diesem Abschnitt soll der Einfluss des Parameters γ näher untersucht werden, der in der Gleichung $x(t+1)=\text{LIMIT}(\gamma x(t)+\alpha Ax(t)+\delta f(0))$ verwendet wird. Mit γ wird also skaliert, wie stark der letzte Statusvektor in die Berechnung des neuen Zustandes des BSB-Netzes mit einfließen soll. Liegt dieser Wert zwischen 0 und 1, lässt sich seine Auswirkung dabei wie folgt interpretieren: γ regelt, wie schnell sich der Statusvektor verkleinern soll, wenn es kein Feedback im neuronalen Netz gibt. Angenommen, die Gewichtsmatrix wäre die Nullmatrix, so würde der Statusvektor exponentiell kleiner werden um die Konstante γ . In der praktischen Anwendung des BSB-Modell wird dieser Wert aber aus einem anderen Grund verwendet. Er kann dazu dienen verrauschte Eingaben von ihren Störeinflüssen zu befreien. Durch einen Gamma-Wert der kleiner als 1 ist, wird nämlich die Attraktor-Struktur vereinfacht. Dies kann man sich an folgenden Überlegungen deutlich machen.

Wenn alle positiven Eigenwerte summiert mit γ kleiner 1 sind, so dehnt sich jeder eingegebene Vektor zu keiner der Ecken des Hyperkubus aus. Da mit jeder Iteration die Potenz der Eigenwerte durch γ kleiner wird, nähert sich jeder Vektor dem Ursprungspunkt an. Reale Anwendungen besitzen hingegen eine kleine Anzahl von Eigenvektoren mit grossen Eigenwerten und eine paar Eigenvektoren mit Eigenwerten kleiner 1. Soll das Netz also Vektoren lediglich mit den Ecken eines Hyperkubus assoziieren können, die sich aus den Eigenvektoren mit den grossen Eigenwerten bilden können, kann der Einfluss der Eigenvektoren mit kleinem Eigenwert sehr gut unterdrückt werden, wenn $\gamma < 1$ gewählt wird. Auf diese Weise lässt sich störendes Rauschen in der Eingabe bereinigen.

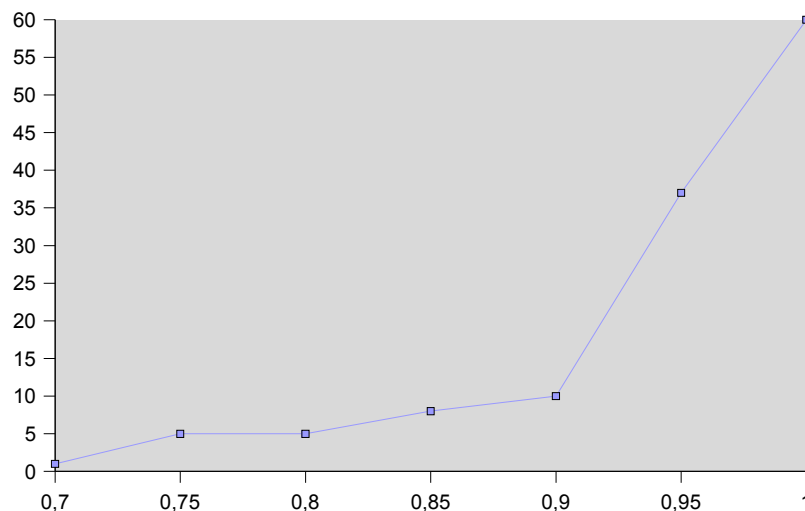


Abb. III.7.1 Anzahl der Attraktoren für verschiedene Verminderungsraten

Abbildung III.7.1 zeigt die Anzahl der verschiedenen Attraktoren in einem BSB-Netz mit 500 Neuronen, das 10 Vektoren assoziieren kann, deren Eigenvektoren Eigenwerte zwischen 0,4 und 0,22 haben. 1000 zufällig ausgewählte Vektoren wurden zum Testen dem neuronalen Netz übergeben und gemessen, wieviele mögliche Attraktoren das Netz sieht. Bei einem Gamma-Wert von 1 wurden 60 verschiedene Attraktoren gemessen. Bei 0,7 hingegen konnten, praktisch alle Vektoren einem Attraktor zugewiesen werden. Wie man der Grafik entnehmen kann, ergab ein Gamma-Wert von 0,9 einen guten Ausschuss an Attraktoren, so dass am Ende die Vektoren 10 Attraktoren zugewiesen werden

konnten. Mit dem Gamma-Wert kann also, je nachdem wie gross die Eigenwerte sind, das Spektrum der möglichen Attraktionsbecken eingeschränkt und damit die Assoziationsgenauigkeit des BSB-Netzes erhöht werden.

IV Bedeutung für die Psychologie und Kognitionswissenschaft

Auch wenn sich der interdisziplinäre Nutzen des BSB-Modell in Grenzen hält und die Kognitionswissenschaft oder die psychologische Untersuchung des menschlichen Denkens nicht direkte Anwendungsgebiete für das BSB-Modell darstellen, so wirft dieses Konzept trotzdem wichtige Fragen über die Funktionsweise unseres Gehirns auf. Insbesondere die Tatsache, dass das BSB-Modell bereits 1977 entwickelt wurde, zeigt, wie früh eigentlich schon die noch junge Neuroinformatik Intelligenz und Künstliche Intelligenz mit emergentem Verhalten in Verbindung brachte, auch wenn zur damaligen Zeit unter KI-Entwicklern die Meinung vorherrschte, Intelligenz liesse sich durch eine breite strukturierte Wissensbasis erzeugen, weshalb noch heute die Sprache Prolog für das automatisierte deduktive Schlussfolgern verwendet wird.

Assoziative neuronale Netze hingegen verfolgen den Ansatz, durch eine Modellierung der biologischen Anatomie des Gehirns scheinbar komplexes und intelligentes Verhalten nachzuahmen und dadurch die Grundfertigkeiten der Intelligenz zu simulieren. Die Grundfertigkeit, die das BSB-Modell dabei zeigt, ist die Vervollständigung antrainierter Muster. Auch wenn die Menge der verschiedenen neuronalen Netzmodelle nach wie vor nur einzelne Fähigkeiten des Gehirn simulieren können, stellt sich für die Neuroinformatik, wie aber auch für die Biologie und die Kognitionswissenschaft die philosophische Frage nach der Materialität von Geist und Intelligenz. Sind die Fähigkeiten des menschlichen Geistes auf transzendente Einflüsse zurück zu führen oder emergente Erscheinungen eines komplexen Netzes am Rande des Chaos? Der amerikanische Philosoph Paul Churchland entwickelte 1997 mit dem Buch „Die Seelenmaschine - Eine philosophische Reise ins Gehirn“ die materialistische Theorie, dass das menschliche Gehirn, seine Persönlichkeit und sein Geist, lediglich emergente Eigenschaften von Synapsenverbindungen sind. Im Rahmen eines Artikels über dieses Buch für Telepolis¹³ erklärt der Journalist Florian Rötzer:

"Nachdem die Neurobiologie das Gehirn als kompliziertes und rückgekoppeltes Netz von chemisch und elektrisch kommunizierenden und sich beeinflussenden Neuronenpopulationen versteht und immer detaillierter beobachten kann, und nachdem sich technisch vereinfachte Netze bauen lassen, die manche der Funktionen von Neuronennetze simulieren können, hat sich die Annahme verdichtet, daß Intelligenz auf einer bestimmten Konfiguration von Verbindungen einfacher Zellen oder Knoten und aus deren Aktivitätsmuster beruht. Aus diesem Grund scheint Intelligenz und alle anderen geistigen Vermögen, einschließlich der Seele und des (Selbst)Bewußtseins, prinzipiell technisch auch auf anderer Grundlage simulierbar oder emulierbar zu sein, wobei sich Neurobiologie und computergestützte Kognitionswissenschaft wechselseitig ergänzen."
Florian Rötzer¹⁴

Neuronale Netze, wie das BSB-Modell, sind darum ein kleiner aber wichtiger Baustein bei der Erforschung realer neuronaler Netze und damit bei der Erforschung der Funktionsweise unseres Gehirn. Eine der grössten Herausforderungen liegt dabei in der Simulation extrem grosser Neuronenpopulationen. Ein menschliches Gehirn besteht ungefähr aus 100 Milliarden Neuronen, die durch 100 Billionen Verbindungen miteinander verknüpft sind. Auch wenn sich neuronale Netze sehr gut Parallelisieren lassen, sind derartig grosse Simulationen noch lange ausserhalb des technisch Machbaren (zumindest

13 <http://www.telepolis.de> - Internetmagazin zum Thema Netz, Kultur und Wissenschaft

14 <http://www.telepolis.de/deutsch/inhalt/buch/2138/1.html> aufgerufen am: 31.10.2004

Bedeutung für die Psychologie und Kognitionswissenschaft
IV Bedeutung für die Psychologie und Kognitionswissenschaft

in vernünftiger Zeit). Auf der anderen Seite ist selbst die Untersuchung kleinerer neuronaler Netze durchaus schwierig und längst nicht so transparent, wie beispielsweise beim BSB-Modell. Dies macht das Verständnis für die inneren Vorgänge schwierig. Letztendlich sind also Simulationen, Testreihen und Experimente mit neuronalen Netzen die Forschungsmethoden der Neuroinformatik. Die Anlehnung an das biologische Vorbild erfordert dabei eine interdisziplinäre Zusammenarbeit zwischen der Informatik, der Biologie, der Kognitionswissenschaft, der Psychologie und vielen anderen Teilbereichen. Die Frage, ob der Mensch nur die Summe seiner Teilchen ist oder ob eventuell mehr hinter unserem Verständnis von Vernunft, Intelligenz und Emotion steckt, kann also von keiner Forschungsdisziplin allein gelöst werden. Simulationen und Modelle, wie das BSB-Modell, stellen deshalb das Zusammenspiel unterschiedlicher wissenschaftlicher Teilbereiche dar, deren Wissen sich gegenseitig bereichert und zu neuen Modellen und Erkenntnissen über die kognitiven Prozesse des menschlichen Gehirns führen kann.

V Zusammenfassung und Ausblick

Auf den voran gegangenen Seiten erfolgte eine Auseinandersetzung mit Autoassoziativen Netzen im Allgemeinen und mit dem BSB-Modell im Speziellen. Das BSB-Modell wurde von Anderson in den 70er Jahren vorgestellt und gehörte damals zu einer ganzen Reihe assoziativer Netze, die eine neue Art der Netz-Modellierung darstellten. Vorrausgegangen war diesen Aktivitäten unter anderem das im Jahr 1969 von Minsky und Papert veröffentlichte Buch „*Perceptrons*“, das eine Sinnkrise der Neuroinformatik mit sich brachte, die erst 1984, nachdem zum dritten Mal der Backpropagation-Algorithmus neu erfunden wurde (in den beiden Entdeckungen davor wurde er offenbar einfach nicht genügend beachtet), die Nutzbarkeit und Praktikabilität neuronaler Netze nachwies. Im Zeitraum von 1969 bis 1984 wurde also überwiegend an assoziativen Netzen (als Ausweg von den bisher bekannten Modellen) geforscht. Ein Höhepunkt auf diesem Gebiet bildet dabei das im Jahre 1982, also nach dem BSB-Modell, vorgestellte Hopfield-Netz von John Hopfield. Ihm gelang die Verbindung zwischen neuronalen Netzen und physikalischen Modellen und erschloss damit die Methoden der theoretischen Physik für die Untersuchung neuronaler Netze. Dabei stellte sich die Beschränkung auf eine speziellere Netzarchitektur als sehr fruchtbar heraus. So wird zum Beispiel die direkte Rückkopplung in Hopfield-Netzen ausgeschlossen. Dem biologischen Vorbild näher ist auch die Tatsache, dass sich das Netz auch asynchron berechnen lässt.

So würde ich, nachdem ich mich eingehend mit dem BSB-Modell beschäftigt habe, resümieren, dass das BSB-Modell interessante Ansätze aufzeigt, wie sich Muster assoziieren können und wie eben diese Fähigkeit im menschlichen Gehirn (vereinfacht) ablaufen. Meinen Recherchen nach zu urteilen scheint das BSB-Modell aber im Kontext seiner anderen Modelle nur eine akademische Relevanz zu besitzen und grade im Schatten des später eingeführten Hopfield-Netzes zu stehen. Nichts desto trotz ist das BSB-Modell ein anschauliches Beispiel für emergentes intelligentes Verhalten, dass auf dem Computer simulierbar ist.

Eine weitere Auseinandersetzung mit diesem Thema könnte zu der Frage führen, welche Vor- und Nachteile andere assoziative Netze im Vergleich zum BSB-Modell haben. Des weiteren wäre es untersuchenswert, was für neuere (also ab 2000) Netz-Modelle zur Bild- oder Mustererkennung entwickelt werden.

Literaturverzeichnis

AND01: Anderson, James A., An Introduction to Neuronal Networks, The MIT Press, 1995

BRA01: Brause, Rüdiger, Neuronale Netze, Teubner, 1995

KRA01: Kratzer, Klaus Peter, Neuronale Netze, Carl Hanser Verlag, 1990

PAT01: Patterson, Dan, Künstliche neuronale Netze, Prentice Hall, 1996

ROJ01: Rojas, Raúl, Theorie der neuronalen Netze, Springer Lehrbuch, 1991

Erklärung der / des Studierenden

Hiermit versichere ich, dass die vorliegende Arbeit selbstständig verfasst worden ist, dass keine anderen Quellen und Hilfsmittel als die angegebenen benutzt worden sind und dass die Stellen der Arbeit, die anderen Werken - auch elektronischen Medien - dem Wortlaut oder Sinn nach entnommen wurden, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht worden sind.

(Unterschrift)