Bericht Nr. 151

Fehlertolerante assoziative Speicherung in neuronalen Netzwerken

Norman Hendrich, Klaus von der Heide

FBI-HH-B-151/91

April 1991

Fachbereich Informatik
Universität Hamburg
Troplowitzstr. 7
2000 Hamburg 54

Abstract

Neural networks are believed to be a new paradigm for parallel and fault-tolerant computations. For example, the extremly simple spinglass-networks of the Hopfield-Gardner type may be used as associative memories. Simple simulations show that these memory networks are fault-tolerant against dilution of synapses.

In this paper the "constant stability" model is presented, which allows for the calculation of the distribution of stabilities as a function of the concentration of damaged synapses given the initial distribution of stabilities in the networks. It is therefore possible to predict the storage properties and the basins of attraction of the networks under the effects of damage.

Additional simulations show the basins of attraction of damaged and diluted memory networks. The basins of attraction are found to be nearly optimal in diluted networks.

Zusammenfassung

Neuronale Netzwerke gelten zunehmend als Paradigma für fehlertolerante massiv-parallele Informationsverarbeitung. So ist mit den extrem einfachen Netzwerken des Hopfield-Gardner Typs fehlertolerante assoziative Speicherung möglich.

In dieser Arbeit wird das „constant-stability" Modell vorgestellt, mit dem ausgehend von der Anfangsverteilung der Stabilitäten im Netzwerk die Berechnung der Verteilung der Stabilitäten als Funktion des Konzentration zerstörter Synapsen möglich ist. Damit können erstmals die Speichereigenschaften und die Einzugsbereiche in zerstörten Netzwerken abgeschätzt werden.

Mit zusätzlichen Simulationen werden die Einzugsbereiche in teilzerstörten und verdünnten Netzwerken untersucht. Die Ergebnisse zeigen, daß schon schwach verdünnte Netzwerke fast optimale Einzugsbereiche besitzen.

Vorwort

Neuronale Netzwerke gelten zunehmend als ein neues Paradigma für fehlertolerante massiv-parallele Informationsverarbeitung. Zumindest einige der faszinierenden Eigenschaften biologischer Nervensysteme lassen sich dabei mit sehr einfachen Modellen nachbilden.

So können Netzwerke des Hopfield-Gardner Typs als assoziative Speicher mit einer hohen Speicherkapazität und beträchtlichen Einzugsbereichen dienen. Schon einfache Simulationen zeigen dabei, daß sich die Netzwerke tatsächlich als fehlertolerant gegenüber der Zerstörung ihrer speichernden Elemente — der Synapsen — erweisen.

In dieser Veröffentlichung wird das „constant-stability"-Modell vorgestellt, mit dem erstmals die Berechnung der statischen Speichereigenschaften von neuronalen Netzwerken unter teilweiser Zerstörung der Synapsen gelingt. Das Modell ermöglicht die Berechnung der Verteilung der Stabilitäten der gespeicherten Muster, so daß auch eine Abschätzung der dynamischen Eigenschaften der teilzertörten Netzwerke gewonnen werden kann.

Die Ergebnisse des „constant-stability"-Modells werden durch Simulationen an teilzerstörten und verdünnten Netzwerken ergänzt, mit denen die dynamischen Eigenschaften verdünnter neuronaler Netzwerke untersucht werden. Insbesondere erweisen sich die Einzugsbereiche der gespeicherten Muster in verdünnten Netzwerken als fast optimal.

Für geringe Konzentration ausgefallener Synapsen sind die Auswirkungen auf die Speichereigenschaften der Netzwerke weit geringer als etwa die Verkürzung der Lernphase. Die für verfügbare Fertigungsprozesse typische Konzentration ausgefallener Transistoren liegt weit unterhalb der von den Netzwerken noch tolerierbaren. Damit wird auch die Realisierung sehr großer derartiger Netzwerke als Wafer-Scale Schaltkreise möglich.

Die in dieser Veröffentlichung vorgestellten Ergebnisse beruhen im wesentlichen auf der Diplomarbeit „Assoziative Speicherung und Lernen im Gardner-Modell — Simulation und Architekturen der technischen Realisierung" von Norman Hendrich.

Norman Hendrich, Klaus von der Heide.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

1.1 Neuronale Modelle und assoziative Speicherung

Trotz großer Anstrengungen ist es der Forschung auf dem Gebiet der künstlichen Intelligenz bisher nur in geringem Umfang gelungen, die faszinierenden Leistungen biologischer Nervensysteme mit Computern nachzuvollziehen.

Zum einen ist es sehr schwierig, aus der Beobachtung des Verhaltens biologischer Systeme auf die zugrundeliegenden Algorithmen zu schließen und diese auf verfügbare Rechner umzusetzen. Außerdem setzt sich immer mehr die Überzeugung durch, daß die Leistungen biologischer Systeme auf massiver Parallelität — schon das Gehirn einer Fliege enthält mehr als 10^7 Nervenzellen — und weniger auf der Verwendung hochkomplexer Algorithmen beruhen.

Es liegt daher nahe, die Eigenschaften von *neuronalen Netzen* zu untersuchen, also Systemen, die Architekturprinzipien biologischer Nervensysteme kopieren. Die verfügbaren, zum Teil wenig präzisen neurophysiologischen Daten deuten allerdings auf ein sehr komplexes Verhalten der einzelnen Nervenzellen, und über das Muster der Verknüpfungen zwischen den Neuronen gibt es nur vereinzelte Angaben.

Es ist deshalb sehr aufwendig und nur für kleine Systeme innerhalb erträglicher Rechenzeiten möglich, Netzwerke aus biologisch auch nur einigermaßen plausiblen Neuronen zu simulieren. Der Vorteil derartiger Modellierung ist, daß die Ergebnisse der Simulationen eventuell mit physiologischen Daten verglichen werden können.

Von größerer Bedeutung ist daher die Untersuchung von wirklich massiv-parallelen Systemen, in denen jedes einzelne Neuron so einfach wie möglich modelliert wird und die Leistung des gesamten Netzwerks als kollektiver Effekt aller Neuronen entsteht. Von besonderem Interesse ist dabei die Beschreibung *assoziativer Speicherung*, die als wichtige Funktion des Cortex im menschlichen Gehirn gilt.

Man findet, daß sich assoziative Speicherung schon mit sehr einfachen neuronalen Netzwerken realisieren läßt. Tatsächlich können die einzelnen Neuronen so modelliert werden, daß die mathematische Formulierung der Netzwerke zur Beschreibung einer bestimmten Klasse magnetischer Festkörper äquivalent ist. Damit können die mächtigen Methoden der statistischen Physik zur Analyse der Modelle benutzt werden.

1.2 Spinglas-Modelle

Die Beschreibung ungeordneter Systeme gehört zu den interessantesten Herausforderungen der modernen Physik. Neben fraktalen und chaotischen Systemen spielen dabei die *Spingläser* eine besondere Rolle. In diesen Modellen für ungeordnete magnetische Festkörper werden die magnetischen Momente der Ionen durch klassische Ising-Spins mit den Werten $S_i = \pm 1$ beschrieben, die über Kopplungen J_{ij} miteinander wechselwirken. Da die Werte der Kopplungen dabei zufällig aus einer Gauß-Verteilung ausgewählt werden, besitzen die Systeme keinen einfachen Grundzustand und erlauben ungewöhnliche dynamische Phänomene.

Ursprünglich zur Erklärung einiger Anomalien der thermodynamischen Eigenschaften spezieller magnetischer Legierungen entwickelt, steigerte sich das Interesse an diesen Modellen beispiellos, als [Hopfield 82] zeigte, daß sie besondere kollektive Eigenschaften entwickeln können.

So ist die zeitdiskrete Dynamik eines Spinglases äquivalent zur Iteration bestimmter paralleler Algorithmen. Gleichzeitig wies Hopfield darauf hin, daß die in den Spinglas-Modellen verwendeten Ising-Spins als abstrakte Modelle von Nervenzellen interpretiert werden können. Tatsächlich ist die Funktion eines Ising-Spins (Summation des lokalen Feldes $h = \sum_{j \neq i} J_{ij} S_j$ und Spin-Flip $S_i(t+1) = -S_i(t)$ wenn $h < 0$) ein, allerdings äußerst abstraktes, Modell der klassischen physiologischen Beschreibung von Neuronen.

Die *auto-assoziative* (inhaltsadressierte) Speicherung läßt sich besonders einfach in Spinglas-Netzwerken realisieren. Der Zustand eines Ising-Modells mit N Spins wird ja gerade durch ein N-Bit Wort $(S_1, \ldots, S_N)$ beschrieben. Die Idee ist, die Grundzustände des Spinglases entsprechend den gewünschten zu speichernden Mustern zu wählen, indem die Kopplungen geeignet eingestellt werden. Unter dem Einfluß seiner Dynamik wird das Spinglas von Zuständen hoher Energie in seine (metastabilen) Grundzustände relaxieren; von den gespeicherten Zuständen etwas abweichende Eingangsdaten $\{S_i\}$ werden unter der Dynamik in die Muster $\{S_f\}$ wandern, Fehler in den Eingangsdaten dabei korrigiert.

Die entscheidenden Parameter zur Charakterisierung eines Spinglases als assoziativer Speicher sind daher die *Speicherkapazität*, also die Anzahl der Muster, die im System gespeichert werden können, sowie die Gestalt und Größe der *Attraktionsgebiete* um die Muster.

Innerhalb einer sehr kurzen Zeitspanne ist es gelungen, die statistische Mechanik des Hopfield-Modells im Rahmen der mean-field Theorie unter Verwendung des Replika-Tricks zu berechnen [Amit *et. al.* 85] [Amit *et. al.* 87]. Dies ist möglich, weil das Hopfield-Modell die Werte der Kopplungen als Funktion der zu speichernden Muster explizit vorschreibt und daher der Hamiltonoperator analytisch bekannt ist.

Man findet, daß ein Hopfield-Netzwerk mit N Neuronen bis zu $P = \alpha_c N$ Muster, mit $\alpha_c \approx 0.138$, fast fehlerfrei speichern kann und daß die Attraktoren beträchtliche Einzugsbereiche besitzen. Zusätzlich existieren aber unerwünschte metastabile Zustände, die die Dynamik des Netzwerks behindern. Durch andere Wahl der Kopplungen gelingt es, die Eigenschaften der Netzwerke zu verbessern. So erreicht das Pseudoinverse-Modell [Personnaz *et. al.* 85] $\alpha_c = 1.0$.

Die exakte Berechnung der maximalen Speicherkapazität eines neuronalen Netzwerks gelang [Gardner 88a] mit der Entwicklung der statistischen Physik im Raum der Synapsen. Für das sogenannte sphärische Modell mit reellwertigen Synapsen ergibt sich $\alpha_c = 2.0$, während das neuronale Netzwerk mit binären Synapsen $J_{ij} = \pm 1$ eine

Speicherkapazität von $\alpha_{cB} \approx 0.83$ erreicht.

1.3 Aufbau und Ziele dieser Arbeit

Für die Abschätzung der Tauglichkeit neuronaler Netzwerke für praktische Anwendungen als assoziative Speicher müssen aber auch die dynamischen Eigenschaften, insbesondere die Größe und Gestalt der Einzugsbereiche untersucht werden. Ein weiterer wichtiger Punkt ist dabei die *Fehlertoleranz* der Netzwerke.

Biologische Nervensysteme erweisen sich als extrem robust gegen den Ausfall einzelner Kopplungen und Neuronen. Dies führt sofort auf die Frage, ob diese Fehlertoleranz sich auch in den neuronalen Modellen widerspiegelt. Das ist gerade auch für Anwendungen interessant: Um die Komplexität elektronischer Schaltkreise weiter steigern zu können, ist es erforderlich, fehlertolerante Designs zu verwenden, damit nicht der Ausfall eines einzelnen Transistors zum Ausfall des gesamten Schaltkreises führt.

In dieser Arbeit werden nach einer Vorstellung der grundlegenden Methoden neuronale Modelle des Hopfield-Gardner Typs als assoziative Speicher im Hinblick auf die dynamischen Eigenschaften und ihre Fehlertoleranz untersucht. Im einzelnen ist die Arbeit folgendermaßen strukturiert:

- In Kapitel 2 werden nach der Definition der benötigten Modelle die Konzepte und Methoden der statistischen Mechanik der Spinglas-Netzwerke in einer knappen Zusammenfassung dargestellt.

  Dieses Kapitel kann damit auch als eine Einführung in die fundamentalen Konzepte der assoziativen Speicherung mit neuronalen Netzwerken dienen.

- Daran schließt sich in Kapitel 3 die Untersuchung der Einzugsbereiche in verdünnten Netzwerken an.

  Nachdem die besonderen Probleme bei der Beschreibung der dynamischen Eigenschaften der Netzwerke und die wichtigsten theoretischen Modelle vorgestellt wurden, werden die Grundlagen der hier präsentierten Simulationen beschrieben. Diese umfassen die Modifikation der iterativen Lernregeln zur Anwendung in verdünnten Netzwerken und die Simulation verschiedener dynamischer Regeln. Dabei werden natürlich die parallele und serielle (asynchrone) Dynamik abgedeckt, zusätzlich aber auch eine Dynamik mit *memory-terms* untersucht. Die Auswirkungen von thermischem Rauschen auf die dynamischen Eigenschaften neuronaler Modelle wurden in vollständig vernetzten Systemen kürzlich von [Nardulli & Pasquariello 90] beschrieben.

- Kapitel 4 beginnt mit einem Überblick über die verschiedenen Modelle für Fehler in neuronalen Modellen. Es zeigt sich, daß die zufällige Zerstörung von Synapsen ein einfaches und sinnvolles Fehlermodell darstellt.

  Daran anschließend wird das „constant-stability“ Modell vorgestellt, mit dem es erstmals gelingt, die Verteilung der Stabilitäten nach der zufälligen Zerstörung von Synapsen zu berechnen und damit die Speichereigenschaften zuverlässig vorherzusagen. Die Einzugsbereiche können dann aus der Verteilung der Stabilitäten im Prinzip mit den in Kapitel 3 vorgestellten Modellen berechnet werden.

- Um Probleme mit den Bezeichnungen zu vermeiden, werden im Anhang A die benötigten Symbole in ihrer hier benutzten Bedeutung erläutert. Außerdem findet sich im Anhang B ein Beweis der Konvergenz der iterativen Lernregel im verdünnten Netzwerk.

2 Theoretische Grundlagen

Neuronale Netze (*neuronale Modelle, neuronale Netzwerke*) sind Systeme zur Informationsverarbeitung, deren Funktion durch das Zusammenspiel einer großen Zahl miteinander verknüpfter Prozessoren, der sogenannten Neuronen, ermöglicht wird. Damit werden einige Architekturprinzipien des (menschlichen) Gehirns kopiert, in der Hoffnung, auch dessen Leistungen zu erreichen.

Einen Überblick über neuronale Netze geben das monumentale Werk der PDP-Research-Group [McClelland & Rumelhardt 86] und die umfassende Sammlung klassischer Originalarbeiten von Anderson und Rosenfeld [Anderson & Rosenfeld 88].

Eine deutschsprachige Einführung in die Theorie der hier untersuchten Spinglas-Netzwerke ist aber bisher nicht erschienen, und deshalb erscheint eine kurze Zusammenfassung grundlegender Resultate und Methoden sinnvoll.

Dieses Kapitel enthält daher neben den grundlegenden Definitionen auch eine Beschreibung der wichtigsten Spinglas-Modelle und ist in folgende Abschnitte gegliedert:

- Zunächst werden in Abschnitt 2.1 der prinzipielle Aufbau und die Funktion biologischer Nervensysteme grob vereinfacht geschildert, um einen Eindruck davon zu vermitteln, welche Bauprinzipien biologischer Nervensysteme in „neuronale“ Modelle übernommen werden.

- Daran schließt sich in Abschnitt 2.2 die allgemeine Definition eines *neuronalen Modells* an. Die hier präsentierte Definition hält sich eng an die von der PDP-Research-Group vorgeschlagene, ist aber in Details etwas formaler und schon auf die Verwendung der Netzwerke als assoziative Speicher ausgerichtet.

- Nach der allgemeinen Definition eines *auto-assoziativen Speichers*, die dem Buch von [Kohonen 84] entnommen ist, werden dann die Begriffe *Speicherkapazität*, *Einzugsbereich* und *Informationskapazität* erklärt. Die Beschreibung des „best-match“ Algorithmus und des von Kohonen vorgeschlagenen linearen Assoziators liefert zwei Beispiele für assoziative Speicher.

- Der folgende Abschnitt 2.4 dient der Vorstellung der Spinglas-Modelle. Die in diesen Modellen zur Beschreibung der Neuronen verwendeten Funktionen sind äquivalent zur Beschreibung bestimmter magnetischer Festkörper, eben den sogenannten Spingläsern. Die Struktur der Wechselwirkungen der Spins führt in diesen Systemen zu einer sehr großen Anzahl metastabiler Grundzustände. Am Beispiel des Hopfield-Modells wird erläutert, wie dies ausgenutzt werden kann, um auto-assoziative Speicherung zu realisieren. Ein einfaches Argument erlaubt eine erste Abschätzung der Speicherkapazität.

- Eine genauere Berechnung der Eigenschaften des Hopfield-Modells gelingt mit den Methoden der statistischen Physik. Trotz der im Detail sehr aufwendigen Mathematik ist das Prinzip der Rechnungen sehr einfach und wird im Abschnitt 2.5 skizziert. Die replika-symmetrische Lösung der mean-field Gleichungen liefert für das Hopfield-Modell eine Speicherkapazität von $\alpha_{cH} =\approx 0.138$. Eine deutliche Verbesserung der Speicherkapazität ermöglicht das ebenso einfache Pseudoinverse-Modell, dessen Definition sich anschließt.

- In Abschnitt 2.6 wird erläutert, wie mit den Methoden der statistischen Physik auch die Berechnung der optimalen Parameter neuronaler Netzwerke ausgeführt werden kann. Dazu erweist sich die Einführung besonderer Parameter, der sogenannten Stabilitäten, als notwendig. Die Berechnung der optimalen Speicherkapazität wird nur für das Spinglas-Modell mit reellwertigen Kopplungen skizziert, kann aber analog auch für Modelle mit binären oder integer-kodierten Kopplungen angewendet werden.

  Die Konstruktion der von den Rechnungen vorausgesetzten Kopplungsmatrizen gelingt mit iterativen Lernregeln, die hier zunächst nur knapp beschrieben werden. Eine gründliche Diskussion der Lernalgorithmen mit dem Beweis der Konvergenz auch in verdünnten Netzwerken findet sich im Anhang B.

2.1 Strukturen biologischer Informationsverarbeitung

In diesem Abschnitt soll ohne jeden Anspruch auf Vollständigkeit der Aufbau von Nervensystemen kurz skizziert werden, um eine Andeutung davon zu vermitteln, welche Strukturen der biologischen Systeme in die mathematischen Modelle neuronaler Netzwerke übernommen werden. Eine Beschreibung der neuroanatomischen und biochemischen Grundlagen von Nervensystemen findet sich etwa in den Büchern von [Alberts *et. al.* 83] und [Katz 87].

Die Nervensysteme aller höheren Tiere sind aus einer großen Anzahl spezieller Zellen, den Neuronen, aufgebaut. Das menschliche Gehirn zum Beispiel enthält etwa 10^{10} dieser Zellen. Nach der klassischen Theorie der Informationsverarbeitung in biologischen Systemen wird jedes Neuron zunächst durch nur einen Parameter, sein *Membranpotential*, beschrieben. Durch aktiven Ionentransport baut jedes Neuron in seinem Zellkörper im Ruhezustand ein Potential von etwa $-70mV$ gegenüber dem Zellzwischenraum auf.

Das Neuron empfängt elektrochemische Impulse von anderen Neuronen an speziellen Strukturen, den *Synapsen*, und leitet diese Signale über einen verästelten Baum von *Dendriten* zum Zellkörper. Die einkommenden Signale verändern das Membranpotential des Neurons. Sobald dieses einen *Schwellwert* von etwa $-50mV$ überschreitet, öffnen sich Ionenkanäle in der Zellmembran und bringen die Zelle kurzzeitig auf ein Potential von etwa $+20mV$. Dieser Ausgangsimpuls von etwa $1msec$ Dauer, das *Aktionspotential*, breitet sich dann entlang einer Fortsetzung des Zellkörpers, dem *Axon*, aus. Ein Axon kann durchaus eine beträchtliche Länge erreichen, bei peripheren Neuronen zum Beispiel bis zu 1m, bevor es sich baumartig verzweigt und Synapsen mit den dort vorhandenen Dendriten anderer Neuronen ausbildet.

Ein einzelnes Neuron kann Synapsen mit 100 bis 100 000 anderen Neuronen besitzen und seine Aktionspotentiale über die Verzweigungen seines Axons an ebenso viele Neuronen vermitteln. Ein Aktionspotential führt an den Synapsen zur Ausschüttung der sogenannten *Neurotransmitter* aus dem Axon. Diese werden an der empfangenden Nervenzelle von großen Eiweißmolekülen, den *Rezeptoren*, aufgenommen. Die Rezeptoren wiederum steuern die Durchlässigkeit von Ionenkanälen im Dendriten und beeinflussen damit das Membranpotential der empfangenden Zelle.

Mindestens zwei Arten von Synapsen müssen in biologischen Systemen unterschieden werden: *Excitorische* wirken erregend, *inhibitorische* hemmend auf die empfangende Zelle. Die Details der Summation der empfangenden Aktionspotentiale in den Dendriten sind allerdings nur sehr schwierig zu messen.

Deshalb wird meistens die diskrete Zeitstruktur der Aktionspotentiale vernachlässigt. Statt dessen wird angenommen, daß das Membranpotential am Zellkörper als Summe der zeitgemittelten Eingangssignale gebildet werden kann.

Die mittlere Aktivität eines Neurons wird dann durch seine *Feuerfrequenz* beschrieben, das ist die Frequenz, mit der das Neuron Aktionspotentiale am Axon erzeugt. Biologische Neuronen sind nicht binär (aktiv oder inaktiv), sondern zeigen über weite Bereiche der Eingangssignale das Verhalten eines Spannungs-Frequenz-Wandlers zwischen einer minimalen *Ruhefrequenz* und einer Maximalfrequenz.

Es ist bisher nicht gelungen, in den Neuronen spezielle Strukturen zu entdecken, die für Informationsspeicherung dienen könnten: Information wird offenbar nicht lokal in den Neuronen gespeichert.

Vielmehr wird die Information in der Stärke der Kopplungen zwischen den Neuronen kodiert — daher auch der Name „konnektivistische Modelle“. Es gibt Hinweise darauf, daß sich die Stärke der Kopplungen beim Lernen durch die Veränderung der Zahl der Rezeptoren in der synaptischen Membran der empfangenden Zelle einstellt.

Grundlegend dafür ist die Hebb'sche Hypothese, daß der Wert der synaptischen Kopplung zwischen zwei Neuronen verstärkt wird, wenn beide Neuronen häufig gleichzeitig aktiv sind. Dies ist plausibel, weil die mittlere Aktivität der Neuronen sehr gering ist — im Gehirn zeigen im Zeitmittel weniger als 0.1% aller Neuronen eine hohe Aktivität. Eine deutlich höhere Aktivität der Neuronen kann nur bei Krankheiten (Epilepsie) beobachtet werden.

Realistische Modelle von biologischen Neuronen, die auch deren Zeitverhalten beachten, bestehen aus Systemen von Integralgleichungen [Heiden 80] und sind mathematisch nur äußerst schwer zu analysieren.

Zudem zeigen einige neuere Befunde, daß Nervenzellen ein noch komplexeres Verhalten besitzen, als nach dem klassischen Modell zu erwarten war. So können z. B. durchaus auch Rückwirkungen von Dendriten auf Axone oder direkte Synapsen zwischen Dendriten beobachtet werden.

2.2 Definition „neuronales Netz“

Neuronale Modelle übernehmen von biologischen Systemen die große Anzahl relativ einfacher Prozessoren, die jeweils nur eine einzige Operation berechnen, dafür aber stark miteinander verknüft sind. Da die Details neuronaler Modelle sich stark unterscheiden können, definieren wir entsprechend [McClelland & Rumelhardt 86](Chapter 2) ein neuronales Netz durch folgende Eigenschaften, wobei allerdings im Hinblick auf die Automatentheorie schon eine diskrete Zeit eingeführt wird:

- Eine Menge von N Prozessoren (den Neuronen). Jedes Neuron wird gekennzeichnet durch die Angabe folgender Funktionen:

- Ein Muster der Verknüpfungen (Synapsen) zwischen den Neuronen, dessen Topologie beschrieben wird durch die *connectivity matrix* C mit $c_{ij} \in \{0, 1\}$. Wenn eine Verbindung von Neuron j nach Neuron i existiert (d. h. $c_{ij} = 1$), gibt der Wert J_{ij} der Kopplungsmatrix J deren Stärke an. Die Wertemenge der Kopplungen ist zunächst beliebig, üblicherweise sind die J_{ij} reellwertig.

- Aktivierungsfunktionen $a_i(t, S, J_i) \in R$, die beschreiben, wie die synaptischen Inputs $c_{ij} J_{ij} \cdot S_j(t)$, die das Neuron i zur Zeit t erreichen, von diesem addiert bzw. integriert werden. Die einfachste, biologisch motivierte Wahl ist $a_i(t) = \sum_{j \neq i} c_{ij} J_{ij} S_j(t)$.

- Ausgangsfunktionen (Übertragungsfunktionen) $f_i(t, a_i)$, die den neuen Zustand $S_i(t+1) \in R$ des Neurons aus dem jeweiligen Wert seiner Aktivierungsfunktion $a_i(t)$ berechnen: $S_i(t+1) = f_i(t)$. Während zur Beschreibung biologisch motivierter Systeme oft reelle Werte für die S_i zugelassen werden, ist in den Spinglas-Modellen die Wahl $S_i \in \{-1, +1\}$ üblich.

- Einen Zustand des Netzes zur Zeit t; das ist der Vektor $\{S\}$ aus den Zuständen $S_i(t) = f_i(t-1)$, $i = 1 \ldots N$.

- Eine Dynamik, die beschreibt, zu welchem Zeitpunkt ein Neuron seine Aktivierungsfunktion berechnet und wann es den Wert seiner Ausgangsfunktion an die anderen Neuronen weitergibt. Verzögerungen bei der Übertragung der Zustände S_i werden nicht modelliert.

- Eine Lernregel, die die Änderung der Werte der Kopplungen (und evtl. der Konnektivität) als Funktion der Zustände des Netzes beschreibt. $J_{ij}(t+1) = J_{ij}(t) + \Delta J_{ij}(t, J_i, S)$. Die Lernregel sollte lokal in Bezug auf die Neuronen sein, J_{ij} darf daher nicht von J_{lj} abhängen.

- Eine „Umwelt“, in der das Netz arbeiten soll. Diese gibt insbesondere Anfangswerte $S_i(t = 0)$ und $J_{ij}(t = 0)$ vor.

Auch nur einigermaßen realistische Modelle für biologische Neuronen erfordern sehr komplexe Funktionen a_i, f_i ..., so daß sich über ganze Netzwerke aus derartigen Neuronen kaum allgemeine Aussagen formulieren lassen. Deshalb werden für die Parameter und Funktionen in relevanten neuronalen Modellen einfache Annahmen gemacht, die mehr oder weniger auch biologisch motiviert sind.

Im Folgenden werden ausschließlich die sogenannten Spinglas-Netze betrachtet. Die Operationen des einzelnen Neurons werden dabei sehr einfach gewählt (insbesondere sind die Funktionen a_i, f_i und die J_{ij} nicht explizit zeitabhängig), und die Auswahl der vom gesamten Netzwerk zu erbringenden Funktion erfolgt nur durch die entsprechende Einstellung der synaptischen Kopplungen. Mit derartigen Modellen wird daher untersucht, wie weit die Funktionen von Nervensystemen erklärt werden können, wenn nur die Synapsen durch Lernen verändert werden.

2.3 Assoziative Speicherung

2.3.1 Definition „assoziative Speicherung"

Ein assoziativer Speicher ist ein System mit Eingangssignalen $\xi = (\xi_1, \xi_2, \ldots, \xi_n) \in A^n$ und Ausgangssignalen $\eta = (\eta_1, \eta_2, \ldots, \eta_n) \in B^n$, wobei die Wertemengen A und B zunächst beliebig (insbesondere $A = R$ oder $A = \{0, 1\}$) sind.

Sei $X = \{\xi^{(1)}, \xi^{(2)}, \ldots, \xi^{(P)}\}$ eine Menge von Eingangsvektoren und $Y = \{\eta^{(1)}, \eta^{(2)}, \ldots, \eta^{(P)}\}$ eine Menge von Ausgangsvektoren. Das System implementiert *assoziative Speicherung*, wenn es folgende Zuordnung ausführt:

$$\xi^{(1)} \longrightarrow \eta^{(1)},\ \xi^{(2)} \longrightarrow \eta^{(2)},\ \ldots,\ \xi^{(P)} \longrightarrow \eta^{(P)}.$$

Ein Spezialfall ist der auto-assoziative Speicher (oder *inhaltsadressierte* Speicher), desssen Aufgabe es ist, zu gestörten oder unvollständigen Eingabedaten das originale Datum zu ermitteln. Zum Beispiel soll der Speicher einen Eintrag wie „S. Kirkpatrick and D. Sherrington, *Phys. Rev.* *B* 17, 4384 (1978)" schon aus stark veränderten Eingaben wie „Kirkpatrick 1978" wiederfinden. Auto-assoziative Speicherung gilt als eine wichtige Funktion des Cortex im menschlichen Gehirn.

Definition 2.1 *[Kohonen 72] An* ideal autoassociative memory *is a system which holds copies of distinct input signal sets $\xi^{(\mu)}$, $\mu = 1, 2, \ldots, P$ in its internal state, and produces the copy of a particular set $x^{(r)} = (\xi_1^{(r)}, \xi_2^{(r)}, \ldots, \xi_n^{(r)})$, $r \in \{1, 2, \ldots, P\}$ to the output, whenever (in the recall mode) the inputs are exited by a set of signals $\xi = (\xi_1, \xi_2, \ldots, \xi_n)$ in which a specified subset of the values ξ_i matches the corresponding subset of the $\xi_i^{(r)}$.*

Die Kapazität des Speichers wird daran gemessen, wie viele verschiedene Daten (Muster) er speichern kann und wie gut unvollständige Eingabemuster zu den am besten passenden gespeicherten Mustern ergänzt werden können.

Für die in dieser Arbeit untersuchten Spinglas-Modelle gilt $A = \{-1, +1\}$, und die Anzahl der Neuronen ist $n = N$. Dies wird für die folgenden Definitionen für die Speicherkapazität und die Einzugsbereiche ausgenutzt:

Definition 2.2 *Die Speicherkapazität eines assoziativen Speichers ist die Anzahl P der Muster (N-bit Worte), die gespeichert werden können. Für Spinglas-Modelle ist es üblich, den Quotienten $\alpha = P/N$ anzugeben.*

Definition 2.3 *Für jedes gespeicherte Muster ξ^μ wird sein Einzugsbereich (oder Attraktionsgebiet) beschrieben durch den mittleren Mindestüberlapp $m_c = N^{-1} \sum_j \xi_j^r \xi_j^\mu$, den Testmuster ξ^r mit diesem Muster haben müssen, um als Ausgangssignal das Muster ξ^μ zu liefern.*

Es ist zu beachten, daß die Speicherkapazität nicht direkt mit der sogenannten *Informationskapazität* des assoziativen Speichers zusammenhängt. Die Informationskapazität ist die Summe über die in allen gespeicherten Mustern enthaltene Information. Für korrelierte binäre Muster, die eine „Magnetisierung“ $a = N^{-1} \sum_j \xi_j \neq 0$ aufweisen (siehe auch Gleichung (18)), gilt

$$I = \frac{N \cdot P}{\ln 2} \cdot \Big[\frac{1}{2}(1+a) \ln\big[\frac{1}{2}(1+a)\big] + \frac{1}{2}(1-a) \ln\big[\frac{1}{2}(1-a)\big] \Big]. \tag{1}$$

2.3.2 Der Algorithmus für auto-assoziative Speicherung

Die Konstruktion eines Algorithmus für auto-assoziative Speicherung ist trivial. Die gewünschten Muster ξ^μ werden in einem Array gespeichert. Ein Eingabemuster $\xi^{(r)}$ wird dann nacheinander mit allen gespeicherten Mustern verglichen und dem gespeicherten Muster mit dem größten Überlapp zugeordnet. Gemäß der Definition 2.3 verfügt dieser Algorithmus daher über maximale Einzugsbereiche der gespeicherten Muster. Die Durchführung dieses Algorithmus erfordert pro zugeordnetem Muster $O(P \cdot N)$ Operationen (Additionen und Multiplikationen, für P Muster der Länge N Bit). Natürlich können die Vergleiche parallel durchgeführt werden. Der Speicherbedarf beträgt ebenfalls $O(P \cdot N)$ Bit.

2.3.3 Lineare Neuronen, der linare Assoziator

Lineare Neuronen werden von Kohonen [Kohonen 84] ausführlich diskutiert. Die Aktivierungsfunktion der Neuronen ist linear, $a_i(t) = \sum_j J_{ij} \cdot s_j$, für die Ausgangsfunktion der Neuronen wird $f_i(a_i) = a_i$ gewählt, und die Dynamik des Netzes ist zeitdiskret und parallel, $S(t+1) = J \cdot S(t)$. Diese Systeme können bis zu N linear unabhängige Muster lernen, wenn die Delta-Lernregel

$$\Delta J_{ij} = \eta\,(t_i - a_i) a_j$$

benutzt wird. Dabei werden die Synapsen so geändert, daß die Differenz zwischen Ausgangssignal a_i und Sollsignalen t_i proportional zu einem Parameter η verkleinert wird.

Eingangsmuster, die eine Linearkombination verschiedener gespeicherter Muster darstellen (etwa $e_i = 0.6 t_i^{(1)} + 0.4 t_i^{(2)}$), führen aber auch im Ausgangssignal zu dieser

Linearkombination: Lineare Modelle können sich nicht für ein Ausgangssignal entscheiden. Daraus resultiert der Mangel der linearen Netze, bestimmte einfache Funktionen (z. B. die XOR-Funktion, Paritätsfunktionen) nicht realisieren zu können.

Deshalb werden in verbesserten Modellen immer Neuronen mit nichtlinearen Ausgangsfunktionen verwendet.

2.4 Spingläser und das Hopfield-Modell

Als Spingläser werden Legierungen bezeichnet, die magnetische Ionen verdünnt und ungeordnet in einer nichtmagnetischen Matrix enthalten, z. B. $Au_{1-x}Fe_x$. Die Wechselwirkung der magnetischen Ionen (Spins) oszilliert als Funktion des Abstandes zwischen ferromagnetischer und antiferromagnetischer Kopplung (RKKY-Wechselwirkung). Da die magnetischen Ionen nicht regelmäßig, sondern zufällig in das Gitter der Matrixsubstanz eingebaut und die Abstände der Ionen daher stochastisch verteilt sind, treten sowohl ferro- als auch antiferromagnetische Kopplungen auf. Die Spins versuchen sich so auszurichten, daß möglichst viele Kopplungen erfüllt sind, es bleiben aber immer einige Kopplungen „frustriert“. In derartigen Systemen gibt es eine große, mit der Zahl der Spins exponentiell wachsende, Anzahl lokaler (im Phasenraum) Minima der Energie. Für Details sei auf den umfangreichen Review von [Binder & Young 86] und das Buch von [Mézard, Parisi & Virasoro 87] verwiesen.

2.4.1 Ising-Spingläser

Die quantenmechanische Behandlung dieser ungeordneten Systeme ist aussichtslos kompliziert, so daß zur theoretischen Beschreibung der Systeme zunächst auf Ising-Modelle mit klassischen Spins $S_i = \pm 1$ vereinfacht wird.

Der Phasenraum eines Systems mit N Spins besteht dann aus den 2^N Zuständen, die den Einstellungen $\{S\} = \{\pm S_1, \pm S_2, \ldots, \pm S_N\}$ entsprechen. Die Energie des Systems ist

$$E = -\frac{1}{2} \sum_{i,j} J_{ij} S_i S_j,$$

wobei J_{ij} die Wechselwirkung von Spin j auf Spin i beschreibt. Im „infinite ranged“ Modell von Sherrington und Kirkpatrick [Kirkpatrick & Sherrington 78] etwa wählt man zufällige Kopplungen J_{ij} mit Mittelwert $\langle J_{ij} \rangle = 0$ und $\langle J_{ij}^2 \rangle = 1/N$.

Zustände lokal minimaler Energie sind solche, deren Energie sich durch Umklappen irgendeines Spins erhöht. Dazu muß jeder Spin parallel zu seinem lokalen Feld stehen: $S_i = \mathrm{sgn}(\sum_j J_{ij} S_j)$. Diese metastabilen Zustände sind daher Fixpunkte unter einer Dynamik, bei der die Spins voneinander unabhängig einzeln umgeklappt werden und werden durch Barrieren höherer Energie voneinander getrennt.

Ein Spinglas kann daher bei Abkühlung $T \to 0$ keinen eindeutigen Grundzustand erreichen, sondern wird abhängig von seiner Vorgeschichte in eines der vielen, fast entarteten Täler der Energie wandern: Bei tiefen Temperaturen wird das System in flachen Minima kurz gefangen, kann tiefere Täler aber nur sehr langsam verlassen. Experimentell findet man deshalb ein ganzes Spektrum von Relaxationszeiten.

Wegen der langen Relaxationszeiten erreichen die Systeme in endlichen Zeiten kein thermisches Gleichgewicht, sie sind nichtergodisch. Die Anwendung der Methoden der Gleichgewichtsthermodynamik und spezieller Rechentricks (Replika-Trick) ist deshalb nicht unproblematisch.

2.4.2 Das Hopfield-Modell

Wenn das Spinglas von einer Anfangskonfiguration $\{S_i\}$ mit hoher Energie ausgehend relaxiert, durchläuft es eine Reihe von Zuständen und setzt sich für $T \to 0$ in einem der benachbarten tiefen lokalen Minima fest, gekennzeichnet durch eine metastabile Endkonfiguration $\{S_f\}$ seiner Spins.

Damit funktioniert das Spinglas wie ein auto-assoziativer Speicher, der die Zustände $\{S_f\}$ speichert und unvollständige Eingabedaten $\{S_i\}$ ergänzen kann. Das Problem besteht jetzt darin, die Kopplungen J_{ij} zwischen den Spins so zu wählen, daß die zu speichernden *Muster* (Konfigurationen) die tiefsten Energien erhalten und als Attraktoren wirken.

Hopfield schlug vor [Hopfield 82], die Hebb-Lernregel [Anderson & Rosenfeld 88] für die Einstellung der Kopplungen zu verwenden, um $P = \alpha \cdot N$ Muster ξ_i^μ, $(i = 1, \ldots, N$, $\mu = 1, \ldots, P)$ zu speichern. Dabei sollen die Muster mittlere Aktivität aufweisen, also die Werte $+1$ und -1 gleich häufig enthalten.

Die Hebb-Regel lautet in diesem Fall für $i \neq j$

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^\mu \xi_j^\mu, \qquad i, j = 1, \ldots, N, \tag{2}$$

und für die Selbstkopplungen wird $J_{ii} = 0$ gefordert.

Als Dynamik der N Neuronen $S_i = \pm 1$ dient ein $T = 0$ Monte-Carlo Prozeß (zum Zeitpunkt $t + i/N$ wird ein Spin i ausgewählt und neu eingestellt):

$$S_i(t + i/N) = \operatorname{sgn}\left(\sum_{j \neq i} J_{ij} S_j\big(t + (i-1)/N\big) \right). \tag{3}$$

Ein ähnliches Modell wurde schon 1974 von [Little 74] vorgeschlagen, allerdings mit einer parallelen (synchronen) Dynamik

$$S(t + 1) = \operatorname{sgn}\left(\sum_{j \neq i} J_{ij} S_j(t) \right), \tag{4}$$

bei der alle Neuronen gleichzeitig neu ausgerichtet werden. Die Wahl der Dynamik zeigt drastische Auswirkungen auf das Verhalten des Modells. Dies wird in Kapitel 3 ausführlich untersucht.

Das vorgestellte Modell beschreibt bisher offenbar ein Spinglas mit einer besonderen Wahl der Kopplungen gemäß (2). Gleichzeitig genügt das System aber vollständig auch der allgemeinen Definition eines neuronalen Modells. Das wird sofort deutlich, wenn für die Aktivierungsfunktion $a_i(t) = \sum_{j \neq i} J_{ij} S_j(t)$, die Ausgangsfunktion $f_i = \mathrm{sgn}(a_i(t))$, sowie für die Lernregel die Vorschrift (2) gewählt wird.

2.4.3 Abschätzung der Speicherkapazität

Eine einfache Abschätzung der lokalen Felder der Neuronen zeigt, warum das Spinglas mit den Kopplungen J_{ij} gemäß (2) als assoziativer Speicher dienen kann: Im Zustand ξ_i^μ wirkt auf das Neuron i das Feld

$$h_{i\mu} = \frac{1}{N} \sum_{j \neq i} \sum_{\nu=1}^{P} \xi_i^\nu \xi_j^\nu \xi_j^\mu = \frac{1}{N} \left((N-1)\xi_i^\mu + \sum_{j \neq i} \sum_{\nu \neq \mu} \xi_i^\nu \xi_j^\nu \xi_j^\mu \right). \tag{5}$$

Der erste Term ist ein Signalterm, der das gewünschte Muster stabilisiert. Der Rauschterm besteht aus unkorrelierten Summanden mit Wert ± 1, kann also durch eine Gauß-Verteilung mit einer Varianz von etwa $\sigma \approx \sqrt{P/N}$ approximiert werden. Das Muster ξ_i^μ wird daher gespeichert sein, wenn der Signalterm den Rauschterm überwiegt, solange also $P \leq \alpha N$ mit $\alpha = O(1)$ oder kleiner ist.

Die Leistung des Systems als assoziativer Speicher hängt dann von folgenden Faktoren ab:

- Der Speicherkapazität, also der maximalen Anzahl $P = \alpha \cdot N$ von Mustern, die sich stabil im Netzwerk speichern lassen.
- Der Größe der Einzugsbereiche um die gespeicherten Zustände.
- Der Zahl und den Eigenschaften von zusätzlichen, dynamisch stabilen aber unerwünschten Zuständen. („spurious states“)

Mit Simulationen konnte Hopfield zeigen, daß das System mit der Hebb-Lernregel bis zu $\alpha \approx 0.15$ Muster sicher und mit beträchtlichen Einzugsbereichen speichern (mit weniger als 1% Fehlern in den Mustern) und rückrufen kann. Oberhalb von $\alpha \approx 0.15$ gelingt es nicht mehr, die Muster zu speichern.

2.5 Statistische Mechanik des Hopfield-Modells

Die einfache Gestalt des Hamiltonoperators — durch die Hebb-Lernregel lassen sich die J_{ij} direkt angeben — gestattet es, für das Hopfield-Modell auch analytische Aussagen über die Struktur der gespeicherten Zustände zu gewinnen. Bahnbrechend waren dazu die Arbeiten von Amit, Gutfreund und Sompolinsky [Amit *et. al.* 85], [Amit *et. al.* 87], in denen das Hopfield-Modell für endliche Zahl von gespeicherten Mustern, sowie für endliches α im Rahmen der mean-field Theorie mit dem Replika-Trick analysiert wird.

2.5.1 Die Lösung des Hopfield-Modells

Das Hopfield-Modell wird durch den Hamiltonoperator

$$H = -\frac{1}{2}\sum_{i,j}\left(\frac{1}{N}\sum_{\mu}\xi_i^{\mu}\xi_j^{\mu}\right)S_iS_j, \tag{6}$$

mit den ξ_i^{μ} als unabhängigen, unkorrelierten ($\langle\xi_i^{\mu}\rangle_i = 0$) Zufallsvariablen beschrieben.

Der Hamilton-Operator läßt sich auch für endliche Temperaturen $\beta = 1/T$ verwenden (äquivalent einem Rauschen in den Ausgangssignalen der Neuronen). Die Verteilung der Konfigurationen $\{S\}$ relaxiert in diesem Fall gegen eine Gibbs-Verteilung

$$P\{S\} \propto \exp\Big(-\beta H\{S\}\Big). \tag{7}$$

Die Stabilität einer Konfiguration gegen alle „single spin flips" ist für $T > 0$ natürlich nicht mehr für ihre dynamische Stabilität ausreichend.

Da die Kopplungen symmetrisch sind ($J_{ij} = J_{ji}$), ist die Hamilton-Funktion gleichzeitig eine Ljapunovfunktion für die Energie: Während der Zeitentwicklung nimmt die Energie des Systems ständig (für $T > 0$ wenigstens im Mittel) ab, bis ein (tiefer) metastabiler Zustand erreicht ist.

Die freie Energie pro Spin kann mit dem Replika-Trick berechnet werden,

$$f = \lim_{n\to 0}\lim_{N\to\infty}\frac{-1}{\beta n N}\Big(\langle\langle Z^n\rangle\rangle - 1\Big). \tag{8}$$

Die Schreibweise $\langle\langle\ldots\rangle\rangle$ steht für die Mittelung über die Verteilung der $\{\xi_i^{\mu}\}$, die als *quenched average* zu verstehen ist: die Mittelung darf nicht in der Zustandssumme ausgeführt werden, sondern nur in der freien Energie, da die ξ_i^{μ} keine dynamischen Variable sind.

Die Berechnung von

$$\langle\langle Z^n\rangle\rangle = \Big\langle\Big\langle \mathrm{Tr}_{S^{\rho}}\exp\Big[\frac{\beta}{2N}\sum_{ij\mu\rho}(\xi_i^{\mu}S_i^{\rho})(\xi_j^{\mu}S_j^{\rho}) - \frac{1}{2}\beta p n + \beta\sum_{\nu}h^{\nu}\sum_{i\rho}\xi_i^{\nu}S_i^{\rho}\Big]\Big\rangle\Big\rangle \tag{9}$$

gelingt mit Gauß-Transformationen zur Entkopplung der Neuronen S_i. Es werden die Ordnungsparameter $m_{\rho}^{\mu} = N^{-1}\langle\langle\sum_i\xi_i^{\mu}\langle S_i^{\rho}\rangle\,\rangle\rangle$, der Edwards-Anderson Parameter $q_{\rho\sigma} = \langle\langle N^{-1}\sum_i\langle S_i^{\rho}\rangle\langle S_i^{\sigma}\rangle\,\rangle\rangle$ und Lagrange-Multiplikatoren $r_{\rho\sigma} = \alpha^{-1}\sum_{\mu=s+1}^{\alpha N}\langle\langle m_{\rho}^{\mu}m_{\sigma}^{\mu}\rangle\rangle$ eingeführt. Das Integral in (9) kann dann im Limes $N\to\infty$ mit einer Sattelpunktsintegration berechnet werden. Der replika-symmetrische Ansatz ($m_{\rho}^{\mu} = m^{\mu}$, $q_{\rho\sigma} = q$ und $r_{\rho\sigma} = r$) führt dann auf folgende Gleichungen für die Ordnungsparameter [Amit *et. al.* 87],

$$\begin{aligned} m^{\mu} &= \int\frac{\mathrm{d}z}{\sqrt{2\pi}}\mathrm{e}^{-\frac{1}{2}z^2}\ \langle\langle\xi^{\mu}\tanh\beta\big[\sqrt{\alpha r}z + (\vec{m}+\vec{h})\cdot\vec{\xi}\big]\rangle\rangle \\ q &= \int\frac{\mathrm{d}z}{\sqrt{2\pi}}\mathrm{e}^{-\frac{1}{2}z^2}\ \langle\langle\tanh^2\beta\big[\sqrt{\alpha r}z + (\vec{m}+\vec{h})\cdot\vec{\xi}\big]\rangle\rangle \\ r &= q/(1-\beta+\beta q)^2. \end{aligned} \tag{10}$$

Die Lösung dieser Gleichungen liefert unter anderem die folgenden wichtigen Ergebnisse:

Für $\alpha < \alpha_{c_H} = 0.138$ sind die Sollmuster stark mit bestimmten Energieminima des Systems, den sogenannten *retrieval states*, korreliert, es gilt $m(\alpha_{c_H}) > 0.967$ und $m(\alpha \approx 0) = 1.0$. Unterhalb von $\alpha = 0.051$ sind diese Zustände die globalen Minima der Energie. Bei $\alpha = \alpha_{c_H}$ verschwindet der Überlapp der entsprechenden Zustände mit den Sollmustern diskontinuierlich (auf $m = 0$), oberhalb von $\alpha > 0.138$ ist also keine Speicherung möglich.

Zusätzlich existieren für alle Werte von α mit den Sollmustern unkorrelierte Spinglas-Zustände, die für $\alpha > 0.051$ die Rolle der globalen Minima von E übernehmen.

Die replika-symmetrische Lösung der Gleichungen ist nicht stabil, der Effekt der Symmetriebrechung ist aber nur klein. Die Ergebnisse ändern sich qualitativ nicht, wenn die Symmetriebrechung berücksichtigt wird, aber die kritische Speicherkapazität verschiebt sich auf etwa $\alpha_c' = 0.145$. Dies stimmt gut mit der aus Simulationen des Hopfield-Modells ermittelten Speicherkapazität überein.

Die Ordnungsparametergleichungen können auch für endliche Temperatur $T > 0$ gelöst werden. Damit läßt sich das α-T Phasendiagramm des Hopfield-Modells angeben. Assoziative Speicherung ist für alle Temperaturen $T < 1$ möglich, darüber gehen sowohl die Spinglas- als auch die Retrieval-Zustände in paramagnetische (zufällige) Zustände über.

2.5.2 Das Pseudoinverse-Modell

Da die zu speichernden Muster ξ_i^μ Fixpunkte der Dynamik sein müssen, liegt es nahe, die Lösungen des Gleichungssystems

$$\xi_i^\mu = \sum_j J_{ij} \xi_j^\mu \tag{11}$$

zu untersuchen [Personnaz *et. al.* 85]. Falls die Muster linear unabhängig sind, ist die optimale Lösung eine Projektion auf den durch die Muster im Phasenraum aufgespannten Unterraum:

$$J_{ij} = 1/N \sum_{\mu,\nu}^{P} \xi_i^\mu c_{\mu\nu}^{-1} \xi_j^\nu, \tag{12}$$

wobei $c_{\mu\nu}$ die *Korrelationsmatrix* der Muster

$$c_{\mu\nu} = 1/N \sum_i^N \xi_i^\mu \xi_i^\nu, \qquad \mu, \nu = 1, \ldots, P \tag{13}$$

und $c_{\mu\nu}^{-1}$ ihre Moore-Penrose Pseudoinverse ist. Für unkorrelierte Muster $c_{\mu\nu} = \delta_{\mu\nu}$ ergibt sich daraus die von Hopfield benutzte Hebb-Lernregel $J_{ij} = 1/N \sum_\mu \xi_i^\mu \xi_j^\mu$.

In diesem Modell sind alle Muster für $\alpha \leq 1$ stabil gespeichert, die Radien der Einzugsbereiche verschwinden allerdings für $\alpha \geq 0.5$. Eine weitere Verbesserung ist daher die Aufhebung der Selbstkopplungen der Neuronen, das heißt die Verwendung von Kopplungen K_{ij} gemäß $K_{ij} = J_{ij} - \alpha\delta_{ij}$, mit denen assoziativer Rückruf für alle $\alpha < 1$ möglich ist [Kanter & Sompolinsky 87].

2.6 Das Gardner-Modell: Statistische Mechanik der Synapsen

2.6.1 Berechnung der optimalen Speicherkapazität

Neuronale Netzwerke können auch untersucht werden, ohne daß eine besondere Einstellung der Synapsen J_{ij} vorgegeben wird. Damit ist die Berechnung optimaler Eigenschaften neuronaler Netze möglich.

Seien etwa P Sollmuster ξ_i^μ vorgegeben. Gibt es dann eine Einstellung der J_{ij}, so daß die Muster gespeichert werden können? Um Einzugsbereiche um die Muster zu garantieren, wird eine strengere Bedingung für die Speicherung der Muster eingeführt: Jeder Spin soll mit einer gewissen *Mindeststabilität* κ parallel zum lokalen Feld stehen:

$$\kappa_{i\mu} = \xi_i^\mu \cdot h_{i\mu} = \xi_i^\mu \cdot \left(N^{-1/2} \sum_j J_{ij} \xi_j^\mu \right) \geq \kappa > 0, \tag{14}$$

für alle Muster und Neuronen $\mu = 1, \ldots, P$, $i = 1, \ldots, N$. Obwohl die parallele Dynamik des Netzwerks gemäß (4) invariant unter einer Skalierung der J_{ij} ist, muß eine Normierung der Synapsen gefordert werden, um eindeutige Werte für die Stabilitäten $\kappa_{i\mu}$ der Muster zu erhalten. Üblich ist die Verwendung der sphärischen Norm $\sum_j J_{ij}^2 = N$.

Gardner [Gardner 87] konnte dieses Problem 1987 lösen. Dabei werden die synaptischen Kopplungen als dynamische Variable betrachtet, nicht aber die Spins S_i. Alle Einstellungen der Synapsen, die die vorgegebenen Muster speichern, bilden ein zusammenhängendes Volumen V_T im Raum der J_{ij}, für das sich einfach ein analytischer Ausdruck angeben läßt. Die Muster sind genau dann im Netzwerk an allen Neuronen S_i stabil gespeichert, wenn die Größe

$$\prod_{\mu,i} \Theta(\xi_i^\mu \cdot h_{i\mu} - \kappa) = 1 \tag{15}$$

ist. Also gilt

$$V_T = \frac{\int \Pi_{i \neq j} \mathrm{d}J_{ij} \; \Pi_{\mu,i} \; \Theta(\xi_i^\mu \cdot h_{i\mu} - \kappa) \; \delta(\sum_j J_{ij}^2 - N)}{\int \Pi_{i \neq j} \mathrm{d}J_{ij} \; \Pi_i \; \delta(\sum_j J_{ij}^2 - N)}. \tag{16}$$

Falls dieses Volumen sich auf einen Punkt zusammenzieht, wird die Einstellung der Synapsen eindeutig. Diese Einstellung entspricht der optimalen Einstellung der Synapsen und damit auch der maximalen Speicherkapazität. Die sehr aufwendige Berechnung des Volumens gelingt mit dem Replika-Trick.

Man findet, daß die optimale Speicherkapazität α_c eines Spinglas-Modells als Funktion von κ für unkorrelierte Muster beschrieben wird durch das Integral

$$\alpha_c(\kappa) = \left[\frac{1}{\sqrt{2\pi}} \int_{-\kappa}^{\infty} \mathrm{d}t \; \mathrm{e}^{-t^2/2} \, (t+\kappa)^2 \right]^{-1} . \tag{17}$$

Die maximale Speicherkapazität eines neuronalen Netzes ist also $\alpha_c(0) = 2$. Diese Abhängigkeit der Speicherkapazität von der Stabilität bleibt auch in verdünnten Netzwerken (oder Netzwerken mit high-order Wechselwirkungen) bestehen, wenn α als Verhältnis der Zahl der Muster zur Zahl der Synapsen pro Neuron definiert wird, $\alpha = P/C$.

Die Rechnungen lassen sich auch für korrelierte Muster $\langle \xi_i^\mu \rangle_i \neq 0$ durchführen [Gardner 88a]. Dazu werden die Muster entsprechend einer „Magnetisierung“ a aus einer Verteilung

$$P(\xi_i^\mu) = \frac{1}{2}(1+a)\delta(\xi_i^\mu - 1) + \frac{1}{2}(1-a)\delta(\xi_i^\mu + 1) \tag{18}$$

ausgewählt. Dies entspricht einer Korrelation $\langle \xi_i \xi_j \rangle = \delta_{ij} + a^2(1-\delta_{ij})$. Die Speicherkapazität ist für $a \neq 0$ immer größer als für unkorrelierte Muster. Für $\kappa = 0$ und $a \ll 1$ gilt

$$\alpha_c(a) = 2(1 + \frac{2}{\pi}a^2 + O(a^4)),$$

und für $\kappa = 0$ und $a \to 1$ ergibt sich sogar

$$\alpha_c(0) = -\frac{1}{(1-a)\ln(1-a)} \qquad \longrightarrow \; \infty.$$

Die Speicherkapazität für Muster mit sehr geringer Aktivität ($\ln N$ Einsen in den Mustern, d. h. $a \approx -(N - 2\ln N)$) divergiert wie $O(N^2/(\ln N)^2)$.

Die Informationskapazität dagegen fällt mit zunehmender Korrelation der Muster leicht ab. Es gilt $I = 2N^2(1 - 0.084a^2)$ für schwach korrelierte Muster $a \ll 1$, und im Grenzfall $a \approx 1$ erhält man $I = N^2/2\ln 2 = 0.721 N^2$.

2.6.2 Die Verteilung der Stabilitäten

Für die Diskussion der Einzugsbereiche wird die Verteilung $\rho(\Lambda)$ der Stabilitäten benötigt. Diese läßt sich ebenfalls berechnen [Kepler & Abbott 88], [Gardner 89a] und für gesättigte Netzwerke (mit Parametern α, $\kappa_c(\alpha)$) erhält man:

$$\rho(\Lambda) = \frac{1}{\sqrt{2\pi}} \exp(-\Lambda^2/2) \, \Theta(\Lambda - \kappa_c) + \frac{1}{2}\Big(1 + \mathrm{erf}(\kappa_c/\sqrt{2})\Big) \delta(\Lambda - \kappa_c). \tag{19}$$

Dabei bedeutet der Wert von $\rho(\Lambda)\mathrm{d}\Lambda$ den (normierten) Anteil der Stabilitäten, die im Intervall $[\Lambda, \Lambda + \mathrm{d}\Lambda]$ liegen.

Kürzlich wurde darauf hingewiesen [Abbott & Kepler 89a], daß neuronale Netzwerke sich nach der Verteilung der Stabilitäten nahe ihrer Sättigung (d. h. mit Stabilitäten $\kappa \approx \kappa_c(\alpha)$) in Universalitätsklassen einteilen lassen und daß das Hopfield- und das Pseudoinverse-Modell jeweils einer eigenen Klasse angehören. Deshalb seien hier noch die entsprechenden normierten Verteilungen angegeben: Im Hopfield-Modell (37) ergibt eine einfache Analyse der lokalen Felder eine Verteilung

$$\rho_H(\Lambda) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\Lambda - \frac{1}{\sqrt{\alpha}} \right)^2 \right] \tag{20}$$

und für das Pseudoinverse-Modell gilt

$$\rho_P(\Lambda) = \delta\left(\Lambda - \sqrt{(1-\alpha)/\alpha} \right). \tag{21}$$

2.6.3 Speicherkapazität spezieller Modelle

Mit Gardners Methoden kann auch die optimale Speicherkapazität von Netzwerk-Modellen berechnet werden, die weitere Beschränkungen in der Wahl der Kopplungen aufweisen.

Im binären Netzwerk etwa fordert man $J_{ij} = \pm 1$ (wirkliches Ising-Modell), die sphärische Normierung ist dann automatisch erfüllt. Der Einfluß der Replikasymmetriebrechung ist extrem groß, die replika-symmetrische Lösung [Gardner & Derrida 88] liefert $\alpha_{c_B} \approx 1.34$, was aus informationstheoretischen Gründen offensichtlich unmöglich ist. Die Berechnung des ersten Schrittes der Symmetriebrechung gelang 1989 [Krauth & Mézard 89] und liefert $\alpha_{c_B} = 0.83$, was mit Simulationen [Krauth & Opper 89] gut übereinstimmt. Außerdem ist dieser Wert konsistent mit der Berechnung der sogenannten *zero entropy line*: Alle größeren Werte von α führen zu einer negativen Entropie des Systems.

Kürzlich ist es gelungen [Gutfreund & Stein 90], die Rechnungen auch für andere Bedingungen an die Synapsen durchzuführen. Insbesondere konnte damit die Speicherkapazität als Funktion von κ für Netzwerke mit integer-kodierten Synapsen ($-L \leq J_{ij} \leq L$) berechnet werden.

Die maximale Speicherkapazität $\alpha_{cL}(0)$ dieser Modelle läßt sich relativ leicht abschätzen. Sei dazu für gegebenes α eine Kopplungsmatrix mit reellwertigen Kopplungen J_{ij} gegeben, die die Muster speichert, das heißt $\xi_i^\mu \sum_{j \neq i} J_{ij} \xi_j^\mu > \kappa_c(\alpha)\sqrt{N}$. Nach der Skalierung $J_{ij} \to LJ_{ij}$ werden fast alle Synapsen im Intervall $[-L, \ldots, L]$ liegen, und können als Summe einer ganzen Zahl $\overline{J_{ij}}$ und einem Rest Δ_{ij} mit $|\Delta_{ij}| < 1/2$ geschrieben werden,

$$LJ_{ij} = \overline{J_{ij}} + \Delta_{ij}.$$

Die Ungleichungen lauten dann,

$$\xi_i^\mu \sum_{j \neq i} \overline{J_{ij}} \xi_j^\mu > L\sqrt{N}\kappa_c(\alpha) - \xi_i^\mu \sum_{j \neq i} \Delta_{ij} \xi_j^\mu. \tag{22}$$

Mit der Abschätzung $\frac{1}{2}\sqrt{N}$ für die Summe über die Δ_{ij} ist die linke Seite nur positiv (die Muster also gespeichert), wenn

$$\kappa_{cL}(\alpha) \geq 1/2L. \tag{23}$$

Der entsprechende Wert von α kann mit Gleichung (17) berechnet werden und gibt in sehr guter Näherung (jedenfalls für $L > 3$) die kritische Speicherkapazität des integer-kodierten Modells an.

2.6.4 Iteratives Lernen

Die Berechnung der maximalen Speicherkapazität ermöglicht keinerlei Aussagen über die Einstellung der Kopplungen. Für das Gardner-Modell mit reellwertigen Kopplungen lassen sich jedoch verschiedene iterative Lernregeln angeben, mit denen sich die Kopplungsmatrizen für die vorgegebenen Sollmuster auch konstruieren lassen [Gardner 88a], [Forrest 88], [Krauth & Mézard 87], [Abbott & Kepler 89a]. Für alle diese Algorithmen kann ein Konvergenzbeweis unter der Voraussetzung angegeben werden, daß sich die vorgegebenen Muster überhaupt speichern lassen (siehe Anhang B für den Konvergenzbeweis im verdünnten Netzwerk).

Die Kopplungen J_{ij} werden dabei gemäß $J_{ij} = J_{ij} + \Delta J_{ij}$ mit

$$\Delta J_{ij} = N^{-1} \; \epsilon_i^\mu \xi_i^\mu \xi_j^\mu \tag{24}$$

eingestellt, wobei die sogenannte *Fehlermaske* ϵ_i^μ dafür sorgt, daß nur die Muster gelernt werden, die noch nicht ausreichend stabilisiert sind:

$$\epsilon_i^\mu = \Theta\left(\kappa\Big(\sum_{j\neq i} J_{ij}^2\Big)^{1/2} - \sum_{j\neq i} J_{ij}\xi_i^\mu\xi_j^\mu\right). \tag{25}$$

Die mit diesem Algorithmus konstruierten Kopplungen werden natürlich im allgemeinen nicht symmetrisch sein. Symmetrische Kopplungen lassen sich aber mit der Wahl $\Delta J_{ij} = (\epsilon_i^\mu + \epsilon_j^\mu)\xi_i^\mu\xi_j^\mu$ erzwingen.

3 Einzugsbereiche in verdünnten Netzwerken

Zur Charakterisierung eines assoziativen Speichers sind seine dynamischen Eigenschaften ebenso wichtig wie die Speicherkapazität: Wie stark dürfen Eingabedaten gestört sein, um noch korrekt gespeicherten Mustern zugeordnet zu werden? Im Fall der Spinglas-Netzwerke mit ihrer Attraktordynamik bedeutet das die Frage nach der Größe und Gestalt der Einzugsbereiche (Attraktionsgebiete) der gespeicherten Muster. Auch die Konvergenzgeschwindigkeit — die Anzahl der Iterationen der Dynamik bis zum Erreichen eines Fixpunktes — und die Struktur eventuell vorhandener dynamisch stabiler aber unerwünschter Zustände (*spurious states*) ist in diesem Zusammenhang wichtig.

Selbst im einfachsten Fall der parallelen Dynamik gemäß (4) ist eine theoretische Behandlung extrem aufwendig. Es muß die stark nichtlineare dynamische Entwicklung von N Neuronen beschrieben werden, und wegen der iterativen Natur der Dynamik sind schon nach dem ersten Zeitschritt Rückkopplungseffekte zu berücksichtigen.

Andere dynamische Regeln, etwa die serielle asynchrone Dynamik (3) erfordern eine noch kompliziertere Beschreibung. Zuverlässige analytische Rechnungen über die Größe der Einzugsbereiche existieren nur für einige spezielle Modelle — insbesondere für das extrem verdünnte Netzwerk mit paralleler Dynamik. Eine Untersuchung der dynamischen Eigenschaften der Netzwerke ist deshalb weitgehend auf Simulationen angewiesen.

Die theoretischen Ergebnisse legen nahe, daß die Einzugsbereiche in verdünnten Netzwerken wesentlich größer sind als in vollständig verknüpften. In extrem verdünnten Netzwerken, mit (im Limes $N \to \infty$ und $C \to \infty$) weniger als $\ln N$ Synapsen pro Neuron, sind die Einzugsbereiche der parallelen Dynamik unterhalb von $\alpha \approx 0.4$ sogar optimal und nur durch die Korrelationen der gespeicherten Muster begrenzt. Das bedeutet auch, daß in diesem Fall keine *spurious states* die Dynamik der Modelle dominieren. In vollständig verknüpften Netzwerken verspricht die Modifikation der Dynamik zumindest eine Verbesserung der Einzugsbereiche.

Dieses Kapitel ist daher der Untersuchung der Einzugsbereiche in verdünnten Netzwerken mit Hilfe von Simulationen gewidmet. Beim Übergang von vollständig verknüpften zu verdünnten Netzwerken wachsen die Einzugsbereiche der Sollmuster stark an. Es zeigt sich, daß die Einzugsbereiche für Verdünnungen $c \approx 0.1$ schon sehr groß (fast optimal) sind. Derartige verdünnte Netzwerke erscheinen deshalb für Anwendungen durchaus vielversprechend.

Um den Einfluß der Dynamik auf die Eigenschaften der Modelle untersuchen zu können, werden die Simulationen außerdem mit verschiedenen dynamischen Regeln ausgeführt.

- Dazu wird zunächst in Abschnitt 3.1 eine Übersicht über die Problematik der Beschreibung der dynamischen Eigenschaften der Spinglas-Modelle gegeben, und es werden die bisher veröffentlichten Simulationen von Spinglas-Netzwerken zusammengestellt.

- Es schließt sich in Abschnitt 3.2 die Beschreibung der wichtigsten theoretischen Modelle zur Berechnung der Einzugsbereiche in Spinglas-Netzwerken mit paralleler Dynamik an. Es ist relativ leicht zu zeigen, daß die Dynamik der Netzwerke unter Transformationen der Art $\xi_i^\mu \to \xi_i^\mu S_i$ invariant ist. Dies schränkt die Zahl der für die Rechnungen in Frage kommenden Parameter stark ein. Trotzdem ist die für die Modelle benötigte Mathematik extrem aufwendig. Daher können nur die wichtigsten Ergebnisse zusammengefaßt und die Ideen der Rechnungen grob skizziert werden.

- Der Zusammenstellung der Grundlagen der Simulationen dient der Abschnitt 3.3. Die Übersicht beginnt mit der Beschreibung der Erzeugung von Sollmustern definierter Korrelation und der Auswahl der einzusetzenden Lernregel. Die Ergebnisse der theoretischen Modelle legen nahe, daß möglichst große Einzugsbereiche durch die Einstellung großer Stabilitäten der Muster erreicht werden können. Im Prinzip garantieren nicht nur der MinOver-Algorithmus, sondern auch die verbesserten lokalen iterativen Lernregeln die Einstellung der optimalen Stabilitäten. Aufgrund von finite-size Effekten können jedoch in den Simulationen nicht immer die theoretischen Werte erreicht werden. Nach der Lernphase werden Testmuster mit definiertem Überlapp zu gespeicherten Mustern erzeugt und bis zum Erreichen eines Fixpunktes (oder eines Zyklus) iteriert. Die Berechnung der Einzugsbereiche erfolgt dann durch Mittelung über das Verhalten sehr vieler derartiger Testmuster.

- Die Diskussion der in dieser Arbeit erhaltenen Ergebnisse beginnt mit Abschnitt 3.4. Dort werden die numerisch ermittelten Einzugsbereiche in verdünnten Spinglas-Modellen unter paralleler Dynamik diskutiert. Die Resulate bestätigen das für verdünnte Netzwerke vorgeschlagene theoretische Modell sehr gut, obwohl die in den Simulationen untersuchten Netzwerke noch sehr klein (und nicht sehr stark verdünnt) sind. Die Einzugsbereiche erweisen sich als unabhängig von der Größe der Netzwerke, jedenfalls innerhalb der hier untersuchten Grenzen. Der Vergleich der Einzugsbereiche im vollständig verknüpften Netzwerk mit dem entsprechenden theoretischen Modell zeigt dagegen Abweichungen auf. Deshalb werden aus den Simulationen die für dieses Modell wichtigen Parameter berechnet und mit den veröffentlichten Werten verglichen.

- Die Untersuchung der vollständig verknüpften oder nur schwach verdünnten Netzwerke zeigt bei hoher Speicherkapazität α nur recht kleine Einzugsbereiche der gespeicherten Muster, obwohl diese weiterhin die Minima der Energiefläche darstellen. Dies ist auf die Existenz von spurious states zurückzuführen, die als unerwünschte Attraktoren für die Testmuster dienen. Eine grobe Untersuchung dieser spurious states lieferte leider keinen interessanten Aufschluß über deren Natur. Bei höheren Speicherkapazitäten $\alpha \geq 0.4$ scheinen sehr viele unerwünschte zusätzliche stabile Zustände mit stark unterschiedlichen Stabilitäten aufzutreten.

- Die Stabilitäten können im Prinzip für jedes gespeicherte Muster individuell eingestellt werden, wenn die Lernregel entsprechend erweitert wird. Abschnitt 3.6 untersucht die Möglichkeiten dieses sogenannten *phase space gardening*, die Einzugsbereiche der Muster einzeln einzustellen. Dies ist selbstverständlich auch für viele Anwendungen interessant.

  Es zeigt sich, daß die Einzugsbereiche der Muster durch Vorgabe einer Stabilität $\kappa_{i\mu}$ tatsächlich innerhalb weiter Grenzen und mit nur schwacher Abhängigkeit von α individuell eingestellt werden können. Die Simulationen zeigen außerdem, daß zwar der Radius der Einzugsbereiche vorgegeben werden kann, ihre Gestalt aber kugelförmig (isotrop) bleibt. Die Einstellung von Einzugsbereichen bestimmter Form gelingt mit den hier betrachteten Lernregeln nicht; die Verwendung des *learning with noise* könnte dies jedoch ermöglichen.

- Der Einfluß der verwendeten Dynamik auf die Eigenschaften des Netzwerks wird dann in Abschnitt 3.7 untersucht. Auf die Verwendung einer Dynamik mit $T > 0$, mit der thermisches Rauschen im Netzwerk modelliert werden kann, wurde dabei verzichtet, da die Simulationen dadurch sehr aufwendig werden. Die Untersuchung wurde vielmehr auf die asynchrone serielle Dynamik und zwei weitere dynamische Regeln beschränkt, die das Verhalten eines Neurons nicht nur vom aktuellen lokalen Feld, sondern auch vom lokalen Feld zu früheren Zeitpunkten abhängig machen (*memory terms*).

  Die Simulationen liefern das interessante Ergebnis, daß die Verwendung der asynchronen Dynamik anstelle der parallelen das Verhalten der Netzwerke kaum verschlechtert. Oberhalb von $\alpha \approx 0.6$ sind die für die Einzugsbereiche ermittelten Werte sogar fast identisch. Wie später dargelegt wird, ist die Konvergenzgeschwindigkeit für serielle Dynamik eventuell sogar deutlich besser.

  Die Verwendung der memory-term Dynamik ermöglicht über weite Bereiche der Speicherkapazität α eine deutliche Vergrößerung der Einzugsbereiche. Allerdings gelingt es bei weitem nicht, in vollständig vernetzten Modellen optimale Einzugsbereiche einzustellen. Die Einzugsbereiche erweisen sich auch unter Verwendung dieser Dynamik als isotrop.

- Abgeschlossen wird dieses Kapitel von einem Abschnitt 3.8 über die Konvergenzgeschwindigkeit für die verschiedenen Dynamiken und vollständig verknüpfte, sowie stark verdünnte Netzwerke. Diese Werte sind nur sehr schwierig zu ermitteln: Zyklen müssen abgebrochen werden, andererseits kann das Erreichen eines Attraktors sehr viele Iterationen der Dynamik erfordern. Die aus den Simulationen ermittelten Werte zeigen ein sehr komplexes Verhalten. Die Fluktuationen sind so groß, daß keine Extrapolation auf das Verhalten viel größerer Netzwerke gewagt werden kann.

3.1 Dynamische Eigenschaften in neuronalen Netzen

Die Beschreibung der Einzugsbereiche in einem neuronalen Modell ist nicht nur wegen der großen Zahl miteinander wechselwirkender Freiheitsgrade sehr schwierig.

Wenn die Sollmuster ξ_i^μ in einem Spinglas-Netzwerk mit Stabilität $\kappa_{i\mu} > \kappa$ gespeichert sind, läßt sich natürlich sofort eine untere Grenze für die Größe der Attraktionsgebiete unter paralleler Dynamik angeben: Alle Anfangszustände mit Anfangsüberlapp $m_0 > 1 - \kappa/(2N^{1/2})$ werden im ersten Schritt der Dynamik in das Muster ξ_i^μ wandern. Diese Abschätzung ist allerdings viel zu pessimistisch, und sie berücksichtigt nur den ersten Schritt der Dynamik. Simulationen zeigen, daß die Einzugsbereiche über weite Bereiche der Speicherkapazität α wesentlich größer — von $O(1)$ — sind.

Auch wenn — wie im Hopfield-Modell — eine Energiefunktion angegeben und zur Konstruktion einer Ljapunovfunktion benutzt werden kann, führt das kaum weiter: Die Analyse der statistischen Mechanik des Hopfield-Modells zeigt, daß die Minima der Energie nach gewünschten Zuständen und *spurious states* klassifiziert werden müssen — während die Zahl der stabilen Mischzustände von Mustern für $\alpha < 1$ klein ist, wächst die Zahl der Spinglaszustände exponentiell mit der Zahl der Neuronen. Für kompliziertere Modelle, vor allem für Netzwerke nach iterativem Lernen, sind die Einstellungen der Kopplungen J_{ij} im allgemeinen nicht analytisch bekannt. Wenn die Lernregel zudem asymmetrische Werte für die Kopplungen zuläßt, kann die Energiefläche kaum noch sinnvoll zur Beschreibung der dynamischen Eigenschaften genutzt werden.

Außer von der verwendeten Lernregel zur Einstellung der Synapsen, sowie von globalen Parametern (etwa der Korrelation der Sollmuster), hängen die Einzugsbereiche natürlich von der verwendeten Dynamik ab. Verschiedene dynamische Regeln können dabei völlig andere Eigenschaften aufweisen, sowohl in der Struktur der Attraktoren (Sollmuster, spurious states, Zyklen) als auch in deren Attraktionsgebieten und der Konvergenzgeschwindigkeit.

In diesem Zusammenhang sind vor allem die einfache parallele Dynamik (4), sowie die asynchrone serielle Dynamik (3) interessant. Die parallele Dynamik ist dabei vor allem für theoretische Modelle geeignet, wie im folgenden Abschnitt 3.2 gezeigt wird. Für die Beschreibung biologischer Systeme, aber auch für technische Realisierungen der Modelle, ist die serielle Dynamik wichtig, weil sie das Verhalten asynchroner Netzwerke modellieren kann. Natürlich muß für die serielle Dynamik zusätzlich noch eine Reihenfolge des Update der Neuronen festgelegt werden.

Messungen an biologischen Nervensystemen zeigen einen hohen Anteil von Rauschen in den elektrischen Signalen der einzelnen Zellen. Die einfachste Möglichkeit, Rauschen in den Spinglas-Modellen zu beschreiben, ist die Verwendung einer modifizierten Dynamik [Little 74]. Dazu gibt $P(S_i(t+1)|S_i(t))$ die Wahrscheinlichkeit an, ausgehend von S_i zur Zeit $t+1$ den Wert $S_i(t+1)$ vorzufinden ($\beta = 1/T$),

$$P\big(S_i(t+1)\,|\,S_i(t)\big) \quad = \quad \Big[1 + \exp\big(-2\beta S_i(t+1)\cdot h_i(t)\big)\Big]^{-1} \tag{26}$$

Im Hopfield-Modell gelingt es, eine endliche Temperatur des Netzwerks auch bei der theoretischen Beschreibung der Attraktoren der Dynamik zuzulassen und das α-T Phasendiagramm zu berechnen. Dies ist in den verbesserten Netzwerken mit iterativen Lernregeln nicht mehr möglich.

Die Untersuchung der Einzugsbereiche bei endlicher Temperatur kann nur mit Simulationen erfolgen. Eine entsprechende Analyse des vollständig vernetzten Gardner-Modells (mit symmetrischen Kopplungen) wurde kürzlich vorgestellt [Nardulli & Pasquariello 90]. Dabei zeigt sich, daß eine Temperatur $T > 0$ des Netzwerks fast keinen Einfluß auf die Größe der Attraktionsgebiete hat. Natürlich wird die Konvergenz des Netzwerks mit steigender Temperatur immer langsamer.

Ein anderer Versuch, die Größe der Einzugsbereiche zu verbessern, ist die Verwendung einer dynamischen Regel mit „Gedächtnistermen" (*memory terms*) [Kanter & Sompolinsky 87], etwa gemäß

$$S_i(t+1) = \mathrm{sgn}\Big(\frac{1}{2}h_i(t) + \frac{1}{2}h_i(t-1)\Big). \tag{27}$$

Die Idee ist, daß eine derartige Dynamik Zyklen der Länge 2 (die unter paralleler Dynamik sehr häufig sind) effektiv unterdrückt und außerdem über winzige lokale Energieminima wegmitteln sollte. Tatsächlich werden die Einzugsbereiche durch Verwendung dieser Dynamik oberhalb von $\alpha > 0.4$ etwas größer. Weil eine Dynamik dieser Art leicht in digitalen Systemen zu implementieren ist, werden ähnliche dynamische Regeln im Folgenden noch diskutiert.

Wegen dieser beträchtlichen Schwierigkeiten bei der Beschreibung von dynamischen Eigenschaften der Spinglas-Netze haben sich die bisherigen analytischen Untersuchungen von Spinglas-Netzwerken fast ausschließlich auf die Frage nach der Speicherkapazität konzentriert. Vollständig lösbar ist die Dynamik nur für extrem verdünnte Netzwerke [Derrida *et. al.* 87], [Gardner 89a].

Die Einzugsbereiche des vollständig verknüpften Hopfield-Modells wurden in [Bruce *et. al.* 87] sowie von [Forrest 88] untersucht. Auch die analytische Behandlung des Pseudoinverse-Modells von [Kanter & Sompolinsky 87] enthält eine Untersuchung der Einzugsbereiche.

Zwei weitere Simulationen in vollständig verknüpften Netzen wurden von [Kepler & Abbott 88] und [Krauth *et. al.* 88] vorgestellt, um theoretische Modelle für die Einzugsbereiche mit paralleler Dynamik zu unterstützen. Die Auswirkungen verschiedener Dynamiken sind nur in [Kanter & Sompolinsky 87] kurz diskutiert worden. Die Einzugsbereiche in schwach verdünnten neuronalen Netzwerken sind bisher nicht untersucht worden.

3.2 Theoretische Modelle für die Einzugsbereiche

Trotz der immensen Schwierigkeiten bei der theoretischen Behandlung der Einzugsbereiche in Spinglas-Modellen ist eine ganze Reihe verschiedener Modelle vorgeschlagen worden. Die grundlegende Idee ist jeweils, ausgehend von einer Anfangskonfiguration mit Überlapp m_0 mit einem Muster ξ_i^μ die Wahrscheinlichkeit für einen Überlapp m_1 nach einem Schritt der Dynamik auszurechnen (gemittelt über alle Anfangszustände und über die Sollmuster). Bei paralleler Dynamik (4) gilt es also,

$$P(m_1|m_0) = \frac{\mathrm{Tr}_S\left[\delta\left(m_1 - \frac{1}{N}\sum_{i=1}^N \xi_i^\mu \mathrm{sgn}\left(\sum_{j\neq i} J_{ij}S_j\right)\right)\delta\left(m_0 - \frac{1}{N}\sum_{j=1}^N S_i\xi_i^\mu\right)\right]}{\mathrm{Tr}_S\left[\delta\left(m_0 - \frac{1}{N}\sum_{i=1}^N S_i\xi_i^\mu\right)\right]} \tag{28}$$

zu berechnen.

Schon 1987 gelang es [Gardner *et. al.* 87] die dynamische Entwicklung für Sherrington-Kirkpatrick Spingläser sowie für das Hopfield-Modell mit paralleler Dynamik für Zeiten $t = 0$ bis $t = 4$ explizit anzugeben. Die Formeln werden aber mit jedem Zeitschritt komplizierter, und die Berechnung der Fixpunkte der Dynamik scheint hoffnungslos.

Im Hopfield-Modell erhielten sie das interessante Resultat, daß die Zeitentwicklung eines Anfangszustandes schließlich in einem Spinglas-Zustand enden kann, obwohl die Dynamik der ersten Zeitschritte eine Konvergenz des Anfangszustandes zum entsprechenden Sollmuster andeutet. Das bedeutet, daß die Attraktionsgebiete den Phasenraum nicht ausfüllen und daß die Grenzen der Attraktionsgebiete nicht glatt sind. Die numerischen Ergebnisse für die Größe der Einzugsbereiche bestätigen dieses Ergebnis. Allerdings sind die Einzugsbereiche (jedenfalls unter paralleler Dynamik) fast isotrop und die „Rauhigkeit“ der Attraktionsgebiete nimmt mit wachsender Netzwerkgröße ab.

Die Zeitentwicklung der parallelen Dynamik vereinfacht sich entscheidend, wenn das zu untersuchende Netzwerk extrem verdünnt ist, mit höchstens $O(\ln N)$ Kopplungen pro Neuron [Derrida *et. al.* 87]. In diesem Fall spielen nämlich Rückkopplungseffekte keine Rolle, da für jedes Neuron i der *tree of ancestors* dieses Neurons, also die Neuronen $S_k(t-1)$, $S_k(t-2)$, ..., die das Neuron i beeinflussen, keine Schleifen enthält. Für einen Anfangszustand mit Überlapp m_0, der im ersten Zeitschritt auf sein Muster zuläuft, $m_1 > m_0$, gilt dann auch $m_2 > m_1$, ...bis zur Konvergenz. Die Größe der Einzugsbereiche kann daher direkt aus dem Verhalten des ersten Schritts der Dynamik ausgerechnet werden.

Dieser erste Schritt der parallelen Dynamik kann auch im Gardner-Modell berechnet werden, wenn die Verteilung der Stabilitäten $\rho(\kappa_{i\mu})$ für Parameter α und κ bekannt ist (19). Ausgehend von einem Anfangszustand S_0 mit Überlapp m_0 mit einem Sollmuster ξ_i^μ erhält man für den Überlapp m_1 nach einem Schritt paralleler Dynamik [Gardner *et. al.* 87], [Gardner 89a], [Kepler & Abbott 88],

$$m_1 = F_\kappa(m_0) = \int_{-\infty}^{\infty} \mathrm{d}\Lambda\; \rho(\Lambda)\,\mathrm{erf}\left(\frac{m_0\Lambda}{\sqrt{2(1-m_0^2)}}\right). \tag{29}$$

Wegen des oben skizzierten Arguments gibt der instabile Fixpunkt

$$m_S = F_\kappa(m_S) \tag{30}$$

damit direkt die Größe der Einzugsbereiche in extrem verdünnten Gardner-Netzwerken an. Die Fixpunktgleichung kann numerisch gelöst werden, wenn die entsprechende Verteilung der Stabilitäten eingesetzt wird.

Das Resultat zeigt, daß die Einzugsbereiche in gesättigten Netzen für $\alpha \leq 0.41$ optimal sind. Das heißt, jedes Testmuster mit makroskopischem Anfangsüberlapp ($m_0 > N^{-1/2}$) wird vom Netzwerk korrekt wiedererkannt. Für $\alpha > 0.41$ vergrößert sich der Fixpunkt m_S langsam von $m_S = 0$ zu Werten nahe $m_S = 1$. Natürlich bleiben die Einzugsbereiche für alle $\alpha < \alpha_c = 2.0$ endlich.

Ausgehend von einer numerischen Analyse der Dynamik der Netzwerke schlugen Kepler und Abbott [Kepler & Abbott 88] vor, den Fixpunkt von

$$\frac{F_\kappa(m_F) - m_F}{1 - m_F} = a_0 \tag{31}$$

zur Beschreibung der Attraktionsgebiete in vollständig verknüpften Netzen zu verwenden, wobei a_0 eine Konstante ist, deren Wert sie mit $a_0 = 1/2$ angaben.

Anders als die Beziehung (29) aber, die von den Simulationen sehr genau bestätigt wird, ist die Verwendung des Fixpunktes (31) nicht sehr präzise und hängt darüber hinaus von der verwendeten Lernregel ab. Insbesondere für $\alpha < 0.2$ fluktuieren die Werte der in die Fixpunktgleichung einzusetzenden Werte von a_0 beträchtlich (etwa $c = 0.4 \ldots 0.7$).

Einen weiteren Weg zur näherungsweisen Berechnung der Attraktionsgebiete zeichneten [Krauth *et. al.* 88] auf. Zunächst merkten sie an, daß die Dynamik des Netzwerks unter den „Eichtransformationen“ $J_{ij} \to J_{ij} S_i S_j, \quad \xi_i^\mu \to \xi_i^\mu S_i$ invariant bleibt, wobei $\{S_i\}$ ein beliebiger Zustand des Netzwerks ist. Deshalb kann die Beschreibung der Einzugsbereiche nur von solchen Parametern abhängen, die ebenfalls invariant unter diesen Transformationen sind. Derartige Größen sind die Stabilitäten $\kappa_{i\mu}$, die Symmetrie η der Kopplungsmatrix, die Selbstkopplungen J_{ii} und einige exotische Größen, etwa $\sum_{j,k,\ldots,m} J_{ij} J_{jk} \cdots J_{mi}$.

Ihre Idee ist deshalb, das Netzwerk nur durch zwei Parameter, die Verteilung der Stabilitäten $\rho(\kappa)$ und die Symmetrie η der Kopplungsmatrix J_{ij} zu beschreiben. Es gelang ihnen, die Zeitentwicklung dieses sogenannten *one pattern models* bis zu $t = 4$ explizit zu berechnen. Ihre Vorhersagen stimmen für Netze mit hoher Symmetrie $\eta \approx 1$ gut mit Simulationen an Netzwerken mit der Simplex-Lernregel überein, etwas schlechter mit Simulationen der iterativen Lernregel. Leider haben sie keine Angaben für Netzwerke mit geringer Symmetrie $\eta \approx 0$ veröffentlicht, die verdünnten Netzwerken entsprechen würden.

3.3 Grundlagen der Simulationen

Anders als die theoretische Beschreibung bereitet die numerische Bestimmung der Einzugsbereiche keine prinzipiellen Schwierigkeiten.

Für jede Wahl der Speicherkapazität α können leicht die Sollmuster $\xi_i^\mu, \mu = 1, \ldots, \alpha \cdot N$ mit beliebiger Magnetisierung a bzw. Korrelation $\langle \xi_i \xi_j \rangle = \delta_{ij} + a^2(1 - \delta_{ij})$

erzeugt werden. Die Konstruktion einer Kopplungsmatrix, die die vorgegebenen Muster speichert, gelingt dann durch Iteration des Lernalgorithmus (24), bis die gewünschten Stabilitäten der Muster erreicht sind.

Allerdings kann aufgrund von finite-size Effekten (auch noch für Netzwerke mit $N = 512$ oder mehr Neuronen) die theoretische Grenze $\kappa_c(\alpha)$ nicht immer streng erreicht werden. Außerdem ist zu beachten, daß die Zahl der zum Lernen nötigen Iterationen nahe der kritischen Stabilität stark ansteigt. Es gilt also, einen Kompromiß zwischen der erreichten Stabilität und dem Rechenzeitbedarf zu finden.

Eine Lösung bildet die Verwendung von verbesserten Lernalgorithmen. Der schnelle Algorithmus von [Abbott & Kepler 89a] verringert die Zahl der nötigen Lernzyklen beträchtlich, und der MinOver-Algorithmus von [Krauth & Mézard 87] garantiert das Auffinden der bestmöglichen Stabilität.

Nach der Lernphase liegt eine Kopplungsmatrix mit den gewünschten Stabilitäten der Sollmuster vor. Weil die Eigenschaften des Netzwerks stark von der verwendeten Dynamik abhängen werden, ist der nächste Schritt die Auswahl und Implementation einer geeignteten Dynamik.

Anders als im Hopfield-Modell ist die einfache Neuberechnung der Synapsen in Gardner-Netzwerken nicht möglich. Daher muß die Kopplungsmatrix des Netzes gespeichert werden. In stark verdünnten Netzen ist es dafür günstig, die Matrix nicht direkt zu speichern (da viele Kopplungen $J_{ij} = 0$ sind), sondern über Zeiger zu verwalten, um so Rechenzeit und Speicherplatz zu sparen.

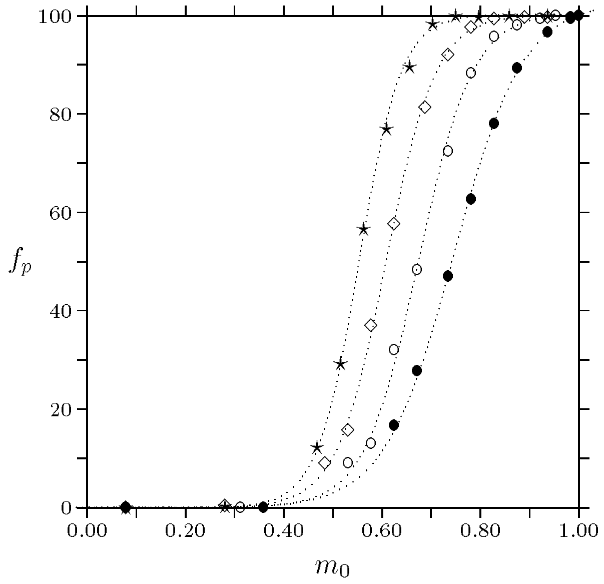
Die Ermittlung der Einzugsbereiche erfolgt dann nach folgendem Algorithmus [Forrest 88]: Es werden Testmuster $\xi_i^{\mu,r}$ mit definiertem Überlapp m_0 mit einem gespeicherten Muster ξ_i^μ erzeugt und unter der gewählten Dynamik iteriert, bis Stabilität oder ein Zyklus erreicht wird. Für jeden Anfangsüberlapp m_0 ergibt sich aus der Mittelung über viele derartige Testmuster der mittlere erreichte Endüberlapp m_f, sowie der Anteil der perfekt erkannten Muster f_p.

Ein Beispiel für die derart erhaltenen Kurven $f_p(m_0, \kappa)$ ist in Abbildung 1 dargestellt. Die Kurven $m_f(m_0, \kappa)$ sind etwa in Abbildung 25 zu erkennen. Für hohen Anfangsüberlapp $m_0 \approx 1$ laufen fast alle Testmuster in die entsprechenden Sollmuster, für kleine Werte von m_0 werden nur wenige oder gar keine Testmuster korrekt zugeordnet.

Es ist deutlich zu erkennen, daß die Größe der Einzugsbereiche (bei konstanter Speicherkapazität α) tatsächlich mit den Stabilitäten κ der Muster wächst. Die Einzugsbereiche sind wesentlich größer als die pessimistische Abschätzung $m_c \approx 1 - \kappa/(2N^{1/2})$ erwarten läßt. Die analytische Beschreibung des Endüberlapps m_f als Funktion des Anfangsüberlapps m_0 erscheint recht kompliziert.

Die Darstellung des Anteils der perfekt erkannten Muster dagegen zeigt ein scharf begrenztes Übergangsgebiet von $f_p = 1$ nach $f_p = 0$, und die Form dieses Übergangs legt eine analytische Darstellung nach

$$f_p(m_0)/(1 - f_p(m_0)) = a_1 \exp(N a_2(m_0 - m_c)). \tag{32}$$



Anteil f_p perfekt erkannter Testmuster als Funktion von m_0 und der Mindeststabilität $\kappa = 0.25\ \bullet$, $0.45\ \circ$, $0.65\ \diamond$, $0.90\ \star$. ($\alpha = 0.4$, $N = 128$)

Abbildung 1: Beispiel für den Anteil perfekt erkannter Testmuster f_p als Funktion von m_0 und κ.

nahe. Dabei sind a_1 und a_2 Konstanten, die von den jeweiligen Parametern α und κ abhängen, und m_c ist der numerische Wert für die Größe des Einzugsbereiches: Alle Testmuster mit $m_0 > m_c$ werden mit hoher Wahrscheinlichkeit erkannt, für $m_0 < m_c$ endet die Zeitentwicklung des Netzwerks nicht im gewünschten Sollmuster. Diese Bestimmung der Größe der Einzugsbereiche benutzt eine ganz andere Methode, als sie dem theoretischen Modell m_F (31) zugrunde liegt, und läßt sich daher auch in verdünnten Netzwerken verwenden: Der Wert m_F dagegen ist nur über die Iteration (29) mit den Einzugsbereichen verknüpft.

Simulationen für verschiedene Netzwerkgrößen zeigen, daß die Breite des Übergangsbereiches mit N tatsächlich gemäß (32) abnimmt. Dies deutet im Grenzfall $N \to \infty$ auf einen Phasenübergang [Forrest 88] und impliziert isotrope Einzugsbereiche. Ein Beispiel dafür, wie scharf der Übergangsbereich schon in kleinen Netzwerken wird, liefern die Diagramme im Abschnitt 3.6.

Es ist zu beachten, daß wegen der kleinen Anzahl von Neuronen der in den Simulationen verwendeten Netzwerke ($N < 512$) die zufälligen Korrelationen der Sollmuster von $O(N^{-1/2})$ nicht vernachlässigt werden können — insbesondere für kleine Werte des Anfangsüberlapps m_0. Ein Testmuster mit Überlapp $m_{0\mu} \approx N^{-1/2}$ mit einem Sollmuster ξ_i^μ hat mit hoher Wahrscheinlichkeit einen größeren Überlapp $m_{0\nu}$ mit irgendeinem anderen Sollmuster ξ_i^ν.

Die Korrektur dieses Effekts erfordert allerdings für jedes Testmuster die Berechnung des Überlapps mit allen Sollmustern und ist daher relativ aufwendig. Es sei darauf hingewiesen, daß [Kanter & Sompolinsky 87] diese Korrektur in ihrer Simulation des Pseudoinverse-Modells durchgeführt haben.

Natürlich lassen sich aus den Simulationen auch alle anderen benötigten oder gewünschten Größen problemlos ermitteln, zum Beispiel die Konvergenzgeschwindigkeit als Funktion von m_0, α und κ oder die für das Modell (31) benötigte Konstante a_0.

3.4 Einzugsbereiche in verdünnten neuronalen Netzen

Die Ergebnisse der Simulationen an verdünnten, fast gesättigten Netzwerken mit paralleler Dynamik sind in Abbildung 2 dargestellt. Zusätzlich sind die theoretischen Modelle m_S (30) für das extrem verdünnte und m_F (31) für das vollständig verknüpfte Netz eingezeichnet. Jeder Datenpunkt entspricht dabei dem Mittelwert aus mehreren Simulationen von Netzwerken mit gegebenen Parametern N, c, α und κ. Für jede einzelne Simulation wurden neue unkorrelierte Sollmuster zufällig erzeugt. Die Werte von m_c entstanden dann aus der Auswertung der Beziehung (32) für sehr viele (mindestens einige tausend) Testmuster. Die statistischen Fehler der Simulationen entsprechen etwa der Größe der Symbole.

Wegen des hohen Rechenzeitbedarfs der Simulationen in großen Netzwerken mußte die Anzahl der untersuchten Testmuster in den Simulationen mit $N \geq 400$ reduziert werden. Die statistischen Fehler dieser Simulationen können zweimal die Größe der Symbole erreichen.

Die hier vorgestellten Simulationen der vollständig verknüpften Netzwerke weisen für $\alpha < 0.5$ etwas größere Einzugsbereiche auf, als sie vom Modell m_F (31) vorhergesagt werden, sind aber in Übereinstimmung mit früheren Simulationen des Pseudoinverse-Modells. Obwohl die Stabilitäten der Muster im Pseudoinverse-Modell kleiner als die im Gardner-Modell erreichten sind, gaben [Kanter & Sompolinsky 87] unter Verwendung einer finite-size Korrektur gemäß

$$R = \left\langle \frac{1 - m_c}{1 - m_0} \right\rangle$$

eine grobe Näherung für die Einzugsbereiche mit modifizierter serieller Dynamik an (dabei wurden die Spins mit $S_i \neq \xi_i^\mu$ zuerst neu eingestellt):

$$R(\alpha) \approx 1 - \alpha.$$

Unter Verwendung paralleler Dynamik erhielten sie sehr ähnliche Werte für die Größe der Einzugsbereiche. Für $\alpha < 0.4$ jedenfalls erwiesen sich die Einzugsbereiche der parallelen Dynamik als nur wenig kleiner gegenüber der modifizierten seriellen Dynamik.

Dies steht im Widerspruch zum Modell (31), stimmt aber unter Berücksichtigung der finite-size Korrektur gut mit den hier präsentierten Resultaten überein. Die hier vorgestellten Daten sind ebenfalls konsistent mit den von [Forrest 88] unter Verwendung serieller Dynamik und einer symmetrischen iterativen Lernregel erhaltenen Werten für $m_c(\alpha = 0.25) \approx 0.44$ und $m_c(\alpha = 0.5) \approx 0.75$ (siehe auch Abbildung 11).

Oberhalb von $\alpha \approx 0.75$ allerdings stimmen die aus den Simulationen ermittelten Werte für die Größe der Einzugsbereiche sehr gut mit den beiden Modellen (30) und (31) überein.

Trotzdem stellt der aus der Iteration (31) erhaltene Wert auch in vollständig verknüpften Netzwerken nur eine grobe Näherung für die Größe der Einzugsbereiche dar. In Abbildung 3 sind die aus den Simulationen berechneten Werte für die in die Iteration (31) einzusetzende Konstante a_0 dargestellt. Dabei wurde die Iteration zusätzlich auf verdünnte Netzwerke ($c = 0.1$) angewendet, um eine Referenz zu erhalten. Wenn im verdünnten Netzwerk tatsächlich $m_k > \cdots > m_1 > m_0$ gilt, sollte die Kurve $f_p(a_0)$ genau bei $a_0 = 0.5$ von $f_p = 0$ auf $f_p = 1$ ansteigen, und dies ist auch deutlich zu erkennen. Dagegen kann der von [Kepler & Abbott 88] angegebene Wert von $a_0 \approx 0.5$ für das vollständig verknüpfte Modell hier nicht bestätigt werden, obwohl der Übergang von $f_p = 0$ zu $f_p = 1$ tatsächlich sehr scharf ausgeprägt ist. Vielmehr legen die numerischen Daten für $c = 1.0$ einen Wert von $a_0 \approx 0.6$ nahe.

Um die Simulationen von Netzwerken mit verschiedener Konnektivität vergleichen zu können, wurde die Speicherkapazität α gemäß $\alpha = P/C$ definiert. Damit gibt α das Verhältnis von der Zahl der Muster zur Zahl der Synapsen pro Neuron an. Mit dieser Bedeutung von α gilt auch für die verdünnten Netze die Beziehung (17) zwischen κ und α.

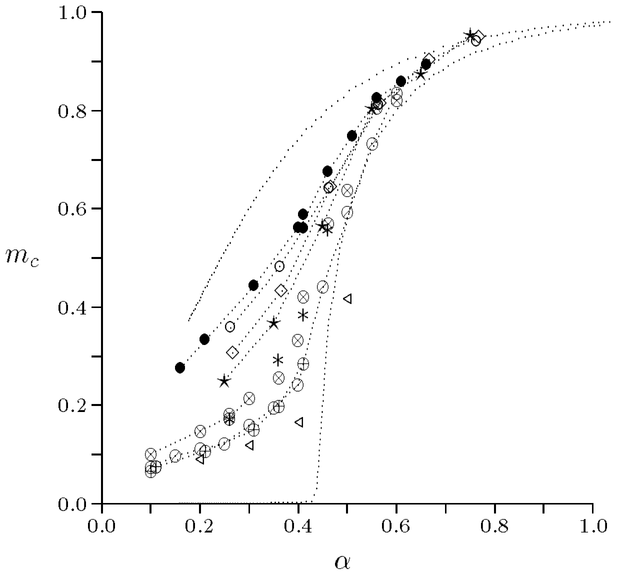
Der Übergang vom vollständig verknüpften Netzwerk mit allen Rückkopplungseffekten zum verdünnten Netz mit Symmetrie $\eta \approx 0$ ohne Rückkopplungseffekte ist gut zu erkennen. Die Simulationen bestätigen das Modell m_S für das extrem verdünnte Netz.

Für die nur gering verdünnten Modelle wurde auf Simulationen bei kleiner Speicherkapazität α weitgehend verzichtet. Die Werte für m_c müssen im Grenzfall $\alpha \rightarrow 0$ offenbar einen Wert nahe der statistischen Korrelation $1/\sqrt{N}$ erreichen, da hier keine finite-size Korrektur auf die Werte von m_c angewendet wurde.

Viel interessanter ist der Bereich der Speicherkapazitäten nahe $\alpha \approx 0.4$, in dem der Übergang vom Verhalten des vollständig verknüpften Netzwerks zum verdünnten deutlich sichtbar wird.

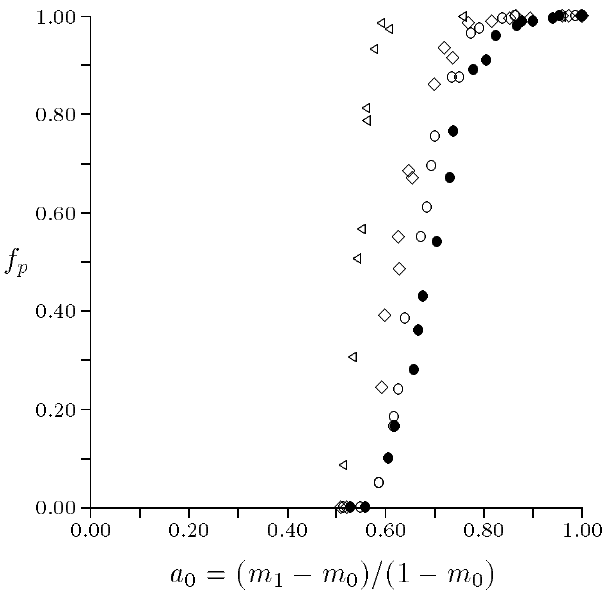
Die fast optimalen Einzugsbereiche $m_c \approx N^{-1/2}$ werden schon für Verdünnungen $c \approx 0.1$ erreicht. Es ist zu beachten, daß die bei starker Verdünnung resultierenden Netzwerke sehr klein sind und daher relativ große finite-size Fluktuationen aufweisen. Dieser Effekt wird im Bereich um $\alpha \approx 0.4$ noch verstärkt, weil Netzwerke mit nur leicht unterschiedlichen Speicherkapazitäten dort durch den starken Anstieg von $m_c(\alpha)$ deutlich differierendes Verhalten zeigen können.

Deshalb erscheint eine Untersuchung des *finite size scaling* sinnvoll, um die Eigenschaften von Netzwerken verschiedener Größe zu beschreiben. Wegen des enormen Rechenzeitbedarfs für die Simulation großer Netzwerke mußte die Anzahl der Neuronen auf $N \leq 512$ begrenzt werden. In Abbildung 4 sind die Einzugsbereiche m_c für



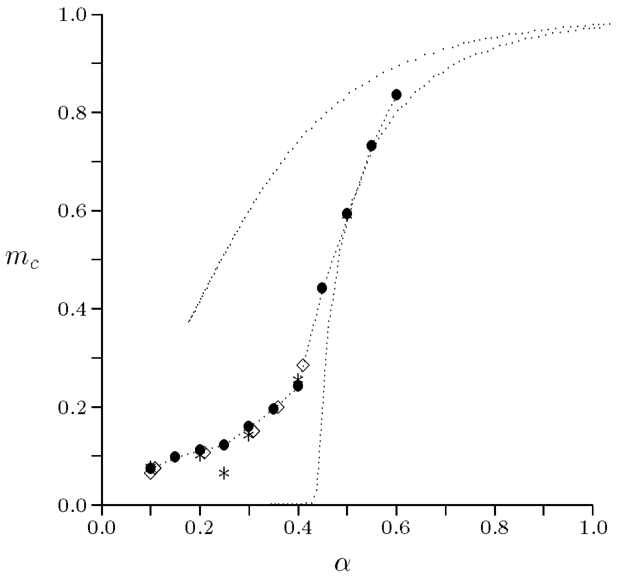
Numerisch ermittelte Einzugsbereiche m_c als Funktion von α in verdünnten Netzwerken, verglichen mit m_F und m_S. ($N = 128$, $c = 1.0$ •, $c = 0.8$ ∘, $c = 0.6$ ⋄, $c = 0.4$ ⋆) ($N = 256$ $c = 0.2$ ⊗, $c = 0.1$ ⊘) ($N = 400$, $c = 0.2$ ∗, $c = 0.1$ ⊕) ($N = 256$, $c = 0.05$ ◃).

Abbildung 2: Einzugsbereiche m_c in verdünnten, fast gesättigten Netzwerken mit paralleler Dynamik als Funktion von $\alpha = P/C$. Die Kurven zeigen die theoretischen Modelle m_F (obere) und m_S (untere).



Anteil perfekt erkannter Testmuster f_p als Funktion des Parameters $a_0 = (m_1 - m_0)/(1 - m_0)$ in vollständig verknüpften und in verdünnten Netzwerken. $N = 128$, $c = 1.0$, $\alpha = 0.20$ •, 0.30 ∘, 0.40 ⋄. $N = 400$, $c = 0.1$, ◃.

Abbildung 3: Numerisch ermittelte Werte der Konstante a_0 für die Fixpunktiteration.



Vergleich der numerisch ermittelten Einzugsbereiche m_c in stark verdünnten Netzwerken mit $c = 0.1$ als Funktion von α und N. $N = 256$ $\bullet$, $N = 400$ $\diamond$, $N = 512$ $*$.
Siehe Text.

Abbildung 4: Numerisch ermittelte Werte von m_c für parallele Dynamik ($N = 256$ bis $N = 512$). Siehe auch Abb. 5.

Netzwerke mit $N = 256, 400$ und 512 Neuronen und Konnektivität $c = 0.1$ (zusammen mit den Modellen m_F und m_S) dargestellt.

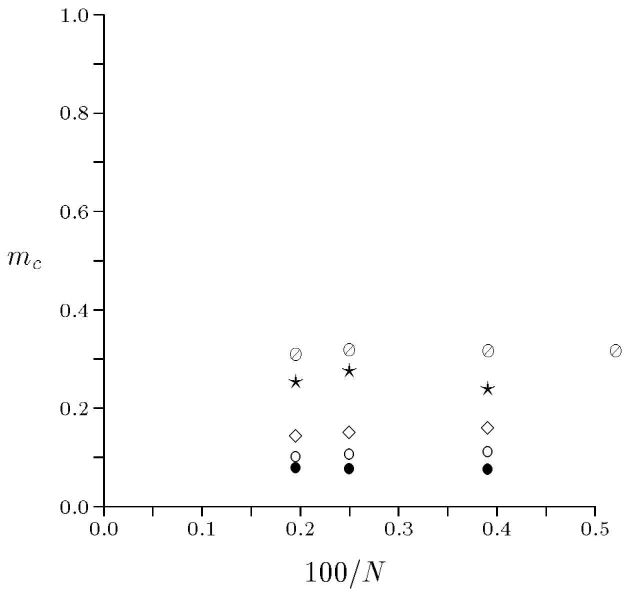
Die entsprechende Darstellung als Funktion der Netzwerkgröße findet sich in Abbildung 5, wobei zusätzlich das Verhalten von m_c für vollständig verknüpfte Netzwerke mit $\alpha = 0.2$ dargestellt ist. Die numerisch ermittelten Einzugsbereiche erweisen sich tatsächlich als weitgehend von der Netzwerkgröße unabhängig. In vollständig verknüpften Netzwerken ist auch die Untersuchung von sehr kleinen Netzwerken mit $N \leq 128$ schon sinnvoll möglich und führt zu identischen Ergebnissen.

Die präsentierten Werte für die Einzugsbereiche sollten sich daher problemlos extrapolieren lassen und können auch für große Netzwerke verwendet werden.

Allerdings treten doch in einzelnen Simulationen wegen zu kleiner Anzahl von ausgewerteten Testmustern „Ausreißer“ auf (siehe etwa Abb. 4, $\alpha = 0.25$). Es wäre daher wünschenswert, das hier präsentierte Verhalten der verdünnten Netzwerke durch Simulationen an viel größeren Systemen verifizieren zu können.

3.5 Die Energielandschaft

Im Hopfield-Modell gelingt mit der statistischen Mechanik die Berechnung der Minima des Hamiltonoperators $E = -(1/2N)\sum_{i,j} J_{ij} S_i S_j$ des Spinglases mit den Hebb-Kopplungen $J_{ij} = (1/N)\sum_{j \neq i} \xi_i^\mu \xi_j^\mu$. Wegen der Symmetrie der Kopplungen sind die



Finite-size scaling der Einzugsbereiche in stark verdünnten Netzwerken ($c = 0.1$, $N = 256, 400, 512$). $\alpha = 0.1$ $\bullet$, 0.2 $\circ$, 0.3 $\diamond$, 0.4 $\star$.
Zusätzlich ist das Verhalten von m_c für vollständig verknüpfte Netzwerke mit $\alpha = 0.2$ gezeigt ($N = 192 \ldots 512$) $\oslash$.

Abbildung 5: Finite-size scaling von m_c für parallele Dynamik.

ermittelten Minima der Energie E gleichzeitig auch stabil unter der seriellen oder parallelen Dynamik.

Die Analyse der Energienminima des Hopfield-Modells zeigt aber [Amit *et. al.* 85] [Gardner 86] [Amit *et. al.* 87], daß die Dynamik durch zusätzliche, unerwünschte *spurious states*, insbesondere Mischzustände (Linearkombinationen der Sollmuster) und spinglas-artige Zustände, behindert wird. Außerdem sind Zyklen als Attraktoren möglich.

Symmetrische Kopplungen $J_{ij} = J_{ji}$ können im Gardner-Netzwerk bei Verwendung der einfachen iterativen Lernregeln nicht garantiert werden (und lassen sich nur durch die Verwendung der Lernregel mit $\Delta J_{ij} = (\epsilon_{i\mu} + \epsilon_{j\mu})\xi_i^\mu \xi_j^\mu$ erzwingen). Natürlich kann nach der Lernphase für jede Konfiguration $\{S_i\}$ des Netzwerks wieder die Energie

$$E = -\frac{1}{2N} \sum_{i,j} \frac{J_{ij}}{\|J_i\|} S_i S_j$$

berechnet werden. Für ein Sollmuster ξ_i^μ fällt E trivial mit den erreichten Stabilitäten zusammen, $E = -(1/2N)\sum_i \kappa_{i\mu}$. Allerdings werden die Minima dieser Energie nicht mehr genau mit den dynamisch stabilen Konfigurationen übereinstimmen, wenn die Kopplungen einen asymmetrischen Anteil aufweisen.

Trotzdem ist es interessant, die durch die Anwendung der iterativen Lernregel erzeugte Energielandschaft zu untersuchen. Zum einen sind die nach iterativem Lernen resultierenden Kopplungsmatrizen (in vollständig vernetzten Netzwerken) fast symmetrisch. Zum anderen zeigen die Rechnungen von [Nardulli & Pasquariello 90], daß sich die optimalen Werte der Stabilitäten auch mit der symmetrischen Lernregel erreichen lassen.

In den hier durchgeführten Simulationen wurden auch mit der einfachen Lernregel in vollständig verknüpften Netzwerken durchweg Werte von $\eta \approx 0.9 \ldots 0.95$ erreicht. Das bedeutet, daß die Minima der Energie zwar nicht genau mit dynamisch stabilen Zuständen übereinstimmen müssen, aber zumindest stark mit diesen korreliert sind. Tatsächlich fand sich in den Simulationen kein Sollmuster, daß in mehr als zwei Positionen (Spins) vom benachbarten lokalen Minimum von E abwich.

Ein Überblick über die Gestalt der Energielandschaft rund um ein jeweils willkürlich herausgegriffenes Sollmuster ist in Abbildung 6 als Funktion des Überlapps mit dem Muster und für verschiedene Speicherkapazitäten α dargestellt. Es ist natürlich unmöglich, die Energie für alle Konfigurationen zu berechnen, die sich von einem Muster in $1, 2, \ldots k$ Positionen unterscheiden: Die Anzahl dieser Zustände wächst mit $\binom{N}{k}$. Deshalb wurden für jedes k Mittelwerte über etwa 1000 zufällig ausgewählte Konfigurationen gebildet. Gleichzeitig wurden für jedes k auch die Minima und Maxima der Energien der untersuchten Konfigurationen gespeichert. Auf die Suche nach den globalen Minima/Maxima in einer Entfernung k vom jeweiligen Muster wurde jedoch verzichtet. (Die Probleme einer derartigen Untersuchung im binären Netzwerk werden von [Fontanari & Köberle 90] geschildert.)

Es ist gut zu erkennen, daß das Minimum von E bei den gezeigten Daten tatsächlich mit dem jeweiligen Sollmuster übereinstimmt. Mit der gewählten Skala kann gleichzeitig die erreichte Stabilität des jeweiligen Sollmusters bequem abgelesen werden. Die Minima von E sind für kleine und mittlere Werte der Speicherkapazität $\alpha \approx 0.05 \ldots 0.4$ sehr deutlich ausgeprägt, für höhere Speicherkapazitäten ist deutlich das Auftreten lokaler Minima — und damit von spurious states — zu erkennen. (Dabei ist zu beachten, daß die gezeigten Punkte die über jeweils etwa 1000 Konfigurationen gemittelte Energie im Netzwerk darstellen.)

Die detaillierte Untersuchung der spurious states ist in den Simulationen problematisch und erfordert in jedem Fall einen enormen Rechenaufwand:

Wenn bei der Iteration eines Testmusters ein dynamisch stabiler Zustand auftritt, der nicht mit dem gewünschten Sollmuster übereinstimmt, so muß zunächst untersucht werden, ob der Endzustand nicht einfach ein anderes Sollmuster ist. Dieses Erkennen des „falschen“ Sollmusters wurde jedoch nur in sehr kleinen Netzwerken (mit $N < 64$) beobachtet. In größeren Netzwerken konnte auch bei sehr kleinem Anfangsüberlapp der Testmuster nur äußerst selten die Konvergenz gegen ein anderes Sollmuster ermittelt werden.

Dagegen treten unter paralleler Dynamik Zyklen häufig als Attraktoren auf. Fast alle ermittelten Zyklen hatten die Länge 2, Zyklen der Länge 4 wurden nur in etwa 0.1% aller Zyklen beobachtet, längere Zyklen nie. Unter Verwendung der seriellen Dynamik wurden keine Zyklen beobachtet (bei asymmetrischen Kopplungen sind Zyklen auch mit serieller Dynamik möglich). Auch bei Verwendung der später zu besprechenden modifizierten Dynamiken mit memory-terms (27) konnten keine Zyklen ermittelt werden.

Die nächste Frage betrifft die Existenz von dynamisch stabilen Mischzuständen der Sollmuster. Die Wahrscheinlichkeit für das Auftreten derartiger Mischzustände ist jedoch (zumindest für $\alpha < 1.0$) extrem gering. Der führende Term entsteht durch eine Linearkombination von drei Sollmustern, und ist proportional zu $\binom{P}{3}\,(6/8)^N$, weil nur 6 der 8 möglichen Einstellungen von ξ_i^a, ξ_i^b und ξ_i^c wieder einen Wert ± 1 ergeben.

Keiner der in den Simulationen aufgefundenen spurious states konnte als Mischzustand identifiziert werden. Dabei wird selbstverständlich nicht versucht, die stabile Endkonfiguration mit allen möglichen Linearkombinationen von 3, 5, ... Mustern zu vergleichen. Vielmehr gelingt die Klassifikation über die Energie des spurious state: Wegen der Linearität der Summation der lokalen Felder ergibt sich für einen Mischzustand dieselbe Energie wie für die gespeicherten Sollmuster.

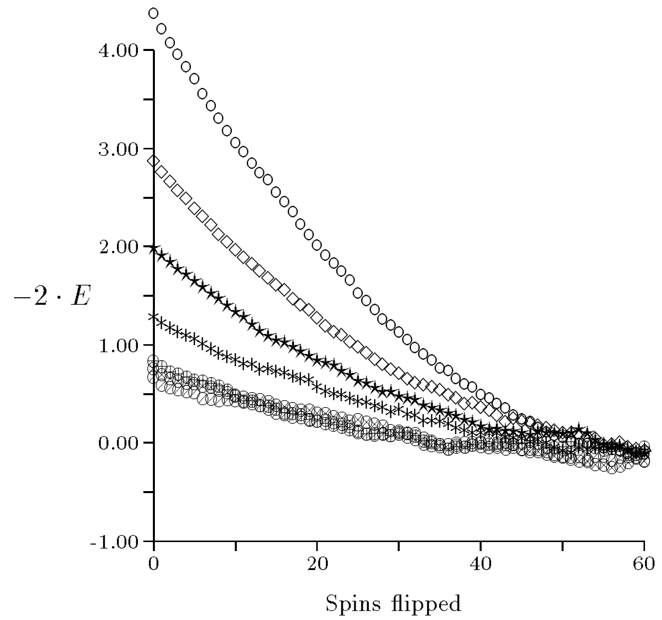
Aus dieser Beziehung resultiert die Darstellung in Abbildung 7. Dort ist die mittlere Energie der Sollmuster und der dynamisch stabilen spurious states als Funktion des Anfangsüberlapps m_0 mit dem jeweiligen Sollmuster für verschiedene Werte von $\alpha = 0.2 \ldots 0.6$ aufgetragen. Alle Datenpunkte sind das Ergebnis einer Mittelung über eine große Anzahl von Testmustern.

Die Energien der Sollmuster sollten natürlich vom Anfangsüberlapp völlig unabhängig sein. Die abgebildeten Daten für die Sollmuster entstanden durch die Mittelung der Energie der vom Netzwerk richtig erkannten Zustände. Es ist zu beachten, daß mit den iterativen Lernregeln die Einstellung wirklich identischer Stabilitäten für die Sollmuster sehr schwierig ist; die nach dem Lernen erreichten Stabilitäten weichen um bis zu $\Delta E \approx 0.1$ voneinander ab.

Weit im Inneren der Einzugsbereiche sind die Werte daher praktisch unabhängig vom Anfangsüberlapp. Nahe der Grenze der Einzugsbereiche werden jedoch einige Muster häufiger erkannt als andere, und daher werden die kleinen Unterschiede in den Stabilitäten der verschiedenen Sollmuster deutlich. ($N = 128$, $\alpha = 0.2\ \otimes$, $\alpha = 0.4\ \diamond$, $\alpha = 0.6\ \bullet$). Außerhalb der Einzugsbereiche sind die Energien für die Sollmuster nicht eingezeichnet, weil diese vom Netz auch nicht mehr erkannt wurden.

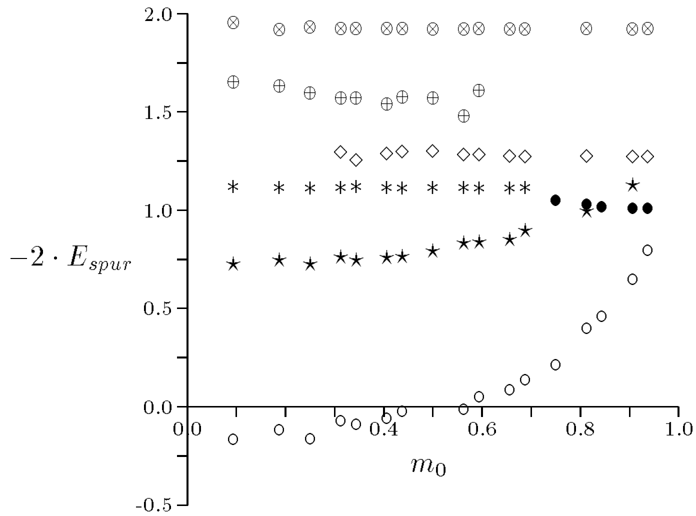
Die ermittelten Energien der spurious states sind für $\alpha = 0.2\ \oplus$ fast konstant (also unabhängig vom Anfangsüberlapp) und nur wenig größer als die Energien der Sollmuster. Außerhalb der Einzugsbereiche zeigen auch die Simulationen für $\alpha = 0.4\ \star$ und $0.6\ \circ$ dieses Verhalten. Allerdings zeigt sich zusätzlich, daß die Energien der spurious states sich an der Grenze der Einzugsbereiche an die Energien der Sollmuster annähern. Dies deutet darauf hin, daß die Energielandschaft (wenigstens oberhalb von $\alpha \approx 0.4$) die Struktur eines *Zentralmassivs* bekommt: Rund um das globale Minimum (das Sollmuster) gibt es eine Anzahl von lokalen Minima mit langsam anwachsenden Energien.

Für $\alpha = 0.4$ ist zusätzlich die Energie der unter Verwendung der memory-term Dynamik (27) erreichten spurious states gezeigt. Diese liegt viel näher an der Energie



Energie der Konfigurationen $\xi_i^{\mu,r}$ um ein gespeichertes Muster ξ_i^{μ} als Funktion des Überlapps und α.
($N = 128$, $\alpha = 0.05$ $\circ$, 0.11 $\diamond$, 0.20 $\star$, 0.40 $*$, 0.72 $\oplus$, 0.82 $\otimes$, 0.92 $\oslash$).

Abbildung 6: Energien $E = -(1/2N)\sum_i h_i S_i$ der Konfiguration $\{S\}$ als Funktion des Überlapps mit einem Sollmuster ξ_i^{μ} für verschiedene Werte $\alpha = 0.05, \ldots 0.92$.



Energie $E = \sum_{i,j} J_{ij} S_i S_j$ der unerwünschten dynamisch stabilen Zustände als Funktion des Anfangsüberlapps mit einem Muster ξ_i^{μ}. Siehe Text. ($N = 128$, $\alpha = 0.2$ E_{spur} $\oplus$, E_{pat} $\otimes$, $\alpha = 0.4$ E_{spur} $\star$, E_{pat} $\diamond$, $E_{spur,memdyn}$ $*$, $\alpha = 0.6$ E_{spur} $\circ$, E_{pat} $\bullet$.

Abbildung 7: Energien $E = -(1/2N)\sum_i h_i S_i$ der spurious states bei paralleler Dynamik als Funktion des Anfangsüberlapps der Testmuster $\xi_i^{\mu,r}$ mit ξ_i^{μ}.

der Sollmuster als die Energie der spurious states unter paralleler Dynamik. Im Fehlen der Datenpunkte oberhalb von $m_0 \approx 0.7$ zeigt sich, daß die memory-term Dynamik außerdem größere Einzugsbereiche ermöglicht, siehe Abschnitt 3.7.

3.6 Phase-Space-Gardening

Unter dem Schlagwort *phase-space-gardening* versteht man Techniken, um die Energielandschaft eines Systems gezielt zu modellieren. In Spinglas-Netzen stellt sich dabei die Frage, ob die Größe der Einzugsbereiche der Sollmuster beim Lernen gezielt eingestellt werden kann und ob ihre Gestalt zu beeinflussen ist.

Dazu kommen vor allem zwei Methoden in Frage. Eine besondere Form der iterativen Lernregel, das sogenannte *learning with noise* lernt nicht nur die Sollmuster selbst, sondern stellt die Kopplungen so ein, daß auch leicht gestörte Versionen der Sollmuster in einem Zeitschritt in das jeweilige Sollmuster wandern (unter paralleler Dynamik). Einige interessante Resultate in Bezug auf die Speicherung von Worten in derartigen Netzwerken werden von [Gardner 89c] diskutiert. Das Ausmaß, bis zu dem der Phasenraum mit dieser Lernregel modelliert werden kann, ist aber bisher noch nicht untersucht worden: Learning with noise ist sehr aufwendig, da dem Netzwerk zu jedem Muster auch viele leicht gestörte Versionen angeboten werden müssen.

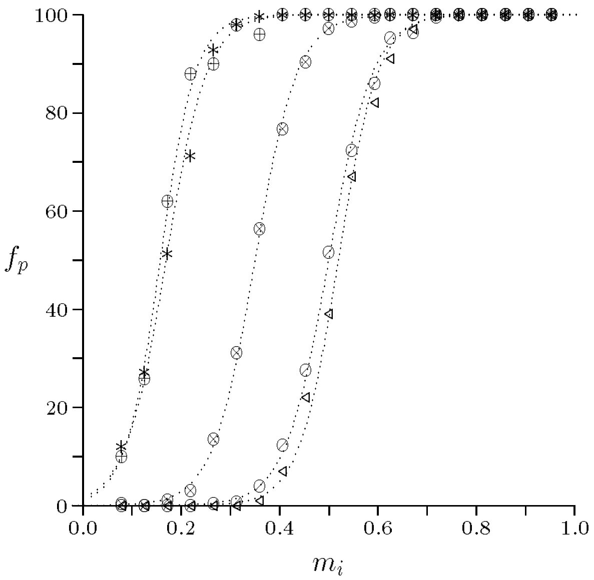
Die einfachere Möglichkeit ist, die gewünschten Stabilitäten beim gewöhnlichen iterativen Lernen nicht alle gleich groß zu wählen, sondern für jedes Sollmuster seine individuelle Stabilität $\kappa_{i\mu}$ zu fordern. Jedes Muster wird dann gelernt, wenn die erreichte Stabilität kleiner als $\kappa_{i\mu}$ ist. Die Konvergenz der Lernregeln ist weiterhin gesichert, solange überhaupt eine Lösung für die Wahl der Synapsen existiert (siehe Anhang B).

Im Prinzip sollte der Einzugsbereich jedes Muster mit einer derartigen Lernregel individuell eingestellt werden können. Natürlich ist zu erwarten, daß die Gestalt der Einzugsbereiche (näherungsweise isotrop) sich nicht ändert, sondern nur deren Radius.

Die Ergebnisse einiger Simulationen mit einer derartigen Lernregel sind in den Abbildungen 8 bis 10 für Speicherkapazitäten $\alpha = 0.2$, 0.4 und 0.6 dargestellt. Die Stabilitäten einiger Muster wurden jeweils willkürlich zu $\kappa_1 = 2.0$ und $\kappa_2 = 1.5$ gewählt, für die übrigen Muster wurde der für Sättigung noch mögliche Wert gefordert. Zur Auswertung wurde dann für jedes Muster ξ_i^μ einzeln sein Einzugsbereich durch Mittelung über viele Testmuster $\xi_i^{\mu,r}$ berechnet.

Es gelingt tatsächlich, die Einzugsbereiche der Muster individuell zu wählen. Durch die Einstellung einer Mindeststabilität $\kappa_1 = 2.0$ für einige Muster ist es selbst bei $\alpha = 0.6$ problemlos möglich, für diese Muster fast optimale Einzugsbereiche zu erzeugen, während die übrigen Muster nur sehr kleine Einzugsbereiche aufweisen, siehe Abbildung 10.

Die Simulationen zeigen, daß die Größe der Einzugsbereiche der Muster mit großen Mindeststabilitäten $\kappa_1 = 2.0$ und $\kappa_2 = 1.5$ fast nicht von der Speicherkapazität α — und damit von den Stabilitäten der übrigen Muster — abhängt. Daher läßt sich tatsächlich für jedes Muster sein Einzugsbereich fast unabhängig von anderen Mustern



Phase-space-gardening: Anteil f_p (%) perfekt erkannter Muster als Funktion des Anfangsüberlapps im gesättigten Netzwerk mit $\alpha = 0.2$ und mehreren Gruppen von Mustern mit folgenden Stabilitäten: $\kappa_1 = 2.0$ $*$, $\oplus$; $\kappa_2 = 1.5$ $\otimes$; $\kappa_3 = 1.0$ $\oslash$, $\triangleleft$; $(N = 128)$.

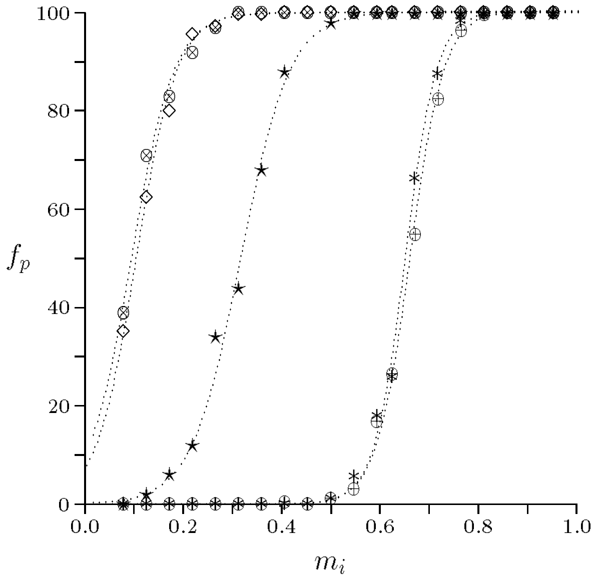
Abbildung 8: Phase-space-gardening für $\alpha = 0.2$.

einstellen, solange das Netzwerk insgesamt unterhalb der Sättigung bleibt.

Die Abbildungen zeigen für identische Werte von κ_i mehrere, leicht gegeneinander verschobene Kurven: Dies ist darauf zurückzuführen, daß sich mit der verwendeten Lernregel in den kleinen Netzwerken die Stabilitäten für verschiedene Muster nicht exakt identisch einstellen lassen. Außerdem sind die Fluktuationen in diesen Simulationen relativ groß, weil nur eine relativ kleine Anzahl von Testmustern untersucht wurde. Es ist aber auch zu erkennen, daß die aus diesen Darstellungen zu ermittelnden Werte für die Größe der Einzugsbereiche trotz der geringen Anzahl von Testmustern schon recht genau ermittelt werden können und daß die kleinen Differenzen der Stabilitäten auch zu nur kleinen Differenzen in der Größe der Einzugsbereiche führen: So stimmen für $\alpha = 0.4$ (Abbildung 9) sowohl für $\kappa = 1.5$ als auch für $\kappa = 0.9$ die entsprechenden Werte von m_c sehr gut (besser als $\Delta m_c \approx 0.02$) überein.

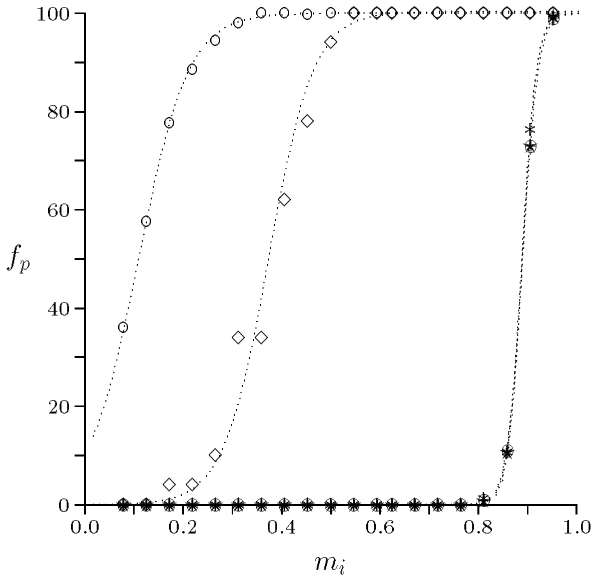
Die Einstellung individueller Einzugsbereiche für verschiedene Muster kann zum Beispiel dazu benutzt werden, einen inaktiven Zustand des Netzwerks hervorzuheben (Auszeichnung eines Musters durch Wahl eines großen Einzugsbereiches, in den auch alle spurious states hineinlaufen).

Es muß angemerkt werden, daß auch der klassische Algorithmus aus Abschnitt 2.3.2 in dieser Weise erweitert werden kann. Dazu reicht es aus, für jedes gespeicherte Muster zusätzlich einen Faktor $0 \leq a_\mu \leq 1$ anzugeben, und das Testmuster $\xi^{(r)}$ dem Muster mit dem größten Überlapp $m_\mu = (a_\mu/N)\sum_j \xi_j^\mu \xi_j^{(r)}$ zuzuordnen



Anteil f_p (%) perfekt erkannter Muster als Funktion des Anfangsüberlapps im gesättigten Netzwerk mit $\alpha = 0.4$ und mehreren Gruppen von Stabilitäten: $\kappa_1 = 2.0$ $\diamond$, $\otimes$; $\kappa_2 = 1.5$ $\star$; $\kappa_3 = 0.9$ $*$, $\oplus$; ($N = 128$).

Abbildung 9: Phase-space-gardening für $\alpha = 0.4$.



Anteil f_p (%) perfekt erkannter Muster als Funktion des Anfangsüberlapps im gesättigten Netzwerk mit $\alpha = 0.6$ und mehreren Gruppen von Stabilitäten: $\kappa_1 = 2.0$ $\circ$; $\kappa_2 = 1.5$ $\diamond$; $\kappa_3 = 0.6$ $\star$, $*$, $\oplus$; ($N = 128$).

Abbildung 10: Phase-space-gardening für $\alpha = 0.6$.

3.7 Asynchrone Dynamik, Memory-Terme

Obwohl eine asynchrone, serielle Dynamik für die Modellierung biologischer Systeme viel realistischer erscheint als die parallele Dynamik, ist die Konstruktion eines theoretischen Modells der dynamischen Eigenschaften der seriellen Dynamik stark erschwert. Schon nach dem Umschalten nur eines einzigen Neurons, ändern sich die lokalen Felder aller anderen Neuronen. Die für parallele Dynamik abgeleiteten Formeln gelten also für asynchrone Dynamik nur bis zum Flip eines Neurons. Zusätzlich kommt hinzu, daß die Reihenfolge der Spinflips natürlich nicht festgelegt sein muß.

Die Ergebnisse von Simulationen großer Ising-Systeme lassen allerdings erwarten, daß die Resultate nicht von den Details der Reihenfolge des Update abhängen [Koehler *et. al.* 89].

Die Simulationen zeigen, daß das Verhalten der Netzwerke unter serieller Dynamik (Update in der Reihenfolge $1, \ldots, N$) sich gegenüber der Verwendung der parallelen Dynamik kaum ändert. Insbesondere läßt sich wieder der Übergang vom Verhalten des vollständig verknüpften Modells zum verdünnten beobachten.

Die Einzugsbereiche der Muster werden mit steigender Verdünnung immer größer und scheinen im Grenzfall starker Verdünnung wieder optimal zu sein, sind allerdings kleiner als unter Verwendung paralleler Dynamik. Der Übergang von der synchronen parallelen zur asynchronen seriellen Dynamik wirkt sich aber nur schwach aus.

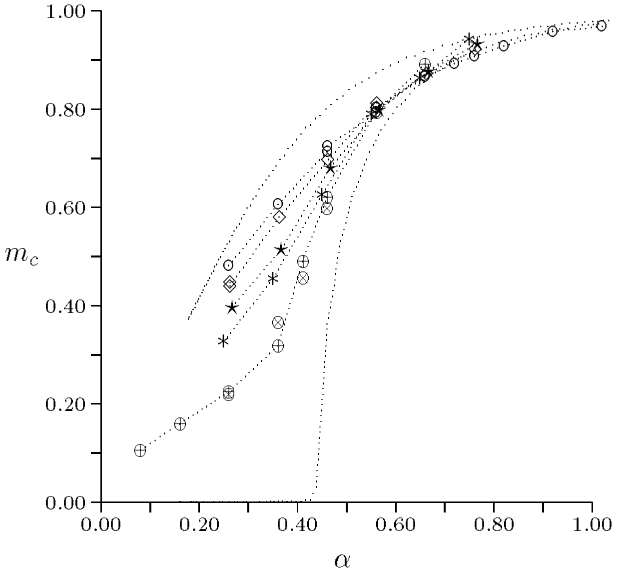
Leider erlauben die numerischen Daten keine Entscheidung darüber, ob die optimalen Einzugsbereiche wieder bei $\alpha \leq 0.41$ auftreten. Auch hier wurde wieder auf Simulationen bei geringer Speicherkapazität α verzichtet. Die Einzugsbereiche dürften bei $\alpha \to 0$ immer größer werden und schließlich einen Wert nahe $1/\sqrt{N}$ erreichen. Die für die vollständig verknüpften Netze ermittelten Werte von m_c stimmen mit den von [Forrest 88] angegebenen Daten gut überein.

Dagegen zeigt die Verwendung einer Dynamik mit Gedächtnistermen starke Veränderungen in der Größe der Einzugsbereiche. Die benutzte *memory2*- Dynamik ist

$$S_i(t+1) = \mathrm{sgn}\Big(\frac{1}{2}h_i(t) + \frac{1}{2}h_i(t-1)\Big). \tag{33}$$

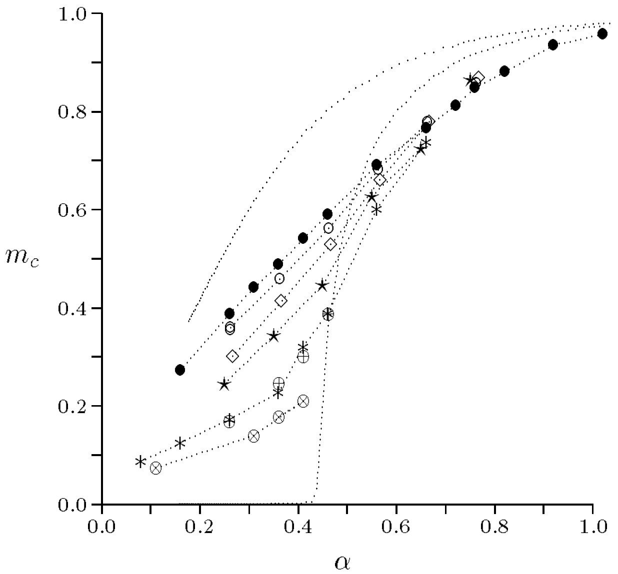
Zum Start der Dynamik mit einem neuen Testmuster wird $S_i(t-1) = S_i(t) = \xi_i^{\mu,r}$ gesetzt. Die Einzugsbereiche sind für alle Werte von α größer als die unter paralleler oder serieller Dynamik ermittelten, siehe Abbildung 12.

Dies ist zum einen darauf zurückzuführen, daß die memory-Dynamik Zyklen unterdrückt, zum anderen auf das Ausmitteln von winzigen lokalen Energieminima. Die unerwünschten stabilen Endzustände unter paralleler (oder serieller) Dynamik werden normalerweise schon nach dem Umschalten weniger Neuronen instabil, und die Verwendung der memory-term Dynamik kann daher eine deutliche Verbesserung bringen. Oberhalb von $\alpha \approx 1.0$ sind die Einzugsbereiche allerdings auch mit dieser Dynamik extrem klein.



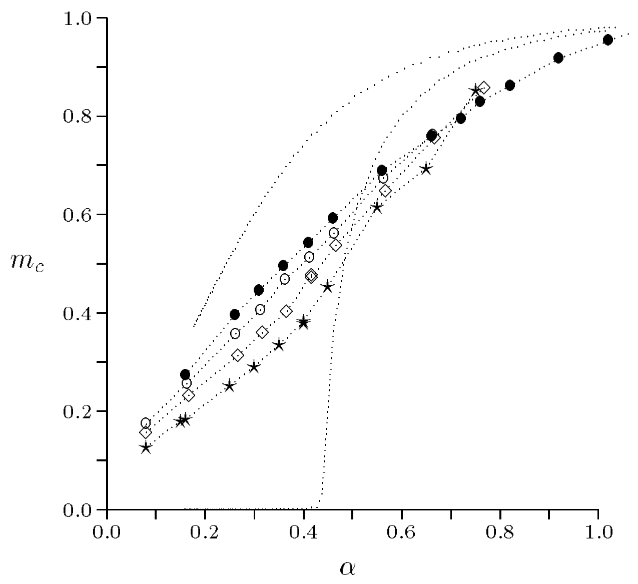
Numerisch ermittelte Einzugsbereiche m_c als Funktion von α in verdünnten Netzwerken mit serieller Dynamik. ($N = 128$, $c = 1.0$ $\circ$, $c = 0.8$ $\diamond$, $c = 0.6$ $\star$, $= 0.4$ $*$) ($N = 256$, $c = 0.2$ $\oplus$) ($N = 400$, $c = 0.2$ $\otimes$).

Abbildung 11: Einzugsbereiche m_c in verdünnten Netzwerken mit serieller Dynamik, verglichen mit m_F und m_S.



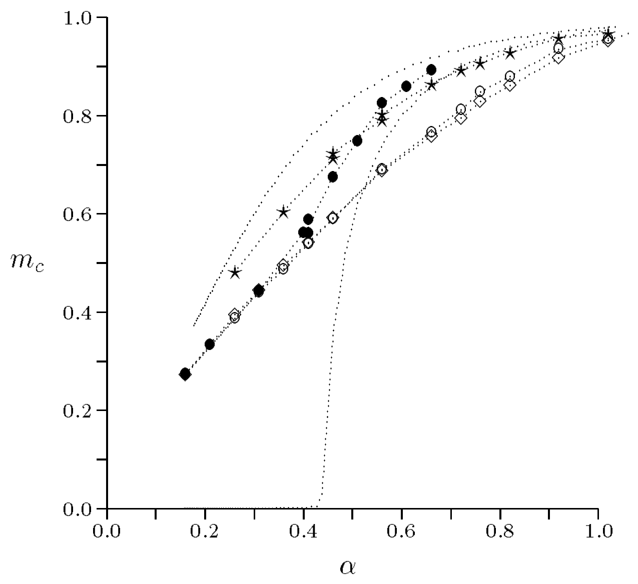
Numerisch ermittelte Einzugsbereiche m_c in verdünnten Netzwerken mit *memory*-Dynamik. ($N = 128$, $c = 1.0$ $\bullet$, $c = 0.8$ $\circ$, $c = 0.6$ $\diamond$, $c = 0.4$ $\star$) ($N = 256$ $c = 0.2$ $*$) ($N = 400$ $c = 0.2$ $\oplus$, $c = 0.1$ $\otimes$).

Abbildung 12: Einzugsbereiche m_c in verdünnten Netzwerken unter Dynamik mit *memory terms*, verglichen mit m_F und m_S.



Numerisch ermittelte Einzugsbereiche m_c als Funktion von α in verdünnten Netzwerken mit *memory-3*-Dynamik. ($N = 128$ $c = 1.0$ $\bullet$, $c = 0.8$ $\circ$, $c = 0.6$ $\diamond$, $c = 0.4$ $\star$).

Abbildung 13: Einzugsbereiche m_c in verdünnten Netzwerken unter Dynamik mit *memory-terms* (3), verglichen mit m_F und m_S.



Vergleich der numerisch ermittelten Einzugsbereiche in vollständig verknüpften Netzwerken als Funktion von α. ($N = 128$)
parallele Dynamik $\bullet$,
memory-Dynamik $\circ$,
memory-3-Dynamik $\diamond$,
serielle Dynamik $\star$.

Abbildung 14: Einzugsbereiche m_c in vollständig vernetzten Netzwerken als Funktion von α für parallele-, memory2- und memory3-Dynamik, verglichen mit m_F und m_S.

Der Übergang zu weiteren memory-Termen (*memory3-Dynamik*),

$$S_i(t+1) = \mathrm{sgn}\Big(\frac{1}{3}h_i(t) + \frac{1}{3}h_i(t-1) + \frac{1}{3}h_i(t-2)\Big). \tag{34}$$

zeigt keine nennenswerte Verbesserung der Eigenschaften des Netzwerks, siehe Abbildung 13. Deshalb wurde auf Simulationen in stärker verdünnten Netzwerken mit memory3-Dynamik verzichtet: Die Einzugsbereiche sind dort ja auch bei Verwendung der einfacheren dynamischen Regeln sehr groß und der Einsatz komplizierterer Dynamik ist nicht notwendig. In Abbildung 12 sind auch einige Simulationen bei kleiner Speicherkapazität $\alpha \leq 0.2$ dargestellt; man erkennt, daß die Einzugsbereiche nicht die maximale Größe erreichen, sondern durch die zufälligen Korrelationen der Testmuster mit den gespeicherten Mustern begrenzt werden.

Die nächste Abbildung 14 zeigt die aus allen Simulationen erhaltenen Ergebnisse im Vergleich der parallelen und seriellen Dynamik mit memory2- und memory3-Dynamik, und zwar für vollständig verknüpfte Netzwerke. Es ist deutlich zu erkennen, daß die Einzugsbereiche nur für $\alpha > 0.4$ vergrößert werden. Die Einzugsbereiche der seriellen Dynamik sind unterhalb von $\alpha \leq 0.4$ am kleinsten, oberhalb von $\alpha \approx 0.4$ aber sogar etwas größer als mit paralleler Dynamik.

Auch in verdünnten Netzwerken bewirkt die Verwendung der Dynamik (33) eine Verbesserung der Einzugsbereiche. Dies ist zum Beispiel in Abbildung 12 leicht daran zu erkennen, daß die Kurve $m_c(\alpha)$ im Gebiet unterhalb der für das Modell (30) berechneten Kurve verläuft. Wie stark sich die Einzugsbereiche im verdünnten Netzwerk durch die Verwendung der memory2-Dynamik (27) vergrößern, (oder ob der Übergang zu optimalen Einzugsbereichen auch wieder bei $\alpha = 0.41$ erfolgt), kann mit den hier vorgestellten Simulationen leider nicht geklärt werden. Um das Verhalten der Netzwerke in diesem Bereich genauer bestimmen zu können, ist die Untersuchung wesentlich größerer Netzwerke nötig.

3.8 Konvergenzgeschwindigkeit

Die Beschreibung der dynamischen Eigenschaften der neuronalen Modelle erfordert neben der Untersuchung der Größe und Gestalt der Einzugsbereiche, sowie der Klassifikation unerwünschter Attraktoren auch noch eine Analyse der Konvergenzgeschwindigkeit.

Unter diesem Begriff soll hier die Anzahl der Iterationen der jeweiligen Dynamik bis zum Erreichen eines Fixpunktes oder eines Zyklus verstanden werden. Die Bezeichnung soll wegen ihrer Prägnanz trotz der offensichtlichen Mängel verwendet werden.

Für jedes neuronale Modell mit Attraktordynamik ist die Anzahl der Iterationen bis zum Erreichen eines dynamisch stabilen Zustandes offenbar ein wichtiger Parameter: Wie schnell erreicht das Netzwerk ausgehend von einer Startkonfiguration den nächstgelegenen Attraktor?

Insbesondere für Anwendungen neuronaler Netzwerke ist dafür die mittlere Anzahl von Iterationen bis zum Erreichen eines Fixpunkts die naheliegende — und zudem allein zugängliche — Größe. Wenn die Dynamik des Netzwerks in einer Testkonfiguration $\{S_i(0)\}$ gestartet wird, ist ja von außen weder abzusehen, wie schnell das Netzwerk dieses Testmuster richtig erkennen wird oder ob es gar einen spurious state oder einen Zyklus erreicht.

Ein Fixpunkt in der Dynamik des Netzwerks läßt sich dabei leicht ermitteln, indem die Zustände $\{S_i(t)\}$ und $\{S_i(t-1)\}$ verglichen werden. Für eine Dynamik mit memory-terms muß dieser Vergleich natürlich entsprechend auch auf $\{S_i(t-2)\}$ usw. ausgedehnt werden.

Dagegen ist es für das Netzwerk (lokal) völlig unmöglich, Zyklen der Dynamik zu erkennen. Auch in den Simulationen ist es sehr aufwendig, nach Zyklen zu suchen. Die einfachste Methode, nämlich der Vergleich von $\{S_i(t)\}$ mit allen vorangegangenen Zuständen $\{S_i(t-m)\}$, reicht zwar aus, um Zyklen zu entdecken, ist aber auch sehr aufwendig: Die Anzahl der nach jeder Iteration durchzuführenden Vergleiche steigt linear mit der Anzahl der schon ausgeführten Iterationen an. Ein besserer Algorithmus besteht darin, die erreichten Konfigurationen zu ordnen (etwa indem das Bitmuster der $S_i(t)$ als Integer betrachtet wird), und die jeweils aktuelle Konfiguration $\{S_i(t)\}$ des Netzwerks mit der größten bisher erreichten zu vergleichen. Da die Dynamik des Netzwerks deterministisch ist, liegt ein Zyklus vor, wenn die größte Konfiguration ein zweites Mal erreicht wird.

Zum Glück treten kurze Zyklen nur unter Verwendung paralleler Dynamik häufig auf, und in den Simulationen konnte deshalb auf die Entdeckung von Zyklen weitgehend verzichtet werden. Die Häufigkeit von Zyklen wurde nur für parallele Dynamik und kleine Netzwerke ($N = 128$) untersucht. Es zeigt sich, daß mehr als 99.8% aller Zyklen unter Verwendung paralleler Dynamik die Länge 2 aufweisen, alle übrigen Zyklen hatten die Länge 4. Zyklen mit ungerader Anzahl von Iterationen oder mit einer Länge von mehr als 4 Iterationen konnten in den Simulationen überhaupt nicht beobachtet werden.

Dabei ist allerdings zu beachten, daß auch sehr lange Zyklen auftreten können, die das Netzwerk durch große Teile seines Phasenraums führen, siehe [Gardner 89a]. Derartige lange Zyklen können mit Hilfe von Simulationen nicht nachgewiesen werden. Tatsächlich mußte in den Simulationen für Testmuster mit kleinen Anfangsüberlapps m_0 häufig eine sehr große Anzahl von Iterationen der Dynamik ausgeführt werden, ohne daß ein Fixpunkt erreicht werden konnte. Die Zeitentwicklung der Dynamik muß daher abgebrochen werden, wenn ein Testmuster nach vielen Iterationen weder einen Fixpunkt noch einen kurzen Zyklus erreicht hat.

Es ist natürlich möglich, die Dynamik der Netzwerke von vornherein willkürlich nach einer konstanten Anzahl von Iterationen abzubrechen, unabhängig davon, ob ein dynamisch stabiler Zustand erreicht wurde oder nicht. Insbesondere wird in einigen Modellen die Dynamik auf eine einzige Iteration beschränkt (etwa im sparse-coding Netzwerk von Willshaw und Palm [Anderson & Rosenfeld 88]). Dies bedingt aber auch ganz andere Werte für die Größe der Einzugsbereiche und soll hier nicht weiter untersucht werden.

Die Untersuchung der Größe der Einzugsbereiche wird dadurch erleichtert, daß das

Modell (32) den Verlauf der Funktionen $f_p(m_0)$ in allen untersuchten Netzwerken — nahe oder fern der Sättigung und unabhängig von der verwendeten Dynamik — nach Anpassung seiner Parameter sehr genau beschreibt: Insbesondere werden in großen Netzwerken fast alle Testmuster mit Anfangsüberlapp $m_0 > m_c$ vom Netz korrekt erkannt werden. Und die Beziehung (32) ist gut erfüllt, obwohl die Testmuster mit mit kleinerem Anfangsüberlapp $m_0 < m_c$ entweder anderen Sollmustern zugeordnet werden können oder in spurious states oder in Zyklen enden. Man könnte daher erwarten, daß sich auch weitere dynamische Eigenschaften der Netzwerke derart einfach beschreiben lassen.

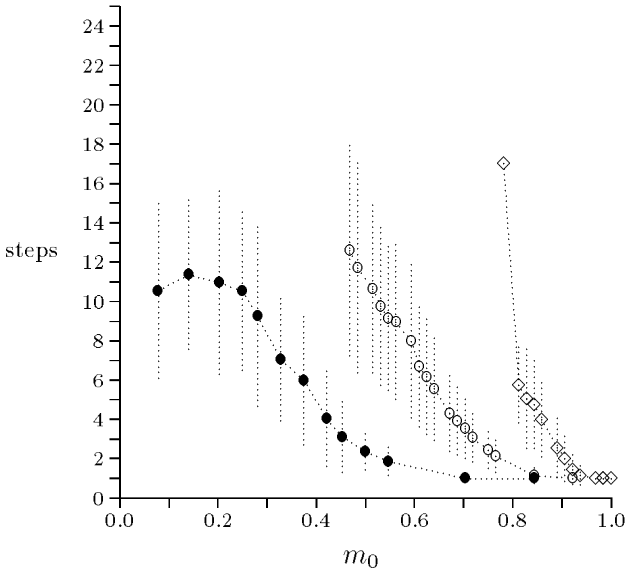
Die Simulationen zeigen für die Anzahl der Iterationen bis zum Erreichen eines Attraktors allerdings ein sehr viel komplexeres Verhalten, als das einfache Ergebnis für die Größe der Einzugsbereiche erwarten läßt.

Ein Bespiel für die Konvergenzgeschwindigkeit in vollständig verknüpften Netzwerken mit $N = 128$ unter paralleler Dynamik zeigt Abbildung 15 für $\alpha = 0.2$, 0.4 und 0.6. Die Iteration der Dynamik wurde dabei nach 25 Schritten willkürlich abgebrochen, und es wurden nur die Testmuster gewertet, die vorher schon einen Fixpunkt erreicht hatten.

Es ist deutlich zu erkennen, daß im Inneren der Einzugsbereiche (für $m_0 > m_c(\alpha)$) eine Iteration der parallelen Dynamik ausreicht, um die Sollmuster zu erreichen. Nahe der Grenze der Einzugsbereiche (das heißt für $m_0 \approx m_c$) beginnt die Anzahl der Iterationen bis zum Erreichen eines Fixpunktes anzusteigen — für kleine Werte der Speicherkapazität eher langsam, für $\alpha \geq 0.5$ dagegen recht steil. In Abschnitt 3.4 wurde zum Beispiel für $\alpha = 0.2$ ein Wert von $m_c(\alpha = 0.2) \approx 0.28$ ermittelt. Die Konvergenzgeschwindigkeit beginnt schon für $\alpha \approx 0.4$ — also ziemlich weit im Inneren der Einzugsbereiche der Muster — deutlich anzusteigen.

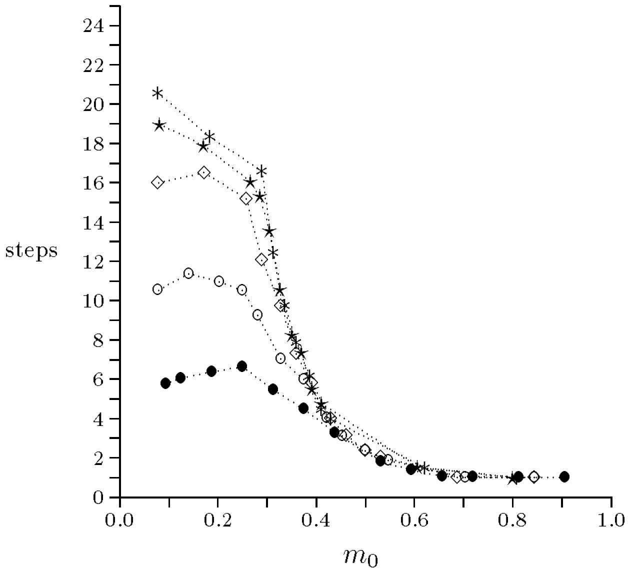
Die angegebenen Maximalwerte der Iterationen sind mit etwas Vorsicht zu verwenden, weil die Dynamik jeweils nach 25 Iterationen abgebrochen wurde. Die Wahrscheinlichkeit, daß ein Testmuster nach mehr als (etwa) 25 Schritten der Dynamik noch ein Sollmuster erreicht, ist allerdings extrem klein. Es ist auch interessant zu sehen, daß die Anzahl der Iterationen für sehr kleine Werte des Anfangsüberlapps m_0 wieder leicht abnimmt. Das ist darauf zurückzuführen, daß viele dieser Testmuster relativ schnell einen spurious state erreichen.

Die in der Abbildung dargestellten Fehlerbalken sind die aus den Daten ermittelten statistischen Fehler. Dabei ist natürlich zu beachten, daß die einzelnen Werte für die Konvergenzgeschwindigkeit offenbar nicht normalverteilt sind: Auch an der Grenze der Einzugsbereiche erreichen einige Testmuster das zugehörige Sollmuster schon nach sehr wenigen, etwa zwei bis vier, Iterationen, während die Anzahl der Iterationen zum Erreichen eines spurious state sehr unterschiedlich ist und die Zeitentwicklung anderer Testmuster abgebrochen werden muß. Da es aber keine Möglichkeit gibt, die dynamische Entwicklung eines Testmusters vorherzusagen, ist, wie oben erläutert, zunächst nur die Betrachtung der mittleren Anzahl von Iterationen sinnvoll. Für $\alpha = 0.6$ ist bei $m_0 = 0.78$ der statistische Fehler sehr klein, weil nur noch sehr wenige Testmuster



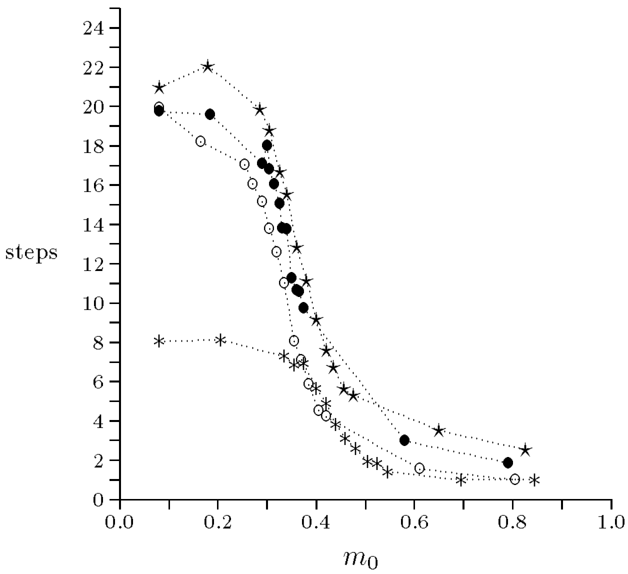
Anzahl der Iterationen bis zum Erreichen eines Fixpunktes im vollständig verknüpften gesättigten Netzwerk für parallele Dynamik als Funktion des Anfangsüberlapps m_0. $N = 128$, $\alpha = 0.2$ $\bullet$, $\alpha = 0.4$ $\circ$, $\alpha = 0.6$ $\diamond$.
Die extrem großen statistischen Fehler weisen darauf hin, daß die Werte nicht gut durch eine Normalverteilung beschrieben werden. Siehe Text.

Abbildung 15: Beispiel für die Anzahl der Iterationen der Dynamik bis zum Erreichen eines Fixpunktes für parallele Dynamik als Funktion von m_0 und α. $N = 128$, $\alpha = 0.2$, 0.4, 0.6



Anzahl der Iterationen bis zum Erreichen eines Fixpunktes für parallele Dynamik im vollständig verknüpften Netzwerk ($c = 1.0$) bei $\alpha = 0.2$. $N = 64$ $\bullet$, $N = 128$ $\circ$, $N = 256$ $\diamond$, $N = 400$ $\star$, $N = 512$ $*$.

Abbildung 16: Anzahl der Iterationen der Dynamik bis zum Erreichen eines Fixpunktes. $N = 64$ bis $N = 512$, $\alpha = 0.2$, $c = 1.0$



Anzahl der Iterationen bis zum Erreichen eines Fixpunktes im vollständig verknüpften gesättigten Netzwerk mit $N = 400$, $\alpha = 0.2$ als Funktion von m_0:
- $\circ$ parallele Dynamik,
- $\bullet$ memory-Dynamik,
- $\star$ memory3-Dynamik,
- $*$ serielle Dynamik.

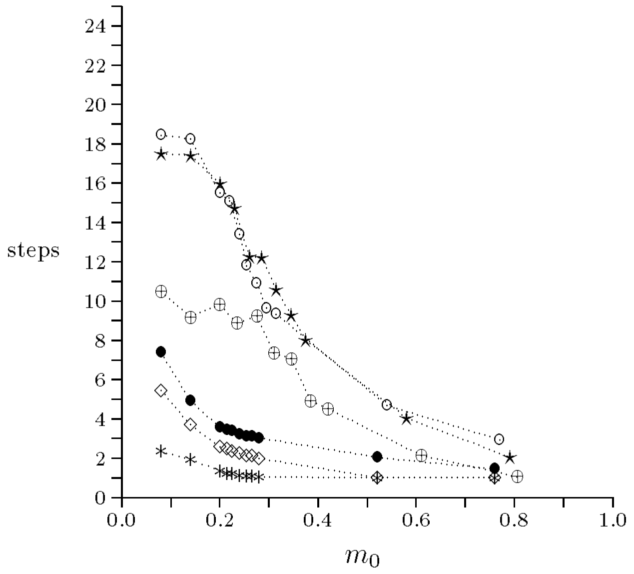
Abbildung 17: Anzahl der Iterationen der Dynamik bis zum Erreichen eines Fixpunktes: parallele, serielle, memory2, und memory3-Dynamik. $N = 400$, $c = 1.0$

überhaupt konvergierten, und dies in sehr ähnlicher Anzahl von Iterationen.

Für die Charakterisierung der dynamischen Eigenschaften der Netzwerke muß auch untersucht werden, wie sich die Konvergenzgeschwindigkeit als Funktion der Netzwerkgröße verhält. Der Einsatz der neuronalen Netzwerke ist offenbar unmöglich, wenn die Fixpunkte in großen Netzwerken nur sehr langsam erreicht werden. Es ist zu erwarten, daß der Anteil der Neuronen, die pro Iteration der Dynamik ihren Zustand ändern, im wesentlichen unabhängig von der Größe des Netzwerks ist.

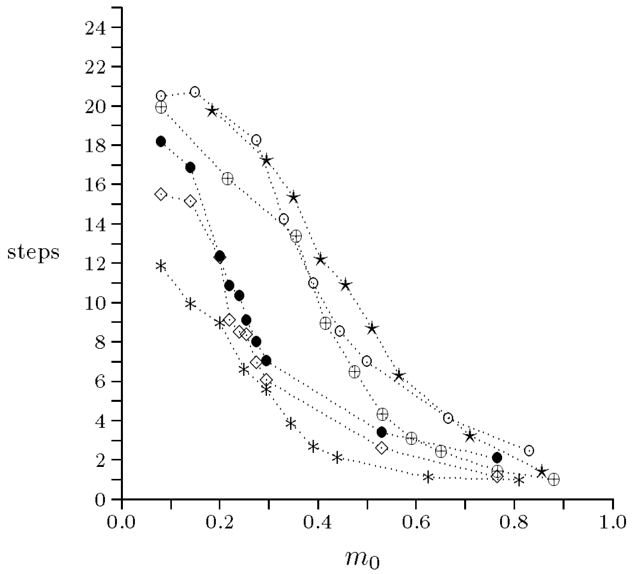
Die Abbildung 16 zeigt die in den Simulationen ermittelten Werte für die Konvergenzgeschwindigkeit in vollständig verknüpften, gesättigten Netzwerken für $N = 64$, 128, 256, 400 und $N = 512$ bei einer Speicherkapazität von $\alpha = 0.2$. Auf die Darstellung der statistischen Fehler wurde verzichtet, um die Übersicht nicht zu gefährden. Die Werte aus Abbildung 15 können als typisch gelten.

Da die Einzugsbereiche mit wachsender Neuronenzahl schärfer begrenzt werden, sollten die Simulationen in größeren Netzwerken eigentlich einen steileren Anstieg in der Konvergenzgeschwindigkeit zeigen, und dieser sollte etwa bei $m_0 \approx m_c$ einsetzen. Dieses Verhalten läßt sich tatsächlich beobachten. Während die Anzahl der Iterationen bis zum Erreichen des Fixpunktes im Inneren der Einzugsbereiche fast unabhängig von der Größe des Netzwerks (und sehr klein) ist, steigt die Konvergenzgeschwindigkeit an der Grenze der Einzugsbereiche mit wachsender Größe des Netzwerks immer stärker



Anzahl der Iterationen bis zum Erreichen eines Fixpunktes für parallele, memory2 und serielle Dynamik. ($N = 400$, $c = 0.1$). $\alpha = 0.1$: $\diamond$, $\bullet$, $*$. $\alpha = 0.4$: $\star$, $\circ$, $\oplus$.

Abbildung 18: Anzahl der Iterationen der Dynamik bis zum Erreichen eines Fixpunktes: parallele, serielle, memory2-Dynamik. $N = 400$, $c = 0.1$



Anzahl der Iterationen bis zum Erreichen eines Fixpunktes für parallele-, memory2- und serielle Dynamik ($N = 400$, $c = 0.2$). $\alpha = 0.2$: $\diamond$, $\bullet$, $*$, $\alpha = 0.4$: $\star$, $\circ$, $\oplus$.

Abbildung 19: Anzahl der Iterationen der Dynamik bis zum Erreichen eines Fixpunktes: parallele, serielle, memory2-Dynamik. $N = 400$, $c = 0.2$

an.

Wegen der großen statistischen Fehler ist es hier völlig unmöglich, ein finite-size scaling der Konvergenzgeschwindigkeiten anzugeben. Aber die Ergebnisse zeigen, daß die Testmuster im Inneren der Einzugsbereiche auch in sehr großen Netzwerken nur wenige Iterationen bis zum Erreichen des Sollmusters benötigen.

Wie zu erwarten war, beeinflußt die Wahl der Dynamik nicht nur die Größe der Einzugsbereiche, sondern auch die Konvergenzgeschwindigkeit stark. Als Beispiel dafür sind in Abbildung 17 die Werte der Konvergenzgeschwindigkeit für das vollständig verknüpfte Netzwerk mit $\alpha = 0.2$ ($N = 400$) für parallele und serielle, sowie für memory2- und memory3-Dynamik dargestellt. Die für die memory-Dynamik zusätzlich nötigen Initialisierungsschritte zur Einstellung von $h_i(t-1)$ (und $h_i(t-2)$ für memory3-Dynamik) wurden dabei mitgezählt. Auf die Darstellung der statistischen Fehler wurde wiederum verzichtet.

Es ist gut zu erkennen, daß die Anzahl der Iterationen bis zum Erreichen eines Fixpunkts für die synchronen Varianten der Dynamik (parallel, memory2, memory3) sehr ähnliche Werte aufweist.

Die asynchrone serielle Dynamik dagegen konvergiert insbesondere für kleine Werte des Anfangsüberlapps sehr viel schneller. Während also die Einzugsbereiche unter Verwendung der seriellen Dynamik gegenüber den synchronen Varianten etwas kleiner ausfallen (siehe die Abschnitte 3.4 und 3.7), werden dafür die Fixpunkte, wenigstens im Mittel, wesentlich schneller erreicht.

Auch in verdünnten Netzwerken ergeben sich keine einfacheren Resultate. Ein Beispiel für die Konvergenzgeschwindigkeit in verdünnten Netzwerken ($c = 0.1$ und $c = 0.2$) ist in Abbildung 18 und 19 dargestellt, und zwar für parallele, serielle und memory2-Dynamik bei $\alpha = 0.1$ und $\alpha = 0.4$.

Die Konvergenz ist für $\alpha = 0.1$ (untere drei Kurven in Abbildung 18) unabhängig von der Dynamik extrem schnell. Bei paralleler Dynamik genügen bis zu etwa $m_0 \geq 0.2$ zwei Iterationen, um den Zustand des Netzwerks zu stabilisieren, bei serieller Dynamik sogar nur einer. Für sehr kleine Werte von m_0 wird die Konvergenz etwas langsamer, die nicht korrekt zugeordneten Sollmuster brauchen länger, um die spurious states oder Zyklen zu erreichen. Die Dynamik mit memory-terms erreicht ihre Fixpunkte etwas langsamer nach bis zu vier Iterationen.

Erstaunlicherweise erreicht die asynchrone Dynamik auch für $\alpha = 0.4$ (obere drei Kurven) wieder am schnellsten ihre Fixpunte, während die parallele und die Dynamik mit memory-terms ein fast gleiches Verhalten zeigen. Der für die memory-Dynamik zusätzlich erforderliche Initialisierungsschritt ist in diesen Abbildungen immer als eine zusätzliche Iteration berücksichtigt worden. Er fällt aber für $\alpha = 0.4$ gegenüber dem Verhalten der parallelen Dynamik nicht mehr ins Gewicht.

Ein ähnliches Verhalten zeigen auch die Simulationen der Konvergenzgeschwindigkeit in Netzwerken mit $N = 400$ und $c = 0.2$, siehe Abbildung 19. Die unteren drei Kurven gelten für $\alpha = 0.2$ und die oberen drei für $\alpha = 0.4$. Wiederum ist die Konvergenz der seriellen Dynamik am schnellsten, während die Anzahl der von paralleler und memory2-Dynamik benötigten Iterationen fast gleich ist.

Es erscheint fast aussichtslos, zu diesen Daten ein einfaches Modell anzugeben, daß die Konvergenzgeschwindigkeit als Funktion des Anfangsüberlapps m_0 in Netzwerken mit Parametern N, c und α (sowie κ) auch nur einigermaßen präzise beschreibt. Innerhalb der Einzugsbereiche erreichen die Testmuster das korrekte Sollmuster meistens schon nach sehr wenigen, ein bis vielleicht vier Schritten der Dynamik. Es wäre daher auch im Hinblick auf Anwendungen interessant, die Dynamik willkürlich nach einer bestimmten Anzahl von Iterationen abzubrechen und das Verhalten der resultierenden Modelle zu untersuchen.

4 Fehlertoleranz in neuronalen Netzwerken

Ein besonders faszinierender Punkt bei der Untersuchung der Eigenschaften neuronaler Netzwerke betrifft die Fehlertoleranz.

Biologische neuronale Netzwerke erweisen sich als extrem robust gegenüber Störungen durch Rauschen und Zerstörung, sowohl von Synapsen als auch ganzen Neuronen. Obwohl in jedem Gehirn ständig Neuronen absterben, leidet doch seine Funktion über sehr lange Zeiträume kaum. Vor allem die Großhirnrinde (der *Cortex*) erweist sich als sehr unempfindlich auch gegen beträchtliche Zerstörungen in der Folge von Verletzungen. Für einige Areale ist es bis heute nicht gelungen, aus Verletzungen auf die Funktion zu schließen.

Außerdem ist Information in neuronalen Netzwerken des Spinglas-Typs nicht lokal, sondern verteilt über alle Synapsen des Netzes gespeichert (*„distributed representation"*), und dies führt zu interessanten Problemen im Umfeld der Kodierungstheorie. Während bei der üblichen digitalen lokalen Speicherung die Auswirkung von Fehlern in Speicherzellen sofort quantifiziert werden kann, ist die Beschreibung von neuronalen Netzwerken unter Fehlern deutlich komplizierter.

Dabei stellt sich die Frage, ob neuronale Netzwerke im Hinblick auf ihre Fehlertoleranz mit den klassischen fehlerkorrigierenden Codes der Kodierungstheorie konkurrieren können.

- Dazu werden in Abschnitt 4.1 zunächst verschiedene Modelle für Fehler in neuronalen Modellen vorgestellt. Schon eine sehr grobe Analyse zeigt, daß die Auswirkungen ausgefallener Neuronen auf die übrigen Neuronen eher uninteressant sind. Der Einsatz einer iterativen Lernregel führt sogar dazu, daß sich die übrigen Neuronen vollständig vom defekten Neuron entkoppeln. Dagegen bewirkt die zufällige Zerstörung von Synapsen interessante Effekte und wird im Rest dieses Kapitels als Fehlermodell verwendet.

- Die Untersuchung der Speicherkapazität im Gardner-Modell zeigt die wichtige Rolle der Stabilitäten der Muster zur Charakterisierung der neuronalen Modelle. In Abschnitt 4.2 wird daher ein Modell entwickelt, um die Verteilung der Stabilitäten in einem Netzwerk unter zufälliger Zerstörung von Synapsen zu berechnen. Aus der Verteilung der Stabilitäten kann dann sehr einfach die Wahrscheinlichkeit berechnet werden, mit dem die Muster nach der Zerstörung gespeichert sind. Simulationen an Netzwerken mit zerstörten Synapsen stimmen sehr gut mit dem Modell überein.

- Für die Abschätzung der Einzugsbereiche können dann die in Kapitel 3 vorgestellten Modelle verwendet werden, die ja ebenfalls auf der Verteilung der Stabilitäten beruhen. Dem Vergleich der aus den Simulationen an teilzerstörten Netzwerken ermittelten Einzugsbereichen mit den Vorhersagen der Modelle m_F (31) und m_S (30) dient der Abschnitt 4.4.

- Die Ergebnisse von Abschnitt 4.5 zeigen, daß das hier entwickelte Modell auch im Netzwerk mit binären Kopplungen verwendet werden kann. Das binäre Netzwerk ist nicht nur im Hinblick auf digitale Implementationen interessant, sondern erlaubt auch Vergleiche mit den klassischen Methoden der Kodierungstheorie: Der Informationsgehalt einer Kopplungsmatrix J mit $J_{ij} \in \{-1, +1\}$ läßt sich leicht angeben. Außerdem erreicht das binäre Netzwerk eine relative Speicherkapazität von $O(1)$, während diese in Netzwerken mit reellwertigen Kopplungen nur $O(1/\ln N)$ beträgt.

- Natürlich liegt eine Klassifikation der neuronalen Modelle in Bezug auf die Kodierungstheorie außerhalb der Möglichkeiten dieser Arbeit; in Abschnitt 4.6 wird dennoch versucht, wenigstens eine grobe Orientierung zu liefern. Dazu werden die Leistungen der neuronalen Netzwerke mit den besonders einfach zu behandelnden Reed-Muller Codes verglichen.

- Eine kurze Zusammenfassung vergleicht die in neuronalen Netzwerken tolerierbaren Konzentrationen zerstörter Synapsen mit den typischen Konzentrationen von Fehlern bei der Produktion elektronischer Schaltkreise.

4.1 Modelle für Zerstörung in Netzwerken

Natürlich lassen sich Modelle von Zerstörungen eines neuronalen Netzwerks auf jeder beliebigen Komplexitätsstufe konstruieren. So kann der Ausfall von einzelnen Synapsen oder Neuronen genauso untersucht werden, wie kompliziertere Fehler, etwa subtile Veränderungen in der Dynamik durch Verzögerungen. Zur realistischen Beschreibung von biologischen neuronalen Netzwerken wäre auch das stochastische Auftreten aller dieser Fehler zu berücksichtigen.

Im Hinblick auf die einfache Struktur der Spinglas-Modelle ist es allerdings sinnvoll, sich zunächst auf die einfachsten möglichen Fehlermodelle zu beschränken.

4.1.1 Stuck-at-1 Neuronen

Dies ist zum einen der Ausfall ganzer Neuronen. Da das Neuron nur den Wert seiner Ausgangsfunktion ($S_i = f_i(t) = \pm 1$ für Spinglas-Netzwerke) an die übrigen Neuronen weitergibt, läßt sich ein ausgefallenes Neuron durch eine Ausgangsfunktion, die permanent im aktiven oder inaktiven Zustand verbleibt, beschreiben. In der Sprache der technischen Informatik könnte man derartige Neuronen als 'stuck-at-1' Neuronen bezeichnen.

Der Einfluß eines permanent aktiven Neurons S_k auf ein neuronales Netz mit iterativer Lernregel ist aber trivial: Während des Lernens werden die Synapsen J_{ik} der übrigen Neuronen gemäß $\Delta J_{ik} = \xi_i^\mu \xi_k^\mu$ verändert. Wenn wegen des Ausfalls von S_k ständig $\xi_k^\mu = 1$ gilt, konvergiert der Wert von J_{ik} offenbar gegen den Mittelwert $\langle \xi_i^\mu \rangle_\mu$,

d. h. die übrigen Neuronen entkoppeln sich vom zerstörten Neuron. Der Effekt eines stuck-at-1 Neurons ist also lediglich ein falsches Ausgangssignal, während sich die übrigen Neuronen zu einem Netzwerk aus $(N-1)$ Neuronen organisieren.

Andere Lernregeln (etwa die Hebb-Lernregel im Hopfield-Modell) führen nicht unbedingt zu dieser Entkopplung. Die Untersuchung von stuck-at-1 Neuronen könnte daher in diesen Netzwerken durchaus interessant sein.

4.1.2 Zerstörte Synapsen

Interessante Konsequenzen hat dagegen die Zerstörung von Synapsen. Das einfachste Modell für den Ausfall von Synapsen ist offenbar, die betroffenen Synapsen vom Wert J_{ij} auf den Wert 0 zu zwingen. Dieses Modell entspricht einer Verdünnung des Netzwerks und läßt sich auch biologisch motivieren. Für digitale neuronale Netzwerke kann es dagegen sinnvoll sein, die Zerstörung der Synapse durch Vorzeichenwechsel $J_{ij} \to -J_{ij}$ zu modellieren, insbesondere für das binäre Netzwerk mit Kopplungen $J_{ij} = \pm 1$.

Besonders wichtig ist dabei die zufällige Zerstörung von Synapsen im vorher vollständig verknüpften Netzwerk, das heißt jede Synapse erhält den Wert

$$J'_{ij} = C_{ij} J_{ij}, \qquad C_{ij} = \begin{cases} 0 & \text{mit Wahrscheinlichkeit } \lambda, \\ 1 & \text{mit Wahrscheinlichkeit } 1-\lambda. \end{cases} \tag{35}$$

Die Matrix C_{ij} (*connecitivity matrix*) beschreibt dabei die Architektur des Netzwerks, indem sie angibt, welche Neuronen miteinander verknüpft sind. Im Mittel ist jedes Neuron nach der Zerstörung mit $C := \langle \sum_j C_{ij} \rangle = (1-\lambda)N$ anderen Neuronen verknüpft.

Die Auswirkungen zufälliger Zerstörung von Synapsen auf die Speicherung und das Wiedererkennen der Sollmuster wurden für das Hopfield-Modell von [Koscielny-Bunde 90] mit Simulationen untersucht. Im Hopfield-Modell ist für bestimmte Einstellungen der Verknüpfungsmatrix C im Rahmen der mean-field Theorie zusätzlich auch die analytische Berechnung der Speicherkapazität möglich. Da die Werte der Synapsen nach der Hebb-Regel $J_{ij} = \sum_{\mu=1}^{\alpha \cdot N} \xi_i^\mu \xi_j^\mu$ berechnet werden, ist die verdünnte Version des Hopfield-Modells völlig äquivalent zum Hopfield-Modell unter zufälliger Zerstörung.

In der Arbeit von [Canning & Gardner 88] wird gezeigt, wie die Speicherkapazität im Rahmen der mean-field Theorie für mehrere symmetrische Hypercube-Architekturen und auch für das zufällig vernetze Modell abgeschätzt werden kann. Die dort gewonnenen Ergebnisse sind daher auch für das Hopfield-Modell unter zufälliger Zerstörung von Synapsen anwendbar.

Das extrem verdünnte Hopfield-Modell (mit $O(\ln N)$ Synapsen pro Neuron) wird in [Derrida *et. al.* 87] behandelt.

4.1.3 Relearning

In Netzwerken mit iterativer Lernregel ist zu beachten, daß die Zerstörung von Synapsen unterschiedliche Konsequenzen haben kann, wenn sie vor oder nach der Lernphase eingeführt wird. Nach der Zerstörung (oder im verdünnten Netz) muß die Lernregel natürlich gemäß $\Delta J_{ij} = N^{-1}\, C_{ij}\epsilon_{i\mu}\xi_i^\mu \xi_j^\mu$ modifiziert werden.

Die Werte der Synapsen in Netzwerken mit iterativer Lernregel hängen über die Fehlermaske $\epsilon_{i\mu}$ nicht nur von den Sollmustern, sondern auch von den lokalen Feldern der Neuronen ab. Da diese sich durch die Zerstörung ändern, kann die Wiederholung der Lernphase nach der teilweisen Zerstörung des Netzes die verbliebenen Kopplungen neu einstellen, bis das Netz die Sollmuster wieder perfekt speichert. Natürlich wird der erreichbare Wert der Stabilität der Muster durch die Zerstörung verringert.

Die Konvergenz der Wiederholung der iterativen Lernregel läßt sich unter der Voraussetzung beweisen, daß das verdünnte Netz die Muster überhaupt speichern kann. Der Beweis erfordert keine wesentliche Änderung des entsprechenden Theorems für das vollständig verknüpfte Netz und wird im Anhang B gegeben.

4.2 Speicherkapazität im zerstörten Netzwerk

Um die Eigenschaften von teilzerstörten neuronalen Netzwerken mit iterativer Lernregel zu untersuchen, müssen letztlich numerische Verfahren verwendet werden, da die genauen Werte der Synapsen nicht analytisch bekannt sind. Nach dem Lernen kann Zerstörung/Verdünnung beliebig in das Netz eingebracht werden, und das Verhalten des resultierenden Netzwerks kann gründlich untersucht werden. Um die derart erhaltenen numerischen Daten bewerten zu können, scheint es notwendig zu sein, wenigstens ein einfaches theoretisches Modell zu konstruieren.

Für die Beschreibung der Stabilität der Sollmuster, aber auch als Ausgangspunkt für eine Untersuchung der Einzugsbereiche, muß dazu zunächst die Verteilung der Stabilitäten der Sollmuster nach der Zerstörung berechnet werden.

4.2.1 Verteilung der Stabilitäten im verdünnten Hopfield-Modell

Besonders leicht läßt sich die Verteilung der Stabilitäten im Hopfield-Modell berechnen, weil die Werte der J_{ij} explizit bekannt sind. Für die folgenden Rechnungen werden unkorrelierte Sollmuster vorausgesetzt.

Sei λ die Wahrscheinlichkeit für $C_{ij} = 0$. Dann wird die Hopfield-Kopplungsmatrix gegeben durch

$$J_{ij} = C_{ij}\frac{1}{\sqrt{\alpha N}}\sum_{\mu=1}^{\alpha N}\xi_i^\mu \xi_j^\mu, \tag{36}$$

wobei der Normierungsfaktor zur Einhaltung der sphärischen Normierung $\sum_{j\neq i} J_{ij}^2 = (1-\lambda)N$ gewählt ist. Im vollständig vernetzten Modell ($\lambda = 0$) erhält man das bekannte

Resultat [Kepler & Abbott 88]

$$\rho_H^{(0)}(\kappa) = \frac{1}{\sqrt{2\pi}} \exp\Big[-\frac{1}{2}\Big(\kappa - \frac{1}{\sqrt{\alpha}}\Big)^2\Big], \tag{37}$$

das ist eine Gauß-Verteilung mit Mittelwert $\kappa_0 = 1/\sqrt{\alpha}$ und einer Varianz $\sigma_0 = \langle\kappa^2\rangle - \langle\kappa\rangle^2 = 1$. Im verdünnten Modell gilt

$$\begin{aligned} \kappa_{i\mu} &= \frac{1}{||J_i||}\Big(\sum_{j\neq i} J_{ij}\xi_j^\mu \xi_i^\mu\Big) \\ &= \frac{1}{\sqrt{(1-\lambda)\alpha N}}\Big(\sum_{j\neq i} C_{ij} + \Delta_i\Big) \\ &= \frac{\sqrt{1-\lambda}}{\sqrt{\alpha}} + \frac{\Delta_i}{\sqrt{(1-\lambda)\alpha N}}, \end{aligned} \tag{38}$$

mit einer Summe

$$\Delta_i = \sum_{j\neq i}\sum_{\nu\neq\mu} C_{ij}\xi_i^\nu \xi_j^\nu \xi_i^\mu \xi_j^\mu, \tag{39}$$

die aus $(1-\lambda)\alpha N^2$ unkorrelierten Termen ± 1 besteht, und deshalb eine Varianz $\langle\delta_i^2\rangle \simeq (1-\lambda)\alpha N^2$ besitzt. Die Verteilung der Stabilitäten im verdünnten Hopfield-Modell ist daher im Limes $N \to \infty$ wieder eine Gauß-Verteilung

$$\rho_H^{(\lambda)}(\kappa) = \frac{1}{\sqrt{2\pi}} \exp\Big[-\frac{1}{2}\Big(\kappa - \frac{\sqrt{1-\lambda}}{\sqrt{\alpha}}\Big)^2\Big] \tag{40}$$

mit Varianz 1, deren Mittelwert aber auf $\kappa_\lambda = \sqrt{(1-\lambda)/\alpha}$ reduziert ist.

4.2.2 Das Netzwerk mit konstanter Stabilität

In Netzwerken mit iterativen Lernregeln sind die Werte der Synapsen nicht analytisch bekannt. Die Verteilung der Stabilitäten nach dem Lernen läßt sich jedoch berechnen; für das Gardner-Modell erhält man die Verteilung $\rho(\kappa)$ (19).

Als natürlichen Weg, um auch komplizierte Verteilungen beschreiben zu können, schlage ich die Untersuchung eines Netzwerks mit konstanter Stabilität vor („constant-stability-model“). Die Verteilung der Stabilitäten nach dem Lernen sei gegeben durch

$$\rho^{(0)}(\kappa_{i\mu}) = \delta(\kappa_{i\mu} - \kappa_0), \tag{41}$$

das heißt für alle Muster am Neuron i gelte

$$\kappa_{i\mu} = \frac{1}{||J_i||}\Big(\sum_{j\neq i}^{N} J_{ij}\xi_j^\mu \xi_i^\mu\Big) = \kappa_0, \tag{42}$$

wobei $||J_i|| = \left(\sum_{j\neq i} J_{ij}^2\right)^{1/2} = N^{1/2}$. Nach einer zufälligen Zerstörung von Synapsen mit Wahrscheinlichkeit λ gilt offenbar

$$\begin{aligned} \kappa_{i\mu}^{(\lambda)\prime} &= \frac{1}{||J_i||'}\Big(\sum_{j\neq i}^{N} C_{ij}J_{ij}\xi_j^\mu\xi_i^\mu\Big) \\ &= \frac{1}{||J_i||'}\Big(\sum_{j\neq i}^{N} J_{ij}\xi_j^\mu\xi_i^\mu - \sum_{o=1}^{\lambda N} J_{il_o}\xi_{l_o}^\mu\xi_i^\mu\Big). \end{aligned} \tag{43}$$

(Dabei versteht sich die Summe über o über die Indices mit $C_{il_o} = 0$). Zur Berechnung der lokalen Felder nach der Zerstörung muß also die Norm $||J_i||'$ berechnet und die Summe über die $C_{ij}J_{ij}\xi_j^\mu\xi_i^\mu$ abgeschätzt werden. Der Wert der Norm nach der Zerstörung ist durch $||J_i||' \simeq \sqrt{1-\lambda}\;||J_i||$ gegeben.

Die Berechnung der Summe in Gleichung (43) gestaltet sich etwas schwieriger. Wegen der Bedingung (42) sind die einzelnen Summanden $J_{ik}\xi_k^\mu\xi_i^\mu$ korreliert, es gilt $\langle J_{ik}\xi_k^\mu\xi_i^\mu\rangle = \kappa_0/\sqrt{N}$. Die zweite Summe hat also den Mittelwert $\kappa_0\lambda\sqrt{N}$ und die mittlere Stabilität im Netzwerk mit Parametern (κ_0, λ) sollte $\kappa_\lambda = \sqrt{1-\lambda}\,\kappa_0$ betragen. In Abbildung 20 wird diese Abhängigkeit mit den Simulationen verglichen. Die Übereinstimmung mit den numerischen Daten ist hervorragend.

Ohne die Korrelation durch die Bedingung (42) wären die einzelnen Summanden unabhängig voneinander und die Summe hätte eine Varianz $\sigma_\lambda^2 \simeq (\lambda/(1-\lambda))$, da die einzelnen Summanden (sowohl im binären wie auch im sphärischen Modell) die Varianz $\langle J_{ik}^2\rangle = 1$ besitzen.

Für kleine Konzentrationen $\lambda \ll 1$ der zerstörten Synapsen sind die Summanden $J_{il_o}\xi_{l_o}^\mu\xi_i^\mu$ fast unkorreliert, und diese Näherung sollte gut erfüllt sein. Für große Konzentrationen von $\lambda \approx 1$ dagegen bleiben nur wenige Summanden in der ersten Summe in (43) übrig, während die zweite Summe fast die Bedingung (42) erfüllt. In diesem Fall läßt sich die Varianz also zu $\sigma_\lambda^2 \simeq 1$ abschätzen. Der exakte Verlauf der Varianz als Funktion von λ muß zwischen diesen Werten interpolieren.

Die numerischen Ergebnisse zeigen, daß die Varianz der Summe sehr gut durch $\sigma_\lambda = \sqrt{\lambda}$ beschrieben wird, siehe Abbildung 21. (Die numerisch ermittelten Werte für σ_λ weichen nach oben bis zu 10% von obigem Modell ab. Dies ist auch zu erwarten, denn die Simulationen gehen nicht von der „constant-stability“ Verteilung $\delta(\kappa_{i\mu}-\kappa_0)$ aus. Die Verteilung der Stabilitäten nach dem Lernen approximiert ja vielmehr die Verteilung (19) und aufgrund von finite-size Effekten treten auch Stabilitäten $\kappa_{i\mu} < \kappa_0$ auf. Eigentlich müßte über die Anfangsverteilung der Stabilitäten integriert werden, und dies ist in der Tat möglich, siehe unten. Die relativ geringen Abweichungen der Werte von σ_λ vom Modell σ_{th} zeigen aber, daß die Beschreibung durch die Verteilung $\delta(\kappa_{i\mu}-\kappa_0)$ jedenfalls für kleine Werte von α oder große λ gut gerechtfertigt ist.)

Im Limes $N \to \infty$ wird die Verteilung der lokalen Felder nach Zerstörung der Synapsen mit Wahrscheinlichkeit λ also durch eine Gauß-Verteilung beschrieben, gemäß

$$\rho^{(\lambda)}(\kappa) = \frac{1}{\sqrt{2\pi}\sigma} \exp\Big[-\frac{1}{2}\frac{(\kappa-\kappa_\lambda)^2}{\sigma_\lambda^2}\Big] \tag{44}$$

mit Parametern $\kappa_\lambda = \sqrt{1-\lambda}\,\kappa_0$ und $\sigma_\lambda = \sqrt{\lambda}$.

Diese Abschätzung (mit $\sigma_\lambda = \sqrt{\lambda}$) für die Verteilung der Stabilitäten nach Zerstörung ist mit dem Hopfield-Modell konsistent: Die Verteilung $\rho^{(0)}(\kappa)$ (37) kann geschrieben werden als Überlagerung von δ-Funktionen,

$$\rho_H^{(0)}(\kappa) = \int \mathrm{d}\Lambda \; \rho_H^{(0)}(\Lambda)\; \delta(\kappa - \Lambda). \tag{45}$$

Nach Zerstörung wird aus jedem δ-Term eine Gauß-Verteilung und es gilt tatsächlich

$$\rho_H^{(\lambda)}(\kappa) = \int \mathrm{d}\Lambda \; \rho_H^{(0)}(\Lambda)\; \rho^{(\lambda)}(\kappa). \tag{46}$$

Das Diagramm 22 zeigt ein Beispiel für die Verteilung der lokalen Felder in einem fast gesättigten Netzwerk ($\alpha = 0.4$) und nach Zerstörung mit $\lambda = 0.05$, 0.1 und 0.15. Der Übergang von der Verteilung gemäß (19) zur Gauß-Verteilung (44) ist deutlich zu erkennen.

4.3 Anteil der vom zerstörten Netz gespeicherten Muster

Da die Sollmuster vor der Zerstörung perfekt (mit Stabilität κ_0) gespeichert sind und die Verteilung der Stabilitäten als Funktion von λ und κ_0 im vorliegenden Modell analytisch bekannt ist, kann der Anteil der nach der Zerstörung noch korrekt gespeicherten Muster berechnet werden.

Der Anteil Γ der lokalen Felder $\kappa_{i\mu} < \kappa_{\min}$ an einem Neuron ist durch das Integral

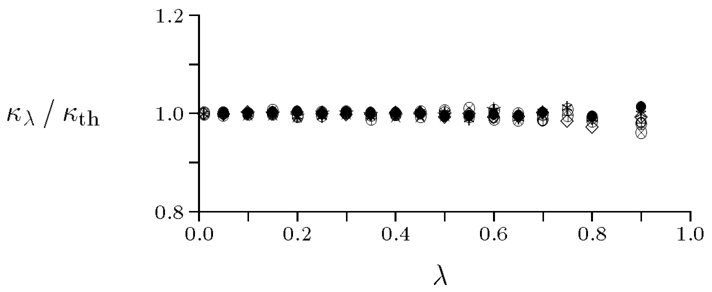
$$\Gamma(\kappa < \kappa_{\min}) = \int_{-\infty}^{\kappa_{\min}} \mathrm{d}\kappa \; \rho^{(\lambda)}(\kappa) \tag{47}$$

gegeben, bzw.

$$\Gamma(\kappa < \kappa_{\min}) = \frac{1}{2}\Big(1 + \mathrm{erf}\Big(\frac{\kappa_{\min} - \kappa_\lambda}{\sigma_\lambda\sqrt{2}}\Big)\Big). \tag{48}$$

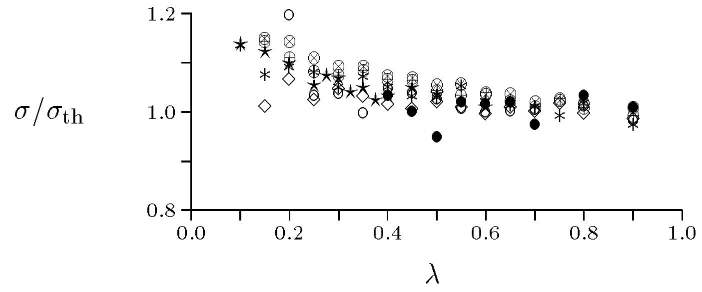
Insbesondere gibt $\Gamma(0)$ den Anteil der vom Neuron nicht mehr stabil gespeicherten Muster an. Zum Beispiel sind für $\alpha = 0.1$ (das entspricht etwa $\kappa_0(\alpha) \approx 3.0$) die Sollmuster mit weniger als 1% Fehlern bis zu einer Konzentration zerstörter Synapsen von $\lambda_c \leq 0.625$ gespeichert.

Die Vorhersagen des einfachen constant-stability Modells stimmen für alle untersuchten Netzwerke mit Speicherkapazitäten $\alpha = 0.1$ bis zu $\alpha = 0.4$ sehr gut mit den Simulationen an den fast gesättigten Gardner-Netzwerken überein. Für $\alpha = 0.5$ bis 0.7 stimmt das constant-stability Modell immer noch qualitativ mit den Simulationen



Mittelwert κ_λ der Verteilung der Stabilitäten $\rho(\kappa)$ als Funktion der Zerstörung λ, normiert auf das Modell $\kappa_{\rm th} = \sqrt{1-\lambda}\,\kappa_0$. ($\alpha = 0.1$ •, 0.2 ○, ..., 0.7 ⊗).

Abbildung 20: Mittelwert κ_λ der Verteilung der Stabilitäten nach Zerstörung.



Breite σ der Verteilung der Stabilitäten $\rho(\kappa)$ als Funktion der Zerstörung, normiert auf das Modell $\sigma_{\rm th} = \sqrt{\lambda}$.

Abbildung 21: Varianz σ der Verteilung der Stabilitäten nach Zerstörung

überein, die Diskrepanzen erreichen etwa 3%, also etwas außerhalb der statistischen Fehler der Simulationen.

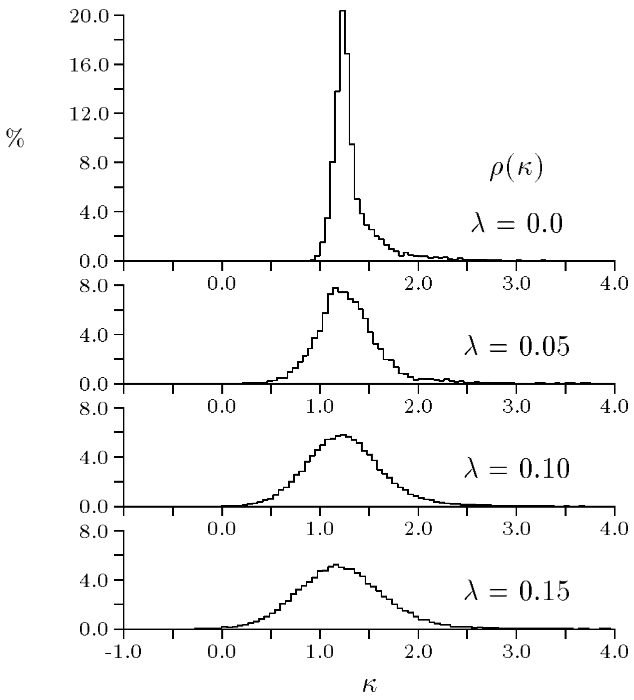
Natürlich kann die Übereinstimmung des Modells mit den Simulationen stark verbessert werden, wenn über die Anfangsverteilung $\rho^{(0)}(\Lambda)$ der Stabilitäten integriert wird, wie oben für das Hopfield-Modell demonstriert,

$$\Gamma(\kappa < \kappa_{\rm min}) = \frac{1}{2}\left(1 + \int {\rm d}\Lambda \;\; \rho^{(0)}(\Lambda) \;\; {\rm erf}\Big(\frac{\kappa_{\rm min} - \kappa_\lambda(\Lambda)}{\sigma_\lambda\,\sqrt{2}}\Big)\right). \tag{49}$$

Die Ergebnisse der Simulationen (an fast gesättigten Netzwerken) und die entsprechenden Vorhersagen (48) sind in Abbildung 23 dargestellt. (Die nicht dargestellten Fehlerbalken der statistischen Fehler erreichen etwa zweimal die Größe der Symbole).

Eine andere Art der Darstellung ergibt sich, wenn der Anteil der im Netz perfekt gespeicherten Muster gegen die Zerstörung λ aufgetragen wird. Wenn ein Muster ξ^μ am Neuron S_i mit Wahrscheinlichkeit $w_1 = 1 - \Gamma(\kappa_{\rm min})$ korrekt gespeichert ist, beträgt die Wahrscheinlichkeit dafür, daß das Muster im ganzen Netz noch korrekt gespeichert ist $w_N = (1 - \Gamma(\kappa_{\rm min}))^N$.

Abbildung 24 zeigt den Anteil der Sollmuster, die an mehr als 99% der Neuronen korrekt gespeichert sind. Die Kurven sind Fits der Form $f_s = 1/2 + \tanh(a \cdot \lambda + b)$.



Beispiel für die Verteilung der Stabilitäten $\rho(\kappa_{i\mu})$ im Gardner-Modell nach dem Lernen ($\alpha = 0.4$, $N = 256$) und nach Zerstörung mit $\lambda = 0.05$, 0.10 und 0.15.

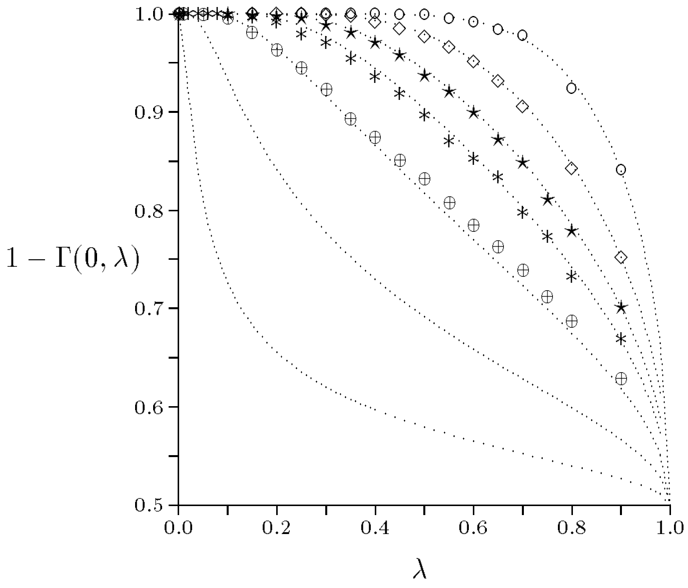
Abbildung 22: Beispiel für die Verteilung der Stabilitäten $\rho(\kappa_{i\mu})$ nach Zerstörung: $\alpha = 0.4$, $\kappa_0 \approx 1.2$, $\lambda = 0$, 0.05, 0.1 und 0.15.

Tatsächlich wächst die Steigung a mit der Größe des Netzwerks an. Im Limes $N \to \infty$ ergibt sich wegen des Exponenten N für jedes $\kappa_{\text{min}} < 0$ (also unter Zulassen eines kleinen Anteils von Fehlern) ein „Phasenübergang“ von Speicherung zum Verlust der Muster.

4.4 Einzugsbereiche im zerstörten Netzwerk

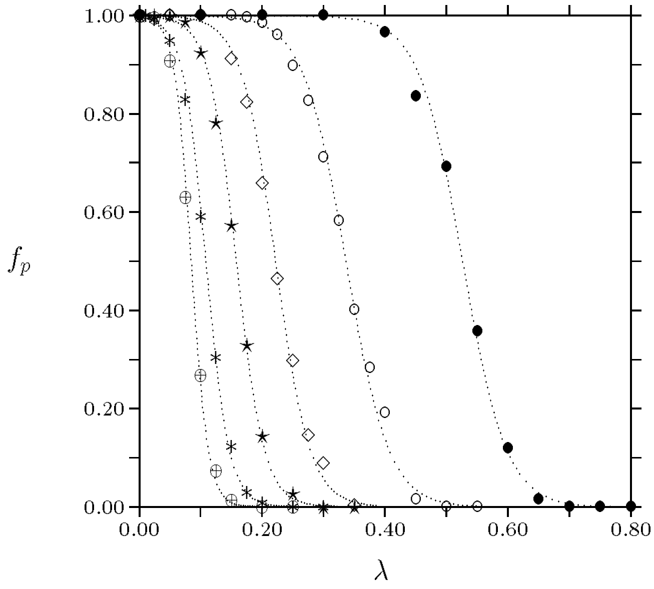
Die Ermittlung der Einzugsbereiche der zerstörten Netzwerke erfolgt nach dem selben Algorithmus wie für verdünnte Netzwerke: Nach dem Lernen und der anschließenden Zerstörung des Netzes werden Testmuster mit definierten Werten des Anfangsüberlapps erzeugt und bis zur Stabilität unter der Dynamik (4) iteriert. Der Endüberlapp der Muster mit den entsprechenden Sollmustern, sowie der Anteil perfekt erkannter Muster wird protokolliert. Fits durch die Kurven $m_f(m_i)$ ergeben dann den Wert des kritischen Anfangsüberlapps m_c und damit die Größe der Einzugsbereiche der Sollmuster.

Da die Muster nach der Zerstörung im allgemeinen nicht mehr perfekt gespeichert sind (s. o.), wurden alle Muster mit einer Abweichung von nicht mehr als 1% vom entsprechenden Sollmuster als perfekt wiedererkannt gewertet.



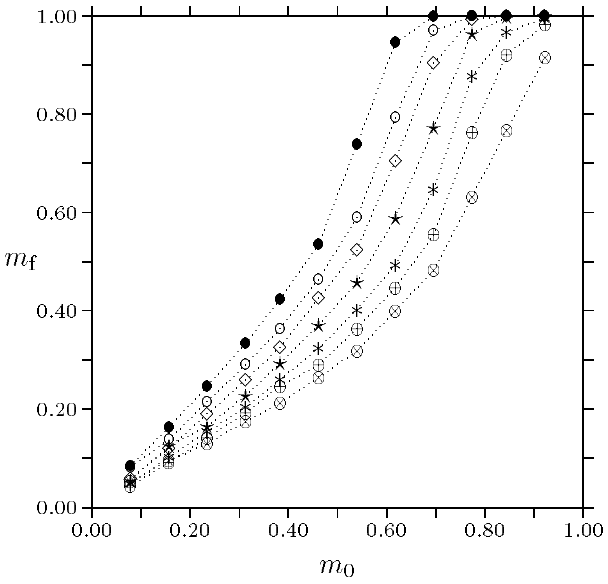
Anteil der pro Neuron stabilisierten Muster, $1-\Gamma(0)$ als Funktion der Zerstörung λ in fast gesättigten Netzwerken. ($N = 256$, $\alpha = 0.1$ $\circ$, 0.2 $\diamond$, 0.3 $\star$, 0.4 $*$, 0.6 $\oplus$). Die Kurven zeigen das "constant-stability" Modell mit (von oben) $\kappa_0 = 2.99, 2.04, 1.58, 1.30, 0.91, 0.5$ und 0.2.

Abbildung 23: Anteil stabiler Muster pro Neuron als Funktion der Zerstörung λ. Simulationen in fast gesättigten Netzwerken: $\alpha = 0.1$ $\circ$, 0.2 $\diamond$, 0.3 $\star$, 0.4 $*$, 0.6 $\oplus$. Die Kurven zeigen das constant- stability Modell.



Anteil f_p der im Netzwerk perfekt gespeicherten Muster als Funktion der Zerstörung λ. $\alpha = 0.1$ $\bullet$, 0.2 $\circ$, 0.3 $\diamond$, 0.4 $\star$, 0.5 $*$, 0.6 $\oplus$. ($N = 256$).

Abbildung 24: Anteil der perfekt gespeicherten Sollmuster ($N = 256$). Siehe Text.



Endüberlapp m_f als Funktion von m_0 in zerstörten Netzwerken. $\alpha = 0.4$, $\kappa \approx 1.1$. $\lambda = 0.0$ $\bullet$, 0.05 $\circ$, 0.075 $\diamond$, 0.10 $\star$, 0.125 $*$, 0.15 $\oplus$, 0.175 $\otimes$.

Abbildung 25: Endüberlapp $m_f(m_0, \lambda)$ und Anteil perfekt erkannter Muster $f_p(m_0, \lambda)$ im Netzwerk mit Zerstörung $\lambda = 0.1, 0.2, 0.3, 0.4$ als Funktion von m_0. ($N = 256$, $\alpha = 0.4$).

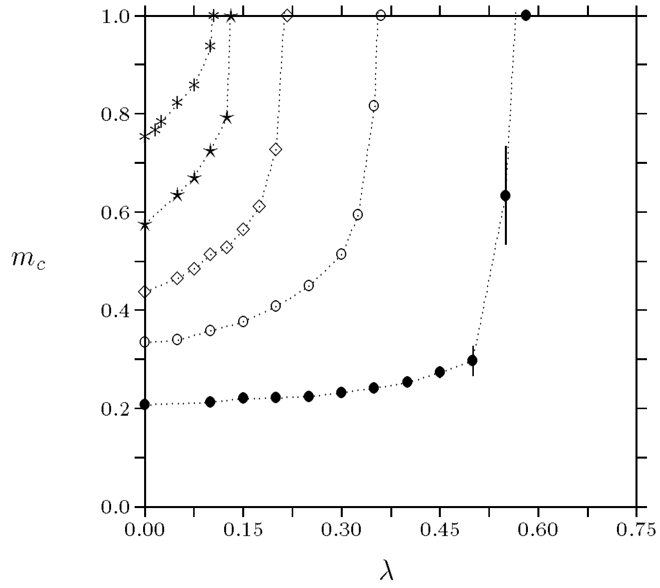
Zusätzlich wurde für jede Simulation die Verteilung der Stabilitäten protokolliert, um anhand der Gleichung (31) die Vorhersagen m_F und m_S mit den numerisch ermittelten Werten vergleichen zu können.

Ein Beispiel für die Gestalt der Funktionen $m_f(m_0, \lambda)$ und $f_p(m_0, \lambda)$ ist in Abbildung 25 für $\alpha = 0.4$ und $\lambda = 0.1, \ldots, 0.5$ dargestellt.

4.4.1 Ergebnisse der Simulationen

Die Ergebnisse der Simulationen an fast gesättigten Netzwerken sind in Diagramm 26 als Funktion von λ dargestellt. Sie zeigen, daß die Funktion des Netzes für kleine Konzentrationen der Zerstörung tatsächlich nicht beeinträchtigt wird: Solange die Verteilung der Stabilitäten ρ eng um κ_0 zentriert ist und kaum Stabilitäten $\kappa_{i\mu} < 0$ auftreten, verkleinern sich die Einzugsbereiche etwas (m_c wird etwas größer), aber nur langsam.

Bei größeren Konzentrationen der Zerstörung, wenn viele lokale Felder $\kappa_{i\mu} < 0$ auftreten, verschlechtern sich die Einzugsbereiche rapide, weil die entsprechenden Sollmuster nicht mehr perfekt gespeichert sind. In diesem Bereich der Konzentration zerstörter Synapsen sind die statistischen Fluktuationen der Werte von m_c extrem groß. Daher kann der Wert von λ_c, der im Limes $N \to \infty$ den Übergang von assoziativer Speicherung zum Verlust aller Muster angibt, nicht genau bestimmt werden. Für die im Diagramm



Numerisch ermittelte Einzugsbereiche m_c als Funktion der Zerstörung λ. $N = 256$, $\alpha = 0.1$ $\bullet$, 0.2 $\circ$, 0.3 $\diamond$, 0.4 $\star$, 0.5 $*$.

Abbildung 26: Numerisch ermittelte Einzugsbereiche m_c nach Zerstörung als Funktion von λ. $\alpha = 0.1$ $\bullet$bis 0.5 $*$. ($N = 256$).

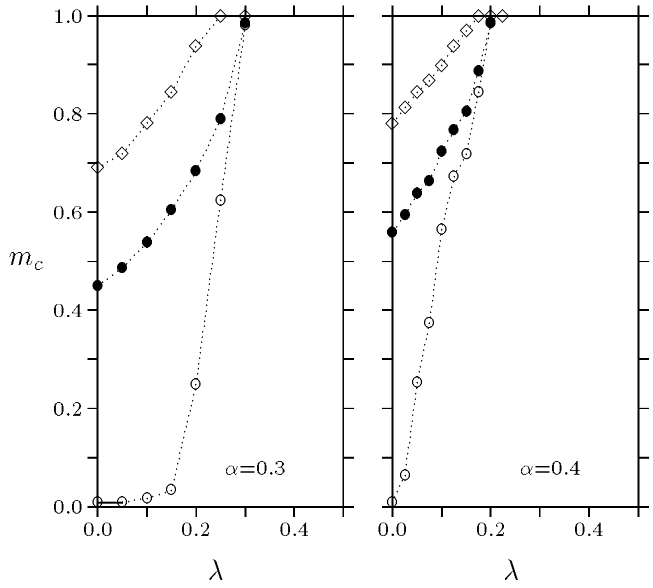
mit $m_c = 1$ gekennzeichneten Werte von λ konnte keine assoziative Speicherung mehr beobachtet werden.

4.4.2 Vergleich mit den theoretischen Modellen

Es ist interessant, die nach dem oben angegebenen Algorithmus ermittelten Einzugsbereiche der Sollmuster mit den theoretischen Modellen für vollständig verknüpfte und extrem verdünnte Netzwerke zu vergleichen. Da die Verteilung $\rho^{(\lambda)}(\kappa)$ der Stabilitäten aus den Simulationen bekannt ist, können die entsprechenden Fixpunkte der Gleichungen (30) und (31) berechnet und mit dem numerisch ermittelten Wert m_c verglichen werden.

Wie schon im Abschnitt über die Einzugsbereiche erläutert, hängt der Wert von m_F aber empfindlich von der Wahl der in die Iteration einzusetzenden Konstante c ab. Die hier numerisch ermittelten Werte von m_c waren immer (für alle Werte von α) größer als die Vorhersage m_F aus der Fixpunktgleichung 31 mit Parameter $c = 1/2$. Da die Netzwerke mit steigender Konzentration zerstörter Synapsen immer weiter von der Sättigung abweichen, sollte die Übereinstimmung mit dem Modell m_F mit steigendem λ immer schlechter werden.

Trotzdem geben beide Modelle ziemlich präzise die Lage der „kritischen Konzentration“ von Zerstörung λ_c wieder, oberhalb der keine assoziative Speicherung mehr



Numerisch ermittelte Einzugsbereiche (critical initial overlap) m_c (•) als Funktion der Zerstörung λ, verglichen mit den Modellen m_S (∘) und m_F (⋄).

Abbildung 27: Vergleich der numerisch ermittelten Einzugsbereiche m_c • mit den Modellen m_F ⋄ und m_S ∘ als Funktion von λ. $\alpha = 0.3, 0.4$. ($N = 128$).

erfolgt. Als Beispiel ist in Abbildung 27 der Verlauf von m_c sowie der Modelle m_F und m_S für Speicherkapazitäten $\alpha = 0.3$ und $\alpha = 0.4$ als Funktion von λ dargestellt.

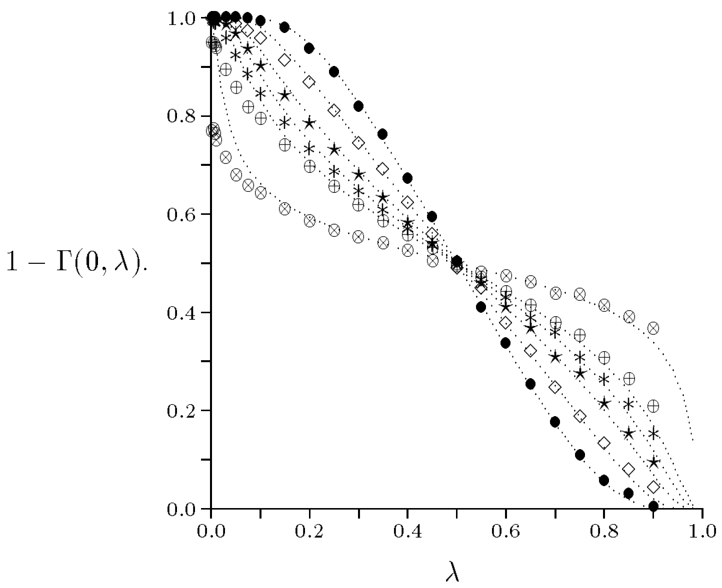
4.5 Zerstörung im binären Netzwerk

Die Untersuchung der Auswirkungen von Fehlern ist gerade auch im Netzwerk mit binären Kopplungen interessant. Dabei tritt neben der Verdünnung ($J_{ij} \to 0$ mit Wahrscheinlichkeit λ) ein zweites wichtiges Modell für die synaptischen Fehler auf: Es ist sinnvoll, die Auswirkungen des Austauschs $J_{ij} \to -J_{ij}$ (mit Wahrscheinlichkeit λ) zu untersuchen.

Die Speicherkapazität des binären Modells unter Verdünnung ist kürzlich mit Gardners Methoden analysiert worden [Bouten *et. al.* 90]. Die Speicherkapazität der *zero-entropy* Lösung liegt etwa bei 1.17, also höher als im binären Modell. Andererseits sind in diesem Netzwerk die Werte $J_{ij} = -1, 0, 1$ für die Kopplungen zugelassen, und die Speicherkapazität sollte daher größer sein als für das binäre Modell $J_{ij} = \pm 1$: Die Informationstheorie liefert für dieses Modell die Grenze $\alpha_c < \log_2 3 = 1.58$.

Für den Vergleich der Leistung der neuronalen Netzwerke (ihrer Fehlertoleranz) mit fehlerkorrigierenden Codes ist es dagegen vorteilhaft, als Modell für synaptische Fehler den Austausch $J_{ij} \to -J_{ij}$ mit Wahrscheinlichkeit λ zu verwenden.

Fast alle Formeln aus Abschnitt 4.2.2 können unverändert auch für dieses Modell



Anteil der pro Neuron stabilisierten Muster $1-\Gamma(0,\lambda)$ als Funktion der Konzentration der zerstörten ("geflippten") Synapsen im binären Netzwerk. ($N = 256$, $\alpha = 0.05$ $\bullet$,0.15 $\diamond$, 0.25 $\star$, 0.30 $*$, 0.35 $\oplus$, 0.40 $\otimes$).

Abbildung 28: Anteil der pro Neuron gespeicherten Muster als Funktion von λ, verglichen mit dem constant-stability Modell (48).

von Fehlern im Netzwerk übernommen werden. Der Mittelwert und die Varianz der Gauß-Verteilung nach der Zerstörung müssen jedoch angepaßt werden. Man erhält,

$$\kappa_\lambda = (1-2\lambda)\kappa_0$$

und

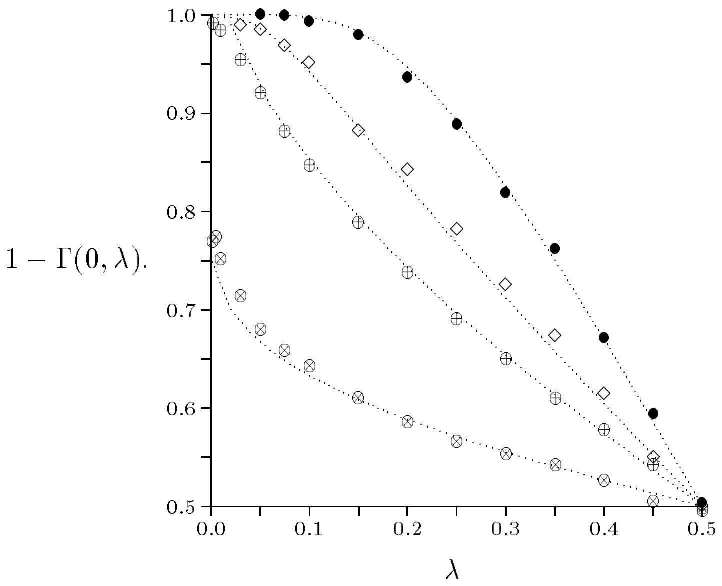
$$\sigma_\lambda = 2\sqrt{\lambda(1-\lambda)}.$$

Da die Stabilitäten unter den Transformationen $J_{ij} \to -J_{ij}$ und $\kappa \to -\kappa$ invariant sind, müssen diese Funktionen natürlich die Symmetrie $|f(\lambda)| = |f(1-\lambda)|$ aufweisen.

Um Simulationen mit dem binären Netzwerk durchführen zu können, muß eine geeignete Lernregel ausgewählt werden. Dies ist ein schwieriges Problem, weil die iterativen Lernalgorithmen für binäre Synapsen versagen. Die geclippte Hopfield-Matrix $J_{ij} = \mathrm{sgn}[\sum_\mu^P \xi_i^\mu \xi_j^\mu]$ erreicht zum Beispiel nur eine Speicherkapazität von etwa $\alpha_{bH} \approx 0.1$.

Deshalb wurde zur Einstellung der Kopplungsmatrix in den Simulationen ein neuer Algorithmus benutzt, der in ähnlicher Form von [Koehler *et. al.* 89] vorgeschlagen wurde und mit dem sich eine Speicherkapazität $\alpha \approx 0.4$ erreichen läßt.

Die Ergebnisse der Simulationen für die Speicherung der Muster als Funktion von λ sind in Abbildung 28 dargestellt und mit dem constant-stability Modell verglichen.



Anteil der pro Neuron stabilisierten Muster $1-\Gamma(0,\lambda)$ als Funktion der Konzentration der zerstörten ("geflippten") Synapsen im binären Netzwerk. ($N = 256$, $\alpha = 0.05$ $\bullet$, 0.15 $\diamond$, 0.30 $\oplus$, 0.40 $\otimes$). Die statistischen Fehler der Simulationen erreichen etwa zweimal die Größe der Symbole.

Abbildung 29: Anteil der pro Neuron gespeicherten Muster als Funktion von λ, verglichen mit dem über die Anfangsverteilung der Stabilitäten integrierten constant-stability Modell (49).

Für kleine Werte der Speicherkapazität ($\alpha \leq 0.2$) erreicht der verwendete Lernalgorithmus relativ hohe Werte für die Stabilitäten der Sollmuster (und speichert diese fehlerfrei); dementsprechend stimmen die Ergebnisse der Simulationen sehr gut mit dem constant-stability Modell überein.

Bei höheren Werten der Speicherkapazität $\alpha \approx 0.2 \ldots 0.4$ gelingt es nicht mehr, alle Muster fehlerfrei zu speichern und die erreichten Stabilitäten sind sehr niedrig. Außerdem wird die Verteilung der Stabilitäten sehr breit. Es verwundert deshalb nicht, wenn die Ergebnisse der Simulationen für $\alpha \approx 0.2 \ldots 0.4$ nicht mehr genau mit dem in der Abbildung gezeigten constant-stability Modell übereinstimmen. Natürlich kann das Modell wieder durch die Integration über die Verteilung $\rho^{(0)}(\kappa)$ der Stabilitäten vor der Zerstörung der Kopplungen verbessert werden, siehe Abbildung 29.

In den Abbildungen 28 und 29 sind zusätzlich Simulationen (bei $\alpha = 0.4$) eingetragen, bei der das Netzwerk nach dem Lernen nur etwa 80% der Muster korrekt speichert. Obwohl die vom verwendeten Lernalgorithmus dabei erreichte Verteilung der Stabilitäten nur einen Mittelwert von etwa $\kappa_0 \approx 0.31$ aufweist und eine sehr große Varianz besitzt (viele Stabilitäten sind negativ), ist sogar die Übereinstimmung mit dem einfachen constant-stability Modell noch recht gut.

4.6 Vergleich mit der Kodierungstheorie

Eine vollständige Analyse neuronaler Modelle mit den Mitteln der Kodierungstheorie liegt außerhalb des Rahmens dieser Arbeit. Trotzdem erscheinen hier zwei kurze Bemerkungen sinnvoll, um wenigstens grob eine Einordnung vorzunehmen. Der interessante Punkt ist, daß das neuronale Netzwerk einen Decoder für einen fehlerkorrigierenden Code (assoziative Speicherung) realisiert, der selbst fehlertolerant ist.

Auto-assoziative Speicherung läßt sich nämlich auch als die Aufgabe eines fehlerkorrigierenden Codes verstehen: Unvollständige (verrauschte) Eingangsdaten werden in den Speicher eingegeben und man erwartet die passenden gespeicherten Muster — die Codewörter — als Ausgangssignale. In der Terminologie der Kodierungstheorie, siehe etwa [Peterson 63], überträgt ein binärer (n, k) Code mit jedem Codevektor der Länge n Bit k Bit Information. Ein Code kann bis zu t Bitfehler in den Eingabedaten korrigieren, wenn er ein *Minimalgewicht*, das ist die minimale Hamming-Distanz zweier Codewörter, von $d \geq 2t + 1$ besitzt.

Die Anzahl der Bitfehler in den Eingabemustern, die von einem Code überhaupt korrigiert werden kann, läßt sich mit der Kodierungstheorie als Funktion von n und k abschätzen. Allerdings lassen sich die theoretisch gewonnenen oberen Schranken nicht alle auch realisieren. Tatsächlich sind viele häufig eingesetzte Codes nicht *optimal*, sondern zeichnen sich vielmehr durch einfache Realisierungen aus. Relativ einfache statistische Überlegungen (siehe [Peterson 63]) führen auf folgende untere Schranke für die Anzahl der Kontrollstellen $n - k$ in einem Code mit Minimalgewicht d (für $n \gg 1$):

$$\frac{n-k}{n} \geq H\Big(d/n\Big),$$

wobei $H(x) = -x\log_2 x - (1-x)\log_2(1-x)$.

Da ein neuronales Netzwerk mit N Neuronen bis zu $P = \alpha_c N$ Muster der Länge N zu speichern vermag, stellt es einen *random code* mit Parametern $(N, \log_2 \alpha N)$ dar, weil die Codevektoren (d. h. die Sollmuster) frei gewählt werden können und $\log_2 \alpha N$ Bit ausreichen, um die gespeicherten Muster zu indizieren.

Die Angabe des Minimalgewichts der Codewörter ist für das neuronale Netzwerk nicht sinnvoll: Das Erkennen eines Eingabemusters hängt ja nicht nur von dessen Hamming-Distanz zu den Sollmustern ab, sondern auch von dynamischen Eigenschaften. Die Zahl der Bitfehler, die vom Netzwerk korrigiert werden können, kann daher nur durch die Betrachtung der Einzugsbereiche ermittelt werden. Unter der Annahme, daß das Netzwerk alle Muster mit $m_0 > m_c$ richtig zuordnet, kann es offenbar bis zu $t < (1 - m_c)\,N/2$ Bitfehler korrigieren. Diese Abschätzung ist nicht exakt (die Simulationen zeigen, daß immer auch einige Testmuster mit $m_0 > m_c$ nicht korrekt erkannt werden), sollte jedoch jedenfalls für großes N eine sehr gute Näherung darstellen.

Zum Vergleich mit dem neuronalen Netzwerk bieten sich die Reed-Muller-Codes an ([Peterson 63]), da ihre Eigenschaften besonders einfach beschrieben werden können, und zwar auch für große Wortlänge N der Codewörter. Diese Codes sind nur in Spezialfällen optimal, erlauben aber elegante Realisierungen. Andere wichtige Codes, die

die Korrektur mehrfacher Fehler erlauben — etwa die BCH-Codes — sind wesentlich schwieriger sowohl zu realisieren als auch zu analysieren. Ein Reed-Muller-Code wird durch seine Wortlänge $N = 2^m$ und einen Parameter r gekennzeichnet, mit

$$k = 1 + \binom{m}{1} + \binom{m}{2} + \cdots + \binom{m}{r},$$

so daß ein (N, k) Code resultiert. Dieser Code hat dann ein Minimalgewicht $d = 2^{m-r}$, kann also bis zu $t \leq 2^{m-r-1} - 1$ Bitfehler korrigieren. Insbesondere mit $r = 0$ ergeben sich die *repetition codes*, die mit jedem N-Bit Wort nur genau 1 Bit übertragen, dafür jedoch bis zu $t < N/2$ Bitfehler korrigieren können. Für $r = 1$ ergeben sich $(N, \log_2 N)$ Codes, die bis zu $t < N/4$ Fehler korrigieren können.

Dies ist durchaus vergleichbar mit der Leistung der neuronalen Netzwerke, die mit Parametern $(N, \log_2 \alpha N)$ bis zu $t \approx N/2$ Bitfehler (jedenfalls für $\alpha < 0.4$) korrigieren können. Die Senderate k/N der Netzwerke ist etwas kleiner, dafür können sie aber mehr Fehler korrigieren als die Reed-Muller-Codes mit $r = 1$. Die Netzwerke stellen damit eine zusätzliche Klasse von random codes mit geringer Senderate dar.

Dabei ist interessant, daß das neuronale Netzwerk als Decoder für diesen fehlerkorrigierenden Code selbst fehlertolerant ist, da nach den Ergebnissen der Abschnitte 4.2 bis 4.5 Zerstörungen des Netzwerks (das heißt, des Decoders) den Code selbst weitgehend funktionsfähig lassen.

Dies ist eine Alternative zur üblichen Betrachtungsweise der Kodierungstheorie: Dort wird fast immer ein verrauschter Übertragungskanal betrachtet, der für alle Störungen verantwortlich ist, während Encoder und Decoder der Codes als fehlerfrei angenommen werden. Da sich mit biologischen Neuronen (höchstwahrscheinlich) überhaupt keine fehlerfreien Schaltungen realsieren lassen, ist diese Art der Betrachtung für die Untersuchung biologischer Systeme nur begrenzt sinnvoll. Vielmehr muß auch eine fehlertolerante Realisierung des Decoders gefordert werden und neuronale Netzwerke liefern eine elegante Lösung.

5 Diskussion

Es wurde das einfache „constant-stability“ Modell für die Beschreibung von neuronalen Netzwerken unter zufälliger Zerstörung von Synapsen vorgestellt.

Die Verteilung der lokalen Felder läßt sich in diesem Modell als Funktion der Anfangsstabilität κ_0 und der Konzentration λ der zerstörten Synapsen berechnen. Damit können sowohl die statischen (Speicherkapazität) als auch die dynamischen (Einzugsbereiche) Eigenschaften der Netzwerke berechnet werden.

Netzwerke mit beliebiger Anfangsverteilung $\rho(\kappa)$ der Stabilitäten können durch Integration über Netzwerke mit konstanter Stabilität κ_0 sehr genau beschrieben werden. In vielen Fällen reicht aber für Speicherkapazitäten $\alpha < 0.4$ auch die Beschreibung der Netzwerke nur durch eine konstante (mittlere) Stabilität aus.

Die Netzwerke erweisen sich als fehlertolerant: Obwohl im Limes $N \to \infty$ jede endliche Konzentration von Fehlern auch zu Fehlern in den Mustern führt, ist die Konzentration von Fehlern bis zu beträchtlichen Werten von λ sehr klein. Für $\alpha = 0.4$ findet man beispielsweise, daß die Muster für $\lambda < 0.224$ mit weniger als 1% Fehlern gespeichert werden.

Die in den Netzwerken möglichen Konzentrationen der Zerstörung sind wesentlich höher als für elektronische Realisierungen charakteristisch: Die Konzentration von Fehlern bei der Fertigung von Mikrochips liegt nicht in der Größenordnung $\lambda \approx 0.1$ sondern ist kleiner als $\lambda \approx 0.001$. Das bedeutet, daß durch Fertigungsfehler bedingte synaptische Fehler in künstlichen neuronalen Netzen kaum zu ernsthaften Störungen der Funktion führen können. Vielmehr ist die Auswirkung von Fehlern mit einer Konzentration $\lambda \approx 0.001$ beträchtlich kleiner als etwa eine Verkürzung der Lernphase.

Literatur

[Abbott & Kepler 89a] L. F. Abbott and T. B. Kepler, Optimal learning in neural network models, *Journal of Physics* A 22, L711–L717 (1989).

[Abbott 90] L. F. Abbott, Learning in neural network memories, *Network* 1, 105–122 (1990).

[Alberts *et. al.* 83] B. Alberts *et. al.*, *Molecular Biology of the Cell*, Garland Publishing, 1983.

[Amit *et. al.* 85] D. J. Amit, H. Gutfreund and H. Sompolinsky, Spin-glass models of neural networks, *Physical Review A* 32, 1007–1018 (1985).

[Amit *et. al.* 87] D. J. Amit, H. Gutfreund and H. Sompolinsky, Statistical Mechanics of Neural Networks near Saturation, *Annals of Physics* 173, 30–67 (1987).

[Amit *et. al.* 87b] D. J. Amit, H. Gutfreund and H. Sompolinsky, Information storage in neural networks with low levels of activity, *Physical Review* A 35, 2293–2303 (1987).

[Amit *et. al.* 89] D. J. Amit, C. Campbell and K. Y. M. Wong, The interaction space of neural networks with sign-constrained synapses, *Journal of Physics* A 22, 4687–4693 (1989).

[Anderson & Rosenfeld 88] J. A. Anderson and E. Rosenfeld, Eds., *Neurocomputing — Foundations of Research*, MIT Press, Cambridge, Mass., 1988

[Binder & Young 86] K. Binder and A. P. Young, Spin Glasses, *Rev. Mod. Phys.* 58-4 (1986).

[Bouten *et. al.* 90] M. Bouten, A. Komoda and R. Serneels, Storage capacity of a diluted neural network with Ising couplings, *Journal of Physics* A 23, 2605–2612 (1990).

[Bruce *et. al.* 86] A. D. Bruce, A. Canning, B. Forrest, E. Gardner and D. J. Wallace, Learning and Memory Properties in Fully Connected Networks, *AIP Conference Proceedings # 151, Neural Networks for Computing*, 1986, 65

[Bruce *et. al.* 87] A. D. Bruce, E. J. Gardner and D. J. Wallace, Dynamics and statistical mechanics of the Hopfield-model, *Journal of Physics* A 20, 2909–2934 (1987).

[Canning & Gardner 88] A. Canning and E. Gardner, Partially connected models of neural networks, *Journal of Physics* A 21, 3275–3284 (1988).

[Derrida *et. al.* 87] B. Derrida, E. Gardner and A. Zippelius, An exactly solvable neural network model, *Europhysics Letters* 4, 481 (1987).

[Diederich & Opper 87] S. Diederich and M. Opper, Learning of Correlated Patterns in Spin-Glass Networks by Local Learning Rules, *Physical Review Letters* 58, 949–952 (1987).

[Fontanari & Köberle 90] J. F. Fontanari and R. Köberle, Landscape statistics of the binary perceptron, *Journal de Physique (France)* 51, 1403–1413 (1990).

[Fontanari & Theumann 90] J. F. Fontanari and W. K. Theumann, On the storage of correlated patterns in Hopfield's model, *Journal de Physique (France)* 51, 375–386 (1990).

[Forrest 88] B. M. Forrest, Content-addressability and learning in neural networks, *Journal of Physics* A 21, 245-255 (1988).

[Gardner 86] E. Gardner, Structure of metastable states in the Hopfield model, *Journal of Physics* A 19, L1047–L1052 (1986).

[Gardner 87] E. Gardner, Maximum Storage Capacity in Neural Networks, *Europhysics Letters* 4, 481–485 (1987).

[Gardner 87b] E. Gardner, Multiconnected neural network models, *Journal of Physics* A 20, 3453–3464 (1987).

[Gardner *et. al.* 87] E. Gardner, B. Derrida and P. Mottishaw, *Journal de Physique (France)* 48, 741 (1987).

[Gardner 88a] E. Gardner, The space of interactions in neural network models, *Journal of Physics* A 21, 257–270 (1988). (Edinburgh Preprint No. 87/396).

[Gardner & Derrida 88] E. Gardner, B. Derrida, Optimal storage properties of neural network models, *Journal of Physics* A 21, 271–284 (1988).

[Gardner 89a] E. Gardner, Optimal basins of attraction in randomly sparse neural network models, *Journal of Physics* A 22, 1969–1974 (1989).

[Gardner 89b] E. Gardner, B. Derrida, Three unfinished works on the optimal storage capacity of networks, *Journal of Physics* A 22, 1983–1994 (1989).

[Gardner 89c] E. J. Gardner, N. Stroud and D. J. Wallace, Training with noise and the storage of correlated patterns in a neural network model, *Journal of Physics* A 22, 2019–2030 (1989).

[Gutfreund & Stein 90] H. Gutfreund and Y. Stein, Capacity of neural networks with discrete synaptic couplings, *Journal of Physics* A 23, 2613–2630 (1990).

[Heiden 80] U. an der Heiden, *Analysis of Neural Networks*, Springer-Verlag, Berlin, 1980

[Hopfield 82] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proc. Nat. Acad. Sci.* 79, 2554–2558 (1982).

[Kanter & Sompolinsky 87] I. Kanter and H. Sompolinsky, Associative recall of memory without errors, *Physical Review* A 35, 380–392 (1987).

[Katz 87] B. Katz, *Nerv, Muskel und Synapse*, Thieme Verlag, 1987.

[Kepler & Abbott 88] T. B. Kepler and L. F. Abbott, Domains of attraction in neural networks, *Journal de Physique (France)* 49, 1657–1662 (1988).

[Kirkpatrick & Sherrington 78] S. Kirkpatrick and D. Sherrington, Infinite ranged models of spin-glasses, *Phys. Rev. B* 17, 4384 (1978)

[Koehler *et. al.* 89] H. Koehler, S. Diederich, W. Kinzel and M. Opper, Learning Algorithm for a Neural Network with Binary Synapses, *private communication* (Uni Giessen preprint).

[Kohonen 72] T. Kohonen, Correlation matrix memories, *IEEE transactions on computers* C-21: 353–359 (1972).

[Kohonen 84] T. Kohonen, *Self-Organization and associative Memory*, Springer-Verlag, Berlin, 1984.

[Koscielny-Bunde 90] E. Koscielny-Bunde, Effect of damage in neural networks, *Journal of Statistical Physics*, 1990.

[Krauth & Mézard 87] W. Krauth and M. Mézard, Learning algorithms with optimal stability in neural networks, *Journal of Physics* A 20, L745–L752 (1987).

[Krauth *et. al.* 88] W. Krauth, J.-P. Nadal and M. Mézard, The roles of stability and symmetry in the dynamics of neural networks, *Journal of Physics* A 21, 2995–3011 (1988).

[Krauth & Opper 89] W. Krauth and M. Opper, Critical storage capacity of the $J = \pm 1$ neural network, *Journal of Physics* A 22, L519–L523 (1989).

[Krauth & Mézard 89] W. Krauth and M. Mézard, Storage capacity of memory networks with binary couplings, *Journal de Physique (France)* 50, 3057–3066 (1989).

[Little 74] W. A. Little, The existence of persistent states in the brain, *Mathematical Biosciences* **19**, 101–120 (1974).

[McClelland & Rumelhardt 86] J. L. McClelland, D. E. Rumelhart and the PDP Research Group, *Parallel Distributed Processing — Explorations in the Microstructure of Cognition*, MIT Press, Cambridge, Mass., 1986.

[McCulloch & Pitts 43] W. S. McCulloch and W. Pitts, A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics* 5, 115–133 (1943).

[Mézard, Parisi & Virasoro 87] M. Mézard, G. Parisi and M. A. Virasoro, *Spin glass theory and beyond*, World Scientific Lecture Notes in Physics Vol. 9, Singapure 1987.

[Minsky & Papert 69] M. Minsky and S. Papert, *Perceptrons*, MIT Press, Cambridge, Mass., 1969.

[Nardulli & Pasquariello 90] G. Nardulli and G. Pasquariello, Domains of attraction of neural networks at finite temperature, *International Neural Network Conference, Paris, 1990*, (Univ. Bari preprint TH/90-66), to appear in *Journal of Physics* A.

[Numerical Recipes] W. H. Pless, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes — The Art of Scientific Computing*, Cambridge University Press, 1986.

[Opper *et. al.* 89] M. Opper, J. Kleinz, H. Koehler and W. Kinzel, Basins of attraction near the critical storage capacity for neural networks with constant stabilities, *Journal of Physics* A 22, L407–L411 (1989).

[Pazmandi & Geszti 89] R. Pazmandi and T. Geszti, Relative stability in the dynamics of a two-pattern neural net, *Journal of Physics* A 22, 5117–5129 (1989).

[Personnaz *et. al.* 85] L. Personnaz, I. Guyon and G. Dreyfus, *Journal de Physique* 46, L-359 (1985).

[Peterson 63] W. W. Peterson, *error-correcting codes*, MIT Press, Camb., 1963.

[Rosenblatt 58] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain, *Psychological Review* 65, 386–408 (1958).

[Sompolinsky & Kanter 86] H. Sompolinsky and I. Kanter, *Phyical Review Letters* 57, 2861–2864 (1986).

[Zippelius & Horner 88] A. Zippelius und H. Horner, *Spingläser und Hirngespinste*, Sommerakademie der Studienstiftung des Deutschen Volkes, 1988.

A Symbolverzeichnis

a	"Magnetisierung" (bzw. Korrelation) der Sollmuster
a_1, a_2	Konstante für f_p-Fit
α	Speicherkapazität, -ausnutzung, P/C
$\alpha_c(\kappa)$	optimale Speicherkapazität im Gardner-Modell
α_{cB}	optimale Speicherkapazität im binären Modell
β	inverse Temperatur
c	Konnektivität, $c = C/N$
$c_{\mu\nu}$	Korrelationsmatrix der Sollmuster
C	Zahl der Synapsen pro Neuron, $C := \langle C_{ij} \rangle$
C_{ij}	Konnektivitätsmatrix, $C_{ij} \in \{0, 1\}$. Kopplung von Neuron j nach Neuron i ist vorhanden, wenn $C_{ij} = 1$
d	Minimalgewicht eines Codes
δ_{ij}	Kronecker-Symbol
$\delta(x - y)$	Dirac-Deltafunktion
δ	positive Konstante (Konvergenzbeweis Lernen)
E	Energie des Netzwerks in einer Konfiguration $\{S\}$, im Hopfield-Modell ist $E = -(1/2N) \sum_{i,j} J_{ij} S_i S_j$
E_i	Wert der "Energie" am Neuron i beim Lernalgorithmus nach [Koehler *et. al.* 89]
$\epsilon_{i\mu}$	Fehlermaske beim iterativen Lernen, $\epsilon_{i\mu} = 1$ wenn Muster μ am Neuron i nicht ausreichend stabilisiert
$\mathrm{erf}(x)$	Fehlerfunktion, $\mathrm{erf}(x) = \sqrt{2/\pi} \int_0^x \mathrm{d}t \exp(-t^2)$
f	freie Energie pro Spin (stat. Mech. des Hopfield-Modells)
f_p	Anteil fehlerfrei erkannter Muster ("fraction recalled perfectly")
$f(x)$	optimale Schrittweite beim iterativen Lernen
$F_\kappa(m_0)$	$m_1 = F_\kappa(m_0, \rho(\kappa))$
$\Gamma(k, \lambda)$	Anteil der pro Neuron mit Stabilität $\kappa_{i\mu} > k$ nach Zerstörung mit Wahrscheinlichkeit λ gespeicherten Muster
h, $h_{i\mu}$	lokales Feld eines Neurons (Neuron i für Muster μ)
J_{ij}	Kopplungsmatrix, Kopplung von Neuron j nach Neuron i, J_{ij} reell, binär oder ganzzahlig.
k	Anzahl der Informationsstellen im (N, k) Code
κ, $\kappa_{i\mu}$	Stabilität $\kappa_{i\mu} = h_{i\mu} \cdot \xi_i^\mu$ am Neuron i für Muster μ
$\kappa_c(\alpha)$	optimale Stabilität im Gardner-Modell
κ_0	Wert der Stabilität im constant-stability Modell (vor der Zerstörung)
λ	Konzentration von zerstörten Synapsen, $C \simeq (1 - \lambda)N$
Λ	Integrationsvariable, Verteilung der Stabilitäten
m	Überlapp zweier Konfigurationen, $m = (1/N) \sum_{j \neq i} S_j^{(1)} S_j^{(2)}$
m_i, m_f	"initial, final overlap"

m_0, m_1	Überlapp vor und nach einem Schritt der Dynamik
m_c	numerisch ermittelter kritischer Anfangsüberlapp: Größe des Einzugsbereiches
m_F, m_S	Modelle für Einzugsbereiche in voll vernetzten (Fully connected) und extrem verdünnten (Sparse connected) Netzwerken
μ	Muster-Index, $\mu = 1, \ldots, P$
n	Anzahl von Replikas
N	Anzahl der Neuronen im Netzwerk
ν	Muster-Index, $\nu = 1, \ldots, P$
P	Anzahl der zu speichernden Sollmuster
$P(y)$	Wahrscheinlichkeitsverteilung von y
$P(a\|b)$	bedingte Wahrscheinlichkeit von a wenn b
$\rho(\kappa)$	Verteilung der Stabilitäten im Netzwerk
R	Radius der Einzugsbereiche, $R = \langle 1 - m_c \rangle$
S_i	Spin/Neuron i, $S_i = \pm 1$
$\{S\}$, $\{S_i\}$, $\{S_f\}$	Konfiguration, $\{S\} = (S_1, \ldots, S_N)$, Anfangs-, Endkonfiguration
σ	Replika-Index, $\sigma = 1, \ldots, n$ Standardabweichung (Breite einer Gauß-Verteilung)
t, t_i	Zeit, Zeitschritt i Anzahl der korrigierbaren Bitfehler in einem Code
T	Temperatur (Rauschen) im Netzwerk
$\Theta(x)$	Heavyside-Funktion
Tr_S	Spur über die Spins S (für die Berechnung von Z^n)
ξ_i^μ	Wert der Sollmusters μ am Neuron i ($\xi_i^\mu = \pm 1$)
$X_i^{(n)}$	Hilfsgröße im Beweis der Konvergenz der Lernregeln
Z	Zustandssumme
$\langle x \rangle$	Mittelung von x über die Neuronen
$\langle\langle x \rangle\rangle$	*quenched average* von x über die Muster ξ_i^μ

B Iteratives Lernen in verdünnten Netzwerken

Die Konvergenz des iterativen Lernalgorithmus (24) im Netzwerk mit Parametern α, κ läßt sich unter der Voraussetzung beweisen, daß überhaupt eine Lösung möglich ist. Eine ausführliche Beschreibung der Rechnungen für das vollständig verknüpfte Netzwerk findet sich etwa in [Gardner 88a].

Eine Verallgemeinerung ermöglicht die Anwendung des Beweises auch für verdünnte Netzwerke. Der Algorithmus (24) besteht in der Iteration von Hebb-Schritten $J_{ij} \to J_{ij} + \Delta J_{ij}$ für die Muster, die am entsprechenden Neuron noch nicht gespeichert sind,

$$\Delta J_{ij} = N^{-1}\, C_{ij}\epsilon_{i\mu}\xi_i^\mu \xi_j^\mu \tag{50}$$

mit der Fehlermaske $\epsilon_{i\mu} = \Theta(\kappa - \kappa_{i\mu})$, das heißt,

$$\epsilon_{i\mu} = \Theta\Big[\kappa\Big(\sum_{j\neq i}(C_{ij}J_{ij})^2\Big)^{1/2} - \sum_{j\neq i} C_{ij}J_{ij}\xi_j^\mu\xi_i^\mu\Big]. \tag{51}$$

Der Algorithmus kann parallel über alle Neuronen, muß aber seriell über die Muster durchgeführt werden.

Sei $J_{ij}^* = C_{ij}J_{ij}^*$ eine Einstellung der Kopplungen, so daß

$$\xi_i^\mu \sum_{j\neq i} C_{ij}J_{ij}^*\xi_j^\mu > (\kappa+\delta)\Big(\sum_{j\neq i}(C_{ij}J_{ij}^*)^2\Big)^{1/2} \tag{52}$$

mit einer positiven Konstante δ für alle Neuronen i und Muster μ.

Da der Algorithmus die Synapsen verschiedener Neuronen S_i unabhängig voneinander einstellt, reicht es natürlich, ein einzelnes Neuron zu betrachten. Sei dazu über

$$(J\cdot U)_i := \sum_{j\neq i} J_{ij}U_{ij} \tag{53}$$

das Skalarprodukt zweier Kopplungsmatrizen J und U am Neuron i definiert und entsprechend

$$\|J\|_i = (J\cdot J)_i^{1/2} \tag{54}$$

die Norm von J am Neuron i. Sei $\{J_{ij}^{(n)}\}$ die Kopplungsmatrix nach n Iterationen der Lernregel (50) und

$$X_i^{(n)} = \frac{(J^{(n)}\cdot J^*)_i}{\|J^{(n)}\|_i\|J^*\|_i.} \tag{55}$$

Die Idee des Beweises ist anzunehmen, daß der Algorithmus nicht in n Iterationen konvergiert. Es wird sich zeigen, daß $X_i^{(n)}$ für genügend großes n größer als 1 wird, und dies widerspricht der Cauchy-Schwarz-Ungleichung. Also muß der Algorithmus konvergieren.

Im folgenden wird zusätzlich $J_{ij}^{(0)} = C_{ij}J_{ij}^{(0)}$ und $J_{ij}^* = C_{ij}J_{ij}^*$ vorausgesetzt, so daß auch $J_{ij}^{(n)} = C_{ij}J_{ij}^{(n)}$ gilt. Die Verknüpfungsmatrizen C_{ij} brauchen dann nicht mehr explizit aufgeführt zu werden.

Die Änderung des Zählers von (55) bei der Iteration n ist

$$\begin{aligned} \Delta(J^{(n)} \cdot J^*)_i &= \epsilon_{i\mu} \sum_{j \neq i} \xi_i^\mu \xi_j^\mu J_{ij}^* \\ &> \Big(\sum_{j\neq i} (J_{ij}^*)^2\Big)^{1/2} (\kappa + \delta) \end{aligned} \tag{56}$$

und daher gilt

$$(J^{(n)} \cdot J^*)_i > \|J^*\|_i(\kappa+\delta)n + (J^{(0)} \cdot J^*)_i. \tag{57}$$

Die Änderung des Nenners kommt durch die Änderung der Norm von $J^{(n)}$ zustande,

$$\begin{aligned} \Delta(J^{(n)} \cdot J^{(n)})_i &= 2\epsilon_{i\mu} \sum_{j\neq i} J_{ij} \xi_j^\mu \xi_i^\mu + C\epsilon_{i\mu} \\ &< \epsilon_{i\mu}(2\kappa\|J^{(n)}\|_i + C), \end{aligned} \tag{58}$$

so daß

$$\Delta\|J^{(n)}\|_i < \kappa + C/2\|J^{(n)}\|_i. \tag{59}$$

Nach einiger Algebra ([Gardner 88a]) erhält man folgende Abschätzung für $X_i^{(n)}$,

$$X_i^{(n)} > \frac{(\kappa+\delta+O(1/n))}{(\kappa + \frac{\ln n}{n}\frac{C}{2(\kappa+\delta)} + O(1/n))}. \tag{60}$$

Also wird $X_i^{(n)}$ nach genügender Anzahl der Lernschritte größer als 1, und dies widerlegt die Behauptung, der Algorithmus konvergiere nicht.

Kürzlich haben [Abbott & Kepler 89a] untersucht, wie groß die einzelnen Lernschritte sein dürfen, ohne daß die Konvergenz gefährdet wird. Die Idee ist, die Synapsen beim Lernschritt nicht gemäß (50) zu modifizieren, sondern mit einer von κ und $\kappa_{i\mu}$ sowie von $\|J^{(n)}\|$ abhängigen Schrittweite,

$$\Delta J_{ij} = N^{-1}\ f(\kappa, \kappa_{i\mu})\|J^{(n)}\|\ \ \cdot C_{ij}\epsilon_{i\mu}\xi_i^\mu \xi_j^\mu. \tag{61}$$

Man findet, daß jede Wahl von $f(\kappa, \kappa_{i\mu})$ mit $0 < f(\kappa_{i\mu}) < 2(\kappa+\delta-\kappa_{i\mu})$ zu konvergentem Lernen führt. Die Konvergenzgeschwindigkeit ist optimal, wenn $f(\kappa_{i\mu}) = \kappa + \delta - \kappa_{i\mu} + [(\kappa+\delta-\kappa_{i\mu})^2 - \delta^2]^{1/2}$ gewählt wird.

Kann dieser Konvergenzbeweis auch auf andere Algorithmen verallgemeinert werden? Für die Implementation in digitalen Systemen ist dabei vor allem wichtig, ob eine andere als die sphärische Norm benutzt werden kann — etwa die Betragsnorm oder eine Maximumnorm. Die numerischen Resultate deuten darauf hin, daß die Verteilung der Synapsen nach dem Lernen sowohl mit sphärischer als auch mit Betragsnorm in sehr guter Näherung durch eine Gauß-Verteilung beschrieben werden kann. In diesem Fall sind die beiden Normen zueinander proportional, und die obigen Abschätzungen können übernommen werden. Die Lernregel konvergiert also auch bei Verwendung der Betragsnorm.