Universität Bremen

Aktive Computation in offenen Systemen

Lerndynamiken in biologischen Systemen:
Vom Netzwerk zum Organismus

Frank Emmert-Streib

Januar 2003

Active computation in open systems

Frank Emmert-Streib

Abstract

In this thesis the action-perception-cycle, which is inherent in every cognitive system, is investigated by several special examples. These examples are taken from different levels of description and are biological motivated. Moreover, a novel Hebb-like learning rule for neural networks is introduced, which has not only interesting features by itself but establishes a connection to the action-perception-cycle and demonstrates the interplay between different levels of description.

The novel Hebb-like learning rule is introduced in chapter 3. The learning rule is based on the introduction of a new degree of freedom, assigned to each neuron in form of a neuron counter. In combination with an external reinforcement signal the synapses which are involved in bad computations are allowed to change according to a stochastic update mechanism which selects synapses according to the counter states of the surrounding neuron counters. The features of the Hebb-like learning rule are investigated in a multilayer neural network which has to learn the exclusive-or (XOR) problem. Different forms of network dynamics are investigated and the results are compared to another Hebb-like learning rule which was introduced by [Klemm et al., 2000]. The results obtained by numerical simulations give similar good results as [Klemm et al., 2000] but under usage of less free parameters. Moreover, a qualitative biological interpretation in form of heterosynaptic long-term potentiation (LTD) can be given.

In chapter 4 a method is developed which allows the assignment of an approximative temperature to a noisy winner-take-all mechanism similar to a temperature in a softmax mechanism. In the second part of the chapter this method is used to assign an approximative temperature at each time step to a multilayer neural network which is trained to learn the exclusive-or (XOR) problem. Results show that the time course of the temperature is annealed the better the neural network solves the problem. This indicates that there is a correspondence between the time course of mean error and the approximative temperature which is annealed in a selforganized way without any additional scheduling besides the learning rule to train the network.

In chapter 5 the generic character of the novel Hebb-like learning rule is demonstrated by training a class of neural networks to learn the problem of timing. The class of neural networks is given by an algorithm of [Watts und Strogatz, 1998] and includes small world networks. Moreover, the influence of the action-selection-mechanism on the convergence time of the neural networks and the interplay with the learning rule is investigated. This establishes a connection to the action-perception-cycle. Besides this, a new error measure for an outer observer is introduced to strictly separate information of the neural network and the outer observer.

The work is completed in chapter 6 by an investigation of an agent in a space like environment. The mathematical model which describes the action-perception-cycle of the agent in a one dimensional grid is presented in detail. Especially the action-selection-mechanisms used and a measure for an outer observer, similar to chapter 5, are thoroughly introduced and discussed. Results show that different kinds of action-selection-mechanisms lead to great differences in the convergence times although all parts of the whole system are identical besides the action-selection-mechanism used. The chapter ends with a biological interpretation of the presented model which can partly be related to the consilidation process of spacial memories in the hippocampus and the neuroanatomical structures therein.

Aktive Computation in offenen Systemen

Lerndynamiken in biologischen Systemen:
Vom Netzwerk zum Organismus

Dissertation

zur Erlangung des Grades

Doktor der Naturwissenschaften

vorgelegt von

Frank Emmert-Streib

Institut für Theoretische Physik

Fachbereich Physik/Elektrotechnik

Universität Bremen

Januar 2003

Diese Dissertation wurde am Institut für Theoretische Physik — Abteilung Bio- und Neurophysik — des Fachbereichs Physik/Elektrotechnik der Universität Bremen in der Arbeitsgruppe von Prof. Dr. Klaus Pawelzik angefertigt.

1. Gutachter: Prof. Dr. Klaus Pawelzik

2. Gutachter: Prof. Dr. Heinz Georg Schuster

Im Gedenken an Per Bak.

(1948–2002)

Inhaltsverzeichnis

1 Einleitung **1**

2 Grundlagen: Lernen in neuronalen Systemen **7**
2.1 Einleitung . . . 7
2.2 Biologische und Modellneuronen . . . 7
2.3 Einteilung von Lernregeln . . . 9
2.3.1 *supervised learning* . . . 9
2.3.1.1 Backpropagation Algorithmus . . . 10
2.3.1.2 Reinforcement Lernen . . . 13
2.3.2 *unsupervised learning* . . . 16
2.3.2.1 Hebbsche Lernregel . . . 16
2.4 Synaptische Plastizität . . . 17

3 Hebbsches Lernen und Erweiterungen **21**
3.1 Einleitung . . . 21
3.2 Lernregel mit Synapsengedächtnis . . . 22
3.2.1 Einleitung . . . 22
3.2.2 Beschreibung des Modells . . . 22
3.2.3 Arbeitsweise des Modells . . . 24
3.2.4 Ergebnisse . . . 26
3.2.5 Biologische Plausibilität . . . 29
3.3 Lernregel mit Neuronengedächtnis . . . 30
3.3.1 Einleitung . . . 30
3.3.2 Umkodierung . . . 31
3.3.3 Extremale Optimierung . . . 33
3.3.4 Modell mit Neuronengedächtnis . . . 35
3.3.5 Arbeitsweise des Modells . . . 36
3.3.6 Ergebnisse . . . 38
3.3.6.1 winner-take-all Mechanismus . . . 38
3.3.6.2 softmax Mechanismus . . . 44
3.3.6.3 verrauschter winner-take-all Mechanismus . . . 50
3.3.7 Biologische Plausibilität . . . 59
3.3.8 Diskussion . . . 62

3.4 Lernregel mit Neuronengedächtnis 2 63
3.4.1 Einleitung 63
3.4.2 Ergebnisse 63
3.4.3 Diskussion 66
3.5 Zusammenfassung 66

4 Lernen in neuronalen Systemen fern vom thermischen Gleichgewicht 69
4.1 Einleitung 69
4.2 Vergleich softmax vs. verrauschter winner-take-all Mechanismus 69
4.2.1 Analytische Lösung für die Verteilung des verrauschten winner-take-all Mechanismus 70
4.2.2 Invarianzeigenschaften beider Modelle 73
4.2.3 Ergebnisse 74
4.3 Temperaturverlauf in mehrschichtigen neuronalen Netzen . . . 79
4.3.1 Verrauschter winner-take-all Mechanismus 79
4.3.2 Ergebnisse 80
4.3.2.1 Lernregel mit Synapsengedächtnis 80
4.3.2.2 Lernregel mit Neuronengedächtnis 83
4.4 Zusammenfassung 86

5 Komplexe Netzwerke 89
5.1 Einleitung 89
5.2 Grundlagen 90
5.2.1 Konstruktion von *small world* Netzwerken 90
5.2.2 Eigenschaften von *small world* Netzwerken 92
5.2.2.1 Clusterungskoeffizient C 93
5.2.2.2 Charakteristische Pfadlänge L 93
5.2.2.3 Zusammenfassung der Eigenschaften 95
5.3 Lerndynamiken in *small world* Netzwerken 95
5.3.1 Lernen von Timing 95
5.3.2 Einleitung 95
5.3.3 Beschreibung des zu lernenden Problems 96
5.3.4 Ergebnisse 99
5.3.4.1 Aktions-Selektions-Mechanismus I. 104
5.3.4.2 Aktions-Selektions-Mechanismus II. 107
5.3.4.3 Diskussion 110
5.3.4.4 Vergleich ASM I. vs. ASM II. 111
5.3.5 Zusammenfassung 115

6 Aktive Computation in offenen Systemen 117
6.1 Einleitung 117
6.2 Überblick experimenteller und theoretischer Arbeiten 118

6.3 Anschauliche Darstellung der Wahrnehmungs-Handlungs-Schleife . . . 123
6.3.1 Prinzipielle Arbeitsweise . . . 124
6.4 Formale Beschreibung der Wahrnehmungs-Handlungs-Schleife 127
6.4.1 Umgebung . . . 127
6.4.2 Dynamik innerer Zustände 1. . . . 128
6.4.3 Interner Evaluierungsmechanismus . . . 130
6.4.4 Dynamik innerer Zustände 2. . . . 131
6.4.5 Belohnungssystem vs. Aktions-Selektions-Mechanismus . . . 132
6.4.6 Evaluierung des Agenten durch einen äußeren Beobachter 137
6.5 Ergebnisse . . . 142
6.6 Diskussion . . . 154
6.6.1 Biologische Plausibilität . . . 156

7 Zusammenfassung **159**

A Mathematischer Anhang **167**
A.1 Pseudoinverse . . . 167
A.1.1 Definition der Pseudoinversen . . . 167
A.1.2 Berechnung der Pseudoinversen . . . 167
A.2 Dijkstra Algorithmus . . . 168
A.3 Anmerkungen zur Simulation . . . 170

Literaturverzeichnis **171**

Kapitel 1

Einleitung

Die Frage nach den Naturgesetzen, nach welchen neuronale Systeme, also Gehirne, ablaufen, hat bereits eine mehrere tausend Jahre dauernde Historie. Die Anfänge liegen mit Aristoteles (384-322 v.Chr.) und Rene Descartes (1596-1650) im Bereich der Philosophie und haben sich über experimentelle Arbeiten von Ramon y Cayal (1852-1934) und Emil du Bois-Reymond (1818-1896) bis hin zu rein theoretischen Untersuchungen von John von Neumann (1903 -1957) [von Neumann, 1958] und J. J. Hopfield [Hopfield, 1982] entwickelt. Es gibt heutzutage sicherlich kein Gebiet, auf dem so viele Wissenschaftler aus unterschiedlichen Disziplinen forschen, wie im Bereich der Neurowissenschaften.
Trotz all dieser Anstrengungen ist es bisher nicht gelungen experimentelle Grundfragen neuronaler System abschließend zu beantworten, wie z. B., ob eine Nervenzelle stochastisch oder deterministisch feuert, oder welche Art von Kodierung im Gehirn verwendet wird. Dies ist ein Grund dafür, dass es bisher noch nicht gelungen ist einen theoretischen Formalismus für neuronale Systeme aufzustellen, welcher z.B. ähnlich leistungsfähig wie der Hamiltonformalismus in der Beschreibung traditioneller physikalischer Phänomene ist. Zwei weitere Problem liegen in der komplexen Zusammensetzung des Gehirns und der Verknüpfung von Gehirn und Körper, der wiederum in eine Umgebung eingebunden ist. Die komplexe Zusammensetzung tritt auf verschiedenen Skalen der Beschreibung auf, z.B. gibt es im menschlichen Gehirn rund 10^{11} Neuronen, die über Synapsen zu einem Netzwerk verbunden sind. Dieses Gesamtnetzwerk ist wiederum in Unternetzwerke unterteilt, denen unterschiedliche Aufgaben bei der Informationsverarbeitung zukommen. Auf der anderen Seite ist auch schon die Beschreibung einzelner Neuronen [Hodgkin und Huxley, 1952] und Synapsen aufwendig, ohne dass man ihren Beitrag zu einer Gesamtfunktion des Gehirns überhaupt berücksichtigt. Aber gerade die Funktionsfähigkeit von Gehirnen hat es der Menschheit ermöglicht sich weiterzuentwickeln. Aus diesem Grund kann man das Gehirn auch nicht verstehen, wenn man es nicht so untersucht, wie es in der Natur vorkommt,

nämlich immer nur zusammen mit einem Körper, der sich in einer Umgebung befindet und mit dieser wechselwirkt. An diesem Punkt sind wir wieder an der Wahrnehmungs-Handlungs-Schleife angelangt, die auch begrifflich besser zum damit meint, wenn man die Arbeitsweise des Gehirns verstehen will.

In dieser Arbeit wird die Wahrnehmungs-Handlungs-Schleife, die insbesondere fester Bestandteil aller kognitiven Systeme ist, anhand einiger spezieller Beispiele, die auf unterschiedlichen Ebenen der Beschreibung liegen und biologisch motiviert sind, untersucht. Desweiteren wurde eine neue Hebb-artige Lernregel für neuronale Netze entwickelt, die nicht nur für sich alleine betrachtet interessante Eigenschaften hat, sondern auch in einen Zusammenhang mit der Wahrnehmungs-Handlungs-Schleife gebracht werden kann, an dem das Zusammenspiel unterschiedlicher Komponenten auf unterschiedlichen Ebenen signifikant zum Ausdruck kommt.
Die Einschränkung auf spezielle Beispiele ist notwendig, weil es derzeit keinen allgemeinen mathematischen Ansatz für dieses Problem gibt. Desweiteren sind alle Beispiele so einfach wie möglich gewählt, da aus theoretischer Sicht der einfachste Fall, welcher gerade noch alle interessierenden Eigenschaften des Systems beschreibt, vollkommen ausreicht. Eine detailgetreuere Darstellung mag vielleicht aus Gründen der Anschauung nützlich sein, für das theoretische Verständnis liefert sie jedoch keine Beitrag.
An dieser Stelle sei jedoch explizit darauf hingewiesen, dass die konkrete Auswahl der Beispiele und deren Bestandteile, mitentscheidend von den zur Zeit zur Verfügung stehenden Computern beeinflusst wurde, da die Simulationszeiten komplexer Systeme leicht ins unermessliche anwächst.

Die vorliegende Arbeit ist wie folgt organisiert: In Kapitel 2. wird ein Überblick verschiedener Formen des Lernens gegeben. Dies betrifft sowohl theoretische als auch experimentelle Arbeiten.

In Kapitel 3. wird eine neue Hebb-artige Lernregel anhand eines mehrschichtigen neuronalen Netzes vorgestellt. Diese Lernregel basiert auf Arbeiten von [Alstrom und Stassinopoulos, 1995; Strassinopoulos und Bak, 1995; Chialvo und Bak, 1999; Klemm et al., 2000; Bak und Chialvo, 2001] und [Boettcher und Percus, 2001] und stellt zum Teil eine Weiter-, zum Teil eine Neuentwicklung dar.
Diese Lernregel verwendet ein Reinforcementsignal, das direkt auf jede Synapse eines neuronalen Netzes einwirkt und die synaptischen Änderungen durch einen lokalen, regelbasierten Mechanismus mitbestimmt, ohne vorherige Berechnung einer globalen Kostenfunktion [Alstrom und Stassinopoulos, 1995; Strassinopoulos und Bak, 1995]. Die synaptischen Änderungen, welche durch die Lernregel bewirkt werden, wirken sich nur in einer Erniedrigung der synaptischen Gewichte aus und entsprechen somit aus biologischer Sicht

einer Form von *long-term depression* (LTD) [Strassinopoulos und Bak, 1995; Chialvo und Bak, 1999; Klemm et al., 2000; Bak und Chialvo, 2001].
Desweiteren verfügt diese Lernregel über ein Gedächtnis, ähnlich wie in [Klemm et al., 2000]. Jedoch bezieht sich dieses Gedächtnis nicht auf die Synapsen [Klemm et al., 2000] sondern auf die Neuronen, wodurch sich eine Reduktion der verwendeten Parametern ergibt, die biologisch als Ressourcenersparnis gedeutet werden kann. Um den Übergang von den Synapsen- auf die Neuronenzähler vollziehen zu können, ist das Prinzip der Extremalen Optimierung (EO) [Boettcher und Percus, 2001] von zentraler Bedeutung, denn hierdurch wird die Möglichkeit eröffnet das Synapsengedächtnis stochastisch durch das Neuronengedächtnis zu approximieren, wodurch der Grundgedanke der Hebb-artigen Lernregel mit Synapsengedächtnis [Klemm et al., 2000] erhalten bleibt. Entscheidenden Einfluss auf den Erfolg dieser stochastischen Approximation hat die Verwendung von algebraischen Verteilungen (*power laws*) auf den die Extremale Optimierung (EO) basiert, da hierdurch große Fluktuationen, wie sie inhärent in Systemen fern vom thermischen Gleichgewicht vorkommen, ausgenutzt werden. Dies wird durch einen expliziten Vergleich mit Exponentialverteilungen demonstriert.
Zusammenfassend ergibt sich hieraus eine regelbasierte, stochastische Lernregel, die sich Hebb-artig interpretieren lässt. Die Eigenschaften dieser Hebb-artigen Lernregel mit Neuronengedächtnis werden in Abhängigkeit verschiedener Parameter, wie der Gedächtnislänge der Neuronen, den Exponenten der algebraischen Verteilungen, dem Rauschen und der verwendeten Netzwerkdynamik in einem mehrschichtigen neuronalen Netze untersucht, mit dem das XOR-Problem gelernt wird. Zusammenhänge und Unterschiede zu bisherigen Ansätzen werden aufgezeigt und diskutiert. Auf experimentelle Arbeiten über synaptische Plastizität wird ausführlich eingegangen und Bezüge werden hergestellt.

In Kapitel 4. werden zwei unterschiedliche Arten von Netzwerkdynamiken in einem mehrschichtigen neuronalen Netz untersucht. Es wird analytisch gezeigt, dass ein softmax Mechanismus nicht identisch zu einem verrauschten winner-take-all Mechanismus ist, bei dem der Rauschterm aus einer Gleichverteilung gezogen wird. Trotz des Unterschieds lässt sich dem verrauschten winner-take-all Mechanismus approximativ eine Temperatur zuordnen, die eine Funktion des Rauschens und der inneren Felder der Neuronen ist.
Die im ersten Teil des Kapitels entwickelte Methode zur approximativen Zuordnung einer Temperatur in einem mehrschichtigen neuronalen Netz mit einer Netzwerkdynamik, welche durch einen verrauschten winner-take-all Mechanismus gegeben ist, wird dazu verwendet, den Temperaturverlauf während eines Lernprozesses zu untersuchen. Hierbei lernt ein mehrschichtiges neuronales Netze das XOR-Problem. Die verwendete Lernregel wird

als zu untersuchender Parameter aufgefasst, so dass die Lerndynamik des neuronalen Netzes einmal durch die Lernregel mit Synapsengedächtnis [Klemm et al., 2000] und einmal durch die Lernregel mit Neuronengedächtnis bestimmt wird.
Abschließend wird eine mögliche Anwendung und Bezüge zur Neurophysiologie und kognitiven Psychologie diskutiert.

In Kapitel 5. wird der generische Charakter der in Kapitel 3. eingeführten Hebb-artigen Lernregel mit Neuronengedächtnis dadurch unterstrichen, dass als zugrundeliegende Netztopologie eine Klasse von Netzwerken gewählt wird, die als *small world* [Watts und Strogatz, 1998] bezeichnet wird. Innerhalb dieser Netzwerktopologie wird untersucht, wie das Problem des Timings, also die sinnvolle Verknüpfung von Input- zu Outputneuronen nach exakt einer vorgegebenen Zeit, gelernt werden kann. Hierfür wird die Lernregel mit Neuronengedächtnis mit der von [Chialvo und Bak, 1999; Bak und Chialvo, 2001] verwendeten Lernregel in Abhängigkeit verschiedener Parameter, wie der Anzahl M der zu lernenden Muster und dem bei *small world* Netzwerken auftretenden Umverknüpfungsparameter p_{rw}, der vereinfacht ausgedrückt ein Maß für die Abweichung von einem regulären Netzwerk ist, untersucht und verglichen. Insbesondere die Einführung von zwei unterschiedlichen Aktions-Selektions-Mechanismen, welche die Muster auswählen, die dem neuronalen Netzwerk im nächsten Zeitschritt als Input präsentiert werden, offenbart einen signifikanten Unterschied zwischen beiden Lernregeln, der nur durch die Ausweitung der Untersuchung auf die Verhaltensebene enthüllt werden konnte. Dadurch wird ein Bezug zur Wahrnehmungs-Handlungs-Schleife aufgedeckt und eine Verknüpfung zwischen Netzwerk- und Verhaltensebene hergestellt.
Da sich biologisch motivierte Untersuchungen von Lerndynamiken neuronaler Systeme durch zusätzliche Nebenbedingungen von Ansätzen anderer Gebiete, wie z.B. dem *machine learning* unterscheiden, die insbesondere die dem neuronalen System zur Verfügung stehende Information betrifft, ist es nicht verwunderlich, dass es unterschiedliche Fehlermaße innerhalb des Gesamtsystems gibt. Aus diesem Grund wird in diesem Kapitel ein Fehlermaß in Form des individuellen Absolutfehlers E_{iabs} eingeführt der die Güte des neuronalen Netzes vom Standpunkt eines äußeren Beobachters, der über die gesamte Information des Systems verfügt, beschreibt. Für dieses Fehlermaß wird der zugrundeliegende stochastische Prozess angegeben und eine Fehlerschwelle festgelegt, deren Unterschreitung die *first-passage* Zeit T_{FPT} und somit im Ensemble deren Verteilung liefert, durch welche sich die Lerndynamik des neuronalen Netzes global charakterisieren lässt. Mathematisch ausgedrückt ergibt sich der individuellen Absolutfehlers E_{iabs} zu jedem Zeitpunkt durch Integration über alle zu lernenden Muster.
Die Einführung des individuellen Absolutfehlers E_{iabs} liegt insbesondere

durch den als Parameter des Gesamtsystems angesehenen Aktions-Selektions-Mechanismus nahe, da Fehlermaße, die nicht die globale Güte des neuronalen Netzes charakterisieren und somit nicht zu jedem Zeitpunkt über alle zu lernenden Muster integrieren, nur eine unzureichende Beschreibung liefern, die in Abhängigkeit des verwendeten Aktions-Selektions-Mechanismus beliebig falsch sein kann.

In Kapitel 6. wird als spezielles Beispiel der Wahrnehmungs-Handlungs-Schleife die Selbstlokalisation eines Agenten in einer Umgebung behandelt. Die Bezeichnung Agent bringt zum Ausdruck, dass das neuronale Netz über Sensorik und Motorik verfügt und somit in einer sehr einfachen Form über einen Körper. Der Agent ist, im Gegensatz zu den bisherigen Untersuchungen, in eine räumliche Umgebung eingebettet (*embodiment*) mit der er wechselwirkt und von der er durch raum-zeitlichen Integration von sensorischem Input und motorischem Output eine interne Repräsentation bilden soll. Metaphorisch lässt sich der Agent in seiner Umgebung als Ratte interpretieren, die ihre Umgebung erkundet.
Zu Beginn des Kapitels wird eine Übersicht experimenteller und theoretischer Arbeiten gegeben, die sich mit der Lokalisation von Ratten in einer Umgebung und deren zugrundeliegenden neuronalen Aktivität im Hippocampus beschäftigen. Dies ist deshalb von Bedeutung, da im Hippocampus von Ratten sogenannte *place cells* gefunden wurden [O'Keefe und Dostrovsky, 1971; O'Keefe, 1976], die immer dann verstärkt feuern, wenn sich die Ratte an einem bestimmten Ort, *place field* genannt, befindet. Da verschiedene *place cells* auch an verschiedenen *place fields* feuern nimmt man an, dass sie zusammengenommen die momentane Position der Ratte kodieren [Wilson und McNaughton, 1993; Zhang et al., 1998].
Nach einer Übersicht weiterer Eigenschaften der *place cells* im Hippocampus [O'Keefe und Nadel, 1978; Taube et al., 1990; Eichenbau, 1996; Wood et al., 1999] wird dessen Neuroanatomie vorgestellt [Cohen und Eichenbaum, 1993] und auf die mögliche funktionelle Verwendung als deklaratives Gedächtnis hingewiesen [Eichenbaum, 1997]. Hierbei versteht man unter deklarativem Gedächtnis all das Wissen, das bewusst verfügbar ist und geäußert werden kann [Squire, 1987].
Danach wird das konkrete mathematische Modell vorgestellt, dass auf einer Arbeit von [Herrmann et al., 1999b] basiert und ein Zusammenhang zu allgemeinen komplexen adaptiven Systemen [Schuster, 2002] hergestellt. Dieses Modell wird so angepasst, dass es dazu verwendet werden kann, einen Agenten auf einem eindimensionalen Gitter mit periodischen Randbedingungen zu beschreiben. Zusätzlich werden noch drei unterschiedliche Aktions-Selektions-Mechanismen definiert, deren Einfluss auf die Lerndynamik von besonderem Interesse ist. Besonders hervorzuheben ist hier der sogenannte Active Walk, der eine adaptive Struktur und somit ein Gedächtnis

besitzt, aufgrund dessen der Agent seine Aktionen auswählt, wodurch die Aktionen an interne Zustände des Agenten gekoppelt werden, was durch den Begriff "Active" zum Ausdruck gebracht wird.
In Analogie zu Kapitel 5. wird auch hier ein Fehlermaß für einen äußeren Beobachter definiert, das der besonderen Problemstellung angepasst ist. Denn die Integration über alle Inputmuster, welche in diesem Fall die beobachtbaren Symbole auf den Gitterzellen sind, kann aufgrund der Topologie des Raumes und somit aufgrund der darin vorkommenden Anordnung der Beobachtungssymbole nicht mehr in einer beliebigen Reihenfolge vorgenommen werden. Aus allgemeinen theoretischen Überlegungen zu solch einem Fehlermaß resultiert der approximierte individuelle Gesamtfehler $\tilde{E}_{iabs}$ des Agenten, der sich praktisch auswerten lässt. Hieraus erhält man für eine Fehlerschwelle wieder die *first-passage* Zeit T_{FPT} und somit im Ensemble deren Verteilung, wodurch sich die Konvergenzeigenschaften des Agenten bewerten lassen.
Im Ergebnisteil wird der Einfluss des verwendeten Aktions-Selektions-Mechanismus, der Gittergröße und des Beobachtungs- und Positionsfehlers auf das Konvergenzverhalten des Agenten bzgl. der Bildung einer internen Repräsentation untersucht.
Zum Abschluss erfolgt eine ausführliche Diskussion der erhaltenen Ergebnisse sowie eine biologisch plausible Interpretation für Teile des vorgestellten mathematischen Modells.

Die Arbeit endet in Kapitel 7. mit einer ausführlichen Zusammenfassung und Diskussion aller Ergebnisse sowie einem Ausblick.

Kapitel 2

Grundlagen: Lernen in neuronalen Systemen

2.1 Einleitung

In diesem Kapitel sollen theoretische und experimentelle Grundlagen des Lernens in neuronalen Systemen vorgestellt werden. Da dies nicht unabhängig vom verwendeten Neuronenmodell vorgenommen werden kann, soll zuvor hierauf kurz eingegangen werden.

2.2 Biologische und Modellneuronen

Seit den Arbeiten von [Cajal, 1904] gilt das Neuron als Grundbaustein des Gehirns. Diese Neuronen, auch Nervenzellen genannt, empfangen elektrische Signale und generieren nach einem bestimmten Mechanismus neue Signale. Die "Leitungen", über die das Neuron die Signale erhält, bezeichnet man als *Dendriten*. Diese Signale werden dann zum Zellkörper, dem sogenannten *Soma*, weitergeleitet. Dort findet an der Zellmembran durch chemische Prozesse eine Verarbeitung des Signals, in Form eines sich ändernden Membranpotentials, statt. Übersteigt das Membranpotential einen bestimmten Wert, so entspringt am Axonhügel, einer speziellen Region des Somas, das Axon, welches ein Aktionspotential zu nachgeschalteten Neuronen weiterleitet. Hierbei ist das Axon über Synapsen mit den Dentriten verbunden. Abbildung 2.1 zeigt eine schematische Darstellung dieser Verhältnisse.

Um diese hoch komplizierten elektro-chemischen Prozesse simplifiziert und abstrahiert beschreiben zu können, wurde eine Reihe von Neuronenmodellen entwickelt. Das Modell, das die wichtigsten biologischen Eigenschaften von Neuronen, also die Erzeugung von Aktionspotentialen in Abhängigkeit von Inputströmen, am besten wiedergibt, ist das Modellneuron von [Hodgkin

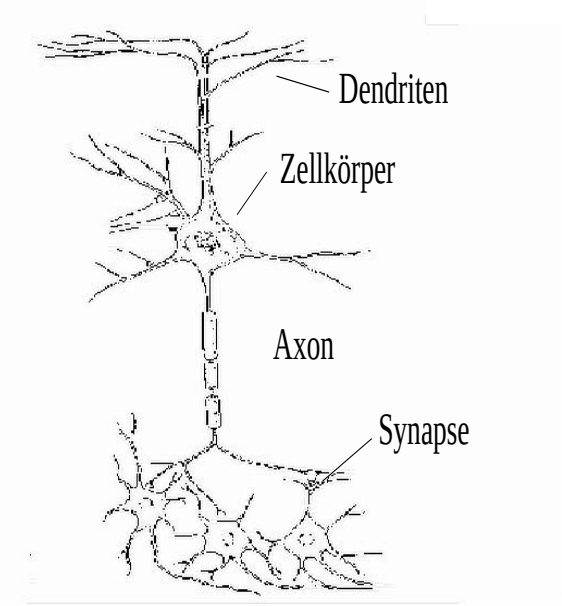


Abbildung 2.1: Darstellung eines biologischen Neurons (aus [Kandel et al., 1991] entnommen).

und Huxley, 1952]. Sie konnten zeigen, dass die Zellmembran durch ein elektrisches Ersatzschaltbild beschrieben werden kann, bei dem die Zellmembran selbst als Kondensator und die Ionenkanäle als Reihenschaltung von Spannungsquelle und Widerstand, für jede Ionenart, interpretiert werden kann. Hierfür wurden sie 1963 mit dem Nobelpreis in Medizin ausgezeichnet. Von [Abbott und Kepler, 1990] wurde gezeigt, dass sich das gekoppelte Differentialgleichungssystem des Hodgkin-Huxley Neurons systematisch reduzieren lässt, wodurch man integrate and fire [Stein, 1967], FitzHugh-Nagumo [FitzHugh, 1961; Nagumo et al., 1962] und sogar eine Form von McCulloch-Pitts [McCulloch und Pitts, 1943] Modellneuronen erhält, die zuvor schon auf anderen Wegen eingeführt und verwendet wurden.

Da in dieser Arbeit nur das McCulloch-Pitts Neuron verwendet wird, beschränke ich meine Ausführungen hierauf. Die Arbeitsweise eines solchen McCulloch-Pitts Neurons besteht darin, dass es die ankommenden Signale aufsummiert und genau dann aktiviert wird, wenn diese Summe einen

bestimmten Schwellwert μ überschreitet.

$$x(t+1) = H(\sum_{i}^{I} w_i(t)x_i(t) - \mu) \tag{2.1}$$

Hierbei ist H die Heaviside Funktion.

$$H(x) = \begin{cases} 1, if & x \geq 0 \\ 0, if & x < 0. \end{cases} \tag{2.2}$$

Allgemein bezeichnet man die Funktion, die auf $\sum_i^I w_i(t)x_i(t) - \mu$ angewandt wird, als Aktivierungsfunktion, da hierdurch die Aktivität des Neurons festgelegt wird.
Das Modellneuron kann also nur die beiden Werte $x \in \{0, 1\}$ annehmen, welche dem Feuern bzw. Nicht-Feuern entsprechen. Entsprechend erhält es auch von den anderen I Neuronen nur die Signale $x_i \in \{0, 1\}$. Da diese jedoch über die Synapsen zu unserem Neuron laufen, wichtet man die Signale symbolisch mit $w_i(t) \in \mathbb{R}$. Welche Prozesse hierbei tatsächlich ablaufen, wird am Ende dieses Kapitels kurz dargestellt.

In Gleichung 2.1 wurde für die synaptischen Gewichte $w_i(t)$ explizit eine Zeitabhängigkeit angegeben. Eine Regel, nach welcher die zeitliche Entwicklung der synaptischen Gewichte $w_i(t)$ erfolgt, bezeichnet man als Lernregel bzw. als Lernregel neuronaler Netze. Im folgenden Abschnitt werden unterschiedliche Formen solcher Lernregeln vorgestellt, die jeweils auf unterschiedlichen Voraussetzungen über die zur Verfügung stehenden Informationen beruhen.

2.3 Einteilung von Lernregeln

Die Einteilung von Lernregeln in unterschiedliche Kategorien kann über die jeweils zur Verfügung stehende Information vorgenommen werden. Da dies nicht immer ganz eindeutig ist, halte ich mich an die Einteilung in [Hertz et al., 1991].
Man unterscheidet zwischen zwei Hauptkategorien des Lernens; überwachtes (*supervised*) und unüberwachtes (*unsupervised*) Lernen.

2.3.1 *supervised learning*

Beim überwachten Lernen ist bekannt, welche Ausgabe des neuronalen Netzes bei einer bestimmten Eingabe erfolgen soll. Hierdurch lassen sich gewünschter und tatsächlicher Output nicht nur qualitativ, sondern auch quantitativ vergleichen. Die hierdurch gewonnene Information kann dann dazu verwendet

werden, die synaptischen Gewichte zu adaptieren, wie dies z.B. beim Backpropagation Algorithmus geschieht.
Eine schwächere Form des überwachten Lernens ist das Reinforcement Lernen. Hier steht nur die Information, ob die Ausgabe des neuronalen Netzes richtig oder falsch war, in Form eines ins neuronale Netz zurückgekoppelten, skalaren Reinforcementsignals r, zur Verfügung. Beide Verfahren sollen im folgenden kurz vorgestellt werden.

2.3.1.1 Backpropagation Algorithmus

Der Backpropagation Algorithmus wurde von [Rumelhart et al., 1986a,b] und [Werbos, 1974] unabhängig voneinander entwickelt[1]. Er eröffnet die Möglichkeit für vorwärtsgerichtete neuronale Netze mit einer beliebigen Anzahl von Schichten die synaptischen Gewichte so zu bestimmen, dass eine vorgegebene Abbildung richtig gelernt werden kann. Historisch gesehen war dies ein enormer Fortschritt, denn bis Mitte der 80'er Jahre des letzten Jahrtausends war nur die Perceptron[2] Lernregel von [Rosenblatt, 1962] bekannt. Mit ihr konnte man jedoch nur einschichtige, vorwärtsgerichtete neuronale Netze trainieren, was insbesondere zu wenig war, um nichtlinear seperable Probleme zu lernen, die mindestens eine verborgene Schicht benötigen [Minsky und Papert, 1969]. Besonders hervorzuheben ist, dass mit Hilfe des Backpropagation Algorithmus ein systematisches Verfahren angegeben werden konnte, welches das XOR-Problem lösen kann [Rumelhart et al., 1986b].

Ich stelle den Backpropagation Algorithmus anhand eines zweischichtigen neuronalen Netzes vor, wie es in Abbildung 2.2 dargestellt ist. Die synaptischen Gewichte sollen nun so eingestellt werden, dass das neuronale Netz den Inputmustern $\mathbf{x}^{\mathbf{I},m}$ die richtigen Outputmuster $\mathbf{y}^{\mathbf{m}}$ für alle $m \in M$ zuordnen kann.
Die Aktivität der zweiten (hidden) und dritten (output) Schicht berechnen sich für das m'te Muster durch

$$h_j^{H,m} = \sum_i^I w_{ji}^H x_i^{I,m} \tag{2.3}$$

$$x_j^{H,m} = g(h_j^{H,m}) \tag{2.4}$$

$$h_k^{O,m} = \sum_j^H w_{kj}^O x_j^{H,m} \tag{2.5}$$

[1] Für eine ausführliche Auflistung der Referenzen siehe [Hertz et al., 1991].
[2] Bei einem Perceptron ist als Aktivierungsfunktion in Gleichung 2.1 das Signum zu verwenden.

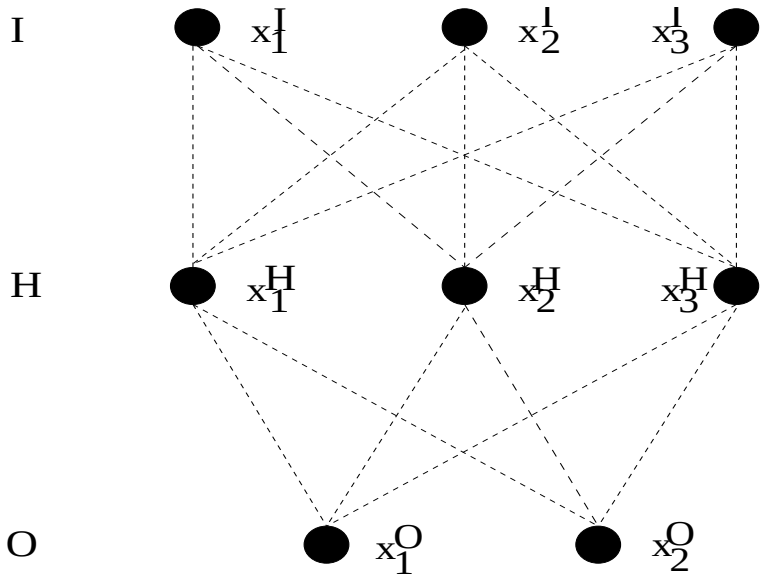


Abbildung 2.2: Darstellung eines zweischichtigen vorwärtsgerichteten neuronalen Netzes.

$$x_k^{O,m} = g(h_k^{O,m}) \tag{2.6}$$

Nun kann man die Ausgabe des Netzes $\boldsymbol{x}^{O,m}$ mit y^m vergleichen. Als Fehlermaß wird der quadratische Fehler

$$E(w^H, w^O) = \frac{1}{2}\sum_{k,m}(y_k^m - x_k^{O,m})^2 \tag{2.7}$$

verwendet. Wenn die verwendete Aktivierungsfunktion g differenzierbar ist, so ist es auch der quadratische Fehler 2.7 und man kann die Gradienten nach den beiden synaptischen Gewichten $\boldsymbol{w}^H$ und $\boldsymbol{w}^O$ bestimmen. Der Backpropagation Algorithmus sucht das Minimum dieses quadratischen Fehlers durch Gradientenabstieg. Nach kurzer Rechnung erhält man

$$\begin{aligned} \triangle w_{k,j}^O &= -\epsilon\frac{\partial E}{\partial w_{k,j}^O} = \epsilon\sum_m (y_k^m - x_k^{O,m})g'(h_k^{O,m})x_j^{H,m} && (2.8)\\ &= \epsilon\sum_m \delta_k^m x_j^{H,m} && (2.9)\end{aligned}$$

und

$$\begin{aligned} \triangle w_{j,i}^H &= -\epsilon\frac{\partial E}{\partial w_{j,i}^H} = -\epsilon\sum_m \frac{\partial E}{\partial x_j^{H,m}}\frac{\partial x_j^{H,m}}{\partial w_{j,i}^H} && (2.10)\\ &= \epsilon\sum_{m,k}(y_k^m - x_k^{O,m})g'(h_k^{O,m})w_{k,j}^O g'(h_j^{H,m})x_i^{I,m} && (2.11)\end{aligned}$$

$$= \epsilon \sum_{m,k} \delta_k^m w_{k,j}^O g'(h_j^{H,m}) x_i^{I,m} \tag{2.12}$$

$$= \epsilon \sum_{m} \delta_j^m x_i^{I,m} \tag{2.13}$$

Die Lernrate ϵ ist positiv und hinreichend klein zu wählen, da der Algorithmus ansonsten nicht konvergieren kann.
Aus Gleichung 2.9 und 2.13 kann man eine allgemeine Regelmäßigkeit ablesen, nämlich

$$\triangle w_{i,j} = \epsilon \sum_{m}^{M} \delta_{\text{output des Layers}} * x_{\text{input des Layers}} \tag{2.14}$$

Aus Gleichung 2.13 erhält man auch die allgemeine Form der δ's, die sich für mehrschichtige Netzwerke ergeben.

$$\delta_j^m = g'(h_j^{H,m}) \sum_k \delta_k^m w_{k,j}^O \tag{2.15}$$

Hierher erhält der Algorithmus auch seinen vollen Namen, *back propagation of errors*, denn der Fehler, also die δ's, werden durch die synaptischen Gewichte propagiert. Jedoch in umgekehrter Richtung wie das Signal, was man unschwer an der Stellung der Indizes in Gleichung 2.15 ablesen kann.

Wenn man sich die Gleichungen 2.9 und 2.13 genau ansieht, erkennt man, dass bei jeder synaptischen Änderung über alle Muster M summiert werden muss. Solch eine Art von Lernen bezeichnet man als *batch* oder *off-line* Lernen. Dies mag für technische Anwendungen kein Problem sein, jedoch ist es aus biologischer Sicht unplausibel, alle vorhandenen Muster auf einmal zu verwenden. Biologische Systeme erhalten sequentiell im Verlauf der Zeit Inputmuster, welche verarbeitet werden müssen, und somit zur Veränderung der synaptischen Gewichte beitragen. Aus diesem Grund gibt es eine *on-line* Variante des oben beschriebenen Algorithmus, bei dem auf die Summation über alle Muster verzichtet wird. Statt dessen zieht man gleichverteilt aus allen Mustern M eines, mit dem man dann die synaptischen Gewichtsänderungen durch die Gleichungen 2.9 und 2.13, jedoch ohne die Summe, bestimmt. Diese Prozedur wird iterativ so lange durchlaufen, bis die Konvergenz der synaptischen Gewichte eingetreten ist.
Diese Argumentation ist zwar soweit richtig, jedoch wird durch das Weglassen der Summe aus dem Backpropagation Algorithmus immer noch kein biologisch plausibler Lernmechanismus. Ungeklärt bleibt immer noch, wie ein biologisches System einen "Sollwert" ($\mathbf{y}^m$) mit einem "Istwert" ($\mathbf{x}^{O,m}$) vergleichen und deren Differenz in Form der δ's rückwärts durch das Netz propagieren soll, wenn es $\mathbf{y}^m$ erst noch lernen muss [Crick, 1989]. Bei einem

Computer, der gewissermaßen als Lehrer des Netzes auftritt, und direkten Zugang zum Ausgabelayer des neuronalen Netzes hat, ist dies kein Problem. Da dieser direkte Zugang bei biologischen Systemen jedoch nicht in der Natur vorkommt (und auch nicht bekannt ist, welche Aktivität dort zu finden sein sollte!) und auch artifiziell nur schwer zu realisieren sein dürfte, stellt der Backpropagation Algorithmus keine Lernregel dar, die biologisch plausibel ist.
Wir sind hier an einem Punkt angekommen, an dem man sich fragen sollte, ob es überhaupt sinnvoll ist, neuronale Netze zu untersuchen, die nicht direkt über einen Körper in eine Umgebung eingebunden sind. Die Diskussion des *embodiment* wird jedoch auf Kapitel 5. und insbesondere Kapitel 6. verschoben, da wir hierdurch zur Wahrnehmungs-Handlungs-Schleife gelangen, die Hauptbestandteil meiner Arbeit ist.

Um mit dem Backpropagation Algorithmus die synaptischen Gewichte für ein vorgegebenes Problem (*on-line* oder *off-line*) zu bestimmen, geht man wie folgt vor:

1. *zufällige Initialisierung aller synaptischen Gewichte und Wahl einer positiven Lernrate* ϵ

2. *a.) ziehe zufällig, gleichverteilt ein Muster* $\mathfrak{m}$
   b.) verwende alle Muster $\mathfrak{M}$

3. *Berechnung der synaptischen Gewichtsänderungen durch 2.9 und 2.13 und somit der neuen synaptischen Gewichte*

4. *gehe zu 2., außer ein Abbruchkriterium ist erfüllt*

2.3.1.2 Reinforcement Lernen

Historisch gesehen liegen die Anfänge des Reinforcement Lernens in der Verhaltenspsychologie [Dayan und Abbott, 2001]. Man unterscheidet hierin zwischen klassischer bzw. Pavlovscher [Pavlov, 1927] und instrumenteller Konditionierung [Thorndike, 1911; Skinner, 1938]. Sie unterscheiden sich darin, dass dem Tier bei der klassischen Konditionierung (auf den Klang einer Glocke folgt Futter) keine Belohnung für ein bestimmtes Verhalten gegeben wird, wohingegen es bei der instrumentellen Konditionierung für eine bestimmte Handlung eine bestimmte Belohnung oder Bestrafung, also ein Feedback, erhält. Bei der klassischen Konditionierung lernt das Tier also, den nächsten Reiz vorher zusagen, wohingegen bei der instrumentellen Konditionierung Handlungen gelernt werden, da die “richtigen” Handlungen zu einer Belohnung führen [Adelle und Alstrom, 2001]. Dies sind auch Grundbestandteile beim Reinforcement Lernen.

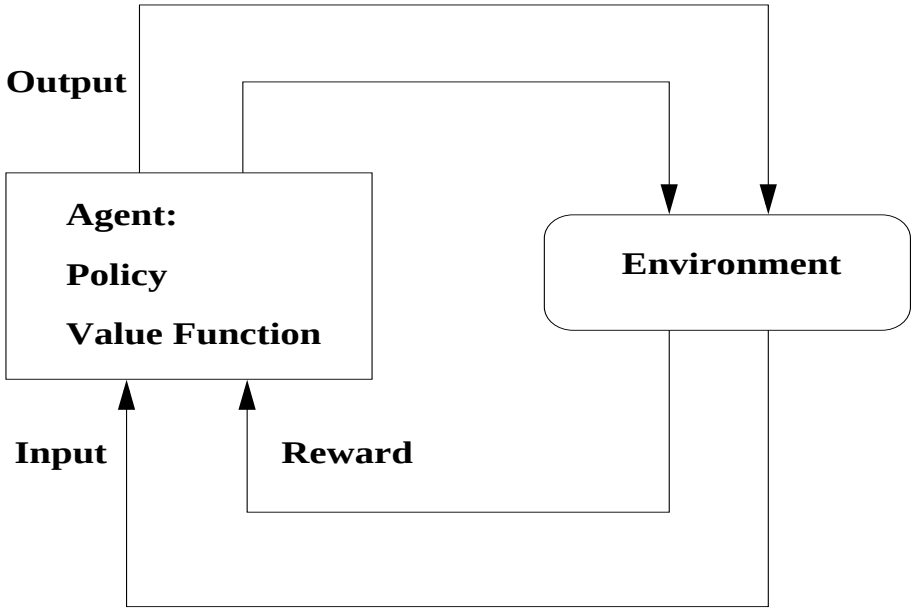


Abbildung 2.3: Darstellung der Problemklasse, mit der sich Reinforcement Lernverfahren befassen.

Im Gegensatz, z.B. zum Backpropagation Algorithmus, stellt Reinforcement Lernen keine mathematische Methode zur Verfügung, um ein gegebenes Problem zu lösen bzw. zu lernen, sondern definiert eine Problemklasse und betrachtet alle mathematischen Methoden, die ein Problem dieser Klasse lernen können, als Methoden des Reinforcement Lernens [Kaebeling et al., 1996; Sutton und Barto, 1998]. Die Problemklasse lässt sich am besten anschaulich an Abbildung 2.3 beschreiben.

Abstrakt kan man sich das Problem so veranschaulichen, dass ein sogenannter Agent, der eine bestimmte Aufgabe lernen soll, mit einer Umgebung in Wechselwirkung steht. Der Agent führt eine Aktion aus und erhält daraufhin von der Umgebung einen neuen Input. Neben diesem Input erhält der Agent direkt oder indirekt (verzögert) ein Reinforcementsignal r. Im Gegensatz zum Backpropagation Algorithmus ist r nur ein qualitativer Feedback, der die Handlungen tendenziell bewertet, da meist auch nicht im vornherein klar ist, welchen Output der Agent denn auf einen bestimmten Input erzeugen soll. Somit kann das Reinforcementsignal r als Belohnung (*Reward*) oder Bestrafung, auf die vom Agenten ausgeführten Handlungen, angesehen werden.

Beim Reinforcement Lernen geht es darum ein bestimmtes Ziel, wie z.B. einen Stab zu balancieren [Barto et al., 1983; Randlov und Alstrom, 2001], Backgammon [Tesauro, 1994] oder Dame [Samuel, 1959] zu spielen, dadurch zu erreichen, dass der Agent die im Laufe der Zeit erhaltenen Reinforcement Signale r, sinnvoll verarbeitet, um dieses Ziel zu erreichen.

Alle konkreten Ansätze, dieses allgemeinen Prinzips des Reinforcement Lernens, verwenden als Grundbestandteil des Agenten eine sogenannte *Policy* [Barto et al., 1983].

1. *Policy* π: Auswahlmechanismus für Aktionen

Durch die *Policy* verfügt der Agent über einen Mechanismus, eine Aktion auswählen zu können, und ist somit in der Lage auf seine Umgebung einzuwirken.

Durch die Verwendung weiterer Komponenten des Agenten, lassen sich die Reinforcement Lernverfahren grob in zwei Klassen einteilen. Die erste Klasse verwendet explizit eine *Value Function*, welche den Erwartungswert, für die zukünftig zu erwartenden *Rewards* darstellt.

2. *Value Function*: $V_t^\pi =< R_t >$

Hierbei bezeichnet R_t die Summe aller zukünftigen Reinforcement Signale. Die *Value Function* bewertet, wie hoch die erwartete Belohnung sein wird, wenn der Agent die zukünftigen Handlungen durch seine derzeit verfügbare *Policy* auswählt.
Diese Klasse unterteilt sich in weitere Subklassen, je nachdem, auf welche Art die *Value Function* approximiert wird. Als einer der wichtigsten Vertreter sei hier *temporal-difference* (TD) Lernen [Sutton, 1988; Schultz et al., 1997] erwähnt.

Die zweite Klasse von Reinforcement Lernverfahren setzt sich explizit als zusätzliche Randbedingung, dass hiermit biologisches Lernen in neuronalen Netzen erklärt werden soll [Alstrom und Stassinopoulos, 1995; Strassinopoulos und Bak, 1995]. Da es in neuronalen Netzen, im Gegensatz zu technisch konstruierten Geräten, keine zentrale Einheit gibt, die den Lernvorgang des neuronalen Netzes überwacht, und in der die *Value Function* näherungsweise berechnet und an die Synapsen weitergeleitet werden kann, werden die Reinforcementsignal r direkt, nach einem regelbasierten Schema, welches sich biologisch plausibel erklären lässt, zur Veränderung der synaptischen Gewichte verwendet. Hierdurch umgeht man das Problem, von einer (globalen) *Value Function* lokale Regeln für die Änderung der synaptischen Gewichte ableiten zu müssen. Um keine Missverständnisse aufkommen zu lassen, soll hier in aller Deutlichkeit gesagt werden, dass der Vorteil nicht in einer Arbeitsersparniss liegt, sondern darin, eine geeignete Approximation für die Ableitung zu finden, denn im allgemeinen wird sich eine naive Ableitung der synaptischen Gewichte aus einer *Value Function* nicht biologisch plausibel erklären lassen.
In diese Klasse von Reinforcement Lernverfahren gehören auch die Arbeiten

von [Chialvo und Bak, 1999; Klemm et al., 2000; Bak und Chialvo, 2001], auf die meine Arbeit in Kapitel 3. bis 5. aufbaut und somit auch meine Arbeit dieser Kapitel.

Zusammenfassend kann man sagen, da Reinforcement Lernen durch eine Problemklasse, wie in Abbildung 2.3 schematisch dargestellt, definiert ist und nicht direkt durch mathematische Methoden, lässt sich nicht allgemein behaupten, Reinforcement Lernen sei biologisch plausibel oder unplausibel. Die Verwendung eines Reinforcementsignals r ist sicherlich biologisch motiviert, jedoch hängt es an den konkreten Bestandteilen des Agenten, um entscheiden zu können, ob es sich aus biologischer Sicht um eine plausible Art des Lernens handelt.

2.3.2 *unsupervised learning*

Beim unüberwachten Lernen steht weder quantitativ noch qualitativ ein Feedbacksignal zur Verfügung, das für den Lernprozess, des zu adaptierenden Systems, verwendet werden könnte. Aus diesem Grund muss es einen intrinsischen Mechanismus geben, der die synaptische Gewichtsänderung, aufgrund einer bestimmten Regel, bewirkt. Der Klassiker solch einer unüberwachten Lernregel, in biologischen neuronalen Netzen, stammt von dem Psychologen Donald Hebb, [Hebb, 1949].

2.3.2.1 Hebbsche Lernregel

Donald Hebb postulierte, dass ein gemeinsames Feuern von prä- x_i und postsynaptischem Neuron x_j zu einer Vertärkung der Synapse führt [Hebb, 1949].

When an axon of cell A is near enough to excite a cell B an repeatedly or persistenty takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficency, as one of the cells firing B, is increased.

Hieraus entstand die mathematische Interpretation,

$$\Delta w_{ij} \sim x_i x_j \tag{2.16}$$

welche als Hebbsche Lernregel bezeichnet wird.

Das Interessante an dieser Formulierung ist, dass die Hebbsche Lernregel ausschließlich mit *lokaler* Information arbeitet, denn nur die Neuronen, welche die Synapse eingrenzen, tragen zur Änderung der Synapsenstärke bei. In zahlreichen theoretischen Untersuchungen wurden die Stärken und

Schwächen dieser einfach aussehenden Lernregel untersucht.

Vielleicht das interessanteste Resultat, das die Anwendbarkeit dieses Postulats demonstriert, stammt von [Hopfield, 1982], der zeigte, dass man durch Uminterpretation eines Isingspinmodells, ein Model für ein neuronales Netze erhält, welches mit der Hebbschen Lernregel zusammen einen Assoziativspeicher ergibt[3]. Als direktes Gegenbeispiel kann man im selben Kontext die Korrelationskatastrophe anführen, die sich bei Hopfieldnetzen unter Verwendung der Hebbschen Lernregel ergibt, wenn man zu viele Muster speichern will [van Hemmen, 1997].
Einen Ausweg aus der Korrelationskatastrophe, basierend auf experimentellen Resultaten von [Markram et al., 1997], der heraus fand, dass die Zeitpunkte und Zeitfenster, zwischen prä- und postsynaptischer Aktivierung, einen Einfluss auf das Vorzeichen der synaptischen Änderung haben, besteht in der Einführung einer asymmetrischen Hebbschen Lernregel, [van Hemmen, 1997].

Ein weiteres Problem, das mit der klassischen Hebbschen Lernregel 2.16 nicht gelöst werden kann, ist das XOR-Problem. [Klemm et al., 2000] schlugen einen Ausweg aus diesem Dilemma vor, indem sie eine Hebb-artige Lernregel mit Synapsenzähler mit einem globalen Reinforcementsignal kombinierten. Da meine Arbeit auf dieser aufbaut, wird diese ausführlich in Kapitel 3. diskutiert.

Nun stellt sich die Frage, wieviele Hebbsche bzw. Hebb-artige Lernregeln es denn gibt? Um dies beantworten zu können, muss man sich einen Überblick, über die in der biologischen Synapse ablaufenden Prozesse, verschaffen.

2.4 Synaptische Plastizität

Wenn man vom Lernen in neuronalen Systemen spricht, dann geht man meist implizit davon aus, dass dies durch eine Änderung der Synapsenstärke, auch synaptische Plastizität gennant, geschieht. Dies ist jedoch nur ein möglicher Mechanismus von vielen, durch den ein Gehirn sich an die Einflüsse einer äußeren Umgebung adaptieren kann. Weitere Mechanismen bestehen z.B. darin, dass ganz neue Synapsen entstehen oder wegfallen können. Im folgenden beschränke ich mich auf die synaptische Plastizität, da davon ausgegangen wird, dass sie eine dominierende Rolle einnimmt, [Koch, 1999].

[3] Per Bak sieht das Hopfield Modell ... *as an example of physicists leading a whole field astray*, [Bak, 1998]

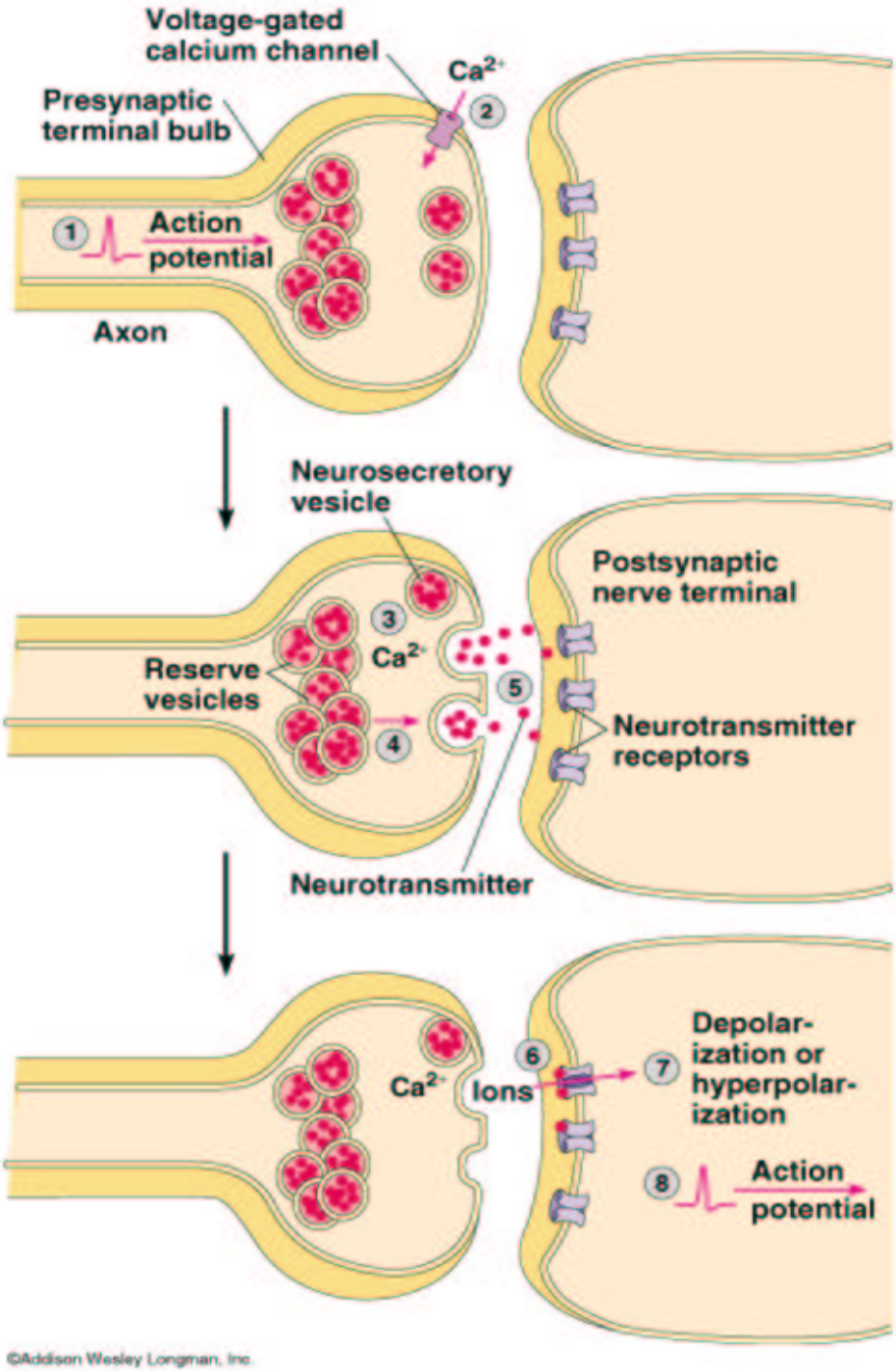


Abbildung 2.4: Schematische Darstellung einer chemischen Synapse und deren Funktion bei der Signalübertragung, aus [Becker et al., 2000].

In der Neurobiologie werden zwei grundlegend verschiedene Arten von Synapsen unterschieden. Die einen bezeichnet man als elektrische (*gab junctions*) und die anderen als chemische Synapsen. Elektrische Synapsen leiten direkt das Aktionspotential vom prä- zum postsynaptischen Neuron. Bei den chemischen Synapsen ist der direkte Übertragungsweg durch den synaptischen Spalt unterbrochen. Wenn das Aktionspotential auf der präsynaptischen Seite ankommt, dann bringt es die Vesikel dazu, mit der Membran zu verschmelzen. Dadurch wird eine Neurotransmittersubstanz, die sich in den Vesikeln befindet, freigesetzt und diffundiert durch den synaptischen Spalt auf die postsynaptische Seite. Dort binden sie sich an Rezptoren, wodurch sich die Ionenkanäle in der Membran öffnen. Dadurch strömen Ionen aus der Umgebung ins Zellinnere und wandeln somit das chemische Signal der Neurotransmitter wieder zurück in ein elektrisches [Katz, 1969]. Abbildung 2.4 veranschaulicht dies durch eine schematischen Darstellung.
Da die Anzahl der chemischen Synapsen im Kortex bei weitem die der elektrischen übersteigt [Koch, 1999], werden im folgenden nur die Eigenschaften der chemischen Synapsen behandelt.
Bevor nun einige experimentelle Resultate über die synaptische Plastizität chemischer Synapsen angeführt werden, muss noch geklärt werden, wie auf biophysikalischer Ebene überhaupt ein synaptisches Gewicht definiert ist. Hierdurch werden dann die folgenden Resultate leichter verständlich.

Die Hauptbestandteile, welche das synaptische Gewicht bestimmen [Koch, 1999], sind die Anzahl der gequantelten *release sites* n, die Wahrscheinlichkeit für die Freisetzung von Neurostransmittern p, und einem Mass q für die Auswirkung der Neurotransmitter auf der postsynaptischen Seite. Somit ergibt sich für die synaptische Stärke

$$R = npq \tag{2.17}$$

In Tabelle 2.1 sind einige Arten synaptischer Plastizität aufgeführt. Sie unterscheiden sich zum einen in deren Wirkungsdauer und zum andern in den Voraussetzungen, welche die jeweilige synaptischer Plastizität induziert.
Von besonderem Interesse sollen hier die langanhaltenden Plastizitätsänderungen sein, also *long-term potentiation* (LTP) und *long-term depression* (LTD). LTP, das von [Bliss und Lomo, 1973] im Hippocampus von Kaninchen entdeckt wurde, wird durch eine Aktivierung des präsynaptischen Neurons induziert, auf das innerhalb eines Zeitfensters von einigen Millisekunden [Markram et al., 1997; Bi und Poo, 1998], das postsynaptische Neuron feuern muss. Dies führt dann zu einer langanhaltenden Erhöhung des synaptischen Gewichts, weshalb dies auch als Beleg für das Hebbsche Postulat angesehen wurde. Bei LTD ist die Reihenfolge, die zu einer Induktion führt, vertauscht, so dass erst das postsynaptische Neuron aktiv sein muss, und dann das

Phänomen	Dauer	Voraussetzung der Induktion
short-term enhancement		
paired-pulse facilitation (PPF)	100 ms	prä
augmentation	10 s	prä
posttetanic potentiation (PTP)	1 min	prä
long-term enhancement		
short-term potentiation (STP)	15 min	post
long-term potentiation (LTP)	>30 min	prä und post
depression		
paired-pulse depression (PPD)	100 ms	prä
depletion	10 s	prä
long-term depression (LTD)	$>$ 30 min	prä und post

Tabelle 2.1: Übersicht von experimentell nachgewiesenen Formen der synaptischen Plastizität, nach [Koch, 1999].

präsynaptische Neuron feuern darf. Als Ursache hierfür nimmt man vereinfacht ausgedrückt an, dass die Wahrscheinlichkeit für die Freisetzung von Neurotransmittern p reduziert wird [Stevens und Wang, 1994].
Über die Vorgänge bei LTP ist man sich noch nicht ganz im klaren. Es gibt Arbeiten, die eine Beeinflussung von allen drei Größen in Gleichung 2.17 für relevant halten, siehe hierfür [Koch, 1999].

Die bisherigen Resultate bezogen sich alle auf sogenannte homosynaptische Änderungen. Das bedeutet, dass nur die Plastizität der Synapse geändert wurde, die zwischen den aktiven prä- und postsynaptischen Neuronen liegt. Dies ist ganz im Sinne des Hebbschen Postulats, das eine lokale Änderung der Gewichte fordert. Darüber hinaus gibt es auch Experimente, die darauf hin deuten, dass es unter bestimmten Voraussetzungen auch eine heterosynaptische Plastizität gibt [Brown et al., 1990]. Bei ihr ist nicht nur die beteiligte Synapse von einer Plastizitätsänderung betroffen, sondern alle Synapsen des Neurons. Für einen Überlick hierüber siehe [Bi und Poo, 2001].

Kapitel 3

Hebbsches Lernen und Erweiterungen

3.1 Einleitung

In diesem Kapitel wird eine neue, Hebb-artige Lernregel mit Neuronengedächtnis anhand eines mehrschichtigen neuronalen Netzes, welches in der Lage ist, das XOR-Problem zu lösen, vorgestellt und diskutiert. Diese Lernregel stellt eine Voraussetzung für die Untersuchungen in Kapitel 5. dar, in der ein Zusammenhang zwischen Aktion, Handlung und Lernregel bzw. Lerndynamik in neuronalen Netzen hergestellt wird.
Zuvor wird in Kapitel 3.2. eine von [Klemm et al., 2000] vorgeschlagene Hebb-artige Lernregel vorgestellt, die auf der Einführung eines Synapsengedächtnisses beruht. In Kapitel 3.3. wird gezeigt, wie man von der Hebb-artigen Lernregel mit Synapsengedächtnis [Klemm et al., 2000] mit dem Prinzip der Extremalen Optimierung [Boettcher und Percus, 2001] auf die Hebb-artige Lernregel mit Neuronengedächtnis kommt. Anhand der Behandlung einiger spezieller Beispiele, die mit anderen Arbeiten verglichen werden, wird deren Arbeitsweise vorgestellt und diskutiert. Darüber hinaus wird auf die biologische Plausibilität der Lernregel mit Neuronengedächtnis, und deren Bezug zu experimentellen Resultaten, ausführlich eingegangen.
In Kapitel 3.4. wird eine abgewandelte Form der Lernregel mit Neuronengedächtnis untersucht. Das zugrundeliegende stochastische Optimierungsverfahren ist nun nicht mehr die Extremale Optimierung [Boettcher und Percus, 2001], sondern ein zu *simulated annealing* [Kirkpatrick et al., 1983] ähnliches Verfahren.
Das Kapitel endet mit einer Zusammenfassung aller Ergebnisse.

3.2 Lernregel mit Synapsengedächtnis

3.2.1 Einleitung

Ein aus theoretischer Sicht großer Nachteil der klassischen Hebbschen Lernregel 2.16 ist sicherlich die Tatsache, dass 2.16 keine universelle Computation erlaubt. Dies bedeutet insbesondere, dass man mit 2.16 kein mehrschichtiges neuronales Netz trainieren kann, welches das XOR-Problem lösen kann [Minsky und Papert, 1969].
Aus diesem Grund wurde von [Klemm et al., 2000] eine modifizierte, Hebb-artige Lernregel vorgeschlagen, die es gestattet, das XOR-Problem in einem mehrschichtigen neuronalen Netz zu lösen, und sich aus biologischer Sicht plausibel interpretieren lässt.
Bevor dieses Modell vorgestellt wird, soll noch kurz auf eine allgemeine Charakterisierung neuronaler Netze eingegangen werden.

3.2.2 Beschreibung des Modells

Wenn man ein konkretes Problem mit neuronalen Netzen lösen will, so muss man neben einer Lernregel noch weitere Bestandteile des Gesamtsystems definieren.

Definition 3.1 (Charakterisierung eines Gesamtsystems:)
1. *Topologie des neuronalen Netzes*
2. *Neuronenmodell*
3. *Netzwerkdynamik*
4. *Lernregel*
5. *Umgebung*
6. *Wechselwirkung zwischen TBM und Umgebung*

Hier wird im Hinblick auf Kapitel 5. und 6., das zu lösende Problem abstrakt aufgefasst, und als Umgebung bezeichnet. Der Grund hierfür wird in Kapitel 5. von selbst klar werden. Die Punkte (1)-(4) in Definition 3.1 legen das Toy-Brain-Model (TBM) fest, welches abkürzend für diese Punkte steht.
Bei dem Gesamtsystem 3.1 ist darauf zu achten, dass nicht einfach einzelne Punkte gegen andere ausgetauscht werden können, ohne das Verhalten des Gesamtsystems möglicherweise zu beeinflussen. Vielmehr stellt das Gesamtsystem eine Art Selbstkonsistenzmechanismus dar [Henkel, 2001], welcher das Potential besitzt, ein gegebenes Problem zufriedenstellend lösen zu können [Bak und Chialvo, 2001]. Dies kann man sich anhand einiger Beispiele klarmachen.

Wechselseitige Beziehung zwischen einzelnen Punkten in Definition 3.1:
(5) $\Rightarrow$ (1)

Beispielsweise ist es nicht möglich mit einem einzelnen Perzeptron das XOR-Problem zu lösen [Minsky und Papert, 1969], man benötigt hierfür eine mehrschichtiges neuronales Netz.
(4) $\Rightarrow$ (5)
Mit der klassischen Hebbschen Lernregel ist es nicht möglich, das XOR-Problem zu lösen.

Das von [Klemm et al., 2000] vorgeschlagene Modell definiert das Gesamtsystem 3.1 wie folgt:

Definition 3.2 (Gesamtsystem nach [Klemm et al., 2000]:)
1. Topologie des neuronalen Netzes: *Dreischichtiges, vorwärtsgerichtetes Netz, mit* I *Input-,* H *Hidden- und* O *Outputneuronen. Aufeinanderfolgende Schichten sind mit Synapsen* $w_{ij} \in \mathbb{R}$ *derart verbunden, dass jedes Neuron einer Schicht mit jedem Neuron der darauffolgenden Schicht verbunden ist.*
2. Neuronenmodell: *Binäre Neuronen* $x_i \in \{0, 1\}$ *mit* $i \in \{1, \ldots, I + H + O\}$
3. Netzwerkdynamik: *Das innere Feld der Neuronen berechnet sich durch* $h_j = \sum_i^{\text{all}} w_{ji} x_i$*, wobei* all *alle Neuronen der vorangegangenen Schicht bezeichnen. Die Aktivität erhält man nun aus einer stochastischen Approximation des winner-take-all Mechanismus, siehe [Yuille und Geiger, 1995].*

$$p_j = Z^{-1} \exp(\beta h_j) \tag{3.1}$$

$$Z = \sum_j \exp(\beta h_j) \tag{3.2}$$

Mit $\beta \in \mathbb{R}$ *und* $\beta(t) = \beta$ *als temperaturartigem Parameter. Es ist zu beachten, dass in jeder Schicht des Netzwerkes nur ein Neuron zu einem Zeitpunkt aktiv sein soll, wie dies von [Chialvo und Bak, 1999] vorgeschlagen wurde und dort als extremale Dynamik (*extremal dynamics*) bezeichnet wird. Aus diesem Grund wird in jeder Schicht, gemäß den ermittelten Wahrscheinlichkeiten 3.1, jeweils nur ein aktives Neuron, durch Ziehen aus eben dieser Verteilung, ermittelt.*
4. Lernregel: *Die Änderung der Synapsen* w_{ij} *erfolgt, falls* $c_{ij} - r > \Theta$ *in 3.4 eingetreten ist. Hierbei ist* $\Theta \in \mathbb{N}$ *eine Schwelle,* $r = \pm 1$ *ein Reinforcementsignal und* c_{ij} *ein Synapsenzähler. In diesem Fall wird das synaptische Gewicht durch*

$$w_{ij} \rightarrow w'_{ij} = w_{ij} - \delta, \tag{3.3}$$

mit $\delta \in \mathbb{R}^+$ *und* $\delta(t) = \delta$ *deprimiert.*
Neben einer Dynamik für die Neuronen gibt es noch eine Dynamik für die Synapsenzähler c_{ij}*, die durch*

$$c_{ij} \rightarrow c'_{ij} = \begin{cases} \Theta, \text{if} & c_{ij} - r > \Theta \quad (*) \\ c_{ij} - r, \text{if} & \Theta \geq c_{ij} - r \geq 0 \\ 0, \text{if} & 0 > c_{ij} - r. \end{cases} \tag{3.4}$$

gegeben ist. Durch Gleichung 3.4 werden immer nur die Synapsenzähler der aktiven Synapsen verändert. Alle anderen Synapsenzähler bleiben ungeändert.
5. Äußere Umgebung: *XOR-Problem*

x_3	x_2	x_1	x_8	x_7
0	*0*	*1*	*1*	*0*
0	*1*	*1*	*0*	*1*
1	*0*	*1*	*0*	*1*
1	*1*	*1*	*1*	*0*

Hierbei dient x_1 als Bias, damit sichergestellt ist, dass die Inputaktivität nie identisch Null werden kann.
6. Wechselwirkung zwischen TBM und Umgebung: *Jedes Inputmuster (aus der äußeren Umgebung) wird unabhängig gewählt und mit gleicher Wahrscheinlichkeit präsentiert.*

3.2.3 Arbeitsweise des Modells

Die innovative Idee des von [Klemm et al., 2000] vorgestellten Modells besteht darin, dass jeder Synapse ein zusätzlicher Freiheitsgrad in Form eines Synapsenzählers c_{ij} zugeordnet wird, der darüber entscheidet, ob ein synaptisches Gewicht w_{ij} verändert werden soll. Die Dynamik des Synapsenzählers wird durch ein globales Reinforcementsignal r determiniert, welches allen Synapsen als Feedback zugänglich ist, wie bereits von [Strassinopoulos und Bak, 1995] vorgeschlagen und verwendet.
Wenn man nun dem neuronalen Netz, gemäß Punkt (5) und (6) der Definition 3.2 ein Inputmuster präsentiert, so berechnet sich durch die Netzwerkdynamik der Neuronen die Aktivität der Outputschicht. Ein äußerer Beobachter kann nun das tatsächliche mit dem gewünschten Resultat, anhand von (5), vergleichen, und dem neuronalen Netz ein skalares Reinforcementsignal zuführen, dass durch $r = 1$ ein richtiges und durch $r = -1$ ein falsches Ergebnis der Berechnung signalisiert. Nun wird durch die Synapsenzählerdynamik, welche durch das Reinforcementsignal r determiniert ist, die neuen Werte für c_{ij} ermittelt, wobei zu beachten ist, dass nur die Synapsenzähler der aktiven Synapsen betroffen sind. Alle anderen bleiben ungeändert. Als aktiv wird eine Synapse genau dann bezeichnet, wenn ihr prä- und postsynaptisches Neuron aktiv ist.
Falls für einen Synapsenzähler der aktiven Synapsen der Fall $c_{ij} - r > \Theta$ eintrit, dann wird das synaptische Gewicht der zugehörigen Synapse gemäß 3.25 deprimiert. Nun beginnt die Prozedur von neuem.

Die zugrunde liegende Idee des Synapsengedächtnisses ist eine Mittelung über einen Teil der Menge der Inputmuster, ähnlich wie beim *batch*

Lernen [Heskes und Wiegerinck, 1996]. Je größer Θ, desto größer ist der Teil der Menge der Inputmuster, über den gemittelt wird. Da die Mittelung nur über einen sehr kurzen Zeitraum genommen wird, $\Theta \sim O(1)$, handelt es sich genauer gesagt um die Ermittlung von relativen Häufigkeiten. Diese Interpretation erhält man aus 3.4 mit der Transformation von $\overline{c}_{ij} = \frac{c_{ij}}{\Theta}$ und $\overline{r} = \frac{r}{\Theta}$, denn hieraus folgt

$$\overline{c}_{ij} \rightarrow \overline{c}'_{ij} = \begin{cases} 1, \text{if} & \overline{c}_{ij} - \overline{r} > 1 \\ \overline{c}_{ij} - \overline{r}, \text{if} & 1 \geq \overline{c}_{ij} - \overline{r} \geq 0 \\ 0, \text{if} & 0 > \overline{c}_{ij} - \overline{r}. \end{cases} \tag{3.5}$$

Aus 3.5 sieht man, dass $\overline{c}_{ij} \in [0, 1]$, hieraus folgt die Interpretation des Zählers als relative Häufigkeit.
Darüber hinaus wird durch die Einführung eines dynamischen Synapsenzählers eine zweite, langsamere Zeitskala eingeführt, auf der die synaptischen Änderungen ablaufen.

Die verwendete Netzwerkdynamik, in Form eines softmax Mechanismus, lässt sich durch einen temperaturartigen Parameter β, kontinuierlich zwischen einem winner-take-all Mechanismus für $\beta \rightarrow \infty$, und einer zufälligen, gleichverteilten Auswahl des aktiven Neurons für $\beta \rightarrow 0$, ineinander überführen.
Die Regulierung der Netzwerkaktivität wird durch das Prinzip der extremalen Dynamik [Chialvo und Bak, 1999] implizit dadurch vorgenommen, dass man in jeder Schicht des neuronalen Netzes immer nur ein aktives Neuron, aus der durch 3.1 festgelegten Verteilung, zieht.
Es sei noch angemerkt, dass die Parameter β und δ nicht voneinander unabhängig sind. Dies sieht man leicht unter Verwendung von Gleichung 3.1 und 3.3.

$$h_j = \sum_i^{all} w_{ji} x_i \tag{3.6}$$

$$h'_j = \sum_i^{all} w'_{ji} x_i = h_j - \delta \sum_i^{all} x_i \tag{3.7}$$

$$\exp(\beta h'_j) = \exp(\beta h_j - \beta\delta \sum_i^{all} x_i) \tag{3.8}$$

Hieraus folgt, dass eine Änderung in der Größe δ keine Veränderung in der Wahrscheinlichkeitsverteilung p'_j hervorruft, solange das Produkt $\beta\delta$ konstant bleibt. Mit einer "Temperaturänderung" durch β lässt sich also eine Veränderung der Größe δ ausgleichen und umgekehrt. Aus diesem Grund wird in allen Simulationen, bei denen der softmax Mechanismus verwendet wird,

$\delta = 1$ gesetzt und der temperaturartige Parameter β als Variable betrachtet, da hierdurch eine "Eichung" von β vorgenommen wird.

Zu 3.3 sei noch angemerkt, dass alle Synapsen einer Schicht um einen konstanten positiven Betrag angehoben werden, sobald ein w_{ij} negativ wird. Die Invarianz der Wahrscheinlichkeitsverteilung 3.1 folgt direkt aus 3.8.

3.2.4 Ergebnisse

Um die Funktionsweise, des in 3.2 vorgestellten Modells von [Klemm et al., 2000] zu verstehen, wurden die dort erhaltenen Ergebnisse nachsimuliert und durch weitere Parametereinstellungen von β und Θ ergänzt. Abbildung 3.1 zeigt die Resultate, die sich für $\delta = 1$ und $\beta = 10$ ergeben. Die Lernkurven

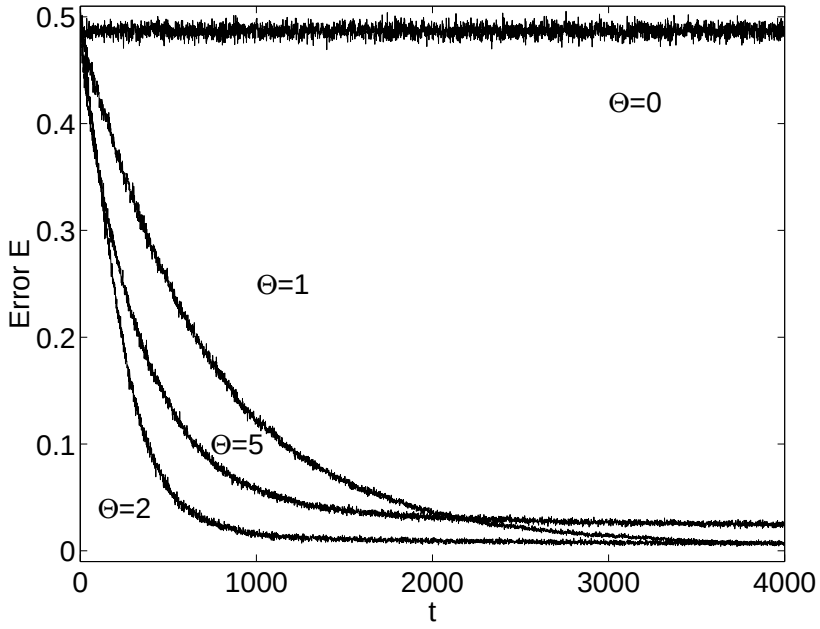


Abbildung 3.1: Mittlerer Fehler E in Abhängigkeit der Zeit t für verschiedene Werte von Θ für das Modell mit Synapsengedächtnis. Als Netzwerkdynamik wurde der softmax Mechanismus mit $\beta = 10$ verwendet.

ergeben sich aus unabhängigen Simulationen eines Ensembles der Größe $N = 10000$, wobei sich der mittlere Fehler E durch

$$E(t) = \frac{1}{N}\sum_{i=1}^{N} e_i(t) \tag{3.9}$$

$$e_i(t) \in \{0, 1\} \tag{3.10}$$

berechnet. Hierbei ist $e_i(t)$ der Fehler der i'ten Realisation zum Zeitpunkt t. In allen Simulationen werden die Synapsengewichte durch $w_{ij} \in [0, 1]$ zufällig,

gleichverteilt initialisiert und alle Synapsenzähler c_{ij} gleich Null gesetzt.
Aus Abbildung 3.1 liest man ab, dass sich für $\Theta = 2$ die beste Lernkurve ergibt.

Den selektiven Einfluss des Synapsengedächtnisses Θ kann man sich an der Abbildung 3.2 klarmachen. Hier ist das neuronale Netz mit seinen drei Schichten gezeigt. Die prinzipiell vorhandenen Synapsen zwischen den Neuronen der verschiedenen Schichten sind gepunktet gezeichnet, wohingegen die Synapsen, welche an der Verarbeitung der letzten beiden Inputmuster beteiligt waren, mit durchgezogenen Linien gezeichnet sind. Anhand der Doppelbeteiligung der Synapse der Hiddenschicht (H) mit der Outputschicht (O), wird die Verschränkung zwischen beiden Inputmustern innerhalb des neuronalen Netzes offensichtlich. Denn die Änderung einer Synapse wirkt

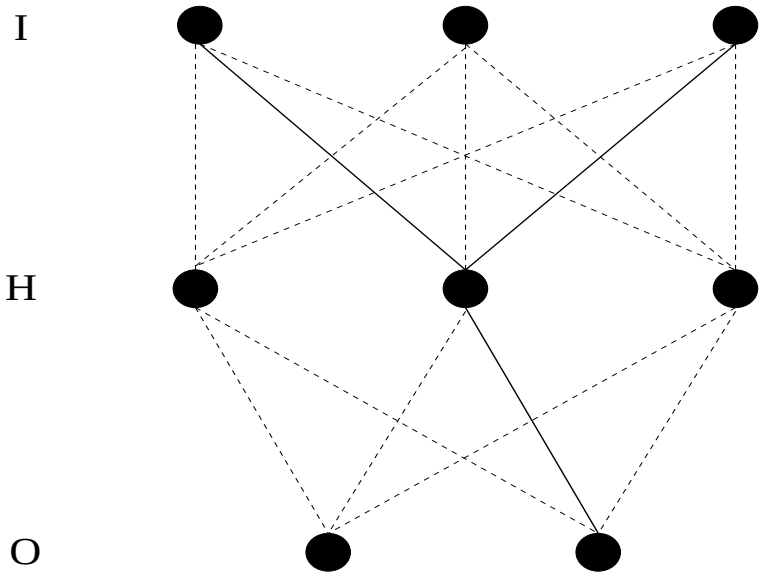


Abbildung 3.2: Netzwerk Topologie, welche als Ansatz für das zu lösende XOR-Problem gewählt wurde. Gezeigt sind die drei Schicht, Input (I), Hidden (H) und Output (O). Die gepunkteten Linien stellen die vorhandenen synaptischen Verbindungen dar. Die durchgezogenen Linien symbolisieren die neuronalen Aktivitätspfade, die von den letzten beiden Inputmustern hinterlassen wurden.

sich nicht nur auf das zuletzt präsentierte Inputmuster, sondern auch auf das vorangegangene aus, da beide Inputmuster von der Hidden- zur Outputschicht den selben Pfad benutzen. Somit ist es sinnvoller den Wert einer Synapse zu verändern, die in der Vergangenheit öfter an falschen Resultaten beteiligt war, als andere Synapsen. Längere Gedächtniseffekte macht man sich analog klar.

Der Einfluss des temperaturartigen Parameters β, der den softmax Me-

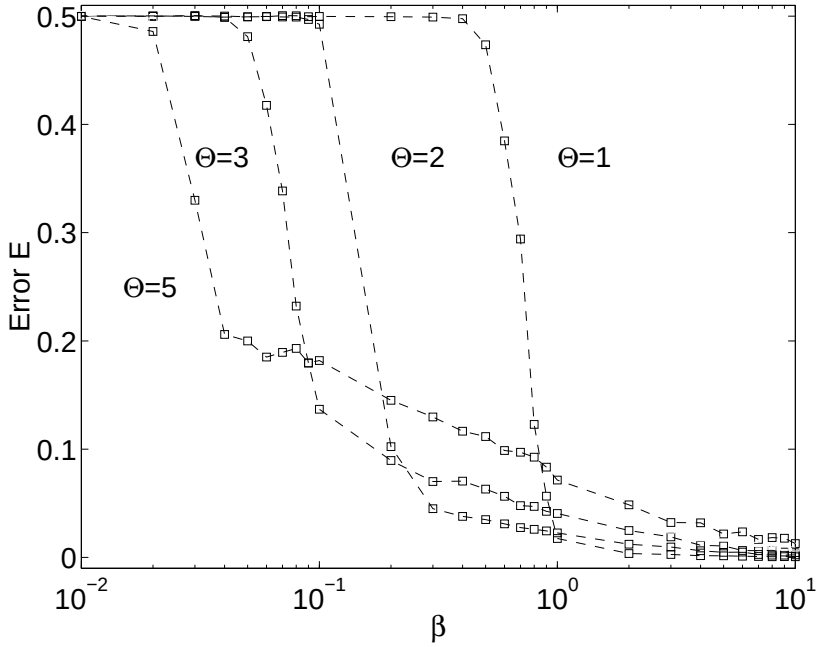


Abbildung 3.3: Mittlerer Fehler E in Abhängigkeit vom temperaturartigen Parameter β für verschiedene Θ Werte. Die Lerndynamik wird durch das Modell mit Synapsengedächtnis bestimmt.

chanismus steuert, ist in Abbildung 3.3 dargestellt. Der dort aufgetragene Fehler E berechnet sich durch

$$E = \frac{1}{N(T_m+1)} \sum_{i=1}^{N} \sum_{t=T-T_m}^{T} e_i(t) \tag{3.11}$$

$$e_i(t) \in \{0,1\} \tag{3.12}$$

Hierbei beträgt die Ensemblegröße $N = 100$ und die Simulationszeit $T = 100000$, wobei über die letzten $T_m = 10000$ Zeitschritte gemittelt wird, da die Lernkurven einen stationären Zustand erreicht haben.
Da für $\beta \to 0$ der softmax Mechanismus gegen eine Gleichverteilung geht, ist für kleine β kein Lernerfolg zu erwarten. Für $\beta \to \infty$ erhält man den deterministischen winner-take-all Mechanismus, welcher das XOR-Problem lernen kann (Fehler unter 10%). Hieraus folgt, dass kein Θ Wert gegenüber anderen ausgezeichnet ist.

Man erkennt, dass hier ein Synapsenzähler der Länge $\Theta > 2$ schon bei niedrigen Werten von β bessere Lernerfolge aufweißt, als für kleinere Θ Werte. Dieses Verhalten wird plausibel, wenn man sich daran erinnert, dass man den Synapsenzähler als relative Häufigkeit des Fehlverhaltens der Synapse, wie in 3.5 angegeben, auffassen kann und β als inverses Rauschlevel für die

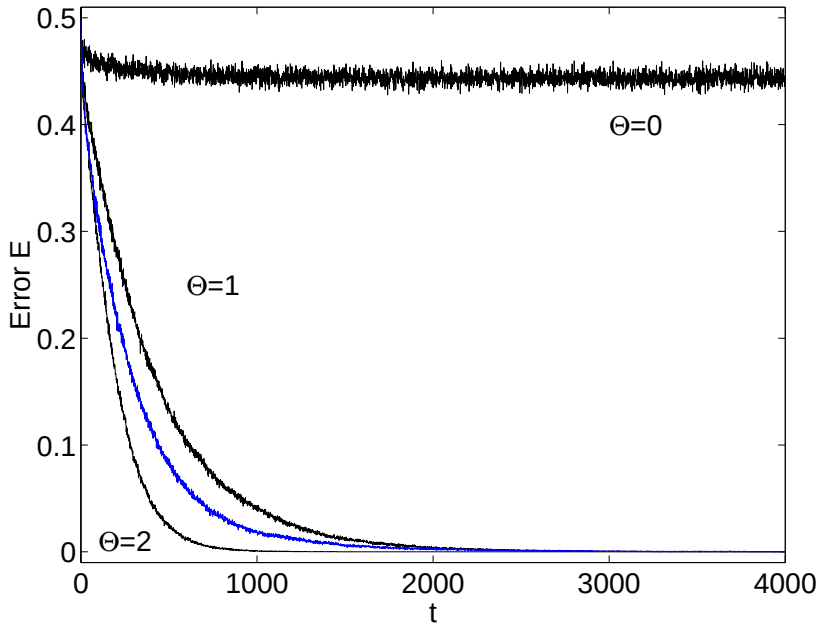


Abbildung 3.4: Mittlerer Fehler E des Modells mit Synapsenzähler in Abhängigkeit der Zeit t und des Schwellwertes Θ für den winner-take-all Mechanismus, $\beta \to \infty$. Jede Lernkurve ist über ein Ensemble der Größe $N = 10000$ gemittelt. Die blaue Kurve entspricht $\Theta = 5$.

Neuronen [Klemm et al., 2000], denn bei höherem Rauschen muss man eine Zeitreihe über einen längeren Zeitraum mitteln, um das Signal bestimmen zu können. Dies entspricht genau der Wirkung von größeren Θ Werten.

In Abbildungen 3.4 sind die zu Abbildung 3.1 analogen Ergebnisse für $\beta \to \infty$ dargestellt, die dem winner-take-all Mechanismus entsprechen. Gemittelt wurde wieder über ein Ensemble der Größe $N = 10000$. Die Kurven für $\Theta = 1$ bis $\Theta = 5$ zeigen[1] auf kurzen Zeitskalen eine geringfügig schnellere Konvergenz als für $\beta = 10$. Für größer werdende Zeiten erfolgt jedoch eine schnellere Konvergenz gegen Null, wie im Fall $\beta = 10$. Dies ist auch verständlich, da β^{-1} als Rauschen interpretiert werden kann.

3.2.5 Biologische Plausibilität

Die Arbeiten von [Chialvo und Bak, 1999; Bak und Chialvo, 2001] und [Klemm et al., 2000], die teilweise aufeinander aufbauen, basieren auf zwei bzw. drei entscheidenden Annahmen, deren biologische Plausibilität nun diskutiert werden soll.

[1] Aus Gründen der Übersicht sind nicht alle dargestellt.

Die ersten beiden Annahmen sind, dass aktive Neuronen über einen gewissen Zeitraum hinweg einen zusammenhängenden Pfad im neuronalen Netz hinterlassen müssen. Der Zeitraum muss gerade so lange sein, bis von außen ein Reinforcementsignal zugeführt wird, welches die am letzten Verarbeitungsschritt beteiligten Synapsen bestraft oder belohnt, je nachdem, wie die Berechnung ausgefallen ist.
Für das "Zusammenheften" der Neuronen wird in [Chialvo und Bak, 1999] eine Arbeit von [Frey und Morris, 1997] angeführt. Frey und Morris führten in vitro Untersuchungen am Hippocampus von Ratten durch. Dabei fanden sie, dass Synapsen, die zu einem früheren Zeitpunkt schon aktiv gewesen waren, eine Art Gedächtnis haben müssen, da gerade diese Synapsen durch erneute Stimulation LTP zeigten. Dieses *synaptic tagging* dauert bis zu 90 Minuten an. Diese Arbeit könnte also die erst Forderung erfüllen. Zu beachten ist jedoch, dass die durchgeführten Experimente LTP induzierten, wohingegen LTD benötigt würde. Somit resultiert aus der Arbeit von [Chialvo und Bak, 1999] die Hypothese, dass es *synaptic tagging* auch im Zusammenhang mit LTD geben müsste.

Für die zweite Forderung wird in [Klemm et al., 2000] eine Arbeit von [Otmakhova und Lisman, 1998] angeführt, bei der der Einfluss eines globalen Signals, in Form von Dopamin, auf LTP und LTD in CA1 des Hippocampus gezeigt wurde. Dies könnte die Rolle des globalen Reinforcementsignals übernehmen.

Die dritte Forderung, die nur von [Klemm et al., 2000] benötigt wird, basiert auf einem Synapsenzähler, der in Abhängigkeit eines globalen Reinforcementsignals verändert werden kann und nur bei Überschreiten einer bestimmten Schwelle die synaptische Änderung nach sich zieht. Hierfür fehlt meiner Ansicht nach eine biologische Rechtfertigung, jedoch könnte eine Kombination der Arbeiten von [Frey und Morris, 1997] und [Otmakhova und Lisman, 1998] unter Umständen so etwas leisten. Dies stellt eine Hypothese dar, die von [Klemm et al., 2000] implizit aufgestellt wird.

3.3 Lernregel mit Neuronengedächtnis

3.3.1 Einleitung

Im vorangegangenen Kapitel 3.2 wurde ein Modell von [Klemm et al., 2000] vorgestellt, welches es erlaubt, das XOR-Problem in einem dreischichtigen neuronalen Netz mit einer Hebb-artigen Lernregel zu lernen.
Aufbauend auf die Arbeit von [Klemm et al., 2000] soll in diesem Kapitel eine neue Hebb-artige Lernregel, basierend auf einem Neuronengedächtnis,

vorgestellt werden, die das XOR-Problem ebenfalls lösen kann, jedoch unter Verwendung von weniger Ressourcen. Das Modell wird anhand einiger spezieller Beispiele untersucht, und mit den Ergebnissen von [Klemm et al., 2000] verglichen.
Am Ende des Kapitels wird dessen biologische Plausibilität diskutiert.

3.3.2 Umkodierung

Da das in Kapitel 3.2. vorgestellte Modell von [Klemm et al., 2000] recht gute Ergebnisse liefert und der zugrundeliegende Gedanke sehr plausibel ist, kann man sich nun die Frage stellen, ob sich dieses Modell noch verbessern lässt?
Ein möglicher Ansatz besteht darin, die vom Modell verwendete Anzahl der freien Parameter zu verringern. Es liegt also nahe zu versuchen, die verwendeten Synapsenzähler gegen andere einzutauschen. Als extremes Beispiel vergleiche man z.B. die Anzahl der Neuronen und Synapsen im Gehirn des Menschen und der Ratte. Dann erhält man beim Menschen 10^{12} Neuronen und 10^{15} Synapsen und bei der Ratte 10^{10} Neuronen und 10^{13} Synapsen[2]. Somit ist die Idee geboren, aus den Synapsen- Neuronenzähler zu machen.

Der entscheidende Punkt ist nun, dass es einen allgemeinen Zusammenhang zwischen dem Synapsen- und Neuronenzähler gibt, der als Zählererhaltung interpretiert werden kann. Für eine gegebene Netzwerktopologie lassen sich die Zählerstände der Neuronen durch die hin- bzw. wegführenden Zähler der Synapsen ausdrücken. Dies wird durch Abbildung 3.5 veranschaulicht.

Exemplarisch sind hier die Erhaltungsregeln für den Neuronenzähler des 5'ten Neurons dargestellt.

$$c_5 = c_{15} + c_{25} + c_{35} \tag{3.13}$$

$$c_5 = c_{57} + c_{58} \tag{3.14}$$

Da das 5'te Neuron nur aktiv gewesen sein kann, wenn Neuron 1,2 oder 3 aktiv gewesen ist, und dementsprechend die Synapsen zwischen Neuron 5 und Neuron 1,2 oder 3, ergibt sich eine Erhaltung für die Summe der in Neuron 5 hineinlaufenden Synapsenzähler durch Gleichung 3.13. Analoges gilt für die herauslaufenden Synapsenzähler von Neuron 5, was durch Gleichung 3.14 beschrieben wird.

Für allgemeine neuronale Netze erhält man eine lineare Abbildung M vom Raum der Synapsenzähler in den Raum der Neuronenzähler.

$$c_n = M c_s \tag{3.15}$$

[2] Aus [Churchland und Sejnowski, 1992] entnommen.

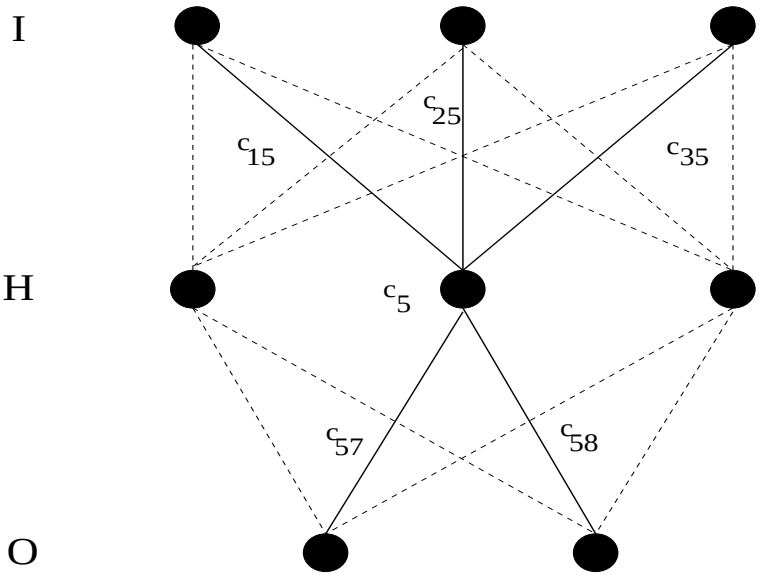


Abbildung 3.5: Netzwerk Topologie für das XOR-Problem. Eingezeichnet sind die Synapsen- bzw. Neuronenzähler die zum bzw. vom 5'ten Neuron führen.

$$c_n \in \mathbb{N}^N \quad , c_s \in \mathbb{N}^S, \quad M \in \mathrm{Mat}(N \times S; \mathbb{R}) \tag{3.16}$$

Hierbei bezeichnet N die Anzahl der Neuronen und S die Anzahl der Synapsen des neuronalen Netzes. Die reellwertige Lineare Abbildung M ist durch die Topologie des neuronalen Netzes vorgegeben und ergibt sich aus den jeweiligen Zählererhaltungen.
Da der neuen Lernregel nur die Neuronenzähler zur Verfügung stehen sollen, sind wir an der Umkehrbeziehung interessiert, welche uns eine Approximation der Synapsen- durch die Neuronenzähler liefert. Für die Lineare Abbildung M in 3.15 existiert jedoch keine Umkehrabbildung, falls $N \neq S$, was den Normalfall von Netztopologien darstellt. Deshalb lässt sich die Umkehrabbildung nur mit Hilfe einer Pseudoinversen erzielen. Hierfür wähle ich die Moore-Penrose Pseudoinverse [Penrose, 1955].
Wenn man sich jedoch die konkreten Ergebnisse[3] für die Netztopologie in Abbildung 3.5 ansieht, so stellt man fest, dass sich die approximierten Synapsenzähler nicht lokal aus den Neuronenzählern ergeben. Das bedeutet, dass nicht nur die Neuronen an der Approximation beteiligt sind, welche die Synapse eingrenzen, sondern auch weiter entfernte.

Um den Einfluss dieser nicht lokalen Information zu unterdrücken, spalte ich die Lineare Abbildung M derart auf, dass die Synapsenzähler nur durch die angrenzenden Neuronenschichten beschrieben werden können. Für die

[3]Nicht dargestellt.

Lineare Abbildung M ergibt sich also

$$M = \begin{pmatrix} M^{l_1 l_2} & N^1 \\ N^2 & M^{l_2 l_3} \end{pmatrix} \tag{3.17}$$

Mit $M^{l_1 l_2} \in \mathrm{Mat}(I + H \times I * H; \mathbb{R})$, $M^{l_2 l_3} \in \mathrm{Mat}(H + O \times H * O; \mathbb{R})$, $N^1 \in \mathrm{Mat}(I \times H * O; \mathbb{R})$ und $N^2 \in \mathrm{Mat}(H \times I * H; \mathbb{R})$[4]. Im speziellen erhält man für die Netztopologie in Abbildung 3.5

$$M^{l_1 l_2} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \tag{3.18}$$

$$M^{l_2 l_3} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \tag{3.19}$$

Für jede dieser Linearen Abbildungen kann man nun wieder die Moore-Penrose Pseudoinverse ausrechnen, die in Anhang A1 dargestellt sind.
Wiederum stellt man fest, dass die approximierten Synapsenzähler im allgemeinen nicht lokal durch die Neuronenzähler approximiert werden können. Es lässt sich auch keine Struktur in den jeweiligen Pseudoinversen erkennen, welche für eine sinnvolle Approximation ausgenutzt werden könnte.

Es sei noch einmal ausdrücklich darauf hingewiesen, dass eine lokale Approximation unbedingt notwendig ist, um sie im Sinne des Postulats von Hebb interpretieren zu können.

Aus diesem Grund schlage ich ein stochastisches Approximationsverfahren, basierend auf dem Prinzip der Extremalen Optimierung von [Boettcher und Percus, 2001] vor.

3.3.3 Extremale Optimierung

Unter diesem Titel wurde von [Boettcher und Percus, 2001, 2000; Boettcher, 2000] ein neues, heuristisches Verfahren zur Lösung harter, diskreter Optimierungsprobleme eingeführt. Das Prinzip der Extremalen Optimierung (EO) wurde durch Arbeiten über *self-organized-criticality* (SOC) [Bak et al., 1987;

[4] Die Matrizen N^1 und N^2 sind Nullmatrizen.

Bak und Sneppen, 1993] inspiriert und versucht, große Fluktuationen, wie sie inhärent in Systemen fern vom Gleichgewicht auftreten, sinnvoll auszunutzen [Boettcher und Percus, 2001].

Das Verfahren ist auf Probleme anwendbar, denen man eine Fitnessfunktion $C(\mathbf{x})$ zuweisen kann, welche sich aus Elementarfitnessfunktionen λ_i zusammensetzt. In [Boettcher und Percus, 2001] werden beispielsweise Spinglässer untersucht, deren Gesamtfitness $C(\mathbf{x})$ der Hamiltonfunktion $H(\mathbf{x})$ entspricht, die sich additiv aus den lokalen Wechselwirkungen $\lambda_i = x_i(\frac{1}{2}\sum_j^n J_{ij}x_j)$ zusammensetzt.

$$H(\mathbf{x}) = -\frac{1}{2}\sum_{ij}^{n} J_{ij}x_i x_j = -\sum_{i=1}^{n} \lambda_i = C(\mathbf{x}) \tag{3.20}$$

Hierin sind $x_i \in \{-1, 1\}$ die Spinvariablen und $J_{ij} \in \{-1, 1\}$ die Kopplungen zwischen den Spinvariablen.

Der EO Algorithmus verläuft nach folgendem Schema:

Algorithmus 3.1 (EO Algorithmus:)

1. *zufällige Initialisierung aller Systemparameter*
2. *berechne elementare Fitness* λ_i
3. *berechne Gesamtfitness* $C(\mathbf{x})$
4. *bestimme eine Rangordnung der elementaren Fitnessfunktionen*

   $$\lambda_{\Pi(1)} \leq \lambda_{\Pi(2)} \cdots \leq \lambda_{\Pi(n)} \tag{3.21}$$

   Hierbei bezeichnet Π *eine geeignete Permutation der Indizes* i.
5. *wähle einen Index* $\Pi(i)$ *aus der Wahrscheinlichkeitsverteilung der Rangordnungen*

   $$P_k \propto k^{-\tau} \tag{3.22}$$
   $$1 \leq k \leq n \tag{3.23}$$
   $$\tau \in \mathbb{R}^+ \tag{3.24}$$
6. *ändere die mit* $\Pi(i)$ *verbundenen Systemparameter. Das bedeutet, dass für* $\Pi(i)$ *eine neue Fitness zugewiesen wird.*
7. *springe zu (2) außer ein Abbruchkriterium ist erfüllt*

Für $\tau \neq 1$ nennt man die Extremale Optimierung auch τ -EO.

Mit dem Parameter τ kann man kontinuierlich zwischen zwei Extremen umschalten. Für $\tau = 0$ erhält man einen random walk in den Indizes i, wohingegen $\tau \to \infty$ einer deterministischen Auswahl des Index mit der schlechtesten Fitness λ_i entspricht.

Der entscheidende Unterschied von EO zu anderen Optimierungsverfahren, wie z.B. simulated annealing [Kirkpatrick et al., 1983] besteht darin, dass EO, durch Steuerung des Exponenten τ, große Fluktuationen in das System einführt, was sich an der Gesamtfitnessfunktion $C(x)$ ablesen lässt [Boettcher, 2000], und dadurch große, aber gerichtete, Sprünge im Raum der Fitnessfunktionen macht. Eine formale Begründung für die genau stattfindenden Abläufe, sind noch Gegenstand aktueller Forschungsarbeiten [Boettcher, 2002].

Nach diesem kleinen Exkurs über die Extremale Optimierung (EO) sind nun alle Einzelbestandteile vorhanden, um die Lernregel mit Neuronengedächtnis präsentieren zu können.

3.3.4 Modell mit Neuronengedächtnis

Am Beginn des Kapitels 3.2. wurde bereits erwähnt, dass nur durch Definition aller Bestandteile des Gesamtsystems 3.1 die Wirkungsweise einer einzelnen Komponente untersucht werden kann, da sich die einzelnen Bestandteile des Gesamtsystems gegenseitig beeinflussen. Aus Gründen der direkten Vergleichbarkeit wird im folgenden die spezielle Definition 3.2 identisch übernommen, bis auf Punkt 4., der die Lernregel definiert. Hierfür wird folgende Lernregel vorgeschlagen:

Definition 3.3 (Hebb-artige Lernregel mit Neuronengedächtnis:)
4. Lernregel: *Die Änderung der Synapsen w_{ij} erfolgt, falls $r = -1$ und $p_{coin} < p^{rank}_{\tilde{c}_{ij}}$ aus 3.28 und 3.31 gilt. In diesem Fall wird das synaptische Gewicht durch*

$$w_{ij} \to w'_{ij} = w_{ij} - \delta, \tag{3.25}$$

mit $\delta \in \mathbb{R}^+$ und $\delta(t) = \delta$ deprimiert.
Neben einer Dynamik für die Neuronen gibt es noch eine Dynamik für die Neuronenzähler c_i, die durch

$$c_i \to c'_i = \begin{cases} \Theta, \text{if} & c_i - r > \Theta \quad (*) \\ c_i - r, \text{if} & \Theta \geq c_i - r \geq 0 \\ 0, \text{if} & 0 > c_i - r. \end{cases} \tag{3.26}$$

gegeben ist. Hierbei ist $\Theta \in \mathbb{N}$ *eine Schwelle,* $r = \pm 1$ *ein Reinforcementsignal und* c_i *ein Neuronenzähler. Durch Gleichung 3.26 werden immer nur die Neuronenzähler der aktiven Neuronen verändert. Alle anderen Neuronenzähler bleiben ungeändert.*

Um entscheiden zu können, wann eine Synapse geändert wird, muss man folgende Prozedur durchlaufen:

1. *berechne die approximierten Synapsenzähler* $\widetilde{c_{ij}}$ *der aktiven Synapsen durch die Neuronenzähler 3.26,*

$$\widetilde{c_{ij}} = c_i + c_j \tag{3.27}$$

2. *da* $c_i \in \mathbb{N}$ *für alle* $i \in \{1, ..., N\} \Rightarrow \widetilde{c_{ij}} \in \mathbb{N}$*, lässt sich jeder aktiven Synapse, ein approximierter Synapsenzähler und somit eine Wahrscheinlichkeit* $p^{rank}_{\widetilde{c_{ij}}}$ *zuordnen, welche durch die Rangordnungsverteilung*

$$P^{rank}_k \propto k^{-\tau} \tag{3.28}$$
$$k \in \{1, \ldots, 2\Theta + 3\} \tag{3.29}$$
$$\tau \in \mathbb{R}^+ \tag{3.30}$$

   festgelegt wird. Hierbei gilt die Abbildung $k = 2\Theta + 3 - \widetilde{c_{ij}}$.

3. *Für* $P(x)_{coin}$ *wird ebenfalls ein Potenzgesetz gewählt*

$$P(x)_{coin} \propto x^{-\alpha} \tag{3.31}$$
$$x \in [0, 1] \tag{3.32}$$
$$\alpha \in \mathbb{R}^+ \tag{3.33}$$

   aus dem für jede aktive Synapse eine Wahrscheinlichkeit p_{coin} *gezogen wird.*

Bemerkung: Es sei ausdrücklich angemerkt, dass im Unterschied zu dem Modell von S. Boettcher, hier nicht die Frage betrachtet wird, welche Fitnessfunktion zu einem bestimmten Zeitpunkt auszuwählen und zu ändern ist, sondern ob sie überhaupt verändert wird! Welche Synapse upgedated wird, ist ja bereits durch die Extremale Dynamik der Netzwerkaktivität definiert. Entscheidend ist die Frage, ob die Synapse auch wirklich geändert wird, oder nicht.

3.3.5 Arbeitsweise des Modells

Um die einzelnen Komponenten, die alle zur Lernregel 3.3 beitragen, besser zu verstehen, sei der Gesamtablauf kurz skizziert.
Dem neuronalen Netze wird ein Muster präsentiert. Danach berechnet sich mit der Netzwerkdynamik die Aktivität der Neuronen. Aufgrund der Aktivität der Neuronen in der Outputschicht, resultiert durch einen äußeren Beobachter

ein Reinforcementsignal r, durch welches das Netzwerk signalisiert bekommt, ob die durchgeführte Berechnung richtig $r = 1$ oder falsch $r = -1$ war. Dieses Reinforcementsignal wird "demokratisch" [Strassinopoulos und Bak, 1995] allen Neuronen zugeführt. Durch die Dynamik der Neuronenzähler werden die Zähler der aktiven Neuronen durch $c_i - r$ geändert. Es sei ausdrücklich darauf hingewiesen, dass das Minimum der Neuronenzähler Null ist, was sich unmittelbar im Falle $c_i - r < 0$ einstellt. Das Maximum hingegen besitzt zwei unterschiedliche Werte, die sich in einen stabilen und einen instabilen aufspalten. Der instabile Maximalwert der Neuronenzähler beträgt $\Theta + 1$ und der stabile Θ[5]. Dieser metastabile Zustand ist in Gleichung 3.26 durch einen Stern (*) gekennzeichnet. Falls sich nun irgendein Wert $c_i - r \not< 0$ einstellt, so geschieht erst einmal mit den Neuronenzählern nichts weiter. Stattdessen wird jeder aktiven Synapse durch die Rangordnungsverteilung 3.28 eine Wahrscheinlichkeit $p^{\text{rank}}_{\widetilde{c_{ij}}}$ zugeordnet, die sich aus den approximierten Synapsenzählern $\widetilde{c_{ij}}$ ergibt. Die Wahrscheinlichkeit $p^{\text{rank}}_{\widetilde{c_{ij}}}$ wird also nicht gezogen, sondern fest zugewiesen. Zusätzlich zu $p^{\text{rank}}_{\widetilde{c_{ij}}}$ wird für jede aktive Synapse eine Wahrscheinlichkeit p_{coin} gezogen. Falls nun die Bedingung $p_{\text{coin}} < p^{\text{rank}}_{\widetilde{c_{ij}}}$ erfüllt ist, so wird diese Synapse deprimiert, ansonsten nicht. Danach "fallen" die Neuronenzähler, falls dies nicht schon der Fall ist, in ihren stabilen Zustand. Dies bedeutet, dass aus $c_i - r > \Theta \implies c_i = \Theta$, wie in Gleichung 3.26 angegeben, folgt. Danach beginnt solch ein Zyklus, durch die Präsentation eines neuen Musters, erneut.

Der Zusammenhang zwischen der Extremalen Optimierung und meiner Lernregel mit Neuronenzähler besteht nun darin, dass die approximierten Synapsenzähler vom maximalen zum minimalen Wert angeordnet werden und ihnen dann durch ein Potenzgesetz in Abhängigkeit vom Exponenten τ, eine Wahrscheinlichkeit $p^{\text{rank}}_{\widetilde{c_{ij}}}$ zugeordnet wird. Je größer der approximierten Synapsenzähler ist, desto häufiger war er an einer falschen Berechnung beteiligt und umgekehrt. Deshalb ist es sinnvoll, den höheren approximierten Synapsenzählern eine größere Wahrscheinlichkeit $p^{\text{rank}}_{\widetilde{c_{ij}}}$ zuzuordnen, als den niedrigeren.
Der Unterschied besteht jedoch darin, dass die Bedingung $p_{\text{coin}} < p^{\text{rank}}_{\widetilde{c_{ij}}}$ eine probabilistische Änderung der Synapsen zur Folge hat, wohingegen bei EO auf jeden Fall in jedem Zeitschritt eine Änderung der elementaren Fitnessfunktionen erfolgt. Es handelt sich also um eine Uminterpretation der EO Prinzipien.

[5]Dies wurde so gewählt, da dies identisch in [Klemm et al., 2000] verwendet wird und ich aus Gründen der Vergleichbarkeit nur die Bestandteile des Modells an den Stellen abändere, die für meinen Neuronenzähler absolut notwendig sind. In Kapitel 5. wird diese Dynamik für die Neuronenzähler vereinfacht.

Zum besseren Verständnis sei noch kurz eine technische Anmerkung zum Definitionsbereich der Rangordnungsverteilung P_k^{rank} gemacht. Da die Neuronenzähler c_i die Werte von 0 bis einschließlich $\Theta+1$ (der instabil ist) annehmen können, kann k, welches die ranggeordneten Werte der approximierten Synapsenzähler c_{ij} repräsentiert, die Werte von 1 bis $2\Theta + 3$ annehmen, wovon man sich durch explizites Niederschreiben der Werte für c_{ij} leicht überzeugt.

Der Grund, warum für die (Münz-)Verteilung $P(x)_{coin}$ gleichfalls ein Potenzgesetz gewählt wurde, besteht darin, dass man für $\alpha = 0$ ohnehin eine Gleichverteilung erhält, welches sicherlich die konventionelle Wahl darstellt, diese jedoch nicht notwendigerweise die besten Resultate im Lernverhalten erzielen muss. Durch diesen zusätzlichen Parameter sind wir also in der Lage, den Einfluss der Verteilung $P(x)_{coin}$ auf das Lernverhalten untersuchen zu können, ohne eine Gleichverteilung von vornherein auszuschließen.

3.3.6 Ergebnisse

In diesem Abschnitt wird das in Kapitel 3.3.4. vorgestellte Modell untersucht und mit den Resultate von [Klemm et al., 2000], welches in Kapitel 3.2.4. vorgestellt wurden, verglichen. Da die in 3.3 definierte Lernregel wesentlich von drei Parametern, nämlich der Gedächtnislänge Θ, dem Exponenten der Rangordnungsverteilung τ und dem Exponenten der (Münz-)Verteilung α abhängt, wird deren Einfluss auf die Lerndynamik des Gesamtsystems untersucht.
Um den Einfluss der verwendeten Netzwerkdynamik[6] zu studieren, wird in den folgenden drei Abschnitten 3.3.6.1 bis 3.3.6.3. erst ein winner-take-all, dann softmax und zum Schluss ein verrauschter winner-take-all Mechanismus für die Netzwerkdynamik verwendet.

3.3.6.1 winner-take-all Mechanismus

Als erstes untersuche ich das Modell für die in 3.2 definierte Netzwerkdynamik für den Fall $\beta \to \infty$, der dem deterministischen winner-take-all Mechanismus entspricht.
Für dieses Modell wurde die Parameterabhängigkeit von Θ untersucht, das die Werte von 0 bis 5 annehmen konnte. Die Abhängigkeit von den Exponenten τ und α wurden im Bereich

$$0 \leq \alpha, \tau \leq 3 \tag{3.34}$$

untersucht. Hierbei wurde eine Schrittweite von 10^{-1} für beide Parameter gewählt.
In den Abbildungen 3.6 bis 3.11 sind die Ergebnisse der Simulation dargestellt.

[6] Hier ist nur die Dynamik der neuronalen Aktivität gemeint.

Hierbei entspricht jeder Punkt der Abbildung einer Simulation, bestehend aus einem Ensemble der Größe $N = 10000$. Die drei Bilder jeder Abbildungen zeigen den mittleren Fehler 3.9 als Funktion des Gedächtnisses Θ zu unterschiedlichen Zeitpunkten. Der mittlere Fehler E wurden nach $t = 500, 1000$ und 1500 Zeitschritten angegeben, da diese die signifikanten Zeiten in Abbildung 3.1 und 3.4 sind.
Man erkennt, dass es Parameterbereiche in dieser durch τ und α aufgespannten zweidimensionalen Ebene gibt, welche sich deutlich in Bezug auf eine wesentlich schnellere Lernzeit von anderen Bereichen abheben. In Tabelle 3.1 sind die Exponenten α und τ angegeben, welche zu den drei Zeitpunkten zu dem niedrigsten mittleren Fehler E_{min} führen. Aus den Abbildungen 3.6 bis 3.11 er-

$t = 500$	Θ	τ	α	E_{min}
	0	2.0	0.4	0.110
	1	2.1	0.6	0.099
	2	1.8	0.9	0.113
	3	2.4	1.7	0.122
	4	2.2	2.8	0.112
	5	1.9	2.4	0.087
$t = 1000$	Θ	τ	α	E_{min}
	0	1.7	0.3	0.024
	1	2.0	0.6	0.021
	2	2.1	1.1	0.029
	3	2.3	1.7	0.032
	4	2.2	3.0	0.022
	5	1.9	3.0	0.018
$t = 1500$	Θ	τ	α	E_{min}
	0	1.4	0.3	0.005
	1	2.2*	0.8	0.004
	2	2.6	1.2	0.007
	3	2.3	1.7	0.009
	4	2.2	3.0	0.004
	5	1.9	3.0	0.003

Tabelle 3.1: Minimaler mittlerer Fehler E_{min} in Abhängigkeit der Exponenten α und τ und des Zählers Θ, welche sich aus den Abbildungen 3.6 bis 3.11 für die jeweiligen Zeitpunkte $t = 500, 1000$ und 1500 ergeben. Der (*) bedeutet, dass es in diesem Fall insgesamt drei Parameterpaare gibt, deren minimaler mittlerer Fehler gleich ist. Die beiden anderen lauten: $\tau = 2.7/3.0$, $\alpha = 0.9/1.0$

kennt man die stetige Abhängigkeit des mittleren Fehlers E von α und τ, denn in Gebieten um die in Tabelle 3.1 angegebenen Werte von α und τ,

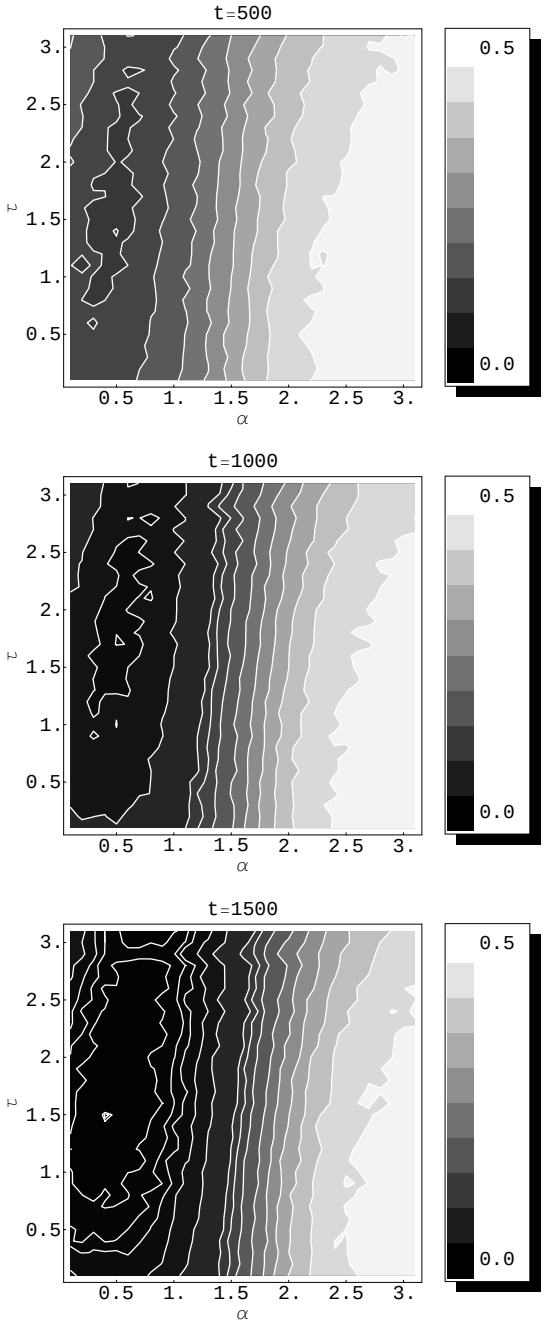


Abbildung 3.6: Mittlerer Fehler E in Abhängigkeit von τ und α zu den Zeitpunkten t=500, 1000 und 1500 für $\Theta = 0$.

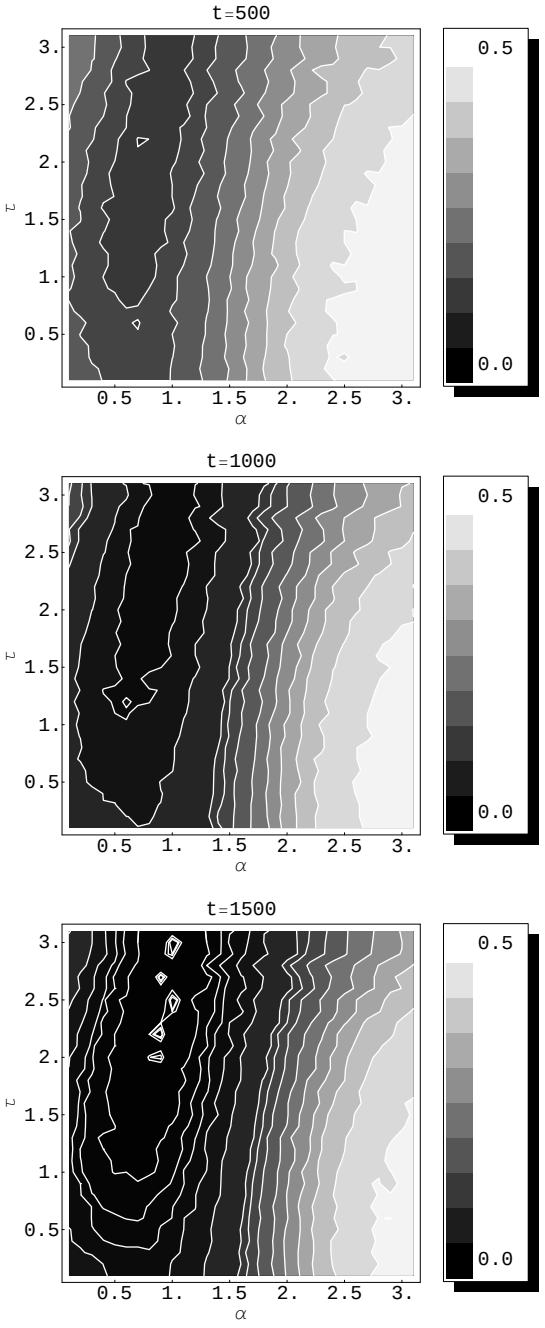
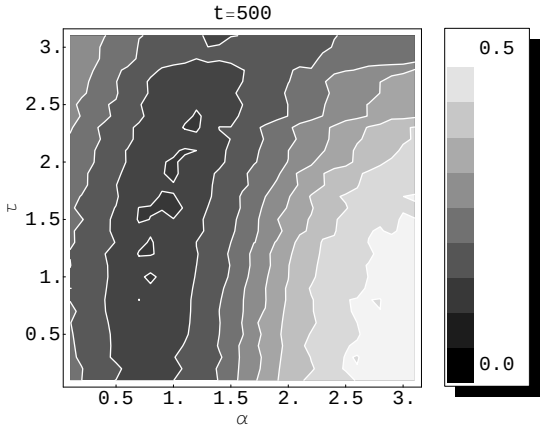
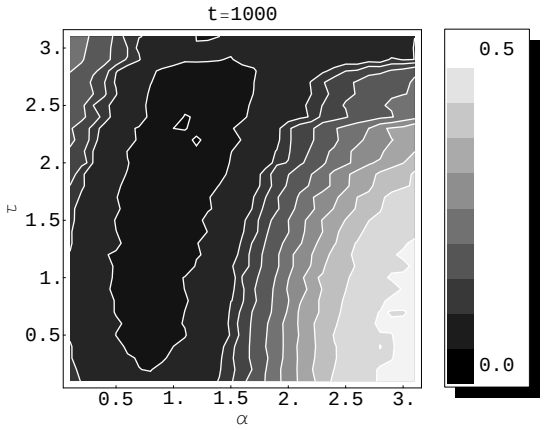


Abbildung 3.7: Mittlerer Fehler E in Abhängigkeit von τ und α zu den Zeitpunkten t=500, 1000 und 1500 für $\Theta = 1$.





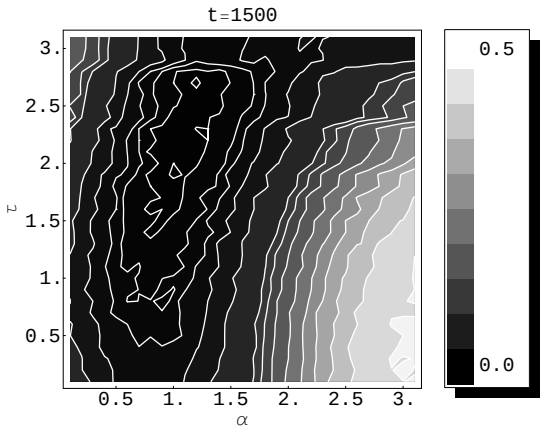
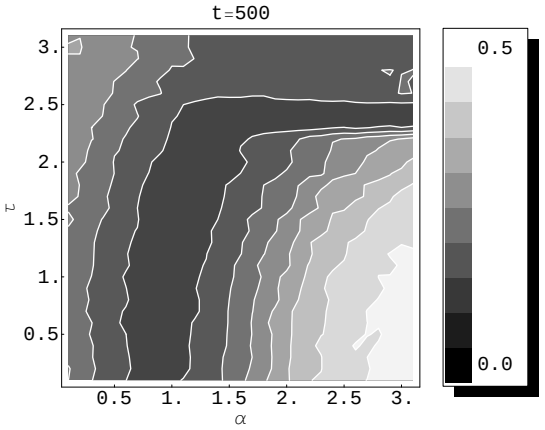
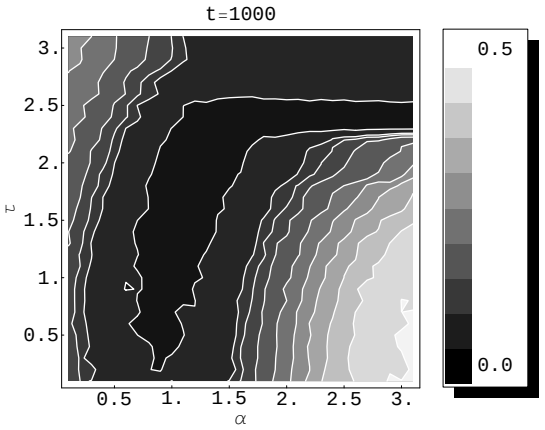


Abbildung 3.8: Mittlerer Fehler E in Abhängigkeit von τ und α zu den Zeitpunkten t=500, 1000 und 1500 für $\Theta = 2$.





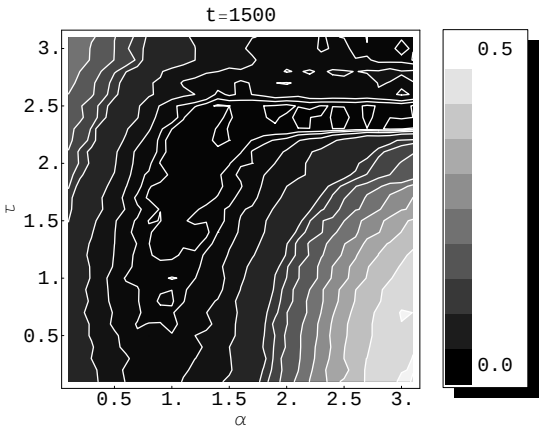
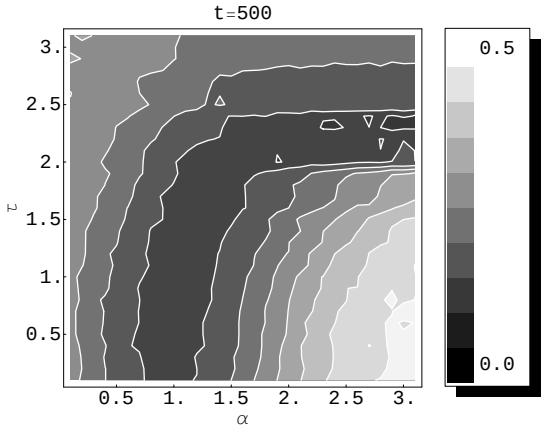
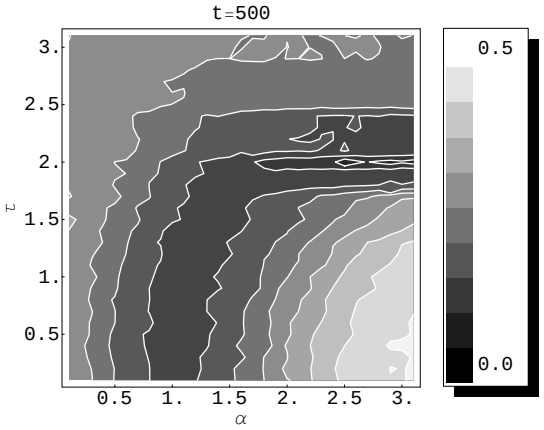
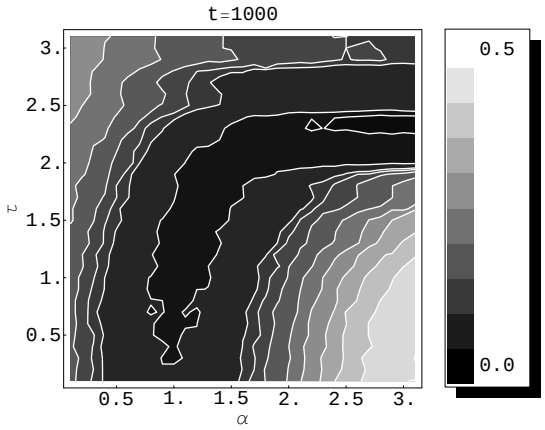
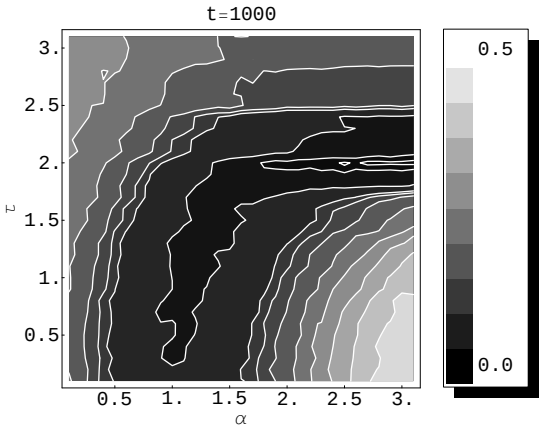


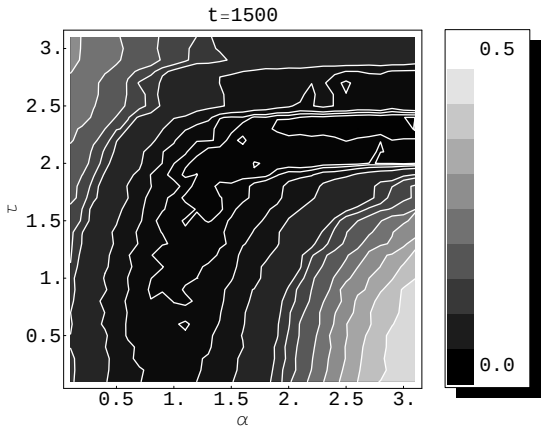
Abbildung 3.9: Mittlerer Fehler E in Abhängigkeit von τ und α zu den Zeitpunkten t=500, 1000 und 1500 für $\Theta = 3$.











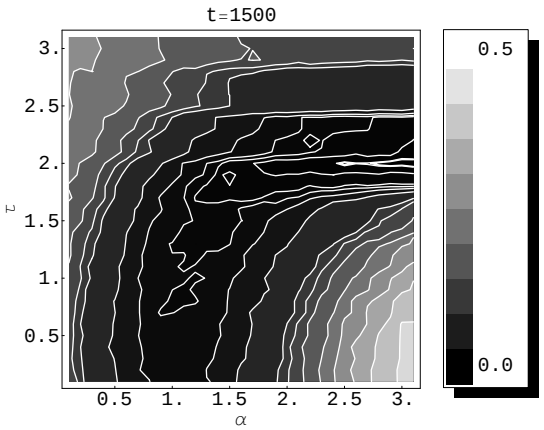


Abbildung 3.10: Mittlerer Fehler E in Abhängigkeit von τ und α zu den Zeitpunkten t=500, 1000 und 1500 für $\Theta = 4$.

Abbildung 3.11: Mittlerer Fehler E in Abhängigkeit von τ und α zu den Zeitpunkten t=500, 1000 und 1500 für $\Theta = 5$.

ist der mittlere Fehler von ähnlicher Größe. Diese Niveaulinien sind auch deutlich in den Abbildungen 3.6 bis 3.11 zu erkennen.
Für den Exponenten τ ergibt sich ein für alle Θ Werte nahezu konstanter bester Wert von ~ 2.0, wohingegen die Werte für α wesentlich stärker schwanken. Das Interessante daran ist, dass der optimale Wert für den Exponenten α für alle Θ Werte immer größer als Null ist, was eine Gleichverteilung ausschließt. Für zunehmende Θ Werte wächst der beste α Wert von ~ 0.4 bis ~ 3.0 an.
Dies kann man sich folgendermaßen klar machen: Wenn Θ erhöht wird, dann wird $p^{\text{rank}}_{\widetilde{c}_{ij}}$ kleiner[7]. Hieraus folgt aber eine Reduktion in der Häufigkeit der synaptischen Änderungen, da $p_{\text{coin}} < p^{\text{rank}}_{\widetilde{c}_{ij}}$ weniger oft erfüllt ist. Wenn nun α erhöht wird, dann wird p_{coin} im Mittel kleiner. Hieraus folgt aber, dass $p_{\text{coin}} < p^{\text{rank}}_{\widetilde{c}_{ij}}$ häufiger erfüllt wird, woraus sich eine Häufigkeitssteigerung in der synaptischen Änderung ergibt.

Durch die drei Parameter Θ, τ und α lässt sich indirekt die Wahrscheinlichkeit verändern, mit der eine synaptische Änderung erfolgt. Dies entspricht genau der von [Klemm et al., 2000] eingeführten Idee, nur dass die konkrete Realisierung auf einem vollständig anderen Mechanismus beruht; deterministisch vs. stochastisch.

In Abbildung 3.12 sind sechs Lernkurven des neuronalen Netzes mit Neuronengedächtnis für $\Theta = 0$ bis $\Theta = 5$ aufgetragen. Für die Exponenten τ und α wurden die Werte aus Tabelle 3.1 für $t = 1500$ gewählt. Zum Vergleich sind zwei Lernkurve aufgetragen, bei denen die Lernregel mit Synapsengedächtnis mit $\theta_{\text{KBS}} = 1$ und $\theta_{\text{KBS}} = 2$ verwendet wurde, die bereits in Abbildung 3.4 dargestellt wurden. Man erkennt, dass die Lernregel mit Synapsengedächtnis stark auf eine Veränderung von Θ reagiert, wohingegen die Lernregel mit Neuronengedächtnis für alle sechs Kurven annähernd die gleichen Ergebnisse liefert.
Hieraus folgt, dass für die Lernregel mit Neuronengedächtnis die Gedächtnislänge Θ seine dominierende Rolle verloren hat, die Θ für die Lernregel mit Synapsengedächtnis einnimmt. Vielmehr kommt es auf das kooperative Zusammenwirken der drei Systemparameter Θ, τ und α an, wodurch die Flexibilität der Lernregel bzgl. einer Änderungen in Θ, τ und α zum Ausdruck kommt. Darüberhinaus geht die anschauliche Interpretation des Synapsengedächtnisses in Bezug auf das Neuronengedächtnis verloren. Dies ist jedoch auch nicht so verwunderlich, wenn man sich daran erinnert, dass die Lernregel mit Synapsengedächtnis durch eine deterministische Abbildung festgelegt wird, wohingegen die Lernregel mit Neuronengedächtnis einen stochastischen Prozess in der Auswahl des Zeitpunktes einer synaptischen

[7] Hier sei τ als konstant angenommen.

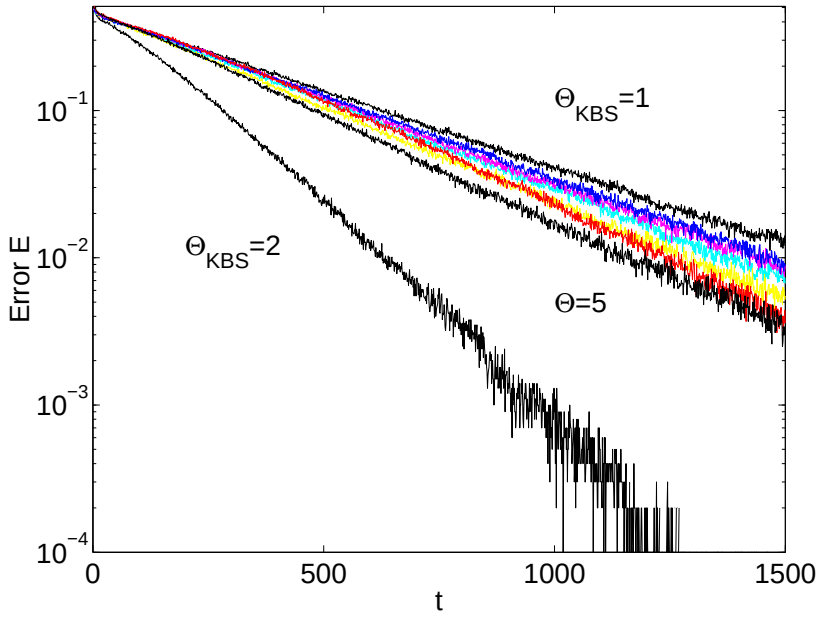


Abbildung 3.12: Vergleich des mittlerer Fehlers E in Abhängigkeit der Zeit t für verschiedene Werte von Θ. Die Bezeichnung "KBS" steht für die Lernregel mit Synapsenzähler. Für die Lernregel mit Neuronenzähler wurde E in Abhängigkeit von $\Theta = 0$ (zyan), $\Theta = 1$ (gelb), $\Theta = 2$ (mangan), $\Theta = 3$ (blau), $\Theta = 4$ (rot), $\Theta = 5$ (schwarz) untersucht. Die Exponenten τ und α sind für die jeweiligen Θ aus Tabelle 3.1 für $t = 1500$ zu entnehmen.

Änderung beschreibt.

3.3.6.2 softmax Mechanismus

Die Netzwerkdynamik des im folgenden untersuchten Modells wird durch einen softmax Mechanismus, mit $\beta = 10$ bestimmt. Die untersuchten Parameterbereiche und die Ensemblegröße, sind mit denen aus Kapitel 3.3.6.1. identisch
In den Abbildungen 3.13 bis 3.18 sind die Simulationsergebnisse für den softmax Mechanismus für $\beta = 10$ abgebildet. Berechnet wurde wieder der mittlerer Fehler E zu den Zeitpunkten $t = 500, 1000$ und 1500 in Abhängigkeit von α und τ.
Man erkennt, dass die mittleren Fehler E, zu allen Zeitpunkten $t = 500, 1000$ und 1500 deutlich größer sind als in 3.6 bis 3.11. Dies liegt an dem stochastischen Einfluss von β, welches zu einer Art verrauschter Signalübertragung in den einzelnen Schichten führt. In Tabelle 3.2 sind die minimalen mittleren Fehler E_{min} mit den zugehörigen Parametern angegeben, die aus den Abbildungen 3.13 bis 3.18 entnommen werden können. Man erkennt, dass die optimalen τ Werte diesmal nicht mehr für alle Θ Werte als ungefähr konstant angenommen

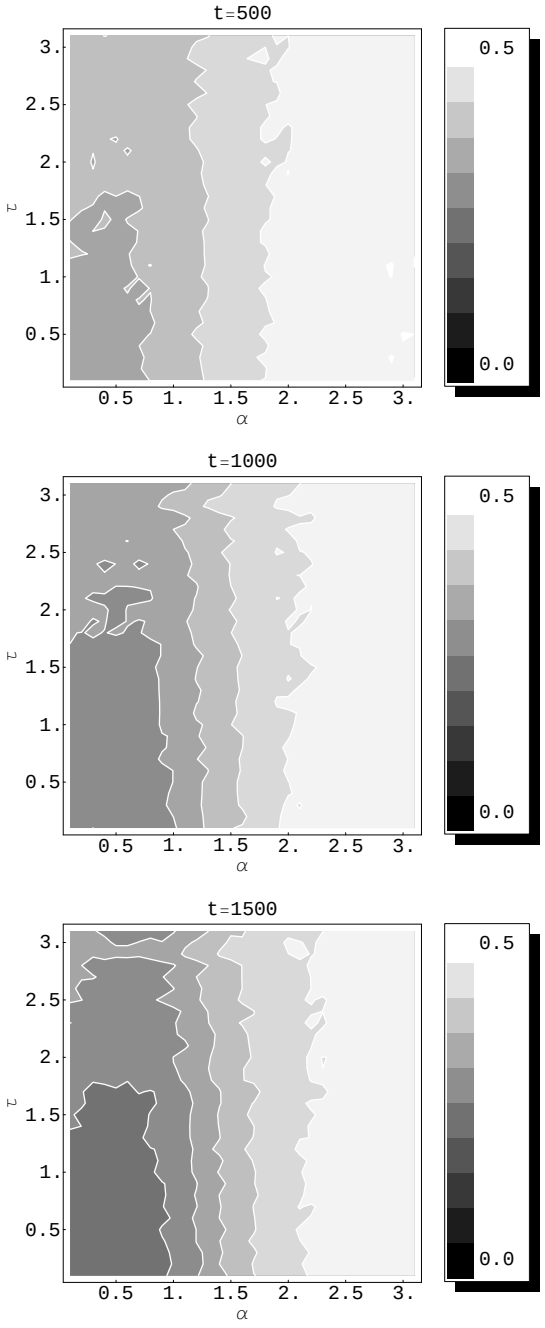


Abbildung 3.13: Mittlerer Fehler E in Abhängigkeit von τ und α zu den Zeitpunkten t=500, 1000 und 1500 für $\Theta = 0$.

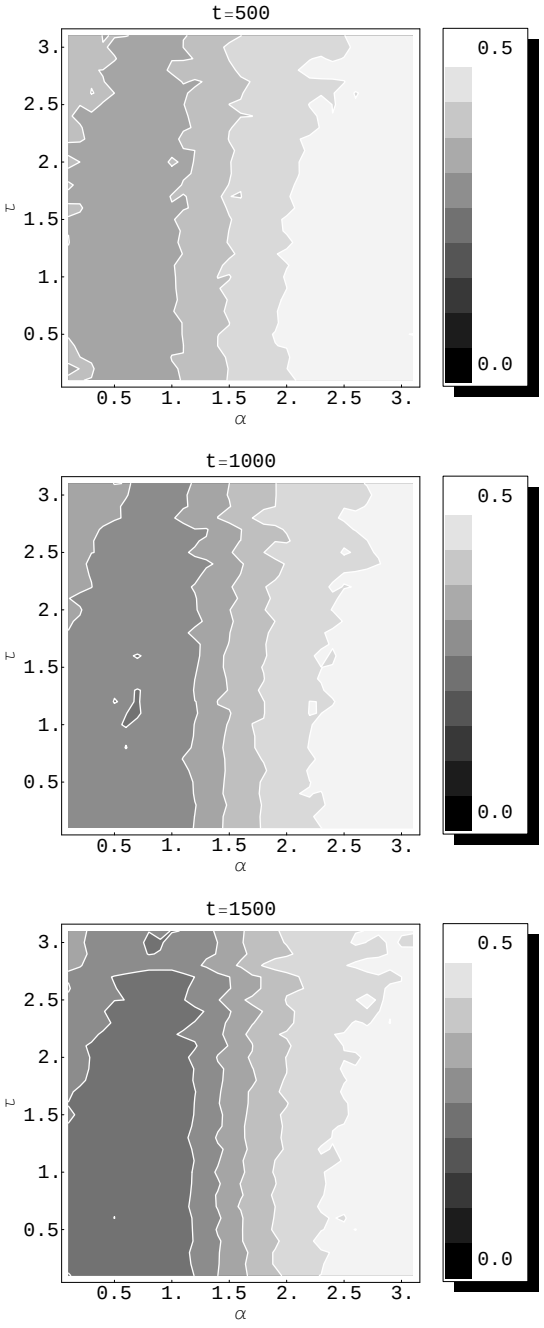
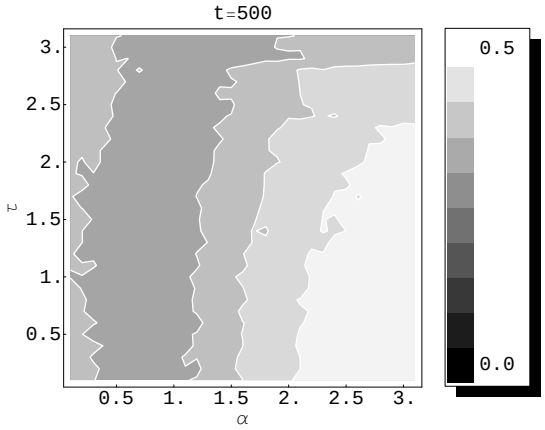
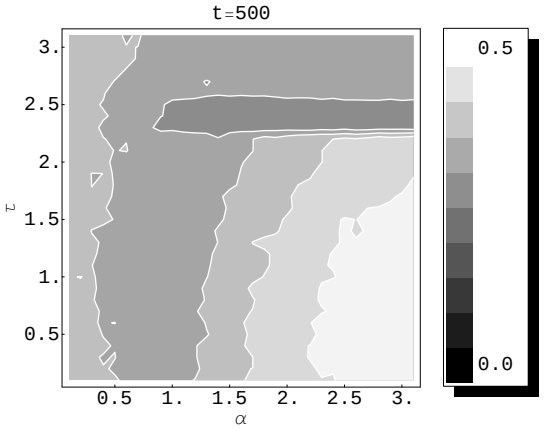
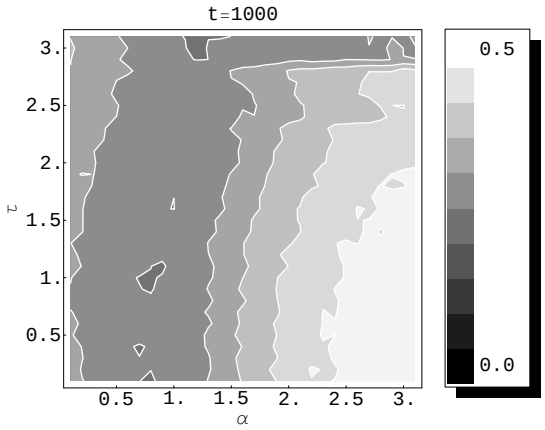
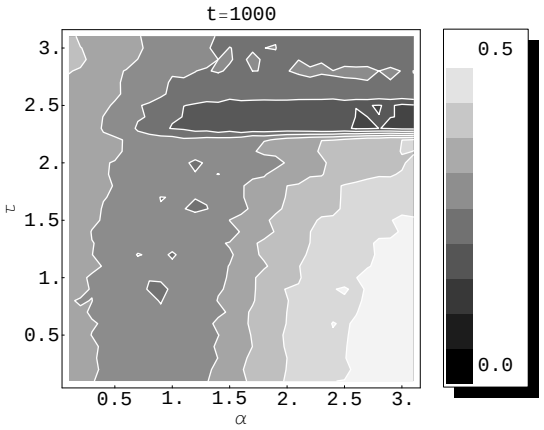


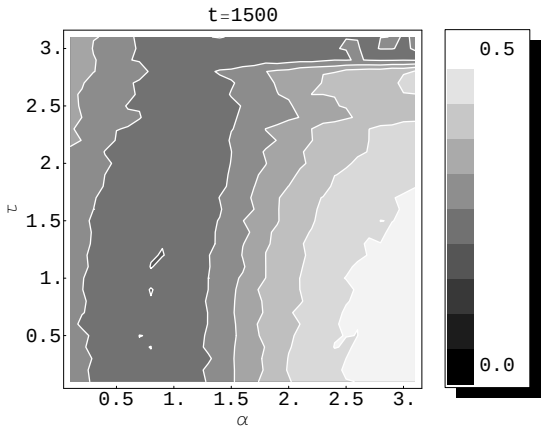
Abbildung 3.14: Mittlerer Fehler E in Abhängigkeit von τ und α zu den Zeitpunkten t=500, 1000 und 1500 für $\Theta = 1$.











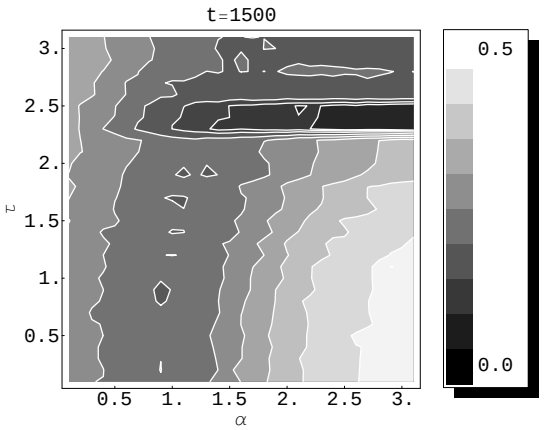


Abbildung 3.15: Mittlerer Fehler E in Abhängigkeit von τ und α zu den Zeitpunkten t=500, 1000 und 1500 für $\Theta = 2$.

Abbildung 3.16: Mittlerer Fehler E in Abhängigkeit von τ und α zu den Zeitpunkten t=500, 1000 und 1500 für $\Theta = 3$.

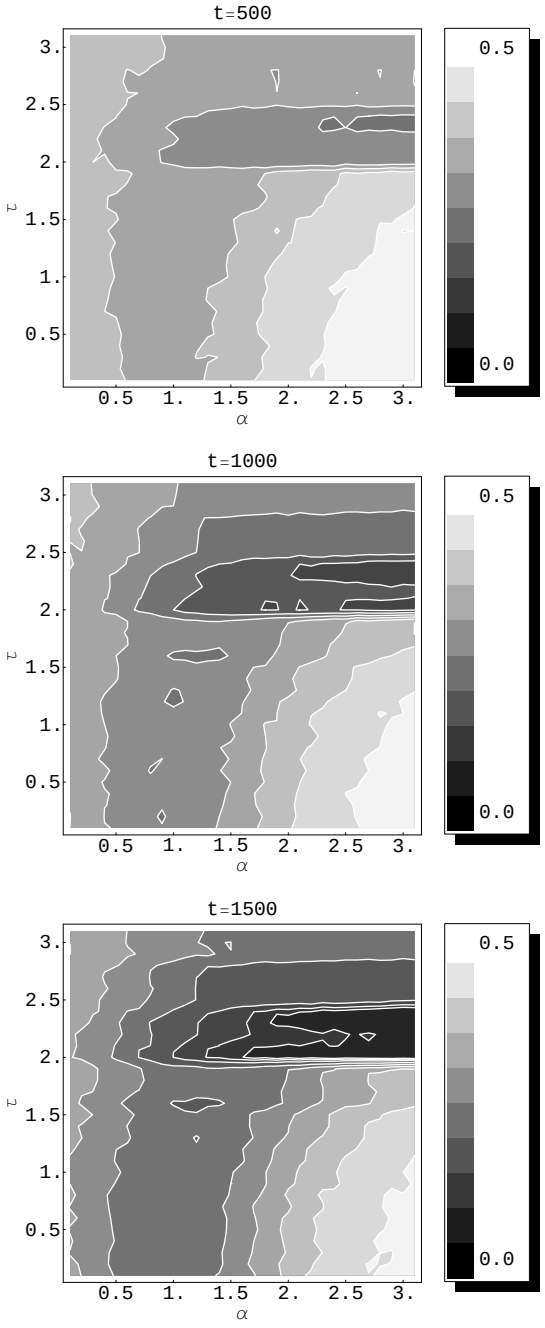


Abbildung 3.17: Mittlerer Fehler E in Abhängigkeit von τ und α zu den Zeitpunkten t=500, 1000 und 1500 für $\Theta = 4$.

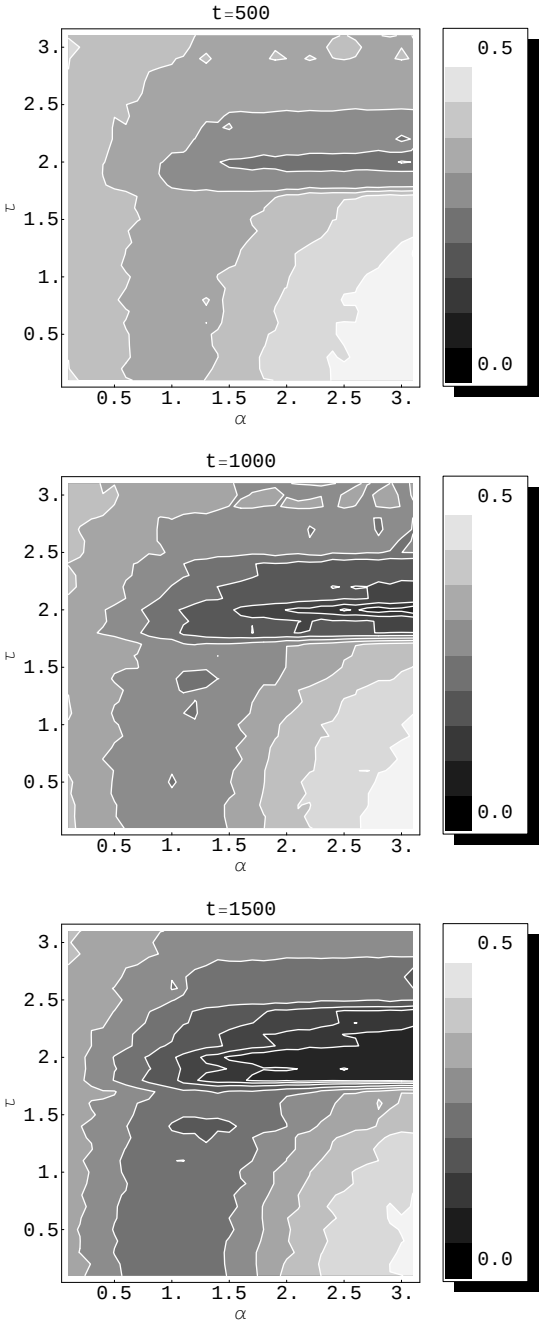


Abbildung 3.18: Mittlerer Fehler E in Abhängigkeit von τ und α zu den Zeitpunkten t=500, 1000 und 1500 für $\Theta = 5$.

$t = 500$	Θ	τ	α	E_{min}
	0	0.2	0.3	0.325
	1	1.8	0.5	0.314
	2	3.0	1.2	0.312
	3	2.4	2.0	0.264
	4	2.2	3.0	0.239
	5	1.9	2.9	0.199
$t = 1000$	Θ	τ	α	E_{min}
	0	0.0	0.2	0.250
	1	1.0	0.6	0.246
	2	3.0	1.1	0.237
	3	2.2	3.0	0.138
	4	2.2	2.9	0.121
	5	1.9	3.0	0.088
$t = 1500$	Θ	τ	α	E_{min}
	0	0.5	0.1	0.200
	1	0.5	0.4	0.200
	2	3.0	1.3	0.197
	3	2.2	3.0	0.075
	4	2.2	3.0	0.069
	5	1.9	2.7	0.055

Tabelle 3.2: Minimaler mittlerer Fehler E_{min} in Abhängigkeit der Exponenten α und τ und des Zählers Θ, welche sich aus den Abbildungen 3.13 bis 3.18 für die jeweiligen Zeitpunkte $t = 500, 1000$ und 1500 ergeben. Als Selektionsmechanismus wurde der softmax Mechanismus mit $\beta = 10$ verwendet.

werden können. Die optimalen Werte für α sind wiederum von Null verschieden, was eine Gleichverteilung für P_{coin} ausschließt, und nehmen wiederum mit zunehmenden Θ Werten zu.
Insgesamt sind die Ergebnisse zu denen für den winner-take-all Mechanismus recht ähnlich. Daraus folgt ein recht robustes Verhalten in der Lerndynamik des neuronalen Netzes, gegenüber dem Einfluss von Rauschen.
Es sei noch angemerkt, dass eine weitere Vergrößerung des Neuronenspeichers Θ keine weitere Verbesserung bzgl. des mittleren Fehlers ergibt (nicht gezeigte Simulationen).

In Abbildung 3.19 sind wieder sechs Lernkurven des neuronalen Netzes mit Neuronengedächtnis für $\Theta = 0$ bis $\Theta = 5$ aufgetragen. Für die Exponenten τ und α wurden die Werte aus Tabelle 3.2 für $t = 1500$ gewählt. Zum Vergleich sind zwei Lernkurve aufgetragen, bei denen die Lernregel mit Synapsengedächtnis mit $\theta_{KBS} = 1$ und $\theta_{KBS} = 2$ verwendet wurde, die bereits in

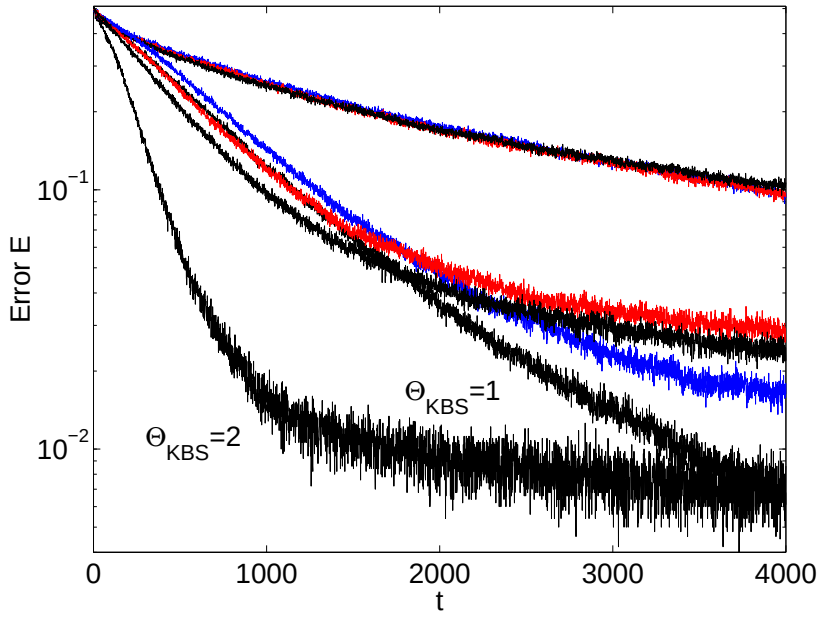


Abbildung 3.19: Vergleich des mittlerer Fehlers E in Abhängigkeit der Zeit t für verschiedene Werte von Θ. Die Bezeichnung "KBS" steht für die Lernregel mit Synapsenzähler. Für die Lernregel mit Neuronenzähler wurde E in Abhängigkeit von $\Theta = 0$ (blau), $\Theta = 1$ (rot), $\Theta = 2$ (schwarz) (obere Gruppe), $\Theta = 3$ (blau), $\Theta = 4$ (rot), $\Theta = 5$ (schwarz) (untere Gruppe) untersucht. Die Exponenten τ und α sind für die jeweiligen Θ aus Tabelle 3.2 für $t = 1500$ zu entnehmen. Bei allen Simulationen wurde ein softmax Mechanismus mit $\beta = 10$ verwendet.

Abbildung 3.1 dargestellt wurden. Man erkennt, dass die Lernregel mit Synapsengedächtnis wieder stark auf eine Veränderung von Θ reagiert. Im Gegensatz zu den Resultaten in Abbildung 3.12 gibt es nun für die Lernregel mit Neuronengedächtnis zwei Gruppen, innerhalb denen nur eine geringe Sensitivität gegenüber Änderungen in Θ zu erkennen ist, wohingegen ein Übergang zwischen den Gruppen zu einer starken Veränderung der Lernkurven führt. Hierbei besteht die eine Gruppe aus $\Theta = 0$ bis $\Theta = 2$ und die andere aus $\Theta = 3$ bis $\Theta = 5$.
Die Spaltung der Θ Werte in zwei Gruppen ist eine Folge von $\beta = 10$ und somit der Temperatur bzw. des Rauschens, denn je größer das Rauschen ist, desto länger muss eine Zeitreihe sein, um das Signal bestimmen zu können. Beim winner-take-all Mechanismus war $\beta = 0.0$, wofür schon $\Theta = 0$ in Kombination mit geeigneten Exponenten τ und α ausreichend war, um gute Konvergenzeigenschaften zu erzielen. Aus diesem Grund konnte man dort auch keine Schwelle in den Θ Werten beobachten, die quantitativ auf große Unterschiede im Lernverhalten des neuronalen Netzes hinweisen würde. Für $\beta > 0$ verschiebt sich die Schwelle aus verständlichen Gründen zu $\Theta > 0$ hin, wo-

durch eine längere Mittelung über die Inputmuster erfolgen kann bevor eine synaptische Änderung stattfindet.

3.3.6.3 verrauschter winner-take-all Mechanismus

In diesem Abschnitt untersuche ich die Arbeitsweise des Modells mit Neuronengedächtnis für einen verrauschten winner-take-all Mechanismus, wie in [Chialvo und Bak, 1999; Bak und Chialvo, 2001] verwendet. Das bedeutet, die Netzwerkdynamik wird nicht mehr durch 3.1 und 3.2 bestimmt, wie in den vorangegangenen beiden Kapiteln, sondern durch

$$\overline{h_j} = h_j + \eta_j \tag{3.35}$$

$$i_{max} = \underset{i}{\operatorname{argmax}}(\overline{h_j}) \tag{3.36}$$

Hierbei wird η_j gleichverteilt und unabhängig aus dem Intervall $[0, \eta]$ gezogen.
Die Gleichungen 3.35 und 3.36 sind so zu verstehen, dass man zu jedem inneren Feld h_j in einer Schicht des neuronalen Netzes, Rauschen η_j, welches aus einer Gleichverteilung aus dem Intervall $[0, \eta]$ gezogen wird, addiert und dann das Neuron auswählt, welches das größte innere Feld $\overline{h_j}$ besitzt. Dieses Neuron ist dann aktiv $\overline{h}_{i_{max}} = 1$, wohingegen alle anderen Neuronen passiv $\overline{h}_{j \neq i_{max}} = 0$ sind.

In diesem Kapitel sollen das Konvergenzverhalten des Modells mit Neuronengedächtnis unter dem Einfluss von Rauschen untersucht werden. Dieses Konvergenzverhalten soll nicht für kurze, sondern auf langen Zeitskalen untersucht werden. Bei den bisherigen Untersuchungen wurde die Frage behandelt, wie schnell ein neuronales Netz das XOR-Problem lernen kann, und die Simulationszeiten lagen im Bereich $T \sim 10^4$. Nun soll die Frage untersucht werden, ob ein neuronales Netz für ein bestimmtes Rauschen das XOR-Problem überhaupt asymptotisch lernen kann. Insofern stellten diese Untersuchungen das Analogon zu den Ergebnissen in Abbildung 3.3 dar.

Für alle durchgeführten Untersuchungen berechnet sich der mittlere Fehler E durch

$$E = \frac{1}{N(T_m + 1)} \sum_{i=1}^{N} \sum_{t=T-T_m}^{T} e_i(t) \tag{3.37}$$

$$e_i(t) \in \{0, 1\} \tag{3.38}$$

Die Ensemblegröße beträgt $N = 100$ und die Simulationszeit $T = 100000$, wobei über die letzten $T_m = 10000$ Zeitschritte gemittelt wird, da der mittlere Fehler bis dahin in einen stationären Zustand übergegangen ist.

In der ersten Simulationsreihe, welche in den Abbildungen 3.20 bis 3.25 dargestellt ist, wurde der mittlere Fehler E in Abhängigkeit des Rauschens η, des Exponenten τ und des Neuronengedächtnisses Θ untersucht. Der Exponent der (Münz-)Verteilung $P(x)_{coin}$ in 3.31 wird für alle Simulationen zu $\alpha = 1.5$ fest gewählt. Die Begründung hierfür stellen die Untersuchungen aus dem letzten Kapitel dar[8]. Aus den Abbildungen 3.13 bis 3.18 erkennt man, dass $\alpha = 1.5$ im Mittel recht gute Ergebnisse erzielt.
Für die linken Abbildungen wurden die Synapsen immer um $\delta = 1 = const.$ geändert, wohingegen für die rechten Abbildungen die Änderung der Synapse δ aus der Verteilung $[0, 20]$ gleichverteilt und unabhängig gezogen wurde. Solch ein stochastischer, synaptischer Update wurde in ähnlicher Form von [Chialvo und Bak, 1999; Bak und Chialvo, 2001] verwendet. Dieser zusätzliche Parameter kann nun ohne Probleme eingeführt werden, da die Temperatur nicht mehr konstant gehalten werden muss, um die Resultate mit der Arbeit von [Klemm et al., 2000] vergleichen zu können. Die genaue Bedeutung einer Temperatur in einem neuronalen Netz mit verrauschtem winner-take-all Mechanismus wird in Kapitel 4. untersucht.

Generell kann man aus den Abbildungen 3.20 bis 3.25 ablesen, dass $\eta \sim 1.0$ bzw. $\eta \sim 10.0$ eine Schwelle zwischen Lernen, $E \sim 0$ und Nicht-Lernen, $E \sim 0.5$ darstellt. Dies hängt natürlich damit zusammen, dass $\delta \sim O(1)$ im Fall $\delta = 1 = \text{const.}$ und $\delta \sim O(10)$ im Fall $\delta \neq \text{const.}$ verwendet wird.

$\delta = 1 = \text{const.}$:
Man erkennt, dass innerhalb jeder Abbildung kleinere Werte für τ entweder bessere oder zumindest gleich gute Ergebnisse liefern, verglichen mit großen τ Werten. Dies kann man sich dadurch erklären, dass für kleinere τ Werte, die Bedingung $p_{coin} < p_{\widetilde{c_{ij}}}^{rank}$ für den synaptischen Update weniger oft erfüllt wird wie für große τ Werte. Weiter sieht man, dass für festgehaltenes τ und größer werdende Θ, die Ergebnisse ebenfalls besser werden. Hierfür gilt die gleiche Begründung, wie man sich leicht klar macht.

$\delta \neq \text{const.}$:
Auch hier liefern innerhalb jeder Abbildung kleinere Werte für τ entweder bessere oder zumindest gleich gute Ergebnisse wie für große τ Werte. Hier wirkt sich jedoch eine Zunahme der Θ Werte kontraproduktiv auf den mittleren Fehler aus. Dies hängt mit der Größenordnung der synaptischen Änderungen zusammen. Da δ aus dem Intervall $[0, 20]$ gleichverteilt gezogen wird, ist die Wahrscheinlichkeit 50% einen Wert von δ zu ziehen, der größer

[8] Der softmax Mechanismus mit $\beta = 10$ stellt auch eine Art verrauschter Netzwerkdynamik dar.

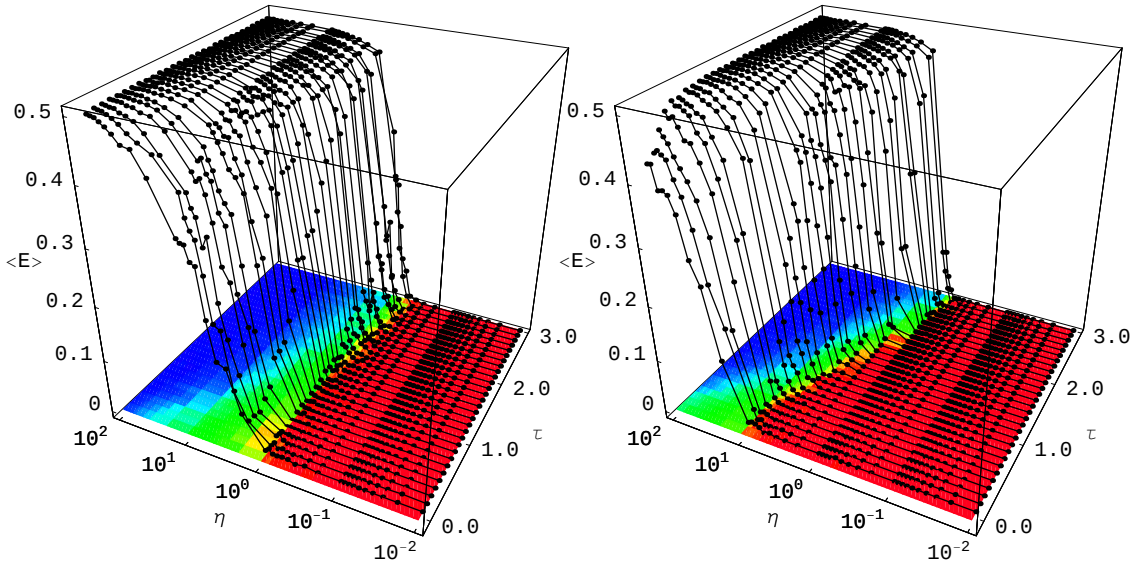


Abbildung 3.20: Mittlerer Fehler E in Abhängigkeit des Rauschens η und des Exponenten τ. Hierbei ist $\alpha = 1.5$ und $\Theta = 0$ fest gewählt. Links: $\delta = 1$. Rechts: $\delta \in [0, 20]$.

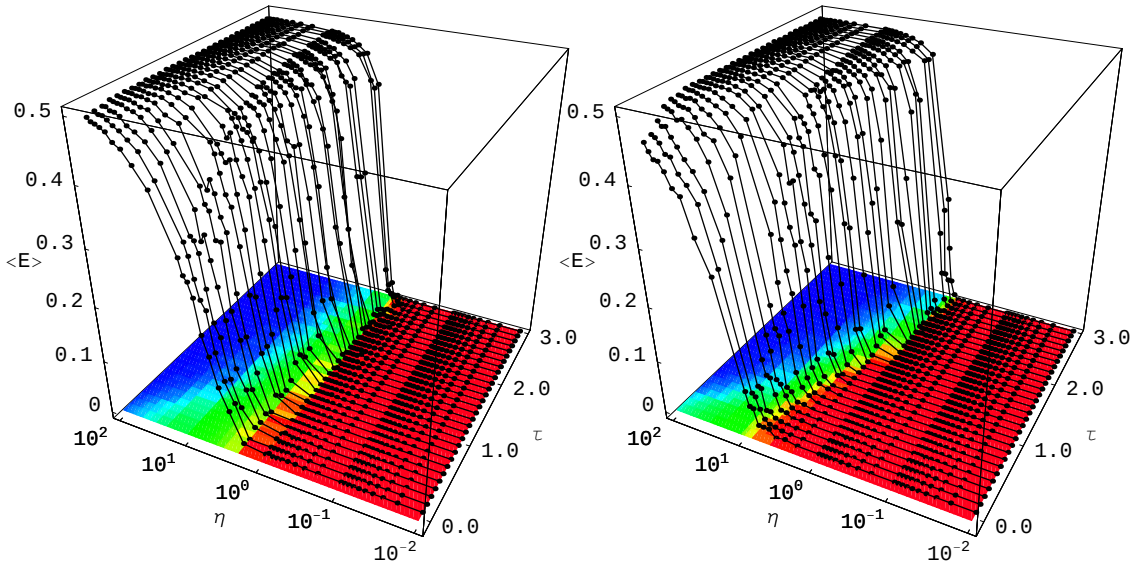


Abbildung 3.21: Mittlerer Fehler E in Abhängigkeit des Rauschens η und des Exponenten τ. Hierbei ist $\alpha = 1.5$ und $\Theta = 1$ fest gewählt. Links: $\delta = 1$. Rechts: $\delta \in [0, 20]$.

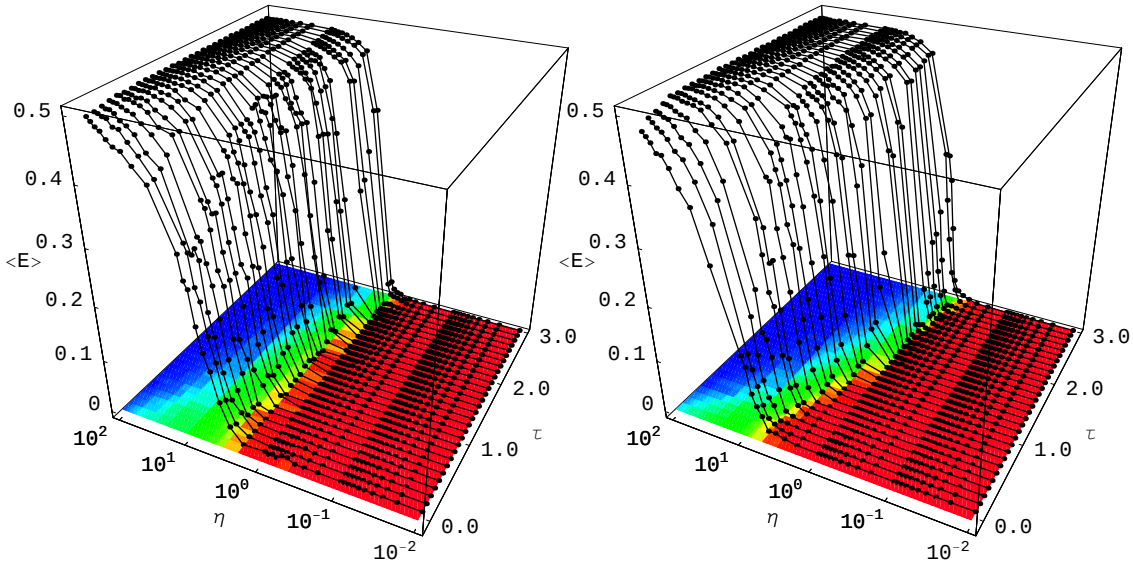


Abbildung 3.22: Mittlerer Fehler E in Abhängigkeit des Rauschens η und des Exponenten τ. Hierbei ist $\alpha = 1.5$ und $\Theta = 2$ fest gewählt. Links: $\delta = 1$. Rechts: $\delta \in [0, 20]$.

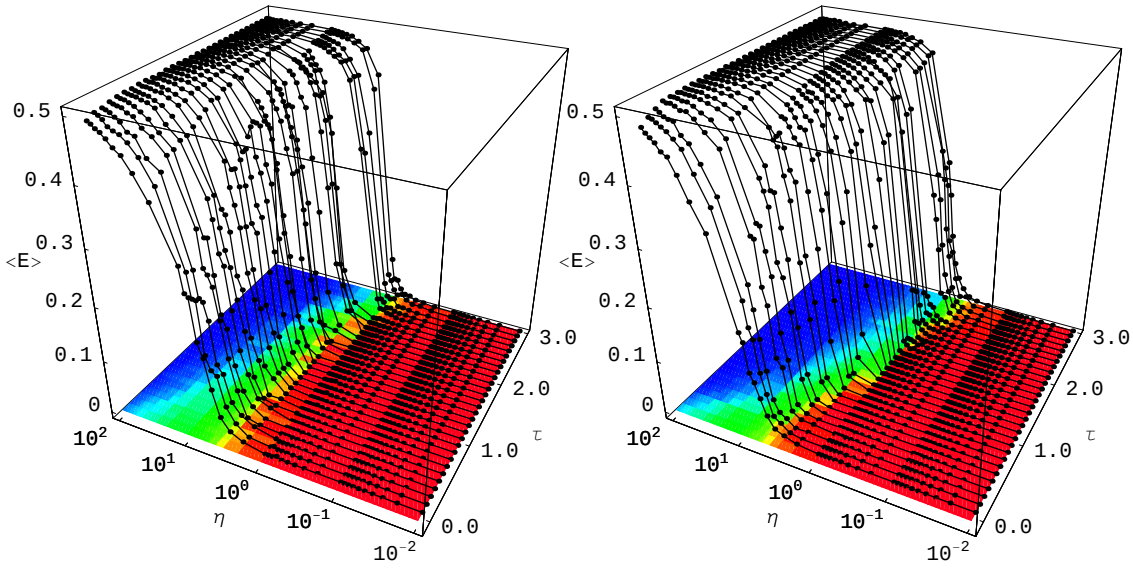


Abbildung 3.23: Mittlerer Fehler E in Abhängigkeit des Rauschens η und des Exponenten τ. Hierbei ist $\alpha = 1.5$ und $\Theta = 3$ fest gewählt. Links: $\delta = 1$. Rechts: $\delta \in [0, 20]$.

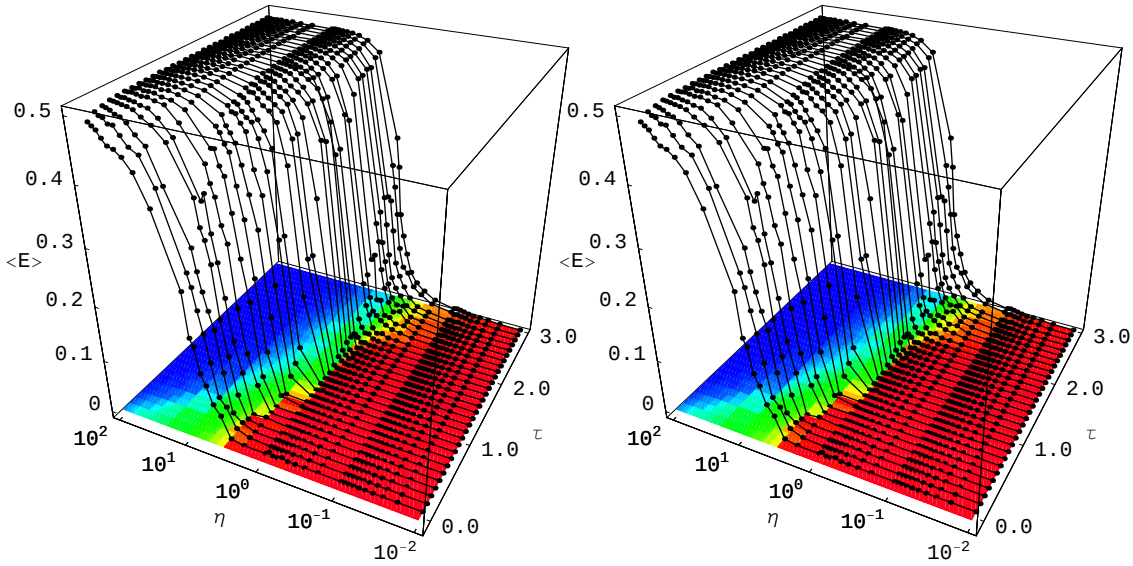


Abbildung 3.24: Mittlerer Fehler E in Abhängigkeit des Rauschens η und des Exponenten τ. Hierbei ist $\alpha = 1.5$ und $\Theta = 4$ fest gewählt. Links: $\delta = 1$. Rechts: $\delta \in [0, 20]$.

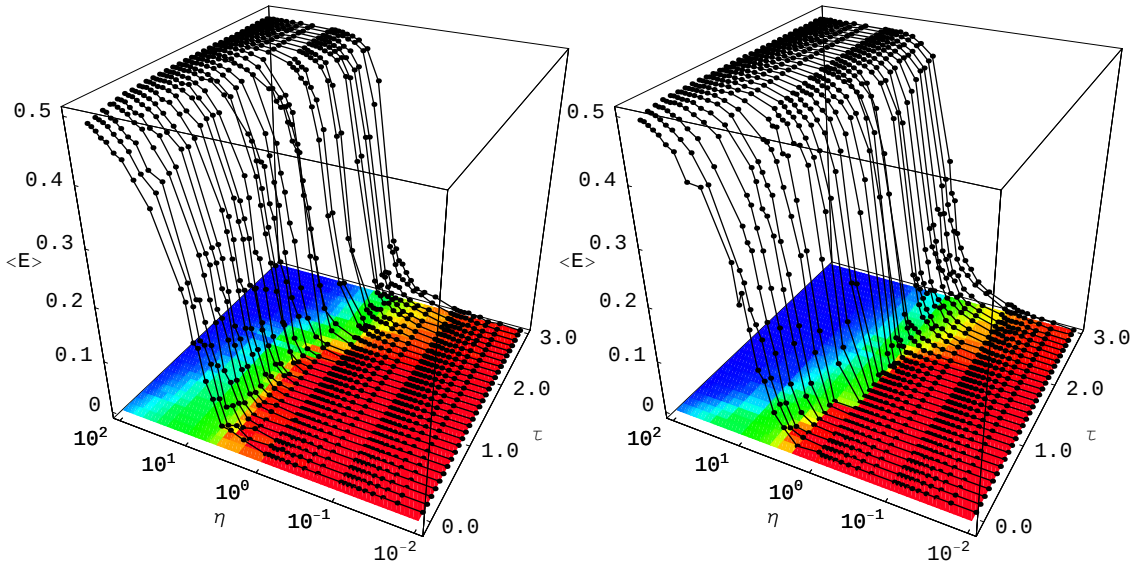


Abbildung 3.25: Mittlerer Fehler E in Abhängigkeit des Rauschens η und des Exponenten τ. Hierbei ist $\alpha = 1.5$ und $\Theta = 5$ fest gewählt. Links: $\delta = 1$. Rechts: $\delta \in [0, 20]$.

gleich 10 ist. Da nun mit zunehmendem Θ die Wahrscheinlichkeit für einen synaptischen Update zunimmt, nimmt infolge der großen δ Werte auch die Wahrscheinlichkeit zu, bereits bestehende und richtig gelernte neuronale Pfade im Netz zu zerstören.

$\Theta = \text{const.}$:
Der Vergleich von Abbildungen mit gleichen Θ Werten ergibt für $\Theta \in \{0, 1, 2\}$ bessere Resultate für $\delta \neq \text{const.}$. Für weiter zunehmende Θ Werten drehen sich die Verhältnisse langsam um, so dass für $\Theta \in \{3, 4, 5\}$ dann $\delta = 1 = \text{const.}$ bessere Resultate liefert. Die Begründung liegt wieder in einer, mit zunehmenden Θ Werten steigenden Wahrscheinlichkeit für die Zerstörung, von bereits richtig gelernten Pfaden.

Insgesamt kann man festhalten, dass die durchgeführten Simulationen solide Ergebnisse liefern. Insbesondere sind keine Totalausfälle in bestimmten Parameterbereichen zu finden, was auf eine Robustheit in den Parametern Θ, τ, α und sogar in der Größe der synaptischen Änderung δ impliziert.

In der zweiten Simulationsreihe, welche in den Abbildungen 3.26 bis 3.31 dargestellt ist, wurde der mittlere Fehler E in Abhängigkeit des Rauschens η, des Exponenten α und des Neuronengedächtnisses Θ untersucht. Der Exponent der Rangordnungsverteilung 3.28 wird für alle Simulationen zu $\tau = 2.0$ fest gewählt. Die Begründung hierfür stellen wieder die Untersuchungen aus der letzten Sektion dar, da man aus den Abbildungen 3.13 bis 3.18 erkennt, dass $\tau = 2.0$ im Mittel recht gute Ergebnisse erzielt.
Für die linken Abbildungen wurden die Synapsen immer um $\delta = 1 = \text{const.}$ geändert, wohingegen für die rechte Abbildung die Änderung der Synapse δ aus der Verteilung $[0, 20]$ gleichverteilt gezogen wurde.

$\delta = 1 = \text{const.}$:
Aus den Abbildungen 3.26 bis 3.31 liest man ab, dass für kleine Θ, kleine α Werte bessere Ergebnisse erzielen, und für große Θ, große α Werte. Dies liegt daran, dass für kleine Θ die Wahrscheinlichkeit $p_{\widetilde{c}_{ij}}^{\text{rank}}$ im Mittel größer ist, wie für große Θ. Wenn man nun für kleine Θ, den Exponenten α verkleinert, dann wird die Bedingung $p_{\text{coin}} < p_{\widetilde{c}_{ij}}^{\text{rank}}$ für den synaptischen Update, weniger häufig erfüllt, wie für größere α. Dies entspricht aber einer längeren Mittelung über das verrauschte Signal.
Wenn man bei großen Θ Werten α erhöht, dann verringert sich die synaptische Updatewahrscheinlichkeit ebenfalls, was den gleichen Vorteil bringt.

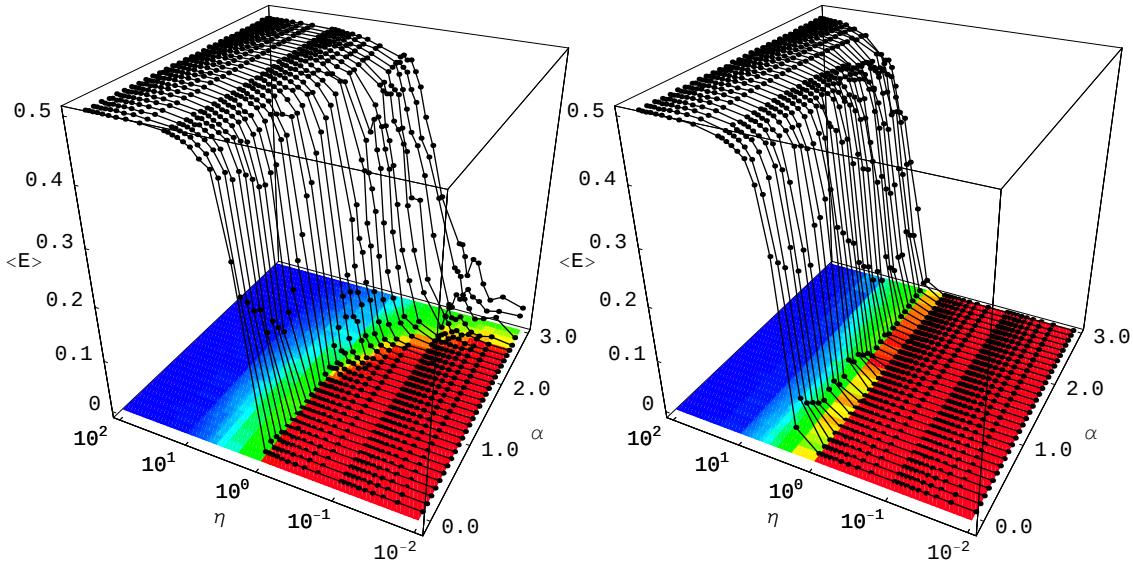


Abbildung 3.26: Mittlerer Fehler E in Abhängigkeit des Rauschens η und des Exponenten α. Hierbei ist $\tau = 2.0$ und $\Theta = 0$ fest gewählt. Links: $\delta = 1$. Rechts: $\delta \in [0, 20]$.

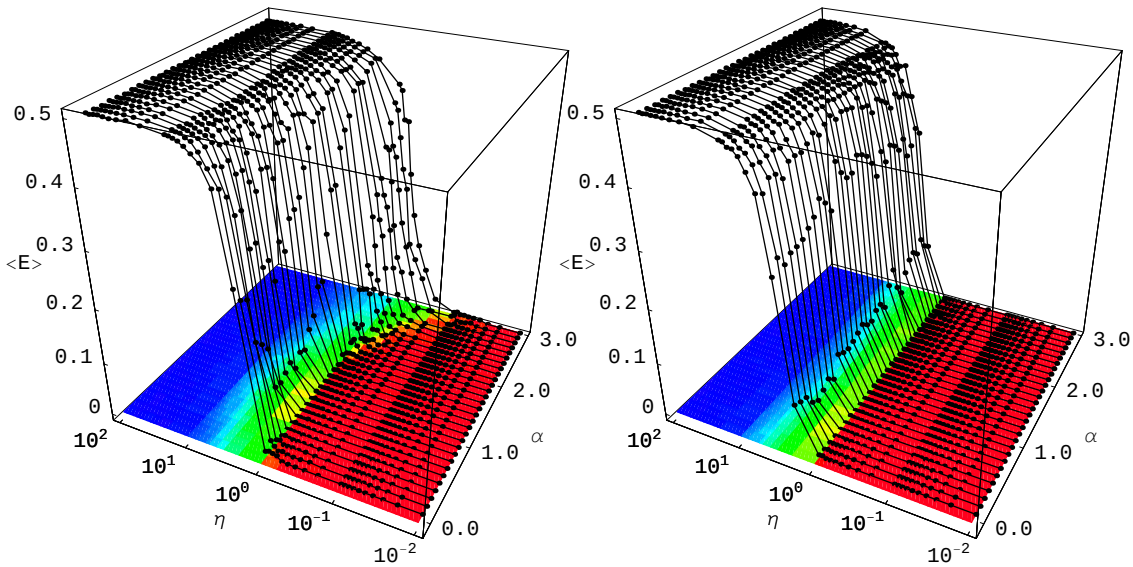


Abbildung 3.27: Mittlerer Fehler E in Abhängigkeit des Rauschens η und des Exponenten α. Hierbei ist $\tau = 2.0$ und $\Theta = 1$ fest gewählt. Links: $\delta = 1$. Rechts: $\delta \in [0, 20]$.

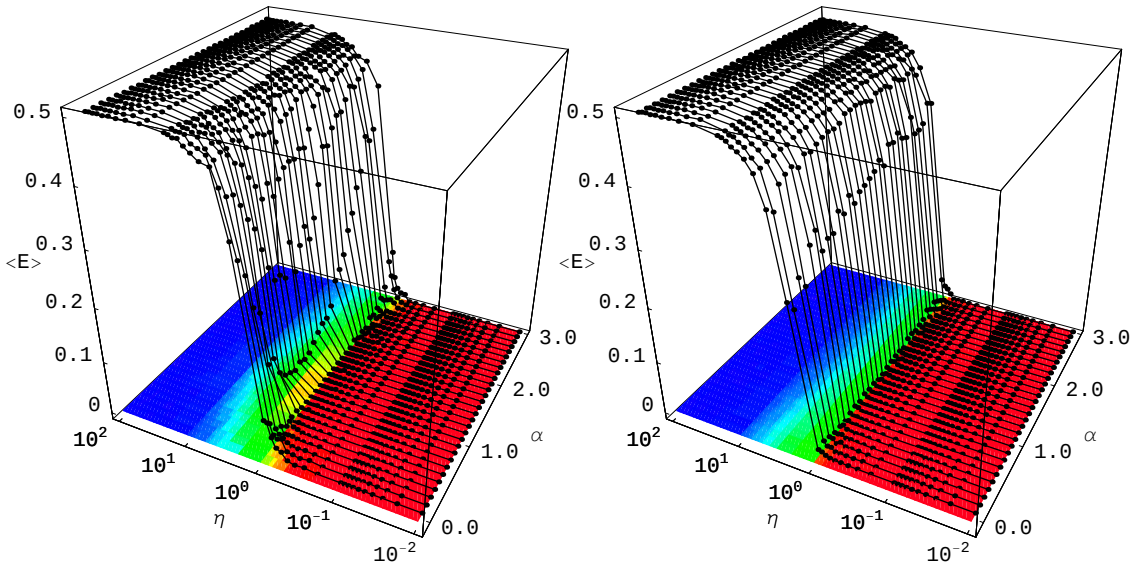


Abbildung 3.28: Mittlerer Fehler E in Abhängigkeit des Rauschens η und des Exponenten α. Hierbei ist $\tau = 2.0$ und $\Theta = 2$ fest gewählt. Links: $\delta = 1$. Rechts: $\delta \in [0, 20]$.

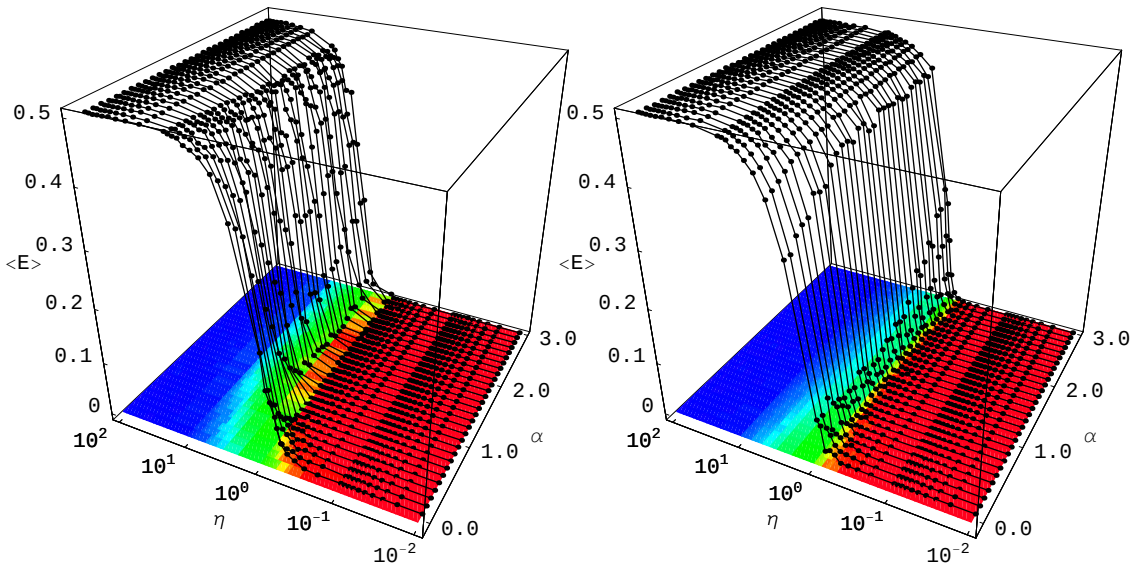


Abbildung 3.29: Mittlerer Fehler E in Abhängigkeit des Rauschens η und des Exponenten α. Hierbei ist $\tau = 2.0$ und $\Theta = 3$ fest gewählt. Links: $\delta = 1$. Rechts: $\delta \in [0, 20]$.

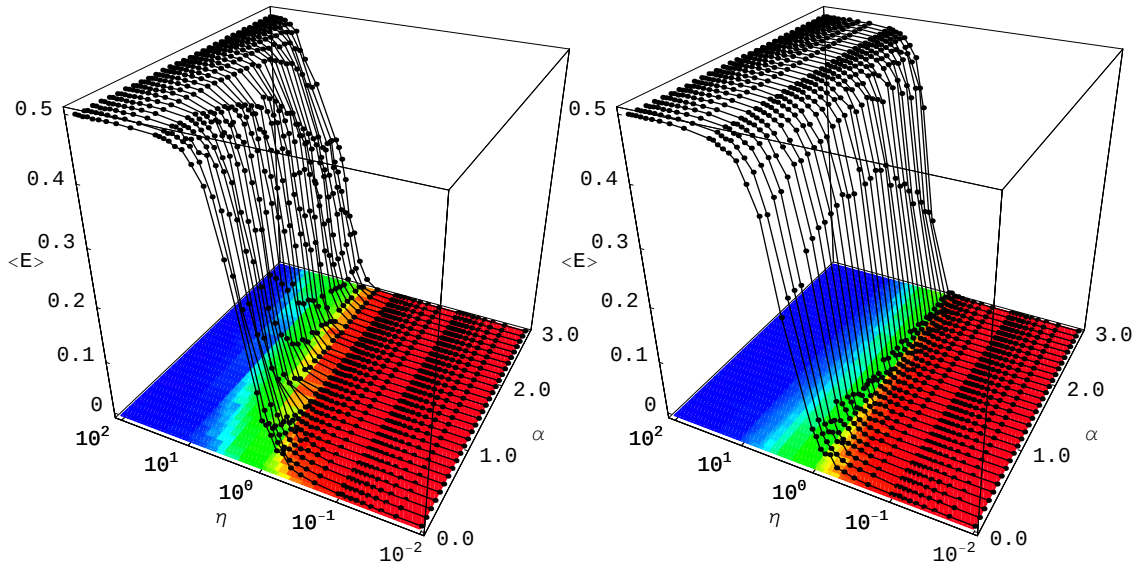


Abbildung 3.30: Mittlerer Fehler E in Abhängigkeit des Rauschens η und des Exponenten α. Hierbei ist $\tau = 2.0$ und $\Theta = 4$ fest gewählt. Links: $\delta = 1$. Rechts: $\delta \in [0, 20]$.

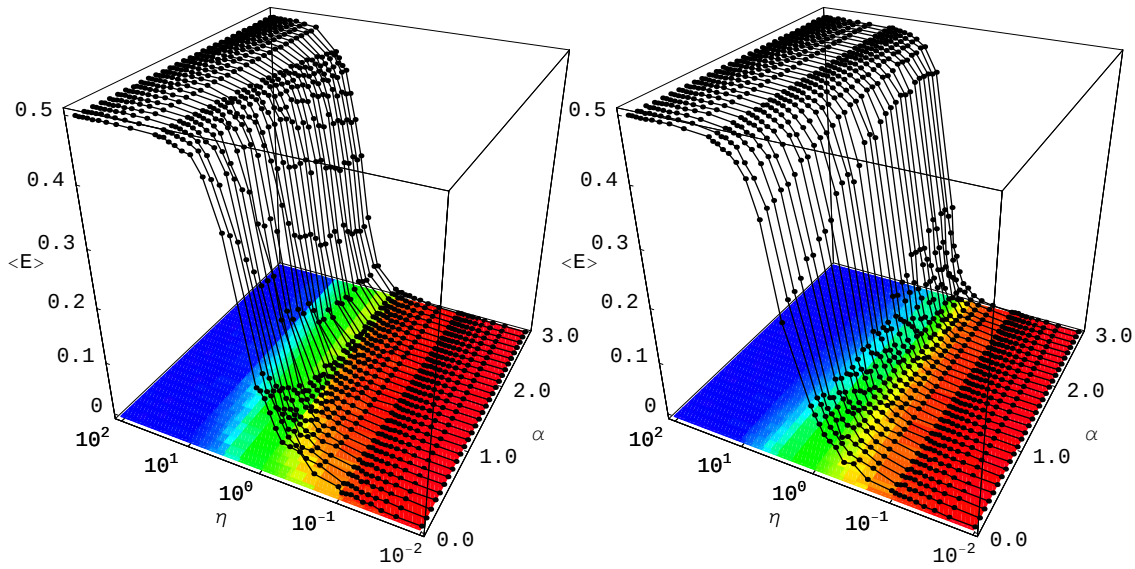


Abbildung 3.31: Mittlerer Fehler E in Abhängigkeit des Rauschens η und des Exponenten α. Hierbei ist $\tau = 2.0$ und $\Theta = 5$ fest gewählt. Links: $\delta = 1$. Rechts: $\delta \in [0, 20]$.

$\delta \neq$ const.:
In diesem Fall sind die Resultate beinahe invariant unter Parameteränderungen in Θ und α. Dies liegt daran, dass die synaptischen Änderungen viel zu groß sind, so dass sie eventuell vorkommende Abhängigkeiten nahezu vollständig überdecken. Ein Indiz hierfür ist, dass die Schwelle zwischen Lernen und Nicht-Lernen kaum an $\eta \sim 10^1$ heran reicht, was aufgrund der Intervallgröße durchaus möglich sein müsste. Anders ausgedrückt, durch große η lassen sich starke Einflüsse des Exponenten α der (Münz-)Verteilung stark dämpfen.

$\Theta =$ const.:
Aus den Abbildungen 3.26 bis 3.31 liest man zwei unterschiedliche Bereiche ab, in denen sich beide Modelle ($\delta =$ const. vs. $\delta \neq$ const.) unterscheiden. Der eine liegt bei niedrigen Θ und hohen α Werten, der andere bei hohen Θ und niedrigen α Werten. Im ersten Bereich erhält man für $\delta \neq$ const. bessere Resultate, weil die zu niedrigen Mittelungszeiten[9] durch eine, in Relation zum Rauschen η, wesentlich größere synaptische Änderung kompensiert werden. Für den zweiten Bereich gilt die analoge Argumentation.

Insgesamt gilt auch hier wieder, dass die Simulationen sehr robust gegenüber einer Veränderung der Parameter sind und somit kein Fein-Tuning erforderlich ist, um asymptotisch eine gute Konvergenz zu erzielen.

3.3.7 Biologische Plausibilität

Die von mir vorgeschlagene Hebb-artige Lernregel mit Neuronenzähler basiert, da sie auf die Arbeiten von [Chialvo und Bak, 1999; Klemm et al., 2000] und [Bak und Chialvo, 2001] aufbaut, ebenfalls auf den in Kapitel 3.2.5. angegebenen und diskutierten Forderungen. Hierbei ist die dritte Forderung insofern zu modifizieren, als dass es ein globales Reinforcementsignal geben müsste, das auf einen Neuronen- und nicht auf einen Synapsenzähler, einwirkt.
Darüberhinaus ergeben sich durch den stochastischen Mechanismus, der für die Selektion der aktiven Synapsen verwendet wird, einige Konsequenzen, die über die in Kapitel 3.2.5. diskutieren Punkte hinausgehen.
In den letzten Jahren gibt es im Bezug auf Untersuchungen über synaptische Plastizität geradezu eine Flut neuer Resultate. Die Ausführungen in Kapitel 2.4. stellen nur die Spitze des Eisbergs dar. Weiterführende experimentelle Resultate weißen immer mehr darauf hin, dass es neben der homosynaptischen Plastizität, die nur auf die Synapse wirkt, die vom aktiven prä- und postsynaptischen Neuron eingegrenzt wird, und somit konform mit dem Postulat

[9] Siehe die Diskussion zu $\delta =$ const..

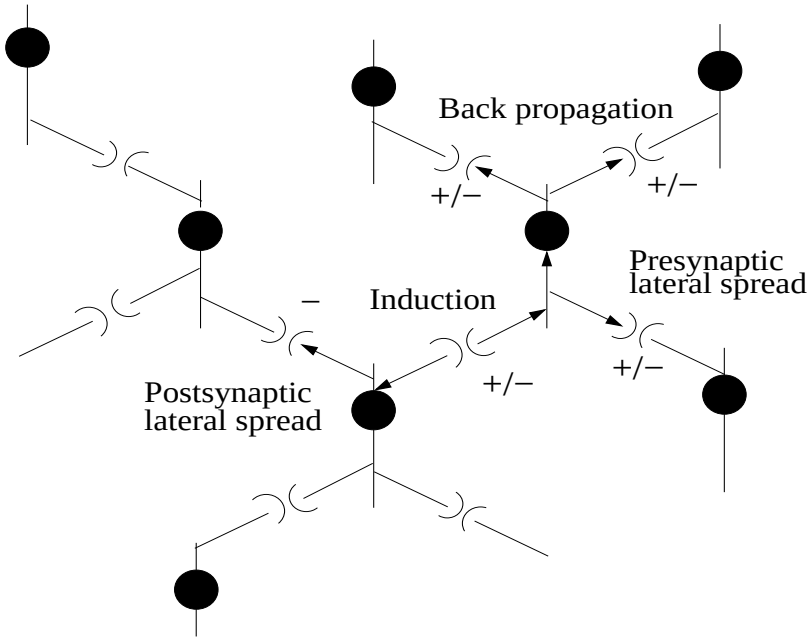


Abbildung 3.32: Schematische Darstellung unterschiedlicher Formen von heterosynaptischem LTP/LTD, nach [Bi und Poo, 2001]. ± bedeutet LTP bzw. LTD. Die Induktion von LTP/LTD an einer Synapse breitet sich sowohl post- und präsynaptisch lateral, als auch präsynaptisch im Dendritenbaum (back-propagation) aus. Hierbei geben die Pfeile die Ausbreitungsrichtung an und die Vorzeichen die bewirkte Veränderung der Synapse.

von Hebb verläuft, auch eine heterosynaptische Plastizität gibt [Lynch et al., 1977; Fitzsimonds et al., 1997]. Bei dieser Form von Plastizität sind nicht nur die Synapsen am Lernvorgang beteiligt, die direkt an der Signalübertragung beteiligt sind, sondern auch andere Synapsen vom prä- und postsynaptischem Neuron. Darüberhinaus sind sogar Synapsen des präsynaptischen Neurons in dessen Dendritenbaum davon betroffen.
In Abbildung 3.32 sind diese Verhältnisse schematisch dargestellt. Ein "+" steht für LTP und ein "-" für LTD. Die Pfeile geben die Ausbreitungsrichtung der heterosynaptischen Plastizität an, die durch die Induktion von LTP/LTD an einer Synapse entsteht.

Die Analogie zur Wirkungsweise des Neuronenzählers, in der von mir vorgestellten Lernregel, wird durch Abbildung 3.33 verständlich. In dieser Abbildung sind die Neuronenzähler mit c_i und die approximierten Synapsenzähler mit $\tilde{c}_{ij}$ bezeichnet. Angenommen, zu einem bestimmten Zeitpunkt t sei die Synapse zwischen Neuron 5 und Neuron 6 an einer Berechnung beteiligt gewesen, deren Ergebnis falsch war. Dann ergibt sich, infolge des negativen Reinforcementsignals r, eine Erhöhung der Neuronenzähler c_5 und

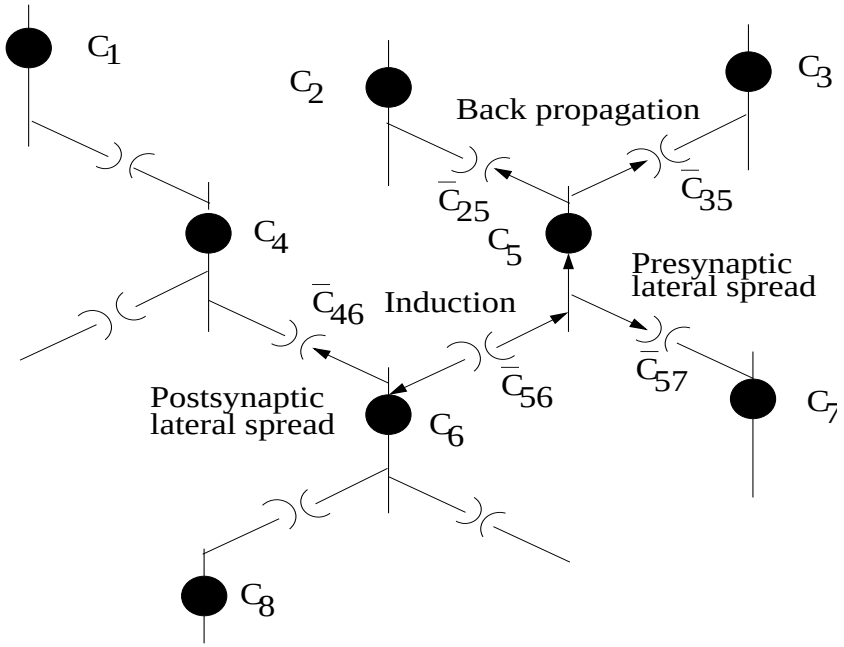


Abbildung 3.33: Veranschaulichung des Zusammenwirkens der Neuronenzähler auf die approximierten Synapsenzähler. Die Neuronenzähler sind hier mit c_i und die approximierten Synapsenzähler mit $\tilde{c}_{ij}$ bezeichnet. Die hier dargestellte Situation ist in Analogie zu Abbildung 3.32 zu sehen.

c_6. Dies hat zur Folge, dass der Wert des approximierten Synapsenzählers $\tilde{c}_{56}$ steigt, da $\tilde{c}_{56} = c_5 + c_6$. Hieraus ergibt sich dann, mit einer gewissen Wahrscheinlichkeit, eine synaptische Änderung, an der, durch Neuron 5 und 6, eingegrenzten Synapse.
Gleichzeitig wirkt sich die Erhöhung der Neuronenzähler c_5 und c_6 auch auf approximierte Synapsenzähler aus, die mit Neuron 5 und 6 verbunden sind. Denn für diese approximierten Synapsenzähler gilt, $\tilde{c}_{5j} = c_5 + c_j$ und entsprechend $\tilde{c}_{6j} = c_6 + c_j$. Wenn nun zu einem späteren Zeitpunkt $t + t_1$ eine dieser Synapsen an einer Berechnung beteiligt ist, die ein falsches Ergebnis produziert, dann ist der approximierten Synapsenzähler dieser Synapse auf einem höheren Wert, als er ohne das Ereignis, zum Zeitpunkt t an der Synapse zwischen Neuron 5 und 6, gewesen wäre. Durch diesen höheren approximierten Synapsenzähler ist jedoch auch die Wahrscheinlichkeit einer synaptischen Änderung an dieser Synapse erhöht. Somit wirken sich die Neuronenzähler heterosynaptisch auf zukünftige synaptische Änderungen angrenzender Synapsen aus.
Diese Auswirkung erfolgt jedoch auch, und das soll hier ausdrücklich betont werden, auf Synapsen des postsynaptischen Neurons, das über sein Axon mit nachgeschalteten Neuronen verbunden ist. Solch eine Form von vorwärts propagierter LTD wurde jedoch experimentell bisher nicht gefunden [Fitzsi-

monds et al., 1997; Tao et al., 2000][10]. Da sich die experimentellen Arbeiten zu diesem Thema jedoch noch im Anfangsstadium befinden, hat man noch keine absolute Gewissheit darüber, ob es solch eine Form von vorwärts propagierter heterosynaptischer Plastizität nicht doch gibt. Somit können die von mir durchgeführten Untersuchungen als Hypothese hierfür angesehen werden.

Abschließend bleibt festzuhalten, dass es schon erstaunlich ist, dass uns die drastische Reduktion der Freiheitsgrade in dem gut funktionierenden Modell von [Klemm et al., 2000], auf ein stochastisches Modell der synaptischen Plastizität geführt hat, das qualitativ mit weiten Teilen der experimentellen Resultate über heterosynaptische LTD in Einklang gebracht werden kann.

3.3.8 Diskussion

Insgesamt kann man festhalten, dass die hier vorgeschlagene Hebb-artige Lernregel mit Neuronengedächtnis das XOR-Problem in einem dreischichtigen neuronalen Netz ähnlich gut und schnell lernen kann, wie die von [Klemm et al., 2000] vorgeschlagene. Darüber hinaus ist sie nicht so empfindlich gegenüber verschiedenen Netzwerkdynamiken, wie z.B. [Chialvo und Bak, 1999; Bak und Chialvo, 2001], die das XOR-Problem für einen verrauschten winner-take-all Mechanismus lösen kann, jedoch nicht für einen softmax Mechanismus [Klemm et al., 2000]. Hierauf wird nochmals ausführlich in Kapitel 4. eingegangen, da es einen interessanten Unterschied zwischen beiden Netzwerkdynamiken gibt.
Durch das sinnvolle Zusammenwirken von zwei algebraischen Verteilungen, der Rangordnungsverteilung P_k^{rank} in 3.28 und der (Münz-)Verteilung $P(x)_{\text{coin}}$ in 3.31, konnte die Grundidee des Synapsenzählers in einen Neuronenzähler umgewandelt werden. Hierdurch wurde jedoch der Charakter der Lernregel vollständig verändert. Aus einer deterministischen Lernregel, die aktive Synapsen nur dann verändert, wenn ein Zähler über einer gewissen Schwelle steigt, wurde eine stochastische, die aktive Synapsen, welche an einer falschen Berechnung beteiligt waren, immer mit einer gewissen Wahrscheinlichkeit verändern kann.
Daran war das Prinzip der Extremalen Optimierung (EO) maßgeblich beteiligt, obwohl es uminterpretiert werden musste, bevor es zum Einsatz kommen konnte. Wie wichtig an diesem Prinzip die Verwendung von algebraischen Verteilungen ist, wird in Kapitel 3.4. untersucht.

Neben all diesen eher formalen Aspekten konnte für die Lernregel mit Neuronengedächtnis eine biologisch plausible Interpretation, in Form einer heterosynaptisch induzierten LTD gegeben werden. Diese Interpretation

[10] Dies gilt auch für vorwärts propagierte LTP.

deckt sich einerseits mit experimentellen Resultaten, welche eine laterale Ausbreitung in post- und präsynaptischem Neuron und sogar eine, in den Dendritenbaum des präsynaptischem Neurons rückpropagierte LTD zeigen (siehe [Bi und Poo, 2001] zur Übersicht). Andererseits widerspricht es den Experimenten von [Fitzsimonds et al., 1997; Tao et al., 2000], die keine vorwärtsgerichtete Propagation von LTP/LTD im postsynaptischen Neuron fanden.
Da die hier eingeführte Lernregel mit Neuronengedächtnis funktionell eingeführt wurde und somit ihre praktische Anwendbarkeit unterstreicht, darf man darauf gespannt sein, ob die hier aufgestellte Hypothese, welche die Existenz von vorwärts propagierter LTD im postsynaptischen Neuron vorhersagt, nicht doch unter bestimmten Bedingungen experimentell gefunden werden kann.

3.4 Lernregel mit Neuronengedächtnis 2

3.4.1 Einleitung

In diesem Abschnitt werden Resultate für das Modell mit Neuronenzähler präsentiert, welche sich ergeben, wenn für die Verteilungen $P_{\widetilde{c_{ij}}}$ und P_{coin} aus 3.28 und 3.31 eine Boltzmannverteilung

$$P_k \propto \exp(-\gamma k) \tag{3.39}$$
$$1 \leq k \leq 2\Theta + 3 \tag{3.40}$$
$$\gamma \in \mathbb{R}^+ \tag{3.41}$$

und eine Gleichverteilung

$$P(x)_{coin} = \text{const.} \tag{3.42}$$
$$0 \leq x \leq 1 \tag{3.43}$$

angenommen wird. Die Untersuchungen werden für den softmax Mechanismus mit $\beta = 10$ durchgeführt. Alle anderen Bedingungen sind die gleichen, wie in Abschnitt 3.3.6.2.
Die Verteilungen 3.39 und 3.42 werden gewählt, weil sie aus naiver Sicht den einfachsten, nicht trivialen Ansatz darstellen. Aus einem Vergleich dieser Ergebnisse mit denen aus den Abschnitt 3.3.6.1. bis 3.3.6.2 erhält man eine relative Qualitätseinschätzung der dort erzielten Ergebnisse.

3.4.2 Ergebnisse

In Abbildung 3.34 sind die Resultate der Simulationen aufgetragen. Der mittlere Fehler E wurde über eine Ensemble der Größe $N = 70000$ gemittelt

und in Abhängigkeit des Exponenten γ dargestellt. Für γ wurden Werte von 0 bis 3.0 zugelassen, wobei die Schrittweite 10^{-1} betrug. Jede der sechs Abbildungen steht für einen festen Wert des Neuronengedächtnisses Θ, wobei in der ersten Spalte Θ von 0 bis 2 und in der zweiten von 3 bis 5 verläuft. Die drei Kurven in jeder Abbildung entsprechen drei unterschiedlichen Zeitpunkten, an denen der mittlere Fehler E bestimmt wurde. Diese Zeitpunkt sind $t = 500, 1000, 1500$ und entsprechen den Kurven von oben nach unten.

$t = 500$	Θ	a	b	c	d
	0	0.41	0.07	0.14	0.13
	1	0.44	0.08	0.22	0.05
	2	0.45	0.08	0.26	0.05
	3	0.45	0.07	0.37	0.07
	4	0.45	0.06	0.42	0.07
	5	0.45	0.05	0.59	0.10
$t = 1000$	Θ	a	b	c	d
	0	0.38	0.12	0.27	0.16
	1	0.40	0.11	0.22	0.07
	2	0.41	0.09	0.33	0.08
	3	0.41	0.07	0.47	0.10
	4	0.41	0.07	0.49	0.10
	5	0.42	0.07	0.44	0.09
$t = 1500$	Θ	a	b	c	d
	0	0.38	0.16	0.21	0.14
	1	0.40	0.15	0.19	0.07
	2	0.41	0.13	0.29	0.08
	3	0.40	0.10	0.36	0.09
	4	0.40	0.08	0.46	0.12
	5	0.40	0.08	0.41	0.10

Tabelle 3.3: Parameter der Funktion 3.44 für die Resultate aus Abbildung 3.34. Die Werte wurden aus einem least square Fit erhalten.

$$E = a - b(1+\gamma)^c \exp(-d\gamma) \tag{3.44}$$

Die gezeigten Kurven in Abbildung 3.34 ergeben sich aus einem least square Fit an die Simulationsergebnisse. Hierbei wurde als Funktion mit den freien Parametern a, b, c, d gewählt, weil es der einfachste Funktionentyp war, an den die Ergebnisse gefittet werden konnten. Die erhaltenen Werte für die Parameter sind in Tabelle 3.3 angegeben.

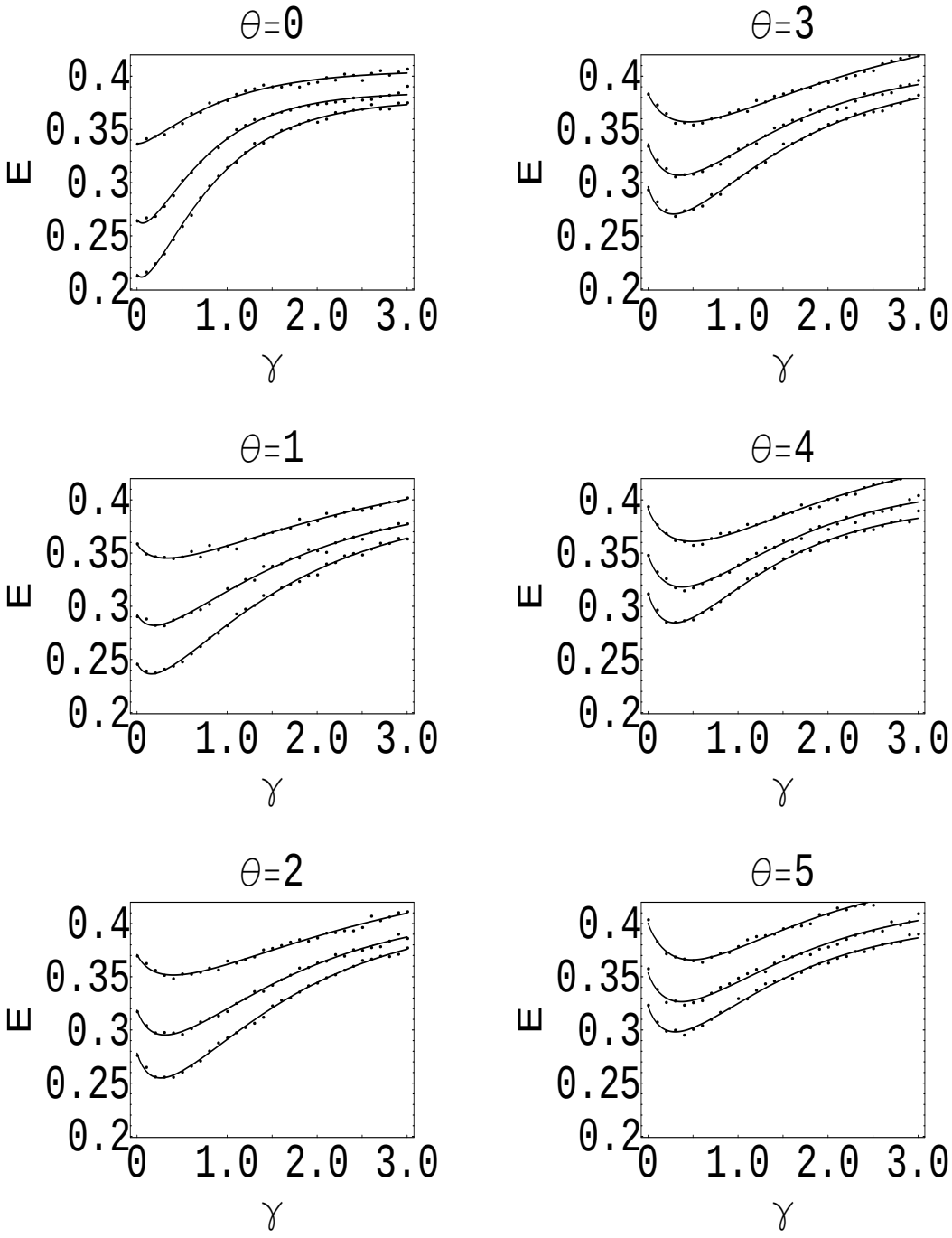


Abbildung 3.34: Mittlerer Fehler E in Abhängigkeit des Exponenten γ der Boltzmannverteilung für unterschiedliche Werte des Neuronengedächtnisses Θ. Die Kurven erhält man aus einem least square Fit an die Funktion 3.44. In jeder Abbildung entsprechen die Kurven von oben nach unten einem mittleren Fehler zu den Zeitpunkten $t = 500$, 1000 und 1500. Für alle Simulationen wurde ein softmax Mechanismus mit $\beta = 10$ verwendet.

Das interessante Ergebnis aus Abbildung 3.34 ist die Tatsache, dass die mittleren Fehler E mit zunehmendem Exponent γ und Neuronengedächtnis Θ immer schlechter werden und näherungsweise [11]gegen die gefitteten Parameter a aus Tabelle 3.3 laufen. Insgesamt sind alle mittleren Fehler E viel schlechter, als die in den Abschnitt 3.3.6.1 bis 3.3.6.3 erhaltenen Resultate. Darüberhinaus wirkt sich die Idee des Neuronengedächtnisses negativ auf den mittleren Fehler E aus, was man an den Parametern a und b aus Tabelle 3.3 ablesen kann. Dies lässt sich darauf zurückführen, dass die Bedingung für einen synaptischen Update $p_{coin} < p_{\widetilde{c}_{ij}}^{rank}$ für größere Werte von Θ, infolge der Boltzmannverteilung, zu selten erfüllt wird, um ein schnelles Lernen zu ermöglichen.

3.4.3 Diskussion

Die Boltzmannverteilung 3.39 stellt also einen kontraproduktiven Beitrag zum Neuronengedächtnis dar und kann somit, verglichen mit einer algebraischen Verteilung, nur wesentlich schlechtere Ergebnisse auf kurzen Zeitskalen liefern. Hieraus sieht man deutlich, dass das Neuronengedächtnis alleine noch kein schnelles Lernen ermöglicht, sondern erst das Zusammenspiel mit einem geeigneten Entscheidungskriterium in Form von algebraischen Verteilungen, welches eine Mittelung der individuellen Fehlerrate unterstützen. Insofern stellt auch das Prinzip der Extremalen Optimierung [Boettcher und Percus, 2001] einen wesentlichen Beitrag für das erfolgreiche Funktionieren der Lernregel mit Neuronengedächtnis dar.
Das konstruktive Zusammenwirken der Bestandteile des Gesamtsystems ist also wesentlich, um insgesamt gute Resultate der zu erhaltenden Lerndynamik zu erzielen.

3.5 Zusammenfassung

In diesem Kapitel wurde die Hebb-artige Lernregel mit Neuronengedächtnis eingeführt. Es wurden Zusammenhänge und Unterschiede zu den Arbeiten von [Chialvo und Bak, 1999; Bak und Chialvo, 2001] und vor allem [Klemm et al., 2000] aufgezeigt und diskutiert, die ebenfalls Hebb-artige Lernregeln für neuronale Netze vorgeschlagen haben. Desweiteren wurde auf die Bedeutung der Extremalen Optimierung (EO) von [Boettcher und Percus, 2001] hingewiesen und diskutiert, welche die Grundlage für die Umkodierung von Synapsen- auf Neuronenzähler darstellt.
Anhand spezieller Beispiele wurde die Wirkungsweise der Hebb-artige Lern-

[11]Für $\gamma \to \infty$ geht $E \to 0.5$, da die Updatebedingung nie mehr erfüllt werden kann. Die Parameterwerte für a aus Tabelle 3.3 sind jedoch für endliche Werte von γ gute Näherungen.

regel mit Neuronengedächtnis untersucht und mit den Arbeiten von [Chialvo und Bak, 1999; Bak und Chialvo, 2001; Klemm et al., 2000] verglichen. Diese ergaben ähnlich gute Resultate, wie in [Klemm et al., 2000], jedoch unter Verwendung von weniger Parametern und wesentlich bessere Resultate, wie in [Chialvo und Bak, 1999; Bak und Chialvo, 2001].
Durch die drei Parameter Gedächtnislänge Θ, Exponent der Rangordnungsverteilung τ und Exponent der (Münz-)Verteilung α, wurde die Lernregel sehr flexibel, was dadurch zum Ausdruck kam, dass es nicht die optimalen Werte für Θ, τ und α gibt, sondern hinreichend gute Kombinationen.

Neben diesen eher theoretischen Untersuchungen wurde eine biologisch plausible Interpretation der Hebb-artige Lernregel mit Neuronengedächtnis gegeben, die qualitative Ähnlichkeit mit experimentellen Resultaten zur heterosynaptischen LTD aufweist.

Kapitel 4

Lernen in neuronalen Systemen fern vom thermischen Gleichgewicht

4.1 Einleitung

Im vorangegangenen Kapitel wurden zwei Modelle zur Lösung des XOR-Problems untersucht und verglichen. Diese Untersuchungen basierten auf einer Definition des Gesamtsystems 3.1. Genauer gesagt waren beide Modelle identisch, bis auf die Lernregel (4) in 3.1 für die synaptischen Gewichte.
In diesem Kapitel wird der Unterschied zwischen zwei verschiedenen Modellen für die Netzwerkdynamik (3) in 3.1 herausgearbeitet. Bei diesen Modellen handelt es sich um den bereits diskutieren softmax Mechanismus in [Klemm et al., 2000] und einen verrauschten winner-take-all Mechanismus der von [Bak und Chialvo, 2001] verwendet wird. Darüberhinaus wird eine Methode eingeführt, mit der man einem verrauschten winner-take-all Mechanismus eine approximierte Temperatur zuordnen kann. In Kapitel 4.3. wird diese Methode dann dazu verwendet, um Unterschiede in den verwendeten Lernregeln aufzuzeigen. Des weiteren wird eine Korrespondenz zwischen mittlerem Fehler eines neuronalen Netzes und approximierter Temperatur demonstriert.

4.2 Vergleich softmax vs. verrauschter winner-take-all Mechanismus

Der verrauschte winner-take-all Mechanismus wird, im Gegensatz zu 3.1 und 3.2, durch

$$\overline{h}_j = h_j + \eta_k \tag{4.1}$$

$$j_{max} = \underset{j}{\text{argmax}}(\overline{h}_j) \tag{4.2}$$

definiert. Hierbei wird η_k gleichverteilt und unabhängig von vorausgegangenen Zügen aus dem Intervall $[0, \eta]$ gezogen [Bak und Chialvo, 2001]. Der Index k bezieht sich hierbei auf die k'te Realisation.
Im Gegensatz zu den Gleichungen 3.1 und 3.2 für den softmax Mechanismus, welche bereits die mittleren Wahrscheinlichkeiten für die Auswahl des i'ten Neurons mit dem inneren Feld h_i ergeben, muss für die "mikroskopischen" Gleichungen 4.1 und 4.2 solch eine gemittelte Verteilungen erst noch berechnet werden.

4.2.1 Analytische Lösung für die Verteilung des verrauschten winner-take-all Mechanismus

Um solch eine Wahrscheinlichkeitsverteilung für den verrauschten winner-take-all Mechanismus berechnen zu können, veranschaulichen wir uns zuerst die Problematik anhand Abbildung 4.1. Hier wurden die inneren Felder h_i der

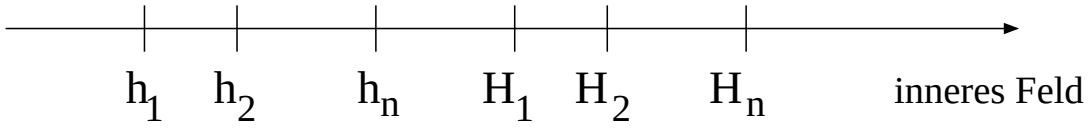


Abbildung 4.1: Anordnung der inneren Felder h_i mit ansteigender Größe. Hierbei bezeichnet $H_i = H_{i,max} = h_i + \eta$ den Wert des maximalen inneren Feldes für Neuron i. Die inneren Felder wurden nach der Permutation aufsteigend umnumeriert. Die Relationen zwischen h_i und H_i stellt nur einen von insgesamt n möglichen Fällen dar, der sich für n Neuronen ergeben kann.

Größe nach angeordnet, von kleinen zu großen Werten, und entsprechend der Permutation neu numeriert. Mit $H_i = H_{i,max} = \max\{h_i + \eta_k\} = h_i + \eta$ wurde das maximale innere Feld des i'ten Neurons bezeichnet, wobei das Rauschen η_k aus der Verteilung $[0, \eta]$ gezogen wurde. Die verrauschten inneren Felder $\overline{h}_i$, für die $h_i \leq \overline{h}_i \leq H_i$ gilt, wurden aus Gründen der Übersicht nicht in Abbildung 4.1 eingezeichnet. Es sei noch angemerkt, dass die Anordnung der maximalen inneren Felder H_i in Abbildung 4.1 bzgl. der inneren Felder h_i nur einen von n möglichen Fällen darstellt, der für n Neuronen auftreten kann.

Somit ergibt sich die Wahrscheinlichkeit

$$p_i(h_1, \ldots, h_n) = P(\overline{h}_i > \overline{h}_j \,, \forall j \neq i \,, \eta_k \in [0, \eta]) \tag{4.3}$$

dafür, dass das i'te Neuron ausgewählt wird dadurch, dass man die Wahrscheinlichkeiten wichtet, mit der die Neuronen $j \neq i$ nicht ausgewählt werden.

$$p_i(h_1,\ldots,h_n) = \int_{h_n}^{h_i+\eta} \rho_i(H)P(x_n \leq H)\ldots P(x_1 \leq H)dH \tag{4.4}$$

$$F_{x_i}(H) = P(x_i \leq H) = \int_0^H \rho_i(H')dH' \tag{4.5}$$

$$\rho_i(H) = \frac{1}{\eta}\Theta(H-h_i)\Theta(h_i+\eta-H) \tag{4.6}$$

Hierbei ist $P(x_i \leq H)$ die Verteilungsfunktion des i'ten Neurons, die speziell für die Verteilung $\rho_i(H)$ in 4.6

$$F_{x_i}(H) = P(x_i \leq H) = \begin{cases} 0 & : \quad H \leq h_i \\ \frac{H-h_i}{\eta} & : \quad h_i \leq H \leq h_i+\eta \\ 1 & : \quad h_i+\eta \leq H \end{cases} \tag{4.7}$$

ergibt. Zu Gleichung 4.4 ist noch zu bemerken, dass der Betrag des i'ten Neurons auf der rechten Seite darin besteht, dass über dessen Wahrscheinlichkeitsdichte ρ_i und nicht über die Verteilungsfunktion $P(x_i \leq H)$ integriert wird.
Da im folgenden der Fall $n = 3$ für uns von besonderem Interesse ist, soll die Lösung für diesen Fall explizit angegeben werden.

Für den Spezialfall n=3 erhält man für das erste Neuron

$$p_1(h_1,h_2,h_3) = \begin{cases} 0 & : \quad H_1 \leq h_3 \\ p_1^1(h_1,h_2,h_3) & : \quad H_1 \geq h_3 \end{cases} \tag{4.8}$$

$$p_1^1(h_1,h_2,h_3) = \frac{1}{\eta^3}\left[\frac{H^3}{3} - \frac{H^2(h_2+h_3)}{2} + h_2h_3H\right]_{h_3}^{h_1+\eta} \tag{4.9}$$

$$= \frac{1}{\eta^3}\left[\frac{(h_1+\eta)^3}{3} - \frac{(h_1+\eta)^2(h_2+h_3)}{2} + \right.$$
$$\left. + h_2h_3(h_1+\eta) - \frac{h_3^3}{3} + \frac{h_3^2(h_2+h_3)}{2} - h_2^2h_3^2\right] \tag{4.10}$$

Die Ausdrücke für p_2 und p_3 berechnen sich analog. Jedoch sind hierbei zusätzlich auftretende Fallunterscheidungen bezüglich der Grenzüberschreitung h_3 zu berücksichtigen. Unter Berücksichtigung dieser Fallunterscheidungen erhält man für p_2 :

$$p_2(h_1,h_2,h_3)=\begin{cases} 0 & : \; H_1 \le h_3; H_2 \le h_3 \\ p_2^1(h_1,h_2,h_3) & : \; H_1 \le h_3; H_2 \ge h_3 \\ p_2^2(h_1,h_2,h_3)+p_2^3(h_1,h_2,h_3) & : \; H_1 \ge h_3; H_2 \ge h_3 \end{cases} \tag{4.11}$$

mit

$$p_2^1(h_1,h_2,h_3) = \frac{1}{\eta^2}\left[\frac{H^2}{2}-h_3H\right]_{h_3}^{h_2+\eta} \tag{4.12}$$

$$= \frac{1}{\eta^2}\left[\frac{(h_2+\eta)^2}{2}-h_3(h_2+\eta)+\frac{h_3^2}{2}\right] \tag{4.13}$$

und

$$p_2^2(h_1,h_2,h_3) = \frac{1}{\eta^3}\left[\frac{H^3}{3}-\frac{H^2}{2}(h_1+h_3)+h_1h_3H\right]_{h_3}^{h_1+\eta} \tag{4.14}$$

$$= \frac{1}{\eta^3}\left[\frac{(h_1+\eta)^3}{3}-\frac{(h_1+\eta)^2}{2}(h_1+h_3)+\right.$$

$$\left.+h_1h_3(h_1+\eta)-\frac{h_3^3}{3}+\frac{h_3^2}{2}(h_1+h_3)-h_1h_3^2\right] \tag{4.15}$$

$$p_2^3(h_1,h_2,h_3) = \frac{1}{\eta^2}\left[\frac{H^2}{2}-h_3H\right]_{h_1+\eta}^{h_2+\eta} \tag{4.16}$$

$$= \frac{1}{\eta^2}\left[\frac{(h_2+\eta)^2}{2}-h_3(h_2+\eta)+\right.$$

$$\left.-\frac{(h_1+\eta)^2}{2}+h_3(h_1+\eta)\right] \tag{4.17}$$

Und für $p_3(h_1,h_2,h_3)$:

$$p_3=\begin{cases} 1 & : H_1 \le h_3; H_2 \le h_3 \\ p_3^1(h_1,h_2,h_3)+p_3^2(h_1,h_2,h_3) & : H_1 \le h_3; H_2 \ge h_3 \\ p_3^3(h_1,h_2,h_3)+p_3^4(h_1,h_2,h_3)+p_3^5(h_1,h_2,h_3) & : H_1 \ge h_3; H_2 \ge h_3 \end{cases} \tag{4.18}$$

Hierbei sind

$$p_3^1(h_1,h_2,h_3) = \frac{1}{\eta^2}\left[\frac{H^2}{2}-h_2H\right]_{h_3}^{h_2+\eta} \tag{4.19}$$

$$= \frac{1}{\eta^2}\left[\frac{(h_2+\eta)^2}{2}-h_2(h_2+\eta)-\frac{h_3^2}{2}+h_2h_3\right] \tag{4.20}$$

$$p_3^2(h_1,h_2,h_3) = \left[\frac{H}{\eta}\right]_{h_2+\eta}^{h_3+\eta} = \frac{1}{\eta}(h_3-h_2) \tag{4.21}$$

$$p_3^3(h_1,h_2,h_3) = \frac{1}{\eta^3}\left[\frac{H^3}{3} - \frac{H^2}{2}(h_2+h_2) + h_1h_2H\right]_{h_3}^{h_1+\eta} \tag{4.22}$$

$$= \frac{1}{\eta^3}\left[\frac{(h_1+\eta)^3}{3} - \frac{(h_1+\eta)^2}{2}(h_1+h_2) + h_1h_2(h_1+\eta) - \frac{h_3^3}{3} + \frac{h_3^2}{2}(h_1+h_2) - h_1h_2h_3\right] \tag{4.23}$$

$$p_3^4(h_1,h_2,h_3) = \frac{1}{\eta^2}\left[\frac{H^2}{2} - h_2H\right]_{h_1+\eta}^{h_2+\eta} \tag{4.24}$$

$$= \frac{1}{\eta^2}\left[\frac{(h_2+\eta)^2}{2} - h_2^2 - \frac{(h_1+\eta)^2}{2} + h_2h_1\right] \tag{4.25}$$

$$p_3^5(h_1,h_2,h_3) = \left[\frac{H}{\eta}\right]_{h_2+\eta}^{h_3+\eta} = \frac{1}{\eta}(h_3-h_2) \tag{4.26}$$

4.2.2 Invarianzeigenschaften beider Modelle

Aufgrund der im letzten Abschnitt erhaltenen analytischen Ergebnisse sind wir nun in der Lage den verrauschten mit dem softmax Mechanismus für verschiedene Parameter h_1, h_2 und h_3 zu berechnen, und zu vergleichen.
Zuerst sei angemerkt, dass sowohl der verrauschte als auch der softmax Mechanismus nicht von der absoluten Größe der inneren Felder h_i, sondern nur von deren relativen Größenverhältnissen abhängt, also invariant gegenüber der Transformation

$$h_i \to h_i' = h_i + h \tag{4.27}$$

mit $h = \text{const.}$ für alle i ist.
Für den verrauschten winner-take-all Mechanismus heißt das, dass Gleichung 4.4 für alle Neuronen invariant gegenüber der Transformation 4.27 sein muss.

$$p_i(h_1+h,\ldots,h_n+h) = \int_{h_n+h}^{h_i+h+\eta} \overline{\rho}_i(H)\overline{P}(x_n \le H)\ldots\overline{P}(x_1 \le H)\,dH \tag{4.28}$$

Durch die Substitution $H' = H - h$ erhält man

$$\int_{h_n}^{h_i+\eta} \overline{\rho}_i(H'+h)\overline{P}(x_n \le H'+h)\ldots\overline{P}(x_1 \le H'+h)\,dH' \tag{4.29}$$

Mit 4.6 und 4.27 folgt

$$\overline{\rho}_i(H'+h) = \rho_i(H') \quad ,\forall i \in \{1,\ldots,n\} \tag{4.30}$$

Mit 4.5, 4.27 und der Substitution $H''' = H'' - h$ folgt

$$\begin{aligned}\overline{P}(x_i \leq H' + h) &= \int_0^{H'+h} \overline{\rho}_i(H'')dH'' = \int_0^{H'} \overline{\rho}_i(H''' + h)dH''' \\ &= \int_0^{H'} \rho_i(H''')dH''' = P(x_i \leq H') \end{aligned} \tag{4.31}$$

Durch Einsetzten der Gleichungen 4.30 und 4.31 in das Integral 4.28 folgt nun

$$p_i(h_1 + h, \ldots, h_n + h) = p_i(h_1, \ldots, h_n) \quad , \forall i \in \{1, \ldots, n\} \tag{4.32}$$

die behauptete Invarianz unter der Transformation 4.27.

Die Invarianz des softmax Mechanismus ergibt sich wesentlich leichter, indem man 4.27 in Gleichung 3.1 und 3.2 einsetzt.

$$p_i' = \frac{\exp(\beta h_i')}{Z'} = \frac{\exp(\beta h_i)\exp(\beta h)}{Z'} = p_i \tag{4.33}$$

Da

$$Z' = \sum_{i=1}^{n} \exp(\beta h_i') = \exp(\beta h) \sum_{i=1}^{n} \exp(\beta h_i) = \exp(\beta h) Z \tag{4.34}$$

4.2.3 Ergebnisse

Aufgrund der Invarianzeigenschaft und Simulationsergebnisse[1] beider Modelle kann man o.B.d.A. die inneren Felder h_i in der absoluten und relativen Größenordnung $O(1)$ wählen.
Abbildung 4.2 zeigt die ausgezeichnete Übereinstimmung der simulierten mit der durch 4.8, 4.11 und 4.18 analytisch berechneten Verteilung des verrauschten winner-take-all Mechanismus in Abhängigkeit des Rauschens η. Für die inneren Felder wurden die Werte $h_1 = 0, h_2 = 0.8$ und $h_3 = 1$ verwendet und die Simulation wurde über ein Ensemble der Größe $N = 10000$ gemittelt.

Wir vergleichen nun den verrauschten mit dem softmax Mechanismus unter der Voraussetzung, dass die inneren Felder $h_1 = 0, h_2 = 0.8$ und $h_3 = 1$ fest gewählt bleiben. Denn dann kann man die Frage stellen, wie groß die Temperatur in einer neuronalen Schicht ist, deren Netzwerkdynamik durch

[1] Bei den durchgeführten Simulationen ergab sich, dass $|h_i - h_j| \sim 1$ im Mittel gilt. Dies hängt natürlich mit dem verwendeten Rauschen bzw. dem temperaturartigen Parameter β zusammen.

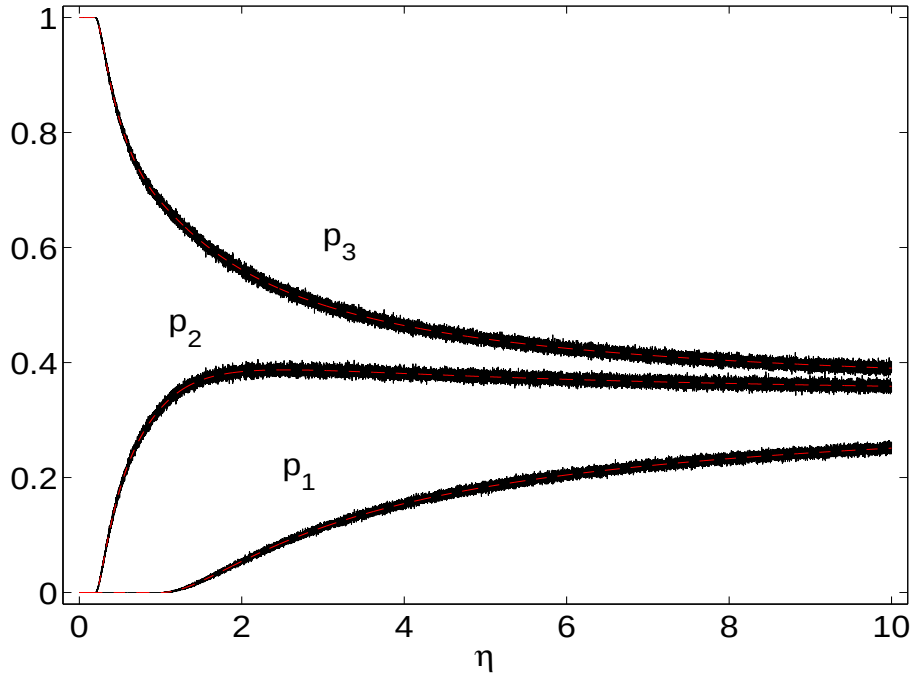


Abbildung 4.2: Verteilung p_i der inneren Felder h_i als Funktion der Rauschgröße η für $h_1 = 0, h_2 = 0.8$ und $h_3 = 1$. Die Simulation ist über ein Ensemble der Größe $N = 10000$ gemittelt (schwarze Kurven). Die roten Kurven zeigen die Ergebnisse der analytischen Berechnungen.

einen verrauschten winner-take-all Mechanismus gegeben ist, für die im voraus zwar ein Rauschlevel η, jedoch keine Temperatur definiert ist. Diese Temperatur erhält man dann aus einem Vergleich der zugehörigen Verteilungen p_i zwischen dem softmax Mechanismus, bei dem die Verteilungen, und somit die Temperatur β^{-1}, durch die Gleichungen 3.1 und 3.2 explizit gegeben sind, und den Gleichungen 4.8, 4.11 und 4.18 für den verrauschten winner-take-all Mechanismus.
Es sei noch einmal in aller Deutlichkeit darauf hingewiesen, dass sowohl die Gleichungen 3.1 und 3.2 als auch 4.8, 4.11 und 4.18 für die Netzwerkdynamik, im Ensemblemittel für eine vorgegebene Temperatur β^{-1} bzw. ein Rauschlevel η, unter der Voraussetzung von fest gewählten inneren Feldern h_i, erhalten wurden.

Wenn man nun den verrauschten mit dem softmax Mechanismus vergleicht, dann gibt es genau zwei Möglichkeiten. Entweder sind beide Verteilungen identisch oder nicht. Im ersten Fall existiert eine eindeutig definierte Temperatur β^{-1}, so dass die Verteilungen 3.1, 3.2, 4.8, 4.11 und 4.18 beider Mechanismen identisch übereinstimmen, im zweiten Fall gibt es nur eine approximierte Temperatur, die den "Abstand" beider Verteilungen minimiert.

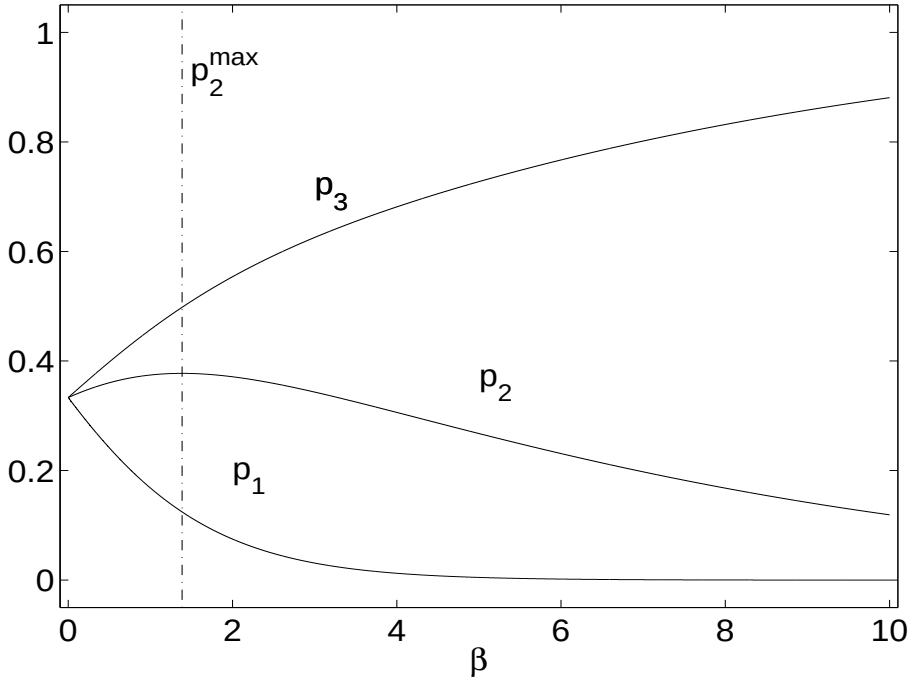


Abbildung 4.3: Verlauf des softmax Mechanismus in Abhängigkeit von β für die inneren Felder $h_1 = 0, h_2 = 0.8$ und $h_3 = 1$, welche fest gehalten wurden.

Abbildung 4.3 zeigt den Verlauf der Verteilung 3.1, 3.2 für den softmax Mechanismus in Abhängigkeit von β. Wie man aus Abbildung 4.3 ablesen kann, ist der Maximalwert $p_2^{max} = 0.377$. Vergleicht man dies mit dem verrauschten winner-take-all Mechanismus aus Abbildung 4.2, so liest man dort einen Maximalwert von $p_2^{max} = 0.387$ ab. Hieraus folgt, dass man Neuron zwei nicht als Bezugspunkt wählen kann, um die Temperatur β^{-1} im verrauschten winner-take-all Mechanismus zu definieren. Denn um den Punkt $p_2^{max} = 0.387$ herum gibt es eine Umgebung, für die keine Temperatur im Sinne des softmax Mechanismus, existiert.

Somit ist nicht nur gezeigt, dass Neuron zwei nicht als Bezugspunkt zur Temperaturdefinition verwendet werden kann, sondern auch, dass verrauschten und softmax Mechanismus nicht identisch sind. Für die beiden anderen Verteilungen p_1 und p_3 aus Abbildung 4.2 gibt es jeweils immer ein β, so dass sie mit der entsprechenden Verteilung in Abbildung 4.3 in Übereinstimmung gebracht werden kann.

Nun wenden wir uns der Frage zu, wie stark sich denn beide Modelle voneinander unterscheiden bzw. wie gut man mit dem einen Modell das andere approximieren kann? Um dies zu beantworten, vergleichen wir die Verteilungen für p_i für die zwei verbleibenden Bezugspunkte der Temperaturdefinition

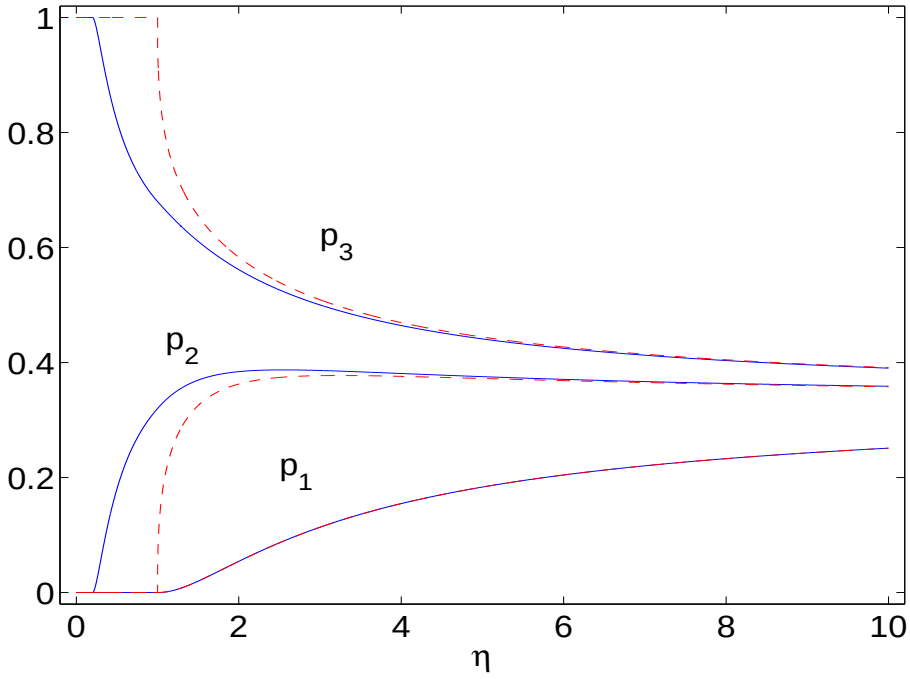


Abbildung 4.4: Vergleich der Verteilungen p_i für den verrauschten winner-take-all (durchgezogene Linie) mit den softmax (gestrichelte Linie) Mechanismus in Abhängigkeit von η für $h_1 = 0, h_2 = 0.8$ und $h_3 = 1$. Als Bezugspunkt zur Bestimmung von β wurde das erste Neuron $i = 1$ verwendet.

β^{-1}, nämlich für Neuron 1 und 3.
Abbildung 4.4 zeigt den Fall, bei dem Neuron 1 als Bezugspunkt dient und Abbildung 4.5 den Fall für Neuron 3. Der verrauschte winner-take-all Mechanismus ist mit durchgezogener Linie dargestellt, wohingegen die stochastische Variante gestrichelt gezeichnet ist. Man erkennt, dass sich beide Modelle für Rauschwerte mit $\eta \gtrsim 3$ in der Größenordnung 10^{-2} voneinander unterscheiden, wobei die Abweichung für $\eta \to \infty$ asymptotisch gegen Null geht da alle Komponenten gegen $p_1 = p_2 = p_3 = \frac{1}{3}$ gehen. Für kleine Rauschwerte $\eta \lesssim 3$ bleibt die Abweichung mit Neuron drei als Bezugspunkt in der Größenordnung 10^{-2}, wohingegen sie für Neuron eins bis zu einer Größenordnung von 10^{-1} ansteigen kann.
Dies liegt daran, dass bei Verwendung von Neuron eins als Bezugspunkt das Rauschen η erst einen Mindestwert annehmen muss, bevor die Wahrscheinlichkeit p_1 größer als Null werden kann. Bis zu dieser Rauschschwelle ist $\beta^{-1} = 0$[2] und aus dem stochastischen wird ein deterministischer winner-take-all Mechanismus für den $p_3 = 1$ und $p_1 = 0$ gilt[3]. Dies geschieht unabhängig davon, wie die Relationen der inneren Felder h_1, h_3 zum Rauschlevel η

[2]Dies liest man aus Abbildung 4.3 ab.
[3]Beachte, dass für die inneren Felder immer noch $h_1 = 0, h_2 = 0.8$ und $h_3 = 1$ gilt.

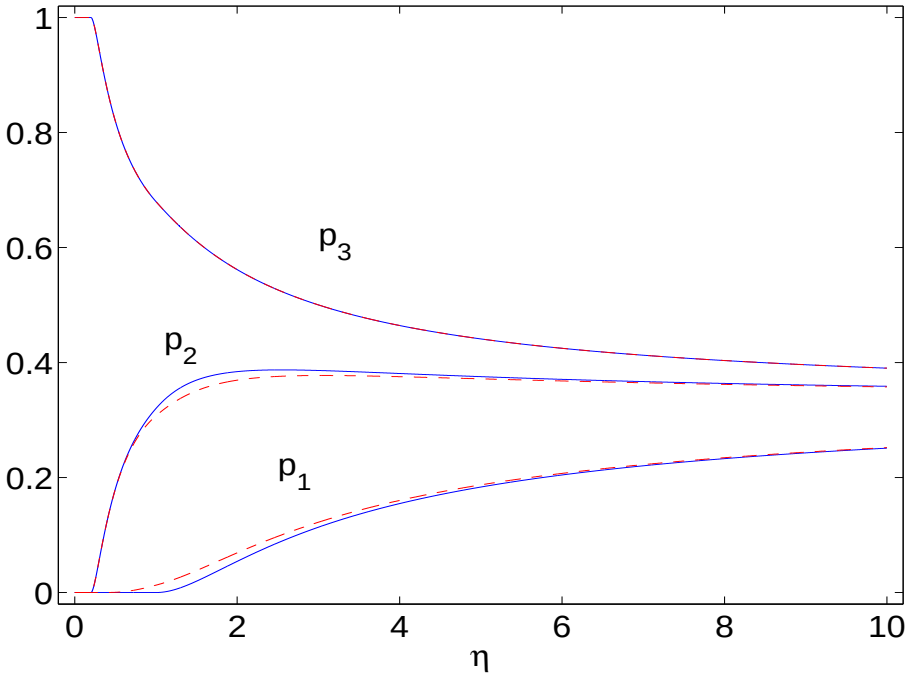


Abbildung 4.5: Vergleich der Verteilungen p_i für den verrauschten winner-take-all (durchgezogene Linie) mit den softmax (gestrichelte Linie) Mechanismus in Abhängigkeit von η für $h_1 = 0, h_2 = 0.8$ und $h_3 = 1$. Als Bezugspunkt zur Bestimmung von β wurde das dritte Neuron $i = 3$ verwendet.

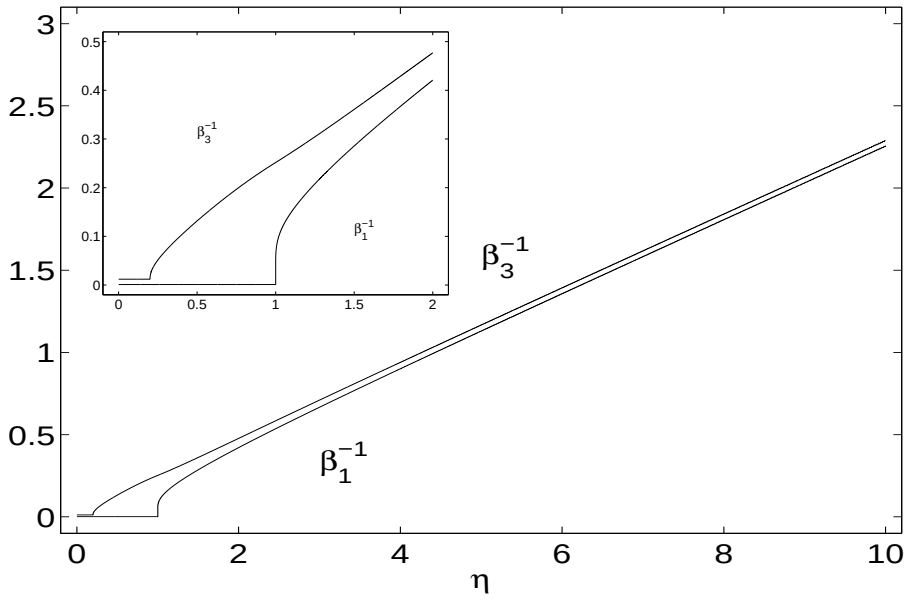


Abbildung 4.6: Darstellung von β_1^{-1} und β_3^{-1} in Abhängigkeit des Rauschens η die sich aus Abbildung 4.4 und 4.5 ergeben.

wirklich sind und ist somit unabhängig von den speziellen Werten, die hier gewählt wurde. Bei der Verwendung von Neuron drei als Bezugspunkt kann dieser Effekt nicht auftreten, da β von Beginn an endliche Werte annimmt und somit wohl definiert ist.
In Abbildung 4.6 sind die für Abbildung 4.4 und 4.5 ermittelten Werte von β^{-1} gegen das Rauschen η aufgetragen. Man sieht, dass sich beide Parameter für kleine Rauschwerte $\eta \lesssim 2$ deutlich voneinander unterscheiden, jedoch asymptotisch zusammenlaufen. Aus den oben genannten Gründen ist $\beta_1^{-1} \leq \beta_3^{-1}$ für $\eta \lesssim 2$.

4.3 Temperaturverlauf in mehrschichtigen neuronalen Netzen

In diesem Abschnitt benutze ich die Ergebnisse von Kapitel 4.2. in dem gezeigt wurde, dass softmax und verrauschter winner-take-all Mechanismus zwar nicht identisch sind, es aber möglich ist dem verrauschten winner-take-all Mechanismus eine approximierte Temperatur β^{-1} zuzuordnen, welche man aus einem Vergleich der gemittelten Ensembleverteilung p_i erhält.
Diese Methode einer approximierten Zuordnung einer Temperatur wird im folgenden dazu verwendet, den Temperaturverlauf in einem dreischichtigen neuronalen Netz in Abhängigkeit der verwendeten Lernregel zu bestimmen, bei dem als Netzwerkdynamik ein verrauschten winner-take-all Mechanismus verwendet wird.

4.3.1 Verrauschter winner-take-all Mechanismus

Die Netzwerkdynamik für das dreischichtige neuronale Netz aus Kapitel 3. wird durch

$$\overline{h}_j = h_j + \eta_k \tag{4.35}$$

$$i_{max} = \underset{i}{\operatorname{argmax}}(\overline{h}_j) \tag{4.36}$$

gegeben, wobei η_k gleichverteilt und unabhängig aus dem Intervall $[0, \eta]$ gezogen wird. Der Parameter η wird für die nachfolgenden Simulationen so gewählt, dass der mittlere Fehler $E(t)$ für das Modell mit Synapsenzähler für $\Theta = 2$ mit einer Netzwerkdynamik, welche durch den softmax Mechanismus für $\beta = 10$ festgelegt wird, näherungsweise mit dem Modell für den verrauschten winner-take-all Mechanismus übereinstimmt. Simulationen zeigen, dass $\eta = 0.45$ geringe Abweichungen zwischen beiden Modellen ergibt. Für die synaptischen Änderungen wurde $\delta = 1 = \text{const.}$ gewählt.

4.3.2 Ergebnisse

4.3.2.1 Lernregel mit Synapsengedächtnis

In Abbildung 4.7 sind die Lernkurven für das Modell mit Synapsengedächtnis in Abhängigkeit verschiedener Werte von Θ dargestellt. Der Fehler $E(t)$ wurde aus einer Mittelung über eine Ensemble der Größe $N = 5000$ erhalten. Man erkennt, dass für zunehmende Werte von Θ die Lernkurven erst besser und dann wieder schlechter werden. Die besten Lernkurven für kurze Zeiten erhält man für $\Theta = 2, 3$. Das Konvergenzverhalten für große Zeiten ist in der

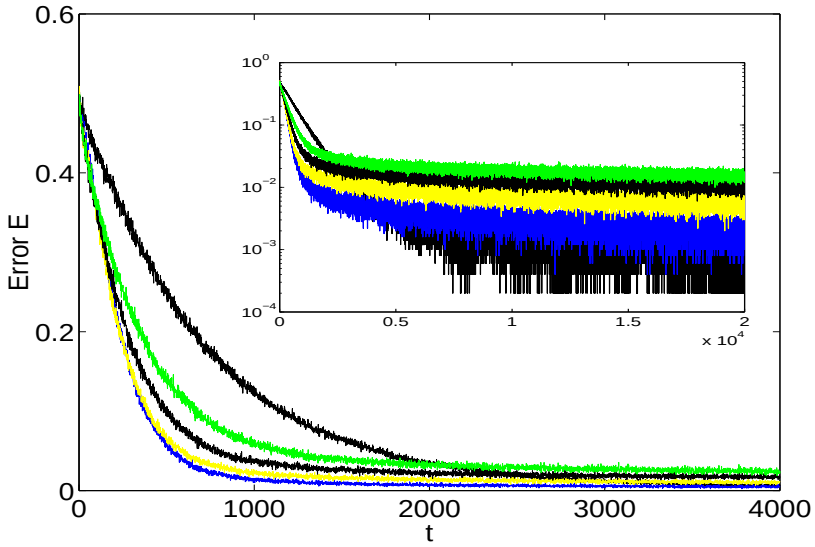


Abbildung 4.7: Mittlerer Fehler E in Abhängigkeit von verschiedenen Werten für Θ für das Modell mit Synapsengedächtnis. Das Rauschen beträgt hierbei $\eta = 0.45$. Die Kurven bedeuten von oben nach unten: $\Theta = 1$ (schwarz), $\Theta = 5$ (grün), $\Theta = 4$ (schwarz), $\Theta = 3$ (gelb), $\Theta = 2$ (blau).

Doppellogabbildung 4.7 in der überlagerten Darstellung gezeigt. Hier erkennt man, dass die Lernkurve für $\Theta = 1$ das beste Konvergenzverhalten besitzt. Insgesamt ordnen sich die Lernkurven auf der rechten Seite der Abbildung nach der Länge des Synapsengedächtnisses Θ an.

In den Abbildungen 4.8 und 4.9 sind die mittleren approximierten Temperaturverläufe β^{-1} in Abhängigkeit der Zeit t von Schicht 2 und 3 des neuronalen Netzes in Abhängigkeit verschiedener Werte für Θ dargestellt. Für die Berechnung der approximierten Temperatur wurde jeweils das Neuron mit dem größten inneren Feld h_i als Bezugspunkt ausgewählt. Beide Abbildungen beginnen erst nach 300 Zeitschritten, da zu Beginn des Lernprozesses oftmals Situationen auftreten, bei denen die inneren Felder h_i der Neuronen annähernd gleich sind. Hieraus folgt, dass β gegen Null

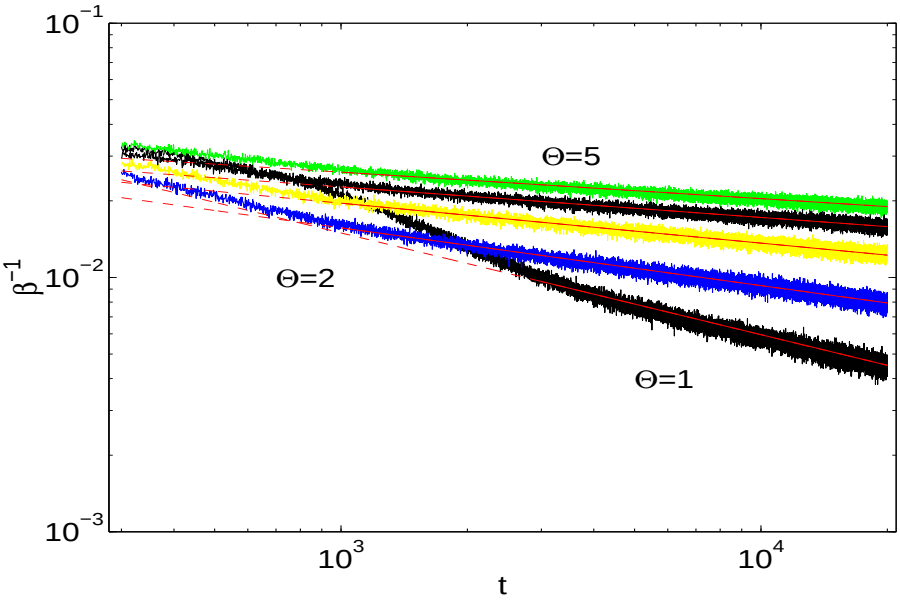


Abbildung 4.8: Mittlere approximierte Temperatur β^{-1} für Schicht 2 in Abhängigkeit der Zeit t. Die Kurven auf der rechten Seite der Abbildung sind von unten nach oben durch $\Theta = 1$ bis $\Theta = 5$ parametrisiert ($\Theta = 2$ (blau), $\Theta = 3$ (gelb), $\Theta = 4$ (schwarz)).

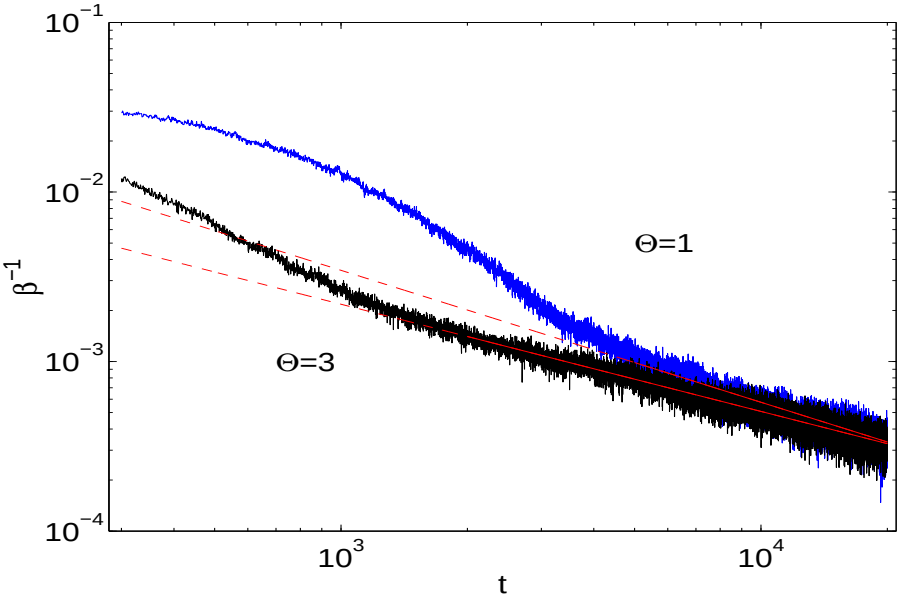


Abbildung 4.9: Mittlere approximierte Temperatur β^{-1} für Schicht 3 in Abhängigkeit der Zeit t für $\Theta = 1$ und $\Theta = 3$. Die Temperaturverläufe für $\Theta = 2$ bis $\Theta = 5$ sind aus Gründen der Übersicht einzeln in Abbildung 4.10 dargestellt.

geht, was einer unendlich großen Temperatur entspräche. Nach einer relativ kurzen Einlernphase verlaufen die Prozesse jedoch so geordnet, so dass man β bzw. β^{-1} wohldefiniert angeben kann. Darüberhinaus sei noch darauf hingewiesen, dass β zwar zu jedem Zeitpunkt wohldefiniert ist, jedoch keinerlei Stetigkeitseigenschaften besitzt. Die recht glatten Kurven aus den Abbildungen 4.8 und 4.9 sind eine Folge der Mittelung über das Ensemble.

	γ	t_{Fit}
$\Theta = 1$	0.40± 0.001	3000
$\Theta = 2$	0.23± 0.0004	1000
$\Theta = 3$	0.16± 0.0003	1000
$\Theta = 4$	0.12± 0.0003	1000
$\Theta = 5$	0.10± 0.0003	1000

Tabelle 4.1: Die Exponenten des Potenzgesetzes für Schicht 2.

	γ	t_{Fit}
$\Theta = 1$	0.78± 0.003	5000
$\Theta = 2$	0.60± 0.002	2000
$\Theta = 3$	0.63± 0.001	2000
$\Theta = 4$	0.66± 0.002	2000
$\Theta = 5$	0.69± 0.001	2000

Tabelle 4.2: Die Exponenten des Potenzgesetzes für Schicht 3.

Konkret sieht man, dass der Temperaturverlauf in jeder Schicht ab dem Zeitschritt $t \sim 2000$ annähernd einem Potenzgesetz $\beta^{-1} \sim t^{-\gamma}$ folgt. Die Exponenten γ sind in Tabelle 4.1 und 4.2 angegeben. In der letzten Spalte sind die Zeitpunkte aufgeführt, ab denen gefittet wurde, denn in der Einlernphase weichen die Temperaturverläufe stark von einem Potenzgesetz ab.
Man erkennt die Ähnlichkeit in der Steigung der Kurven $\Theta = 2$ bis $\Theta = 5$. Lediglich die Kurve $\Theta = 1$ ist etwas steiler. Abbildung 4.10 zeigt die Lernkurven für $\Theta = 2$ bis $\Theta = 5$, bei denen auffällt, dass nicht nur die Exponenten, sondern auch die Absolutwerte der Temperatur für die gesamte Simulationszeit näherungsweise übereinstimmen. Hieraus folgt der Umkehrschluss, dass man die approximierten Temperaturen in Schicht 3 nicht dazu verwenden kann, auf die Gedächtnislänge der Lernregel mit Synapsengedächtnis zurück zu schließen.

Diese Aussagen korrespondieren mit den Resultaten, welche wir aus der

Doppellogabbildung 4.7 gewonnen haben, nämlich dass die Lernkurve mit $\Theta = 1$ für große Zeiten am besten konvergiert und alle anderen Kurven annähernd gleiches Konvergenzverhalten zeigen[4]. Die Steigung der mittleren approximierten Temperatur ist also ein guter Indikator für die Konvergenzeigenschaften einer Lernkurve.
Darüberhinaus entsprechen die Absolutwerte der mittleren approximierten Temperatur für Schicht 2 tendenziell der Größe des Fehlers in Abbildung 4.7.

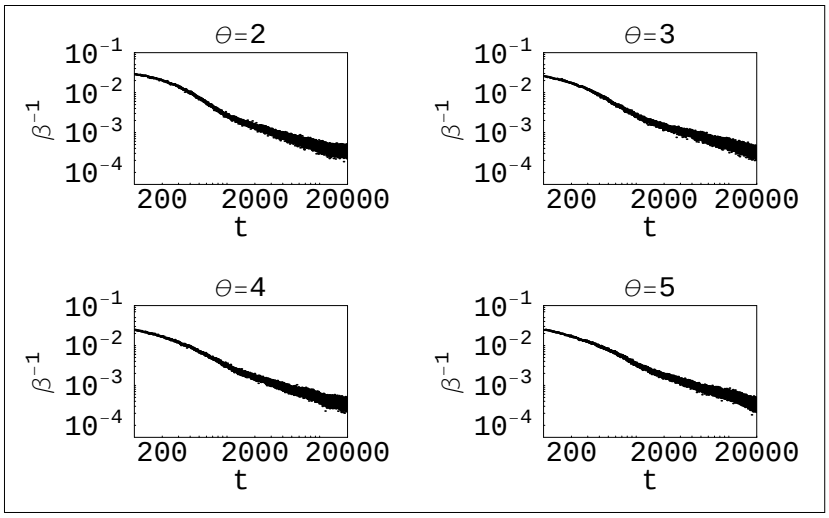


Abbildung 4.10: Mittlere approximierte Temperatur β^{-1} für Schicht 3 für Θ 2 bis 5.

4.3.2.2 Lernregel mit Neuronengedächtnis

In Abbildung 4.11 sind die Lernkurven für das Modell mit Neuronengedächtnis in Abhängigkeit verschiedener Werte von Θ dargestellt. Hierbei wurden die Exponenten der Rangordnungs- und (Münz-)Verteilung $\tau = 2.0$ und $\alpha = 1.2$ für alle Simulationen fest gewählt. Der Fehler $E(t)$ wurde aus einer Mittelung über ein Ensemble der Größe $N = 5000$ erhalten. Man erkennt auch hier, dass für zunehmende Werte von Θ die Lernkurven für kurze Zeiten erst besser und dann für lange Zeiten wieder schlechter werden. Die besten Lernkurven für kurze Zeiten erhält man für $\Theta = 4$.
Das Konvergenzverhalten für große Zeiten ist in der Doppellogabbildung 4.11 dargestellt. Hier erkennt man, dass die Lernkurve für $\Theta = 2$ das beste Konvergenzverhalten besitzt.
Insgesamt stellt man fest, dass alle Lernkurven nach 20000 Zeitschritten

[4]Es sei angemerkt, dass die Exponenten der mittleren approximierten Temperatur von Schicht 2 und 3 nicht mit den gefitteten Exponenten des Fehlers (nicht angegeben) in Abbildung 4.7 übereinstimmen.

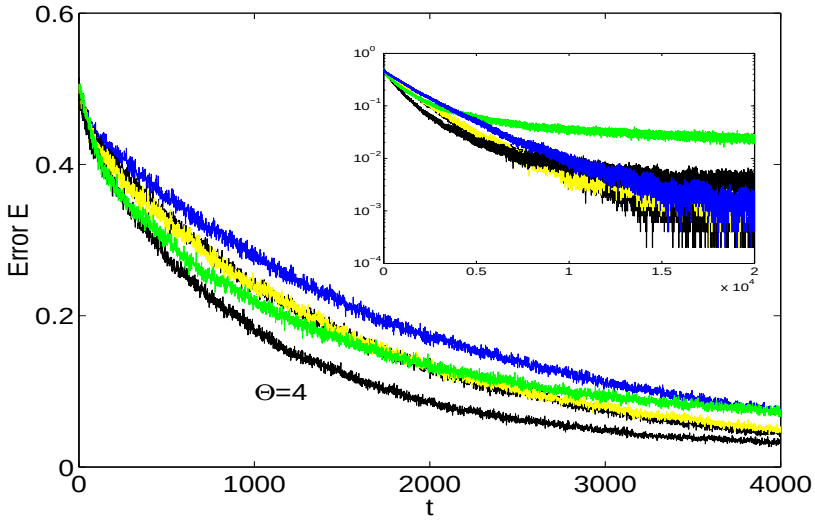


Abbildung 4.11: Mittlerer Fehler E in Abhängigkeit der Zeit t für verschiedene Werte von Θ für das Modell mit Neuronengedächtnis. Das Rauschen beträgt hierbei $\eta = 0.45$ und die Größe der synaptischen Änderung beträgt $\delta = 1 = \text{const.}$. Die Exponenten der Rangordnungs- und (Münz-)Verteilung wurden zu $\tau = 2.0$ und $\alpha = 1.2$ fest gewählt. Die Lernkurven entsprechen: $\Theta = 1$ (blau), $\Theta = 2$ (schwarz), $\Theta = 3$ (gelb), $\Theta = 4$ (schwarz) und $\Theta = 5$ (grün).

unter einem Fehler von 5% liegen, worin sich die durchweg gute Konvergenz widerspiegelt. Das interessante Ergebnis dieser Untersuchungen besteht wieder darin, dass es von der betrachteten Zeitskala abhängt, welche Θ-Werte die besten Resultate liefern. Für kurze Zeitskalen waren es $\Theta = 4$ und für große $\Theta = 2$.

In den Abbildungen 4.12 und 4.13 sind die mittleren approximierten Temperaturverläufe β^{-1} in Abhängigkeit der Zeit t für verschiedene Θ dargestellt. Als Bezugspunkt dieser Berechnungen wurde wieder jeweils das Neuron mit dem größten inneren Feld h_i gewählt.
Man erkennt auch hier, dass ab einem gewissen Zeitpunkt der Verlauf der approximierten Temperatur wieder einem Potenzgesetz $\beta^{-1} \sim t^{-\gamma}$ folgt. Die Exponenten γ sind in Tabelle 4.3 und 4.4 angegeben. In der letzten Spalte sind die Zeitpunkte aufgeführt, ab denen gefittet wurde.

Im Unterschied zum Modell mit Synapsengedächtnis sind die approximierten Temperaturen β^{-1} für Schicht 2 und 3 für die jeweiligen Θ-Werte sehr ähnlich. Auf der anderen Seite sind die Exponenten des Potenzgesetzes für unterschiedliche Θ-Werte auch unterschiedlich. Hieraus kann man erkennen, dass die beiden Modelle eine unterschiedliche Organisationsstruktur bzgl.

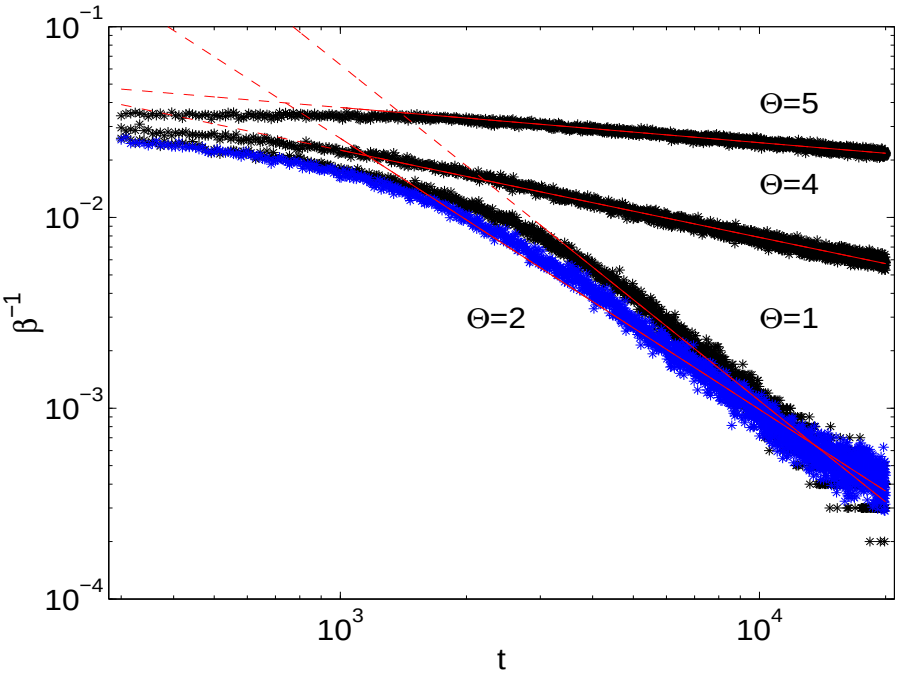


Abbildung 4.12: Mittlere approximierte Temperatur β^{-1} in Abhängigkeit der Zeit t für Schicht 2.

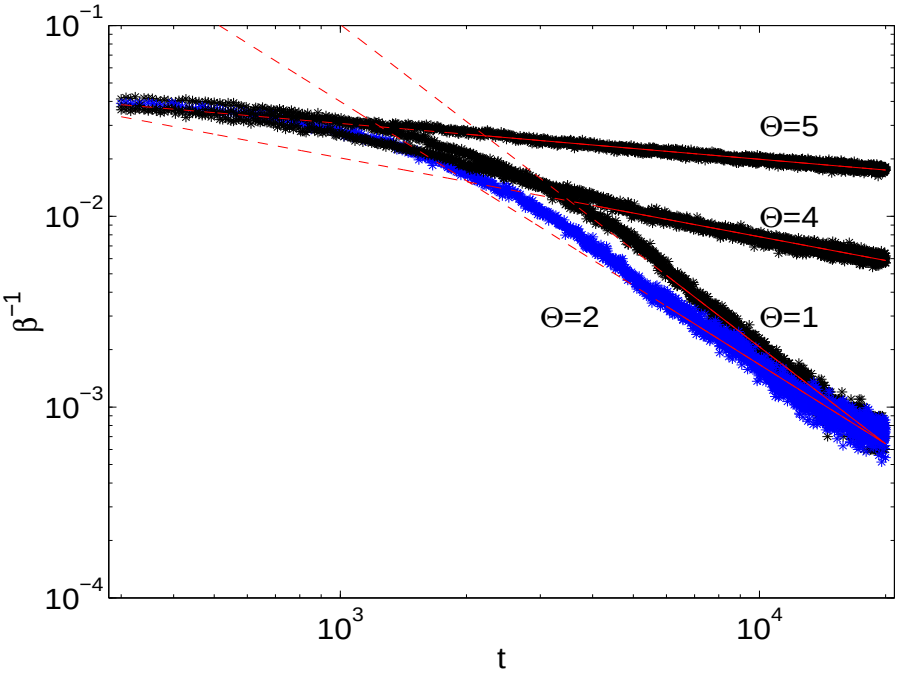


Abbildung 4.13: Mittlere approximierte Temperatur β^{-1} in Abhängigkeit der Zeit t für Schicht 3.

	γ	t_{Fit}
$\Theta = 1$	1.76 ± 0.003	5000
$\Theta = 2$	1.42 ± 0.003	5000
$\Theta = 3$	1.44 ± 0.002	2000
$\Theta = 4$	0.46 ± 0.001	2000
$\Theta = 5$	0.18 ± 0.002	2000

Tabelle 4.3: Die Exponenten des Potenzgesetzes für Schicht 2.

	γ	t_{Fit}
$\Theta = 1$	1.69 ± 0.002	6000
$\Theta = 2$	1.38 ± 0.003	6000
$\Theta = 3$	1.26 ± 0.002	5000
$\Theta = 4$	0.41 ± 0.001	5000
$\Theta = 5$	0.19 ± 0.001	2000

Tabelle 4.4: Die Exponenten des Potenzgesetzes für Schicht 3.

der Lerndynamik besitzen, die sich auch in einem unterschiedlichen Verlauf der mittleren approximierten Temperatur ablesen lässt.
Für die Beziehungen zwischen mittlerer approximierter Temperatur und dem mittleren Fehler in 4.11 gelten die gleichen Aussagen wie beim Modell mit Synapsengedächtnis, was das Konvergenzverhalten und die absolute Größe des mittleren Fehlers angeht. Darüberhinaus trifft diese Korrespondenz nicht nur auf Schicht 2, sondern diesmal auch auf Schicht 3 zu.

4.4 Zusammenfassung

In Kapitel 4.2. wurde analytisch gezeigt, dass softmax und verrauschter winner-take-all Mechanismus nicht identisch sind und deren Differenz im allgemeinen Fall eine Funktion der inneren Felder h_i der Neuronen, des Rauschlevels η und der Temperatur β ist. Weiter wurde eine Methode vorgeschlagen, mit der man einem verrauschten winner-take-all Mechanismus eine approximierte Temperatur zuordnen kann. Diese approximierte Temperatur ist eine Funktion der inneren Felder h_i der Neuronen und des Rauschlevels η des verrauschter winner-take-all Mechanismus.

Aufbauend auf die in Kapitel 4.2. vorgeschlagene Methode, wurde jeder Schicht eines dreischichtigen neuronalen Netzes während des Lernens des XOR-Problems, eine approximierte Temperatur zugeordnet. Als Lerndynamik wurde die Lernregel mit Synapsen- bzw. Neuronengedächtnis verwendet.

Das interessante Resultat bestand darin, dass man für beide Lernregeln in jeder Schicht des neuronalen Netzes einen Abkühlvorgang (*annealing*) in der approximierten Temperatur beobachten kann. Dieses Abkühlen korrespondiert mit den Konvergenzeigenschaften des mittleren Fehlers und tritt selbstorganisiert ohne äußeren Einfluss, infolge der Lernregel auf. Dieses Verhalten unterscheidet sich insbesondere von *simulated annealing*[Kirkpatrick et al., 1983], einem stochastischen Optimierungsverfahren, bei dem explizit von außen der Verlauf der Temperatur vorgegeben werden muss, um ein bestimmtes Optimierungsproblem zu lösen.
Weiterhin ist aufgrund des Abkühlens während des Lernvorgangs insbesondere die approximierte Temperatur zeitlich nicht konstant. Daraus folgt, dass bei Verwendung eines verrauschten winner-take-all Mechanismus für die Netzwerkdynamik die einzelnen Schichten eines vorwärtsgerichteten Netzes sich thermisch nicht im Gleichgewicht befinden. Dies steht im Gegensatz zum softmax Mechanismus, der definitionsgemäß während des gesamten Lernvorgangs die konstante Temperatur β^{-1} vorgegeben bekommt[5], da als Größe der synaptischen Änderung $\delta = 1 = \text{const.}$ verwendet wurde.
Ein Unterschied in den beiden Lernregeln bestand darin, dass sich für die Lernregel mit Synapsengedächtnis die approximierten Temperaturen in Schicht 2 und 3 für unterschiedliche Werte von Θ qualitativ sehr unterscheiden, wohingegen sie für die Lernregel mit Neuronengedächtnis qualitativ und sogar quantitativ sehr ähnlich sind. Dies spiegelt die unterschiedliche Struktur beider Lernregeln wider.

Als mögliche Anwendung im Bereich der Neurobiologie und kognitiven Psychologie könnte man die in diesem Kapitel vorgestellte Methode der approximativen Zuordnung einer Temperatur dazu verwenden, die physiologische mit der behavioristischen Ebene zu koppeln. Dies könnte dadurch erreicht werden, dass man die Korrespondenz zwischen (beobachtetem) mittleren Fehler und Abkühlverhalten (in jeder Schicht des neuronalen Netzes) ausnutzt. Denn in meinem abstrakten, simplifizierten Modell kodiert der zeitliche Verlauf der approximierten Temperatur den zeitlichen Verlauf der Güte des Verhaltens. Anders ausgedrückt: die Richtung der Temperaturänderung gibt die Richtung des Lernvorgangs an. Somit ließe sich physiologischen Messungen an Neuronen, die sich lateral inhibieren (winner-take-all Mechanismus), eine approximierte Temperatur zuordnen deren zeitlicher Verlauf dann mit dem zeitlichen Verlauf der Güte des Verhaltens verglichen werden könnte.
Wenn meine Hypothese richtig ist, dann müsste die Richtung der Änderung in Temperatur und Güte übereinstimmen.

[5]Siehe die Diskussion in Kapitel 3.2.3 über die Abhängigkeit der Parameter β und δ beim softmax Mechanismus.

Kapitel 5

Komplexe Netzwerke

5.1 Einleitung

In den vorangegangenen Kapiteln wurde eine Hebb-artige Lernregel mit Neuronengedächtnis eingeführt. Deren Eigenschaften wurden anhand spezieller Beispiele untersucht und mit anderen Lernregeln verglichen. Neben dem guten Konvergenzverhalten und der Flexibilität bzgl. der drei die Lernregel konstituierenden Parameter Θ, τ und α, konnte die Lernregel mit Neuronengedächtnis qualitativ als heterosynaptische LTD biologisch plausibel interpretiert werden.

In diesem Kapitel werden weitere Eigenschaften der Lernregel mit Neuronengedächtnis untersucht, die sich ergeben, wenn man den Einfluss der Netzwerktopologie, und des verwendeten Aktions-Selektions-Mechanismus, auf die Konvergenzeigenschaften untersucht. Als spezielle Netzwerktopologie wird hier eine ganze Klasse von Graphen untersucht, die als *small world* Netze bezeichnet wird. Hierdurch soll der generische Charakter der Lernregel mit Neuronengedächtnis unterstrichen werden. Der Einfluss des verwendeten Aktions-Selektions-Mechanismus stellt eine Verbindung zur Wahrnehmungs-Handlungs-Schleife her und offenbart einen interessanten Unterschied zwischen verschiedenen Lernregeln.
Als zugrunde liegende Aufgabe, die das neuronale Netze lernen soll, wird in diesem Kapitel nicht das XOR-Problem verwendet. Statt dessen soll Timing zwischen Input- und Outputmustern in seiner einfachsten Form gelernt werden.

Zuerst wird in Kapitel 5.2. der Konstruktionsalgorithmus von *small world* Netzen und deren Eigenschaften vorgestellt und diskutiert. Danach wird in Kapitel 5.3. der Einfluss der Netzwerktopologie und des verwendeten Aktions-Selektions-Mechanismus auf verschiedene Lernregeln untersucht.

Zum Schluss werden die erhaltenen Resultate interpretiert und Konsequenzen diskutiert.

5.2 Grundlagen

In den vergangenen Jahren hat ein besonderer Netzwerktyp mehr und mehr Aufmerksamkeit erregt. Dieser Netzwerktyp wird als *small world* bezeichnet. Das besondere an dieser Netzwerktopologie, welche durch solch ein *small world* Netz beschrieben wird, ist die Tatsache, dass es sowohl Eigenschaften von regulären (hoher Clusterungskoeffizient C) als auch von zufälligen Netzwerken (niedrige charakteristische Pfadlänge L) vereint.
Die ursprüngliche Idee solcher Netztopologien reicht in die 50'er und 60'er Jahre des letzten Jahrtausends zurück, als sich der Theoretiker Anatol Rapoport mit Krankheitsausbreitungen in Population beschäftigte [Rapoport, 1953a,b], bzw. der Psychologe Stanley Milgram Bekanntheitsgrade in der Gesellschaft untersuchte [Milgram, 1967]. Seit dem haben Netzwerktopologien, die weder rein regulär, noch zufällig sind, zunehmend Verwendung bei der Beschreibung scheinbar unterschiedlicher Phänomene im Bereich biologischer (neuronaler Netze, Protein Faltung), sowie sozialer (wissenschaftliche und künstlerische[1] Kollaboration) und kommunikations Netze (Internet, *World-Wide Web*) gefunden. Für einen Überblick siehe [Watts, 1999; Albert und Barabasi, 2002].
Die Renaissance dieser *small world* Netze in den letzten Jahren geht auf [Watts und Strogatz, 1998] zurück, da es ihnen gelungen ist, ein sehr einfaches Modell, welches nur einen einzigen Parameter p_{rw} besitzt, vorzustellen, um kontinuierlich von einem regulären zu einem zufälligen Netzwerk überzugehen.
Mit der Konstruktion dieses Modells und dessen Eigenschaften befassen sich die Kapitel 5.2.1 und 5.2.2.

5.2.1 Konstruktion von *small world* Netzwerken

Ausgangspunkt der Konstruktion von *small world* Netzwerken bildet ein eindimensionaler, regulär verknüpfter Ring. Dieser Ring besteht aus N Knoten und jeder dieser N Knoten besitzt K Kanten. Die Kanten stellen die Verbindungen zwischen den einzelnen Knoten dar.
Um solch einen regulär verknüpften, eindimensionalen Ring zu erhalten, geht man von einem gedachten Ring aus, auf dem die N Knoten (äquidistant[2]) verteilt sind. Jeden dieser Knoten verbindet man dann mit seinen $\frac{K}{2}$ nächsten Nachbarn auf jeder Seite. Hieraus entsteht ein Netzwerk, wie es in Abbildung

[1] Filmschauspieler
[2] Aus ästhetischen Gründen, dies ist jedoch für die Funktion ohne Bedeutung.

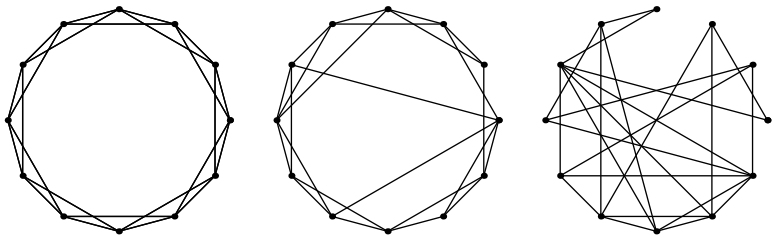


Abbildung 5.1: Darstellung verschiedener Netzwerktopologien, welche sich aus einer Neuverknüpfungen der Kanten durch das von [Watts und Strogatz, 1998] vorgeschlagene Verfahren ergeben. Links: Reguläre Verknüpfung, $p_{rw} = 0.0$. Mitte: *small world* Verknüpfung, $p_{rw} = 0.2$. Rechts: Random Verknüpfung, $p_{rw} = 1.0$.

5.1 auf der linken Seite dargestellt ist. Nun kann man den von [Watts und Strogatz, 1998] vorgeschlagenen Algorithmus verwenden, um eine *small world* Topologie zu erhalten.

Algorithmus 5.1 (Konstruktion von *small world* Netzwerken:)

1. *wähle mit der Wahrscheinlichkeit* $\frac{1}{N}$ *einen Knoten* $n \in N$ *aus*
2. *initialisiere einen Zähler* c*, der den Abstand zum nächsten Nachbarn (Knoten) angibt;* $c = 1$
3. *löse die Verbindung vom* n*'ten Knoten zum* c*'ten nächsten Nachbarn mit der Umverknüpfungswahrscheinlichkeit* p_{rw} *(im Uhrzeigersinn)*
4. *falls die Verbindung gelöst wurde, stelle eine neue Verbindung mit einem anderen Knoten her. Die Wahrscheinlichkeit, dass ein Knoten ausgewählt wird, ist für alle Knoten gleich. Doppel- und Selbstverbindungen sind verboten. Falls der ausgewählte Knoten zu einer Doppel- oder Selbstverbindung führen würde, bleibt die ausgewählte Verbindung unverändert bestehen (Kantenerhaltung)*
5. *bestimme den nächsten Knoten durch* $n = n + 1 \bmod N$*. Falls der neue Knoten bereits ausgewählt wurde, erhöhe den Abstandszähler des nächsten Nachbarn;* $c = c + 1$.
6. *falls* $c > \frac{K}{2}$ *beende das Programm, falls nicht, springe zu Punkt 3.*

Dieser Konstruktionsalgorithmus besitzt nur einen einzigen Parameter p_{rw}. Der Parameter p_{rw} gibt die Umverknüpfungswahrscheinlichkeit an, mit der

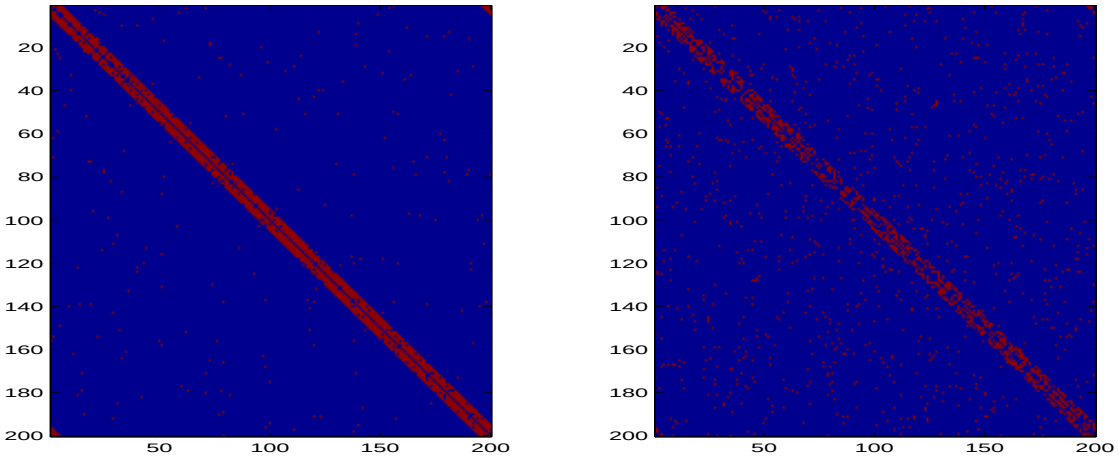


Abbildung 5.2: Darstellung der Verknüpfungsmatrix für $N = 200$ Knoten und $K = 10$ Kanten je Knoten, die durch Algorithmus 5.1 erzeugt wurde. Blau bedeutet, dass es keine Verbindung zwischen den entsprechenden Knoten (i'te Zeile zu j'ter Spalte) gibt. Bei rot existiert eine solche Verbindung. Links: Umverknüpfungsparameter $p_{rw} = 0.1$. Rechts: Umverknüpfungsparameter $p_{rw} = 0.5$.

eine bestehende Verbindung zwischen zwei Knoten gelöst und eine neue gesucht wird (*rewiring*).
Die Auswirkung des Parameters p_{rw} sind in Abbildung 5.1 für drei verschiedene Werte exemplarisch dargestellt. In der mittleren Darstellung ist $p_{rw} = 0.2$. Man erkennt, dass nur sehr wenige Verbindungen, verglichen mit der linken Abbildung, geändert wurden. Erhöht man den Parameter weiter, so nimmt die Anzahl der geänderten Verknüpfungen zu und führt für $p_{rw} = 1.0$ auf ein zufällig verknüpftes Netzwerk. Dies ist in der rechten Abbildung in 5.1 dargestellt.
Durch den Parameter p_{rw} ist es also möglich, ein reguläres Netzwerk ($p_{rw} = 0.0$) in ein zufälliges Netzwerk ($p_{rw} = 1.0$) kontinuierlich überzuführen. In Abbildung 5.2 ist die Verknüpfungsmatrix zwischen den Knoten für zwei unterschiedliche Umverknüpfungswahrscheinlichkeiten p_{rw} graphisch dargestellt. Man erkennt, wie mit zunehmendem Umverknüpfungsparameter p_{rw}, die "Unordnung" immer weiter zunimmt.

5.2.2 Eigenschaften von *small world* Netzwerken

Um die Eigenschaften von *small world* Netzwerken charakterisieren zu können, wurden von D. Watts und S. Strogatz zwei Maße vorgeschlagen [Watts und Strogatz, 1998]: Der Clusterungskoeffizient C und die charakteristische Pfadlänge L.

5.2.2.1 Clusterungskoeffizient C

Der Clusterungskoeffizient C beschreibt eine lokale Eigenschaft von Netzwerken. Er berechnet sich aus den gemittelten lokalen Clusterungskoeffizient C_i für jeden Knoten i des Netzes,

$$C_i = \frac{\#\text{der Kanten in } G_i}{\text{maximale } \# \text{ aller Kanten in } G_i} \tag{5.1}$$

Hierbei bezeichnet G_i den Untergraphen des gesamten Netzwerkes, welcher nur aus dem Knoten i und den mit diesem verbundenen Knoten, einschließlich all ihrer Kanten untereinander, besteht. Für einen Knoten i mit k_i Verbindungen zu anderen Knoten, gibt es demnach

$$\text{maximale } \# \text{ aller Kanten in } G_i = \frac{k_i(k_i - 1)}{2} \tag{5.2}$$

Verbindungen. Der Faktor 2 bedeutet, dass es sich um einen ungerichteten Graphen handelt. Anschauliche kann man den Clusterungskoeffizient C in einem sozialen Netzwerk Interpretieren, wenn man sich vorstellt, dass ein Freund deines Freundes mit einer gewissen Wahrscheinlichkeit auch wieder dein Freund ist.
In Abbildung 5.3 ist der normierte Clusterungskoeffizient C für ein Netz mit $N = 200$ Knoten und $K = 10$ Verbindungen pro Knoten in Abhängigkeit vom Umverknüpfungsparameter p_{rw} dargestellt. Die Normierung ist auf $p_{rw} = 0$ bezogen, da ein reguläres Netz die größte Clusterung besitzt. Für zunehmende Werte von p_{rw} wird der Clusterungskoeffizient C immer kleiner, da die Wahrscheinlichkeit einer Neuverknüpfung, mit einem weiter entfernt liegenden Knoten, immer mehr zunimmt.

5.2.2.2 Charakteristische Pfadlänge L

Die charakteristische Pfadlänge L ist als der Mittelwert der kürzesten Verbindungslänge d_{ij}, zwischen Knoten i und Knoten j, in einem Netzwerk G definiert.

$$L = \frac{1}{N(N-1)} \sum_{i \neq j \in G} d_{ij} \tag{5.3}$$

Für einen ungewichteten Graphen entspricht die kürzeste Verbindung d_{ij}, von Knoten i zu Knoten j, exakt der minimalen Anzahl von Kanten, welche auf dem Weg von i nach j überquert werden muss.
Damit die charakteristische Pfadlänge eine wohldefinierte Größe bleibt, muss sichergestellt sein, dass zwischen zwei beliebigen Knoten ein Verbindungsweg existiert. Graphen, für die solch ein Verbindungsweg für alle Paare von Knoten existiert, bezeichnet man als verknüpft (*connected*). Für random Graphen

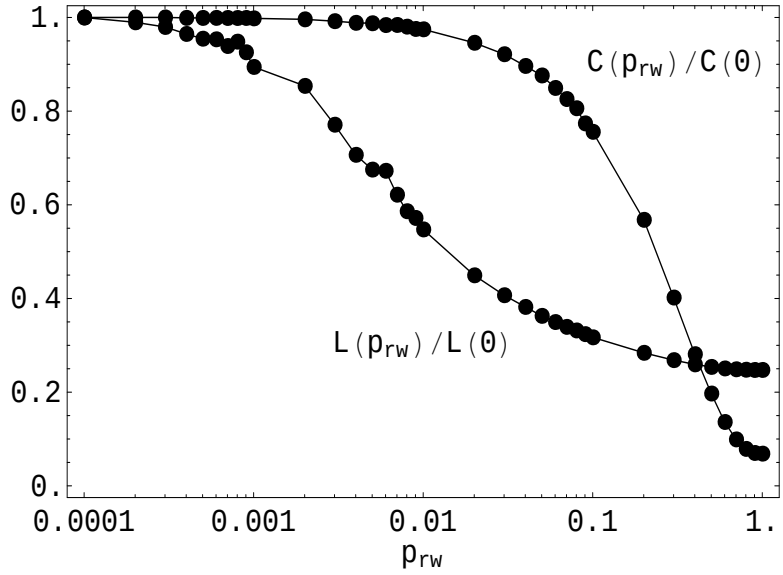


Abbildung 5.3: Darstellung des auf $p_{rw=0}$ normierten Clusterungskoeffizienten C und der charakteristischen Pfadlänge L in Abhängigkeit verschiedener Werte für den Umverknüpfungsparameter p_{rw}. Das untersuche Netzwerk besteht aus $N = 200$ Knoten und $K = 10$ Verbindungen pro Knoten. Die Bezugspunkte der Normierung sind: $C(0) = 1.01$ und $L(0) = 10.17$.

gibt es von Erdös und Rényi ein Theorem [Erdös und Rényi, 1959], welches garantiert, dass "fast alle" Zufallsgraphen für die $K \geq \ln(N)$ gilt, verknüpft sind.
Um die kürzeste Verbindung zwischen zwei Knoten effektiv berechnen zu können, wurden verschiedene Algorithmen entwickelt. Bei meinen Berechnungen habe ich den Algorithmus von [Dijkstra, 1959] verwendet. Eine Beschreibung der Arbeitsweise findet sich in Anhang A.2.
In Abbildung 5.3 ist die normierte charakteristische Pfadlänge L für ein Netz mit $N = 200$ Knoten und $K = 10$ Verbindungen pro Knoten in Abhängigkeit vom Umverknüpfungsparameter p_{rw} dargestellt. Die Normierung ist auf $p_{rw} = 0$ bezogen, da ein reguläres Netz die größte charakteristische Pfadlänge L besitzt. Dies liegt daran, dass durch Erhöhung der Umverknüpfungswahrscheinlichkeit p_{rw} die Wahrscheinlichkeit dafür steigt, dass es eine "Abkürzung" zwischen zwei, anfänglich weit "entfernter" Knoten, gibt[3].

[3]Die Anführungszeichen sollen darauf hinweisen, dass metrische Aussagen, wie "Abkürzung" oder "entfernter", nur für reguläre Netzwerke der Anschauung zugängig sind. Für zunehmende Werte von p_{rw} sind diese Begriffe nur noch abstrakt definiert und die Analogie zu regulären Netzwerken geht immer mehr verloren.

5.2.2.3 Zusammenfassung der Eigenschaften

Die Eigenschaften von *small world* Netzwerken kann man zusammenfassend so beschreiben, dass sie einen hohen Clusterungskoeffizient C und eine niedrige charakteristische Pfadlänge L, bezogen auf die Werte von regulären Netzen, also für $p_{rw} = 0$, aufweisen. Dies lässt sich sehr schön aus Abbildung 5.3 ablesen. Der Startwert von p_{rw}, für den dieses *small world* Verhalten einsetzt, wurde von D. Watts durch die Bedingung $p_{rw} \geq \frac{2}{NK}$ charakterisiert. Diese Bedingung garantiert die Existenz von mindestens einer Abkürzung zwischen zwei Knoten. Für das in Abbildung 5.3 dargestellte Netzwerk ergibt diese Bedingung $p_{rw} \geq 0.001$.

5.3 Lerndynamiken in *small world* Netzwerken

In den folgenden Abschnitten wird die Lerndynamik in *small world* Netzwerken untersucht. Hierzu wird die Lernregel mit Neuronengedächtnis mit der von P. Bak und D. Chialvo vorgeschlagenen Lernregel [Chialvo und Bak, 1999; Bak und Chialvo, 2001] verglichen. Es wird die Abhängigkeit von verschieden Parametern, wie der Umverknüpfungswahrscheinlichkeit p_{rw}, der Anzahl der zu lernenden Muster M und zwei unterschiedlichen Aktions-Selektions-Mechanismen untersucht. Die zugrundeliegende Aufgabe, welche gelernt werden soll, ist nicht mehr das XOR-Problem. Statt dessen soll das neuronale Netz Timing zwischen Input und Output, in seiner einfachsten Form lernen.
Hierbei ist der Vergleich des Einflusses der Aktions-Selektions-Mechanismen von besonderem Interesse, da dieser Parameter die behavioristische und somit die makroskopische Ebene beeinflusst. In den vorangegangenen Kapiteln wurden immer nur Parameter verändert, welche die mesoskopische Ebene des Gesamtsystems 3.1 betrafen, also eine direkte Veränderung innerhalb des neuronalen Netzes zur Folge hatten. Es ist natürlich unmittelbar einleuchtend, dass eine Veränderung innerhalb des neuronalen Netzes, Einfluss auf dessen Lerndynamik hat. Wie aber wirkt sich eine Beeinflussung von außen auf die Lerndynamik eines neuronalen Netzes aus?

5.3.1 Lernen von Timing

5.3.2 Einleitung

Um das generischen Verhalten der Lernregel mit Neuronengedächtnis zu demonstrieren, wird in den folgenden Abschnitt deren Anwendbarkeit auf *small world* Netzwerke untersucht. Hierfür gibt es zwei Gründe. Zum einen wurde in [Chialvo und Bak, 1999; Bak und Chialvo, 2001] die von P. Bak

und D. Chialvo vorgeschlagene Lernregel auf Zufallsnetze angewendet, welche sich aus dem Konstruktionsalgorithmus 5.1 von *small world* Netzen für den Umverknüpfungsparameter $p_{rw} = 1.0$ ergeben. Somit stellen die hier durchgeführten Untersuchungen eine Verallgemeinerung vorangegangener Arbeiten dar und besitzen zu diesen einen wohldefinierten Bezugspunkt.
Zum anderen weisen neuere Arbeiten eindeutig darauf hin, dass sowohl Cortico-corticale Verbindungen in Makaken und Katzen [Latora und Marchiori, 2001, 2002] basierend auf Daten aus [Scannel, 1997; Scannel et al., 1995], als auch Neuron-Neuron Kopplungen im *C. elegans* [Watts und Strogatz, 1998] basierend auf Daten aus [Achacoso und Yamamoto, 1992], Eigenschaften von *small world* Netzen aufweisen.

Die Untersuchung von *small world* Netzwerken liefert also einerseits Aussagen, welche mit den theoretischen Arbeiten von P. Bak und D. Chialvo verglichen werden können und sogar darüber hinaus gehen, und andererseits stellen sie einen Kontakt zur Neuroanatomie biologischer Systeme her, welche experimentell ermittelt wurde.

5.3.3 Beschreibung des zu lernenden Problems

Die Spezifizierung des Gesamtsystems, gemäß Definition 3.1 für das hier zu lernende Problem, ergibt:

Definition 5.1 (Gesamtsystem: Lernen in *small world* Netzen)
1. Topologie des Neuronalen Netzes: *Ein durch den Konstruktionsalgorithmus 5.1 erzeugtes* small world *Netzwerk mit* $N = 200$ *Neuronen und* $K = 10$ *Verbindungen je Neuron in Abhängigkeit des Umverknüpfungsparameters* p_{rw}.
2. Neuronenmodell: *Binäre Neuronen* $x_i \in \{0, 1\}$ *mit* $i \in \{1, \ldots, N\}$
3. Netzwerkdynamik: *Das innere Feld der Neuronen berechnet sich durch* $h_j = \sum_i^{all} w_{ji} x_i$ *,wobei "all" alle Neuronen bezeichnet, welche mit dem* j*'ten Neuron verbunden sind. Die Aktivität ergibt sich aus einem winner-take-all Mechanismus und entspricht somit einer extremalen Dynamik.*
4. Lernregel: *Die Änderung der Synapsen erfolgt durch*
- 4.a) Lernregel mit Synapsengedächtnis mit $\Theta = 0$*, gemäß Punkt 4. in Definition 3.2 und* $\delta \in [0, 1]$
- 4.b) Lernregel mit Neuronengedächtnis mit $\Theta = 3$, $\tau = 2.0$ *und* $\alpha = 0$*, gemäß Definition 3.3 und* $\delta \in [0, 1]$
5. Äußere Umgebung: *Lerne eine Abbildung von Inputneuron* $x_{5(m-1)+1}$ *auf Outputneuron* $x_{100+(m-1)5}$ *für* $m \in M$ *Muster in exakt* $T = 4$ *Zeitschritten.*
6. Wechselwirkung zwischen TBM und Umgebung:
- 6.a) Aktions-Selektions-Mechanismus I.: Jedes Inputmuster wird unabhängig gewählt und mit gleicher Wahrscheinlichkeit präsentiert.
- 6.b) Aktions-Selektions-Mechanismus II.(siehe Definition 5.2)

Zu dieser Definition des Gesamtsystems sollen einige Bemerkungen gemacht werden, um die einzelnen Komponenten in 5.1 klarer zu machen.

Die Lernregel mit Synapsengedächtnis 4.a) stellt für den Spezialfall $\Theta = 0$ und $\delta \in [0, 1]$ die von [Chialvo und Bak, 1999; Bak und Chialvo, 2001] verwendete Lernregel dar, die jede Synapse bei Beteiligung an einer falschen Berechnung des neuronalen Netzes infolge des Reinforcementsignals deprimiert.

Die äußere Umgebung, welche der zu lernenden Abbildung entspricht, besteht darin, dass es für jedes Inputmuster m ein Neuron $x_{5(m-1)+1}$ gibt, welches hierdurch aktiviert wird. Nach exakt $T = 4$ Zeitschritten muss dann das zugehörige Outputneuron $x_{100+(m-1)5}$ aktiv sein. Es sei ausdrücklich darauf hingewiesen, dass die Abbildung auch dann als falsch gelernt angesehen wird, wenn zwar nach exakt T Zeitschritten das vorher definierte Zielneuron aktiv ist, jedoch zuvor ein anderes, zu einem anderen Inputmuster gehörende Outputneuron, aktiv gewesen war. Salopp gesprochen soll so etwas wie Timing gelernt werden, da es nicht nur wichtig ist irgendwann das Richtige zu tun, sondern zu einem bestimmten Zeitpunkt.
Die verwendete Netzwerkdynamik ist ein winner-take-all Mechanismus, der auf eine extremale Dynamik, wie in den vorangegangenen Kapiteln auch, führt. Hieraus folgt, dass das Outputneuron zum Zeitschritt $t - 1$ und das Inputneuron zum Zeitpunkt t nicht gleichzeitig aktiv sind. Somit wird das neuronale Netz weder als Attraktornetzwerk[4] verwendet, bei dem so lange iteriert wird, bis die Netzwerkdynamik in einen Fixpunkt läuft, noch als rekurrentes Netz, welches die Aktivität der Inputneuronen mit denen der Interneuronen überlagert und daraus eine resultierende Aktivität berechnet.
In den Arbeiten von [Chialvo und Bak, 1999; Bak und Chialvo, 2001] wurde ein ähnliches Problem behandelt. Dort wurde eine Abbildung als gelernt betrachtet, wenn nach maximal T_f Zeitschritten[5] ein vorher definiertes Zielneuron erreicht wurde. Dies bedeutet, dass die Abbildung als gelernt angesehen wird, gleichgültig ob das Zielneuron nach einem oder nach T_f Zeitschritten feuert. Zwar ist es einleuchtend, dass bis zu einem gewissen Zeitschritt T_f eine Antwort des neuronalen Netzes erfolgen soll, jedoch ist nicht klar, warum es möglich sein soll, instantan eine Antwort geben zu können. Vielmehr muss man davon ausgehen, dass zwischen Input- und Outputneuronen keine direkte Verbindung besteht, so dass es einiger Verarbeitungsschritte bedarf, um einen Output erzeugen zu können. Aus diesem Grund wurde das Problem auch so umdefiniert, dass dies berücksichtigt wird.
Die Problematik lässt sich an einem anschaulichen Szenario darstellen. Ein Affe bekommt unterschiedliche Symbole präsentiert und soll durch drücken

[4] Wie z.B. ein Hopfieldnetz [Hopfield, 1982].
[5] Die genaue Angabe der Größe T_f fehlt jedoch!

einer Taste entscheiden, um was für ein Symbol es sich gehandelt hat. Durch einen Lehrer bekommt er mitgeteilt, ob seine Antwort richtig oder falsch war [Strassinopoulos und Bak, 1995].

Die Wechselwirkung zwischen TBM und Umgebung wird auf zwei unterschiedliche Arten realisiert. Im einen Fall werden die Inputmuster unabhängig und gleichwahrscheinlich gewählt und präsentiert (Punkt 6.a). Im andern Fall wird ein Aktions-Selektions-Mechanismus verwendet, um gezielt Inputmuster auswählen zu können (Punkt 6.b). Der Begriff "Aktion" soll hier hervorheben, dass das neuronale Netz nicht nur passiv seine Umgebung "beobachtet", sondern aktiv selektieren kann, welche Sinnesreize als nächstes verarbeitet werden sollen. Man könnte sagen, dass dies gewissermaßen eine Verkörperung (*embodiment*) des reinen neuronalen Netzes darstellt, indem man diesen Aktions-Selektions-Mechanismus als Bewegungsapparat und Entscheidungsfähigkeit eines Tieres interpretiert. Metaphorisch ausgedrückt erhält der im vorigen Abschnitt eingeführte Affe Arme und Hände, um sich die Inputmuster selbst auswählen zu können.
Hierdurch werden jetzt die bereits in Kapitel 3. in Definition 3.1 eingeführten Bezeichnungen klarer, da biologische neuronale Netze immer an Körper (Tiere oder Menschen) gekoppelt sind, welche mit ihrer Umgebung in Wechselwirkung stehen. Hierdurch gelangt man auf natürliche Weise zur Wahrnehmungs-Handlungs-Schleife, die in Kapitel 6. ausgiebig diskutiert wird.

Die Klasse der möglichen Aktions-Selektions-Mechanismen umfasst alle zeitabhängigen Wahrscheinlichkeitsverteilungen über der Menge der Inputmuster. Deshalb kann man insbesondere die gleichverteilte Auswahl der Inputmuster als Aktions-Selektions-Mechanismus bezeichnen, wie unter Punkt 6.a) auch geschehen. Dies ist jedoch nur ein Spezialfall, was man schon alleine daran ablesen kann, dass die Wahrscheinlichkeit, Muster m auszuwählen, zeitunabhängig $p(m) = \frac{1}{M}$ ist. Man kann dies als Aktions-Selektions-Mechanismus ohne Gedächtnis interpretieren, denn die Wahrscheinlichkeit $p(m)$ ist nicht adaptiv bzgl. irgendeiner internen oder externen Größe.
Als spezielle Wahl für den zweiten Mechanismus dient der Algorithmus, der in 5.2 definiert wird. Dieser Aktions-Selektions-Mechanismus (ASM) II. ist, im Gegensatz zu ASM I., adaptiv und besitzt somit ein Gedächtnis. Dieser Adaptionsmechanismus ist durch die *If*-Abfragen unter 3. und 4. realisiert.
Bei diesem Pseudocode bezeichnet pc (*pattern counter*) einen Musterzähler, der angibt, wieviele Muster bereits richtig gelernt wurden. Die Präsentation der numerierten Muster pattern erfolgt also immer sukzessive bis zum $pc + 1$'ten Muster. Falls $pc = M$ Muster richtig gelernt wurden, wird der Lernvorgang abgebrochen und die bis dahin vergangene Zeit T_{FPT} ausgegeben. Hierbei

bezeichnet T_{FPT} (*first-passage time*) den Zeitpunkt, an dem das neuronale Netz zum ersten mal alle M Muster richtig gelernt hat. Dies wird im nächsten Abschnitt genauer diskutiert.
In zeitabhängigen Wahrscheinlichkeiten ausgedrückt kann man dies als sukzessiv verschobenes Kronecker Delta δ auf den numerierten Mustern betrachten, wobei die Verschiebung gemäß $\text{pattern} = \text{pattern} + 1 \bmod (pc + 1)$ erfolgt. Dies ändert jedoch nichts an der notwendigen *If*-Abfrage, die durchgeführt werden muss um festzustellen, wieviele Muster bisher gelernt wurden. Aus diesem Grund erscheint die algorithmische Angabe in Pseudocode sinnvoller.

Definition 5.2 (Aktions-Selektions-Mechanismus II.:)

1. *Initialisierung: pc=0, pattern=1*
2. *Loop*
3. *If (pc=M) Then*

   3.a) Exit $\Longrightarrow$ *Ausgabe von* T_{FPT}

   3.b) Else

   pattern=pattern+1 mod (pc+1)

   End
4. *If (alle Muster bis einschließlich das pc+1'te richtig gelernt) Then*

   pc=pc+1

   End
5. *End Loop (springe zu 2.)*

5.3.4 Ergebnisse

In diesem Abschnitt werden die Resultate der Simulationen vorgestellt. Zuvor soll jedoch kurz auf die Methoden eingegangen werden, die zur Auswertung der Simulationsergebnisse verwendet werden.
Im Gegensatz zu den bisherigen Kapiteln wird hier nicht der mittlere Fehler 3.9 und 3.10 zur Charakterisierung des Konvergenzverhaltens des neuronalen Netzes verwendet, sondern die *first-passage time* T_{FPT}. Hierfür gibt es theoretische und praktische Gründe. Aus theoretischer Sicht stellt die *first-passage time* T_{FPT} eine Verbindung zur Theorie der stochastischen Prozesse dar, wie unten ausführlich dargestellt. Und aus Sicht der praktischen Handhabbarkeit der Ensemblesimulationen ist es effektiver nur einen Wert je Ensemblemitglied

abzuspeichern wie die ganze Zeitreihe der Fehler $e(t)$ bis einschließlich T_{FPT}. Insbesondere im Hinblick darauf, dass die hier durchgeführten Simulation sich über $\sim 10^7$ Zeitschritte erstrecken, wohingegen in den bisherigen Kapiteln im Mittel bis $\sim 10^4$ Zeitschritte simuliert wurde.
Um die *first-passage time* T_{FPT} zu erhalten, wird das durch 5.1 definierte System in Abhängigkeit der zu untersuchende Parameter so lange iteriert, bis alle Inputmuster auf die richtigen Outputneuronen abgebildet werden. Danach wird die Simulation abgebrochen, da das gestellte Problem gelernt wurde. Die Anzahl der Iterationsschritte die vergehen, bis die Abbildung richtig gelernt wurde, wird als *first-passage-time* T_{FPT} bezeichnet. Sie wird durch den jeweiligen Aktions-Selektions-Mechanismus, siehe Definition 5.2, ausgegeben. Somit wählt der Aktions-Selektions-Mechanismus nicht nur die Muster aus, die dem neuronalen Netz präsentiert werden, sondern entscheidet auch darüber, wann eine Simulation abgebrochen wird. Das Anhalten der Simulation wird also als Aktion aufgefasst. Falls das Neuronale Netz die Abbildung nicht lernen kann, wird die Simulation nach $T = 10^7$ Zeitschritten abgebrochen.

Die Charakterisierung physikalischer, chemischer und biologischer Systeme durch die *first-passage time* T_{FPT} bis zum Erreichen einer bestimmten Schwelle, stellt ein aus formaler Sicht fundiertes Konzept dar, da es eine direkte Verknüpfung mit der Theorie der Stochastischen Prozesse gestattet[6] [Redner, 2001; Honerkamp, 1990; Orr und Leen, 1993]. In unserem speziellen Fall ist der zugrundeliegende stochastische Prozess der individuelle Absolutfehler eines neuronalen Netzes,

$$E(t)_{iabs} = \frac{\#\text{der bereits gelernten Muster zum Zeitpunkt } t}{M} \tag{5.4}$$

Hierbei bezeichnet M die Anzahl aller zu lernenden Muster.
Da das neuronale Netz zufällig initialisiert wird, bezüglich Synapsenstärke und Verknüpfungen zwischen den Neuronen, und die Änderung der Synapsenstärke während des Lernprozesses ebenfalls stochastisch erfolgt, da die Größe der synaptischen Gewichte δ gleichverteilt aus dem Intervall $[0, 1]$ gezogen wird, wird hierdurch der dem individuellen Absolutfehler $E(t)_{iabs}$ zugrundeliegende stochastische Prozess definiert. Aufgrund der Normierung in 5.4 wird $E(t)_{iabs}$ auf diskrete Werte zwischen Null und eins beschränkt. Hierdurch wird die Bezeichnung als Fehler gerechtfertigt. In Abbildung 5.4 (d) ist $E(t)_{iabs}$ exemplarisch für ein spezielles Beispiel dargestellt.

Wenn man nun den Zeitpunkt des stochastischen Prozesses wissen möchte, an dem der individuelle Absolutfehler $E(t)_{iabs}$ zum ersten Mal den Wert Null annimmt, so wird dieser Zeitpunkt T_{FPT} durch diesen Prozess zu einer Zufallsvariable. Für die Zufallsvariable der *first-passage time* T_{FPT} gibt

[6] Dies gilt sicherlich nicht für den in Gleichung 3.9 und 3.10 definierten mittleren Fehler.

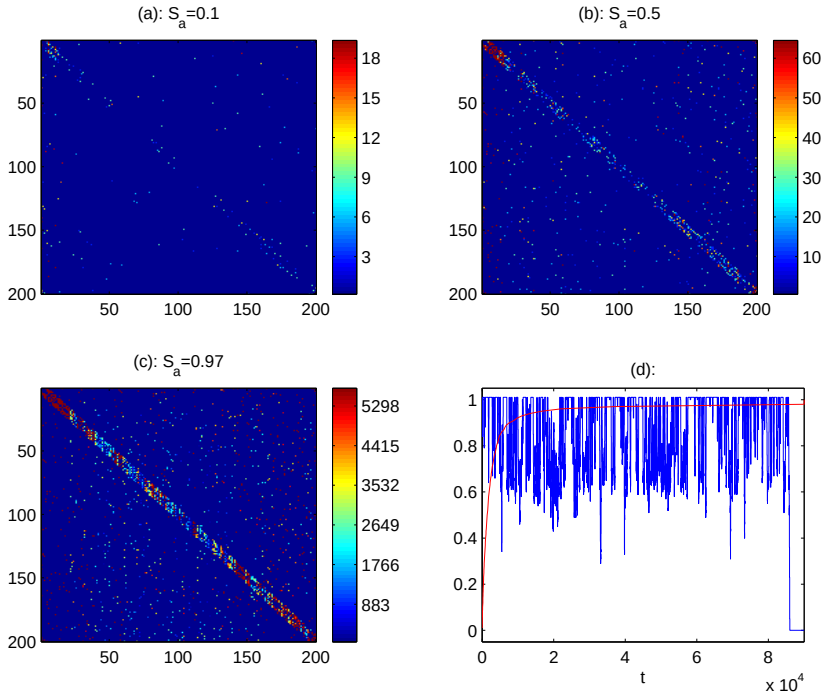


Abbildung 5.4: Darstellung eines Lernprozesses für Lernregel 4.b), $M = 3$ Muster, Aktions-Selektions-Mechanismus I. und Umverknüpfungsparameter $p_{rw} = 0.5$. (d): Darstellung des über 100 Zeitschritte gemittelten individuellen Absolutfehlers $E(t)_{iabs}$ (blaue Kurve) und der relativen Anzahl der am Lernprozess bisher beteiligten Synapsen S_a (rote Kurve) in Abhängigkeit der Zeit t. (a) Häufigkeit der am Lernprozess für $S_a = 0.1/t = 151$ beteiligten Synapsen. (b) Häufigkeit der am Lernprozess für $S_a = 0.5/t = 1419$ beteiligten Synapsen. (c) Häufigkeit der am Lernprozess für $S_a = 0.97/t = 85995$ beteiligten Synapsen.

es also eine Wahrscheinlichkeitsverteilung p^{FPT}, die wiederum durch den zugrundeliegenden stochastischen Prozess $E(t)_{iabs}$ bestimmt ist und sich aus den Simulationen numerisch bestimmen lässt. In Abbildung 5.5 ist die Verteilung p^{FPT} exemplarisch für den Fall $p_{rw} = 1.0$ und $M = 3$ dargestellt. Die Muster wurden gemäß Aktions-Selektions-Mechanismus I. präsentiert und die Synapsen des neuronale Netz durch die Lernregel 4.b) verändert. Typisch für alle Resultate (nicht gezeigt) ist der schnelle Anstieg und das sehr langsame Abklingen der Verteilung. Speziell bei dieser Simulation lag der größte Wert einer *first-passage time* bei $T_{FPT} = 374072$ und somit außerhalb des dargestellten Bereiches.

Mit Hilfe dieser Verteilung p^{FPT} lassen sich aussagekräftige Größen berechnen, durch die man ein besseres Verständnis für die Eigenschaften des stochastischen Prozesses $E(t)_{iabs}$, und somit für das Lernenverhalten neurona-

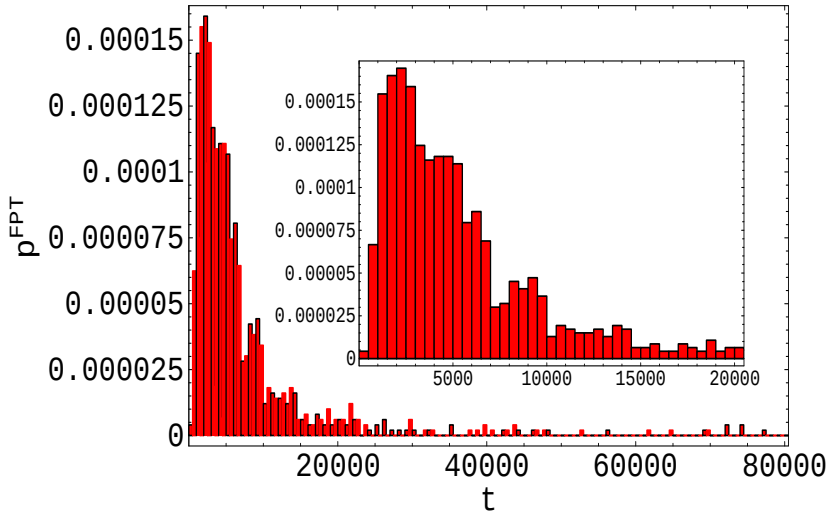


Abbildung 5.5: Darstellung der Verteilung p^{FPT} der *first-passage* Zeiten für den Umverknüpfungsparameter $p_{rw} = 1.0$ und $M = 3$ Muster, welche nach ASM I. präsentiert und nach Lernregel 4.b) gelernt wurden. Die Histogramme, deren Bingröße 500 Zeitschritte beträgt, wurden aus unabhängigen Simulationen eines Ensembles der Größe $N = 1000$ ermittelt. Die innere Abbildung stellt eine Vergrößerung der Verteilung dar.

ler Netze bekommt.

Verteilungsfunktion P_T^E
Die erste Größe gibt die Wahrscheinlichkeit P_∞^E dafür an, dass die zu passierende Schwelle E überhaupt erreicht wird. Sie berechnet sich durch Aufsummation der Wahrscheinlichkeitsverteilung p^{FPT}.

$$P_\infty^E = \sum_t^\infty p_t^{FPT} \tag{5.5}$$

Da bei den durchgeführten Simulationen natürlich nicht bis unendlich, sondern immer nur bis zu endlichen Zeiten T simuliert werden kann, lässt sich nur

$$P_T^E = \sum_t^T p_t^{FPT} \tag{5.6}$$

berechnen. Dies ist die Wahrscheinlichkeit, dass die Schwelle E bis zum Zeitpunkt T erreicht wurde. Das stellt so lange kein Problem dar, wie die gesamte Verteilung p^{FPT} vollständig im Intervall $[0, T]$ liegt, da in diesem Fall

die beiden Größen P_∞^E und P_T^E identisch sind. Für Verteilungen p^{FPT}, welche diese Voraussetzung nicht erfüllen, kann P_∞^E durch P_T^E nur abgeschätzt werden.

mean first-passage time $< T_{FPT} >$
Die zweite wichtige Größe ist die mittlere Zeit, *mean first-passage time* $< T_{FPT} >_\infty$, bis zum erreichen der Schwelle E. Sie ergibt sich direkt aus der Verteilung p^{FPT} durch Wichtung aller Zeitpunkte.

$$< T_{FPT} >_\infty = \sum_t^\infty t p_t^{FPT} \tag{5.7}$$

Wiederum gelten die gleichen Bemerkungen zu $< T_{FPT} >_\infty$ wie oben für P_∞^E aufgrund der endlichen Simulationszeit T, so dass die *mean first-passage time* $< T_{FPT} >_T$ im Fall einer Verteilung p^{FPT}, die nicht auf $[0, T]$ beschränkt ist, lediglich eine Abschätzung nach unten darstellt. Im folgenden wird aus Gründen der Einfachheit immer $< T_{FPT} >=< T_{FPT} >_T$ verwendet.

Bevor nun die Ergebnisse präsentiert werden, soll anhand Abbildung 5.4 die Dynamik des neuronalen Netzes für ein Beispiel exemplarisch veranschaulicht werden. Mit Hilfe von Lernregel 4.b) und Aktions-Selektions-Mechanismus I. wurde ein neuronales Netzwerk mit $p_{rw} = 0.5$ so trainiert, dass es $M = 3$ Muster richtig abbilden kann. Die rote Kurve in 5.4 (d) gibt den relativen Anteil der Synapsen an, der bis zu diesem Zeitpunkt bereits aktiv gewesen ist.

$$S_a = \frac{\#\text{der bis zu zum Zeipunkt t bereits aktiv gewesenen Synapsen}}{\#\text{aller Synapsen}} \tag{5.8}$$

Mit aktiv ist hier die Beteiligung an der Weiterleitung des Signales innerhalb des neuronalen Netzes gemeint. Hierdurch ist S_a auf das Intervall $[0, 1]$ beschränkt. In den Abbildungen 5.4 (a) - (c) ist die Häufigkeit der aktiven Synapsen in Abhängigkeit von S_a zu den jeweiligen Zeitpunkten t_a aufgetragen. Insgesamt sieht man, dass recht schnell ($t \sim 20000$), verglichen mit der *first-passage time*, die für dieses Beispiel $T_{FPT} = 85995$ beträgt, sehr viele Synapsen ($S_a \sim 0.9$) an der Dynamik beteiligt waren. Dies hat den Vorteil, dass schnell ein großer Bereich nach richtigen Pfaden zwischen Input- und Outputneuronen "abgesucht" wird. Ein Nachteil könnte darin liegen, dass bereits bestehende, richtig gelernte Pfade, durch zu schnelles Ändern der Synapsenstärken wieder verlernt bzw. zerstört werden.

p_{rw}	M=3, ASM I.	M=5, ASM I.	M=3, ASM II.	M=5, ASM II.
0.1	1000/52.7	100/0	1000/57.4	500/1.0
0.2	1000/94.7	100/3.5	1000/97	500/7.2
0.3	1000/99.9	100/12	1000/99.9	500/30.5
0.4	1000/100	100/30.9	1000/100	500/53.3
0.5	1000/100	500/48	1000/100	500/76.6
0.6	1000/100	500/63.2	1000/100	500/85.6
0.7	1000/100	500/80	1000/100	500/92.4
0.8	1000/100	500/87.1	1000/100	500/97.4
0.9	1000/100	500/94.7	1000/100	500/97.4
1.0	1000/100	500/96.2	1000/100	500/99.0

p_{rw}	M=3, ASM I.	M=5, ASM I.	M=3, ASM II.	M=5, ASM II.
0.1	500/23.6	-	500/31.2	-
0.2	500/75.0	-	500/74.4	-
0.3	500/92.6	-	500/91.4	-
0.4	500/97.8	-	500/97.6	-
0.5	500/99.4	-	500/99.8	-
0.6	500/99.8	-	500/99.6	100/0.0
0.7	500/100	-	500/99.8	100/1.0
0.8	500/100	100/0.0	500/100	100/4.0
0.9	500/100	100/3.7	500/100	100/3.0
1.0	500/100	100/4.4	500/100	100/6.0

Tabelle 5.1: Auflistung der verwendeten Ensemblegröße und der Prozentzahl der neuronalen Netze, die nach $T = 10^7$ Zeitschritten die vorgegebene Abbildung richtig gelernt haben (Ensemblegröße/Prozent). Obere Tabelle: Lernregel 4.b) wurde zur Änderung der Synapsen verwendet. Untere Tabelle: Hier wurde Lernregel 4.a) verwendet. Alle Ergebnisse sind in Abhängigkeit vom Umverknüpfungsparameter p_{rw}, der Anzahl der Muster M und dem verwendeten Aktions-Selektions-Mechanismus (ASM) angegeben.

5.3.4.1 Aktions-Selektions-Mechanismus I.

In diesem Abschnitt werden die Ergebnisse für den Aktions-Selektions-Mechanismus I. in Abhängigkeit von der Anzahl der zu lernenden Muster M, dem Umverknüpfungsparameter p_{rw}, und der verwendeten Lernregel 4.a) bzw. 4.b) dargestellt. Die Ensemblegrößen der Simulationen sind für die jeweiligen Parameter in Tabelle 5.1 angegeben.

Lerne $M = 3$ Muster:
In Abbildung 5.6 ist die kumulative Verteilungsfunktion 5.6 in Abhängigkeit

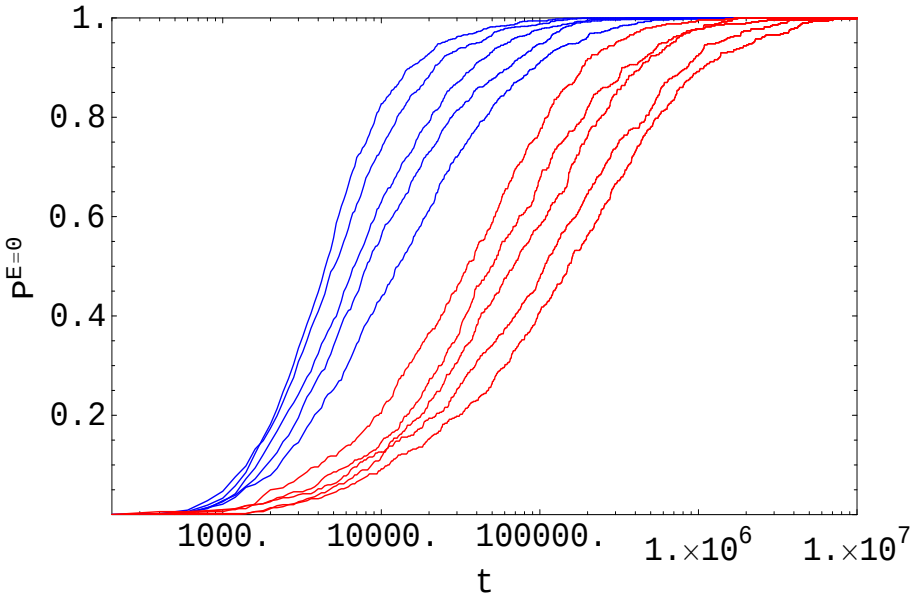


Abbildung 5.6: Darstellung der Verteilungsfunktion $P^{E=0}$ in Abhängigkeit der Zeit für $M = 3$, die mit Lernregel 4.a) (rot) bzw. Lernregel 4.b) (blau) ermittelt wurde. Die Kurven sind jeweils von oben nach unten durch $p_{rw} = 1.0$ bis $p_{rw} = 0.6$ parametrisiert. Verwendet wurde jeweils ASM I.

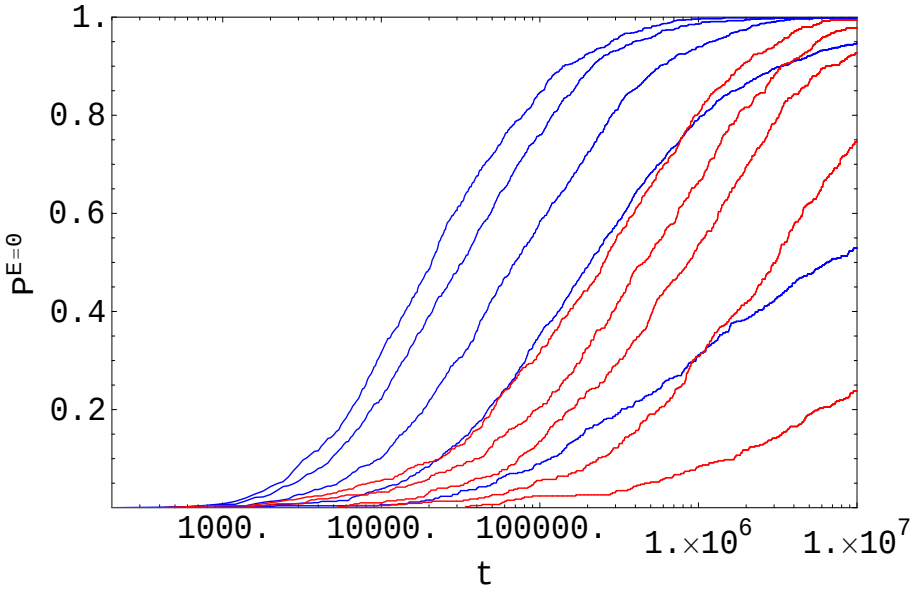


Abbildung 5.7: Darstellung der Verteilungsfunktion $P^{E=0}$ in Abhängigkeit der Zeit für $M = 3$, die mit Lernregel 4.a) (rot) bzw. Lernregel 4.b) (blau) ermittelt wurde. Die Kurven sind jeweils von oben nach unten durch $p_{rw} = 0.5$ bis $p_{rw} = 0.1$ parametrisiert. Verwendet wurde jeweils ASM I.

der Zeit aufgetragen. Bei den blauen Kurven wurde das neuronale Netz mit Lernregel 4.b) und bei den roten mit Lernregel 4.a) trainiert. Die Kurven sind jeweils von oben nach unten durch $p_{rw} = 1.0$ bis $p_{rw} = 0.6$ parametrisiert.
Man erkennt, dass Lernregel 4.b) deutlich bessere Resultate als Lernregel 4.a) erzielt. Darüber hinaus kann jedes Netzwerk, welches mit Lernregel 4.b) trainiert wurde, die vorgegebene Abbildung innerhalb der Simulationszeit lernen. Dies trifft auf Lernregel 4.a) nicht zu. Für $p_{rw} = 0.6$ haben nach $T = 10^7$ Zeitschritten nur 99.8% der trainierten Netze die Abbildung vollständig gelernt, siehe Tabelle 5.1.
Die Resultate für die Parameterwerte $p_{rw} = 0.5$ bis $p_{rw} = 0.1$ sind in Abbildung 5.7 dargestellt. Wiederum gilt für alle Parameterwerte von p_{rw}, dass Lernregel 4.b) deutlich besser ist als Lernregel 4.a). Ab $p_{rw} = 0.3$ kann jedoch auch mit Lernregel 4.b) innerhalb der vorgegebenen Simulationszeit die Abbildung nicht für das gesamte Ensemble richtig gelernt werden, siehe Tabelle 5.1.

Lerne $M = 5$ Muster:
In Abbildung 5.8 sind die zu Abbildung 5.6 und 5.7 analogen Ergebnisse für $M = 5$ Muster dargestellt. Die zu Lernregel 4.b) gehörenden Kurven sind wiederum blau und die Kurven für Lernregel 4.a) sind rot eingezeichnet. Hier sind die Ergebnisse sogar noch eindeutiger, da bis zum Ende der Simulationszeit nur wenige Prozent der neuronalen Netze, welche mit Lernregel 4.a) trainiert wurden, überhaupt die Abbildung richtig lernen konnten. Aus diesem Grund sind auch nur die Verteilungsfunktionen P_t^E für die Parameterwerte $p_{rw} = 1.0$ und $p_{rw} = 0.9$ eingezeichnet, da für weiter abnehmende Werte von p_{rw} die Resultate so schlecht werden, dass ein Lernen der neuronalen Netze gar nicht mehr innerhalb der Simulationszeit stattfindet, siehe Tabelle 5.1.
Hier deutet sich zum ersten Mal ein gravierender Unterschied zwischen beiden Lernregeln an. Dieser Unterschied besteht nicht darin, dass die eine Lernregel ein Netzwerk schneller trainieren kann als die andere. Der eigentliche Unterschied liegt darin, dass möglicherweise eine vorgegebene Abbildung unter bestimmten Bedingungen mit Lernregeln 4.a) gar nicht richtig gelernt werden kann. Um dies zu untersuchen, wurde das Langzeitverhalten der Lerndynamik einzelner neuronaler Netze für $p_{rw} = 1.0$ gesondert untersucht. Hierfür wurde die Simulationszeit auf $T = 10^9$ Schritte ausgedehnt[7].
Für Lernregel 4.b) konnte kein Fall gefunden werden, für den die Abbildung nicht gelernt werden konnte. Die größten vorkommenden *first-passage* Zeiten lagen in der Größenordnung bei $T_{FPT} \sim 10^8$. Für Lernregel 4.a) sieht dies

[7] Aus diesem Grund wurden auch nur einzelne neuronale Netze untersucht und keine Ensemble, da im Rahmen der zur Verfügung stehenden Rechner die Rechenzeit sonst auf nicht mehr tolerierbare Größenordnungen ansteigt.

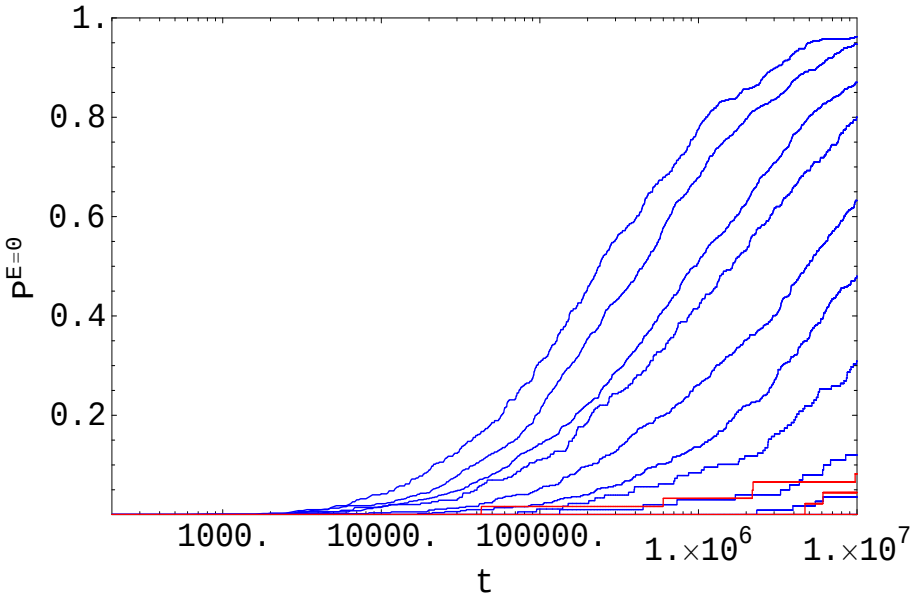


Abbildung 5.8: Darstellung der Verteilungsfunktion $P^{E=0}$ in Abhängigkeit der Zeit für $M = 5$, die mit Lernregel 4.a) (rot) bzw. Lernregel 4.b) (blau) ermittelt wurde. Die Kurven sind für Lernregel 4.b). von oben nach unten durch $p_{rw} = 1.0$ bis $p_{rw} = 0.1$ parametrisiert. Für Lernregel 4.a) sind nur $p_{rw} = 1.0$ und $p_{rw} = 0.9$ dargestellt. Verwendet wurde jeweils ASM I.

anders aus. Auch nach einer Simulationszeit von $T = 10^9$ Schritten gibt es neuronale Netze, welche die vorgegebene Abbildung nicht richtig lernen konnten. Prozentuale Angaben können hier nicht gemacht werden, da, wie oben bereits erwähnt, kein Ensemble untersucht wurde, sondern nur einzelne neuronale Netze.

Somit kann man festhalten, dass es bis zu praktikablen Simulationszeiten von $T = 10^9$ Zeitschritten immer neuronale Netze gibt, welche mit Lernregel 4.a) trainiert wurden, die eine vorgegebene Abbildung nicht richtig lernen konnten. Dies gilt für Lernregel 4.b) nicht. Dies stellt einen signifikanten Unterschied zwischen beiden Lernregeln dar. Der Fall für $T \to \infty$ wird in der Diskussion behandelt.

5.3.4.2 Aktions-Selektions-Mechanismus II.

In diesem Abschnitt werden die Ergebnisse für den Aktions-Selektions-Mechanismus II. in Abhängigkeit von der Anzahl der zu lernenden Muster M, dem Umverknüpfungsparameter p_{rw} und der verwendeten Lernregel 4.a) bzw. 4.b) dargestellt. Die Ensemblegrößen der Simulationen sind in Tabelle 5.1 angegeben.

Lerne $M = 3$ Muster:
In Abbildung 5.9 ist die Verteilungsfunktion 5.6 in Abhängigkeit der Zeit aufgetragen. Bei den blauen Kurven wurde das neuronale Netz mit Lernregel 4.b) und bei den roten mit Lernregel 4.a) trainiert. Die Kurven sind jeweils von oben nach unten durch $p_{rw} = 1.0$ bis $p_{rw} = 0.6$ parametrisiert.

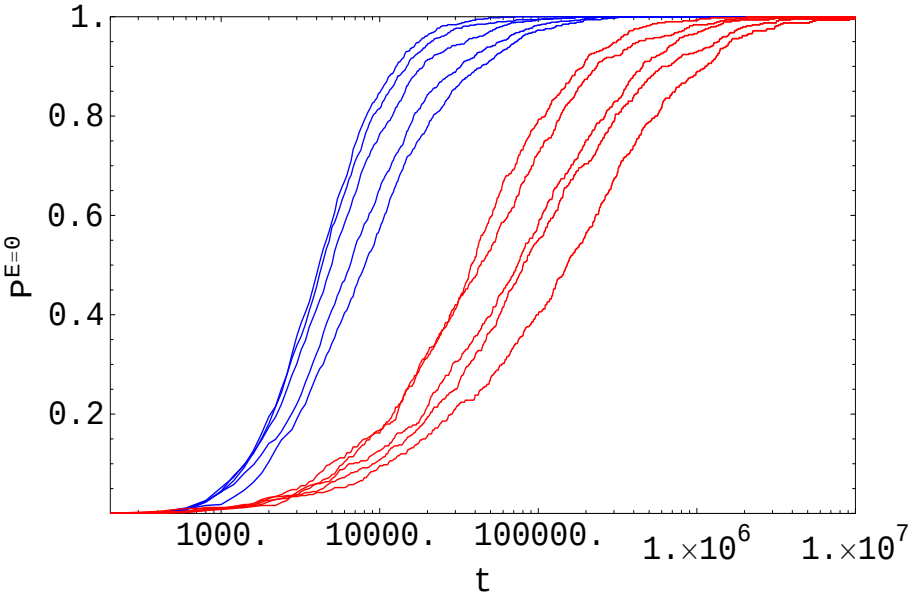


Abbildung 5.9: Darstellung der Verteilungsfunktion $P^{E=0}$ in Abhängigkeit der Zeit für $M = 3$, die mit Lernregel 4.a) (rot) bzw. Lernregel 4.b) (blau) ermittelt wurde. Die Kurven sind jeweils von oben nach unten durch $p_{rw} = 1.0$ bis $p_{rw} = 0.6$ parametrisiert. Verwendet wurde jeweils ASM II.

Man erkennt, dass Lernregel 4.b) wieder deutlich bessere Resultate erzielt, als Lernregel 4.a). Darüber hinaus kann jedes neuronale Netz, welches mit Lernregel 4.b) trainiert wurde, die vorgegebene Abbildung innerhalb der Simulationszeit lernen. Dies trifft auf Lernregel 4.a) wieder nicht zu. Für $p_{rw} = 0.7$ bzw. $p_{rw} = 0.6$ haben nach $T = 10^7$ Zeitschritten nur 99.8% bzw. 99.6% der neuronalen Netze des Ensembles die Abbildung vollständig gelernt.
Die Resultate für die Parameterwerte $p_{rw} = 0.5$ bis $p_{rw} = 0.1$ sind in Abbildung 5.10 dargestellt. Wiederum gilt für alle Parameterwerte von p_{rw}, dass Lernregel 4.b) deutlich besser ist als Lernregel 4.a). Ab $p_{rw} = 0.3$ kann jedoch auch mit Lernregel 4.b) innerhalb der vorgegebenen Simulationszeit die Abbildung nicht für das gesamte Ensemble richtig gelernt werden, siehe Tabelle 5.1.

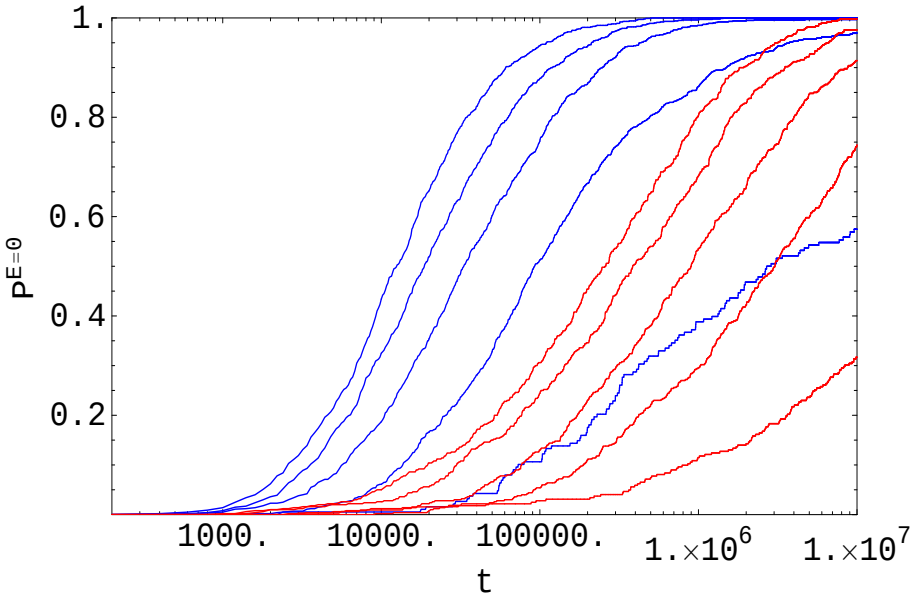


Abbildung 5.10: Darstellung der Verteilungsfunktion $P^{E=0}$ in Abhängigkeit der Zeit für $M = 3$, die mit Lernregel 4.a) (rot) bzw. Lernregel 4.b) (blau) ermittelt wurde. Die Kurven sind jeweils von oben nach unten durch $p_{rw} = 0.5$ bis $p_{rw} = 0.1$ parametrisiert. Verwendet wurde jeweils ASM II.

Lerne $M = 5$ Muster:
In Abbildung 5.11 ist das zu Abbildung 5.8 analoge Ergebnis für Aktions-Selektions-Mechanismus II. dargestellt. Die zu Lernregel 4.b) gehörenden Kurven, welche in Abbildung 5.11 blau eingezeichnet sind, sind für die Umverknüpfungsparameter $p_{rw} = 1.0$ bis $p_{rw} = 0.1$ dargestellt. Für Lernregel 4.a) (rote Kurven) sind nur die Parameter $p_{rw} = 1.0$ bis $p_{rw} = 0.7$ eingezeichnet, da für noch kleinere Parameterwerte von p_{rw} kein Lernverhalten innerhalb der vorgegebenen Simulationszeit mehr zu beobachten ist.
Auch hier wurden einzelne neuronale Netze mit $p_{rw} = 1.0$ gesondert untersucht und die Simulationszeit auf $T = 10^9$ ausgedehnt. Die erzielten Resultate stimmen mit denen überein, die für Aktions-Selektions-Mechanismus I. erhalten wurden. Somit gibt es neuronale Netze, die mit Lernregel 4.a) trainiert wurden, welche eine vorgegebene Abbildung auch nach einer Simulationszeit von $T = 10^9$ nicht lernen können.

Ein interessanter Unterschied zwischen Aktions-Selektions-Mechanismus I. und II. für Lernregel 4.b) ist, dass die neuronalen Netze für die ASM II. verwendet wurde, schneller konvergieren als neuronalen Netze mit ASM I. Diese Eigenschaft wird in Kapitel 5.3.4.4. genauer untersucht.

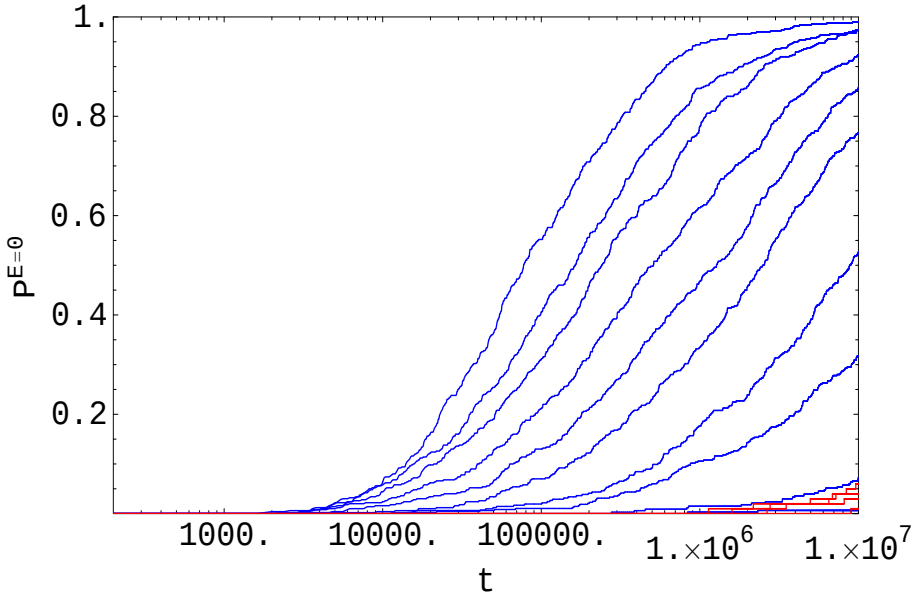


Abbildung 5.11: Darstellung der Verteilungsfunktion $P^{E=0}$ in Abhängigkeit der Zeit für $M = 5$, die mit Lernregel 4.a) (rot) bzw. Lernregel 4.b) (blau) ermittelt wurde. Die Kurven sind für Lernregel 4.b) von oben nach unten durch $p_{rw} = 1.0$ bis $p_{rw} = 0.1$ parametrisiert. Für Lernregel 4.a) sind nur $p_{rw} = 1.0$ bis $p_{rw} = 0.7$ dargestellt. Verwendet wurde jeweils ASM II.

5.3.4.3 Diskussion

Als zusammenfassende Resultate kann man festhalten, dass Lernregel 4.b) statistisch gesehen immer besser ist als Lernregel 4.a), für korrespondierende Parameterwerte und sogar darüberhinaus. Dies impliziert insbesondere, dass die Hebb-artige Lernregel mit Neuronengedächtnis nicht nur für die Netzwerktopologie aus Kapitel 3. und 4. anwendbar ist, sondern auch für viel allgemeinere Netze, die sich durch den Konstruktionsalgorithmus 5.1 realisieren lassen. Interessant ist, dass sich *small world* Netze, also Netze im Parameterbereich $0.1 \leq p_{rw} \leq 0.3$, durch Lernregel 4.b) viel besser lernen lassen, als mit Lernregel 4.a). Dies ist deshalb von Interesse, da biologische neuronale Netze eher solch einer Topologie zuzuordnen sind als einem Zufallsnetzwerk [Watts und Strogatz, 1998; Latora und Marchiori, 2001].
Ein weiteres Resultat besteht darin, dass es für bestimmte Parameterwerte, nämlich $M = 5$ Muster, Umverknüpfungsparameter $p_{rw} \leq 1.0$ und ASM I. und II., immer neuronale Netze gibt, welche innerhalb der Simulationszeit eine vorgegebene Abbildung nicht durch Verwendung von Lernregel 4.a) lernen können. Dagegen blieb ungeklärt, ob dies nur daran lag, dass die Simulationszeit immer noch zu niedrig war, oder ob dies auch für unendlich lange Simulationszeiten gilt.

Ich vertrete den Standpunkt, dass alle neuronalen Netze, welche mit Lernregel 4.a) trainiert wurden, die Abbildung für oben angegebene Parameter im Limes gegen unendlich lernen können. Die Begründung hierfür wird durch Abbildung 5.4 (d) gegeben, welche in sehr ähnlicher Form die Verhältnisse für Lernregel 4.a) beschreibt. Die relative Anzahl der am Lernprozess beteiligten Synapsen S_a beträgt schon nach wenigen Zeitschritten, verglichen mit der Gesamtsimulationszeit, über 50%. Für Lernregel 4.a) verläuft dieser Prozess sogar noch schneller, da immer alle falschen Synapsen verändert werden und nicht nur einige. Somit besitzt Lernregel 4.a) das Potential, den Raum aller möglichen Pfade in einem neuronalen Netz von den Input- zu den Outputneuronen zu durchsuchen. In Kombination mit dem negativen Reinforcementsignal ergibt sich hieraus sogar eine gerichtete Suche, die den Raum aller möglichen Pfade nicht nur zufällig durchläuft.
Diese Argumentation charakterisiert meiner Meinung nach die Stärke der von [Chialvo und Bak, 1999; Bak und Chialvo, 2001] vorgeschlagenen Lernregel weitgehend richtig. Tatsache ist jedoch auch, dass diese Lernregel, verglichen mit Lernregel 4.b), für endliche Zeiten beliebig schlechte Ergebnisse liefern kann.

5.3.4.4 Vergleich ASM I. vs. ASM II.

In Abbildung 5.12 ist die *mean first-passage time* $< T_{FPT} >$ in Abhängigkeit vom Umverknüpfungsparameter p_{rw} aufgetragen. Diese wurde aus den erhaltenen Wahrscheinlichkeitsverteilungen p^{FPT} der bisherigen Simulationen berechnet. Bei den blauen Kurven wurde Aktions-Selektions-Mechanismus I. bei den roten ASM II. verwendet. Die "$\Box$" entsprechen Verteilungen p_{FPT}, die nicht auf das Simulatonsintervall $[0, T]$ beschränkt waren, wohingegen die "$*$" für beschränkte Verteilungen stehen. Zu Lernregel 4.b) gehören die unteren beiden Kurven, die für $M = 3$ Muster erhalten wurden und die oberen beiden, bei denen das neuronale Netz $M = 5$ Muster lernen musste. Die beiden mittleren Kurven wurden mit Lernregel 4.a) für $M = 3$ Muster erhalten. Aus Abbildung 5.12 folgen zwei bemerkenswerte Eigenschaften.

Erstens erkennt man deutlich, dass für Lernregel 4.a) der jeweils verwendete Aktions-Selektions-Mechanismus keinen Einfluss auf die mittleren *first-passage* $< T_{FPT} >$ Zeiten hat, wohingegen Lernregel 4.b) klar unterschiedliche Resultate für ASM I. und ASM II. liefert. Hieraus folgt, dass ein Aktions-Selektions-Mechanismus ein zusätzlicher Parameter des Gesamtsystems sein kann, der einen signifikanten Einfluss auf die Lerndynamik von neuronalen Netzen haben kann.
Durch die Einführung eines zusätzlichen Parameters in Form des Aktions-Selektions-Mechanismus konnte also ein charakteristischer Unterschied zwischen beiden Lernregeln herausgearbeitet werden, der normalerweise

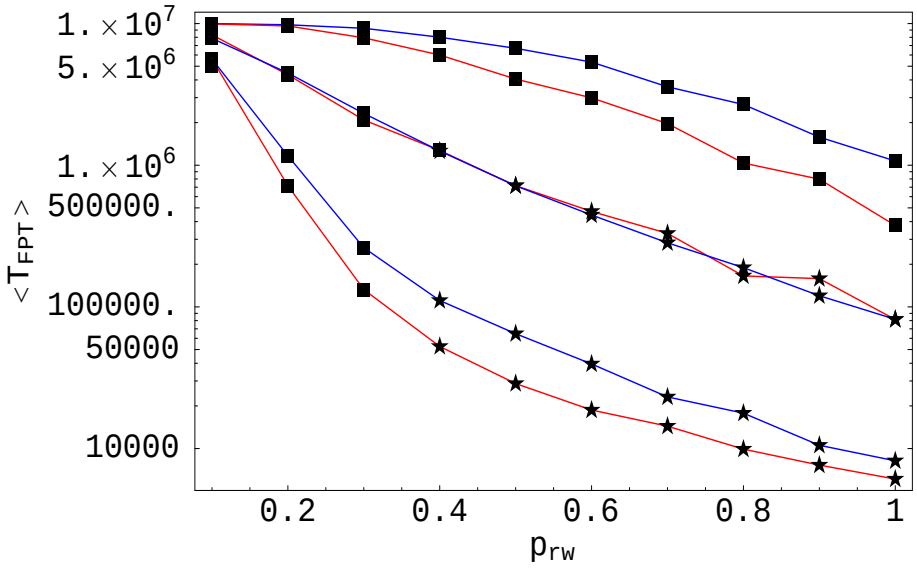


Abbildung 5.12: Darstellung der *mean first-passage time* $< T_{FPT} >$ in Abhängigkeit des Umverknüpfungsparameter p_{rw}. Die blauen Kurven resultieren aus Simulationen mit ASM I., die roten aus Simulationen mit ASM II. Für die unteren beiden Kurven wurde Lernregel 4.b) und $M = 3$ Muster verwendet, für die mittleren beiden Lernregel 4.a) und $M = 3$ Muster und für die oberen beiden Lernregel 4.b) und $M = 5$ Muster. "$*$" bedeuten, dass die Verteilung P_T^E ganz im Simulationsintervall liegt, wohingegen "$\square$" daraufhindeuten, dass dies nicht der Fall ist und die eingezeichneten Werte nur Abschätzungen nach unten darstellen.

vergessen wird, da in den allermeisten Fällen nur ASM I. zum Training neuronaler Netze verwendet wird.
Der Grund, warum beide ASM die gleichen Simulationsergebnisse für Lernregel 4.a) liefern, liegt darin, dass beide ASM (annähernd) die gleiche Statistik haben, mit der die Muster präsentiert werden. Sie unterscheiden sich nur in der Reihenfolge, mit der sie präsentiert werden. Da Lernregel 4.a) jedoch kein Gedächtnis besitzt, kann es diese Eigenschaft nicht "erkennen"[8].
Für Lernregel 4.b) gilt dies nicht. Sie ist aufgrund ihres Neuronengedächtnisses in der Lage, sensitiv auf eine geänderte Reihenfolge der präsentierten Muster zu reagieren, indem die Neuronenzähler gezielt erniedrigt werden und somit die Wahrscheinlichkeit einer synaptischen Änderung im Falle einer falschen Berechnung reduziert wird.
Dieses Verhalten ist auch nicht unplausibel, wenn man die oben beschriebenen Aktions-Selektions-Mechanismen mit einem praktischen Beispiel, wie

[8]Die durchgeführten Simulationen haben gezeigt, dass überproportional viel Zeit für das Lernen des letzten Musters verwendet wird (nicht präsentiert).

etwa dem Lernen von Vokabeln, in Bezug setzt. ASM I. entspräche dem zufälligen Auswählen der Vokabeln, wohingegen ASM II. immer erst eine neue Vokabel hinzunimmt, wenn es alle alten richtig gelernt hat. Welche der beiden Strategien erfolgreicher ist, kann individuell verschieden sein, jedoch wird niemand bestreiten, dass es nicht egal ist, wie man beim Lernen, speziell von Vokabeln oder bei allgemeinen Problemen, vorgeht.

Auf der Basis der hier durchgeführten Simulationen stelle ich die Hypothese auf, dass ein biologisches System, ein Tier oder ein Mensch, seinen Aktions-Selektions-Mechanismus so adaptiert, dass er möglichst gut mit der Lernregel seines neuronalen Systems zusammenpasst. Hieraus folgt, dass man aufgrund eines bestimmten Verhaltens des biologischen Systems, bestimmte Lernregeln ausschließen könnte, da Simulationen deren Ineffektivität zeigen.
Das Neuronengedächtnis liefert also einen Mechanismus, mit dem unterschiedliche Beschreibungsebenen miteinander verknüpft werden können. Zwar ist die Synapsenänderung auf einer mesoskopischen Ebene angesiedelt[9] und das Verhalten des gesamten biologischen Systems auf einer makroskopischen Ebene, jedoch ist durch die Beeinflussung der Lerndynamik durch das Verhalten des biologischen Systems eine Art Kommunikation hergestellt. Diese Kommunikation basiert wesentlich auf dem Vorhandensein eines Reinforcementsignals und der Möglichkeit, dieses auch sinnvoll verwenden zu können, da ansonsten nicht beurteilt werden könnte, ob es besser ist, ein Problem in kurzer oder langer Zeit zu lösen. Für ein solches System wären diese beiden Aktions-Selektions-Mechanismen äquivalent.

Zusammenfassend kann man festhalten, dass nicht nur die synaptischen Änderungen zwischen den Neuronen bzw. allgemein die neuronale Dynamik einen Einfluss auf das Verhalten von Tieren und Menschen hat, sondern dass auch umgekehrt das Verhalten von Tieren und Menschen deren synaptischen Änderungen zwischen den Neuronen und hierdurch allgemein deren neuronale Dynamik beeinflusst. Dieser geschlossene Wechselwirkungszyklus wird als Wahrnehmungs-Handlungs-Schleife (*Action-Perception-Cycle*) bezeichnet. Aufgrund der herausragenden Bedeutung der Wahrnehmungs-Handlungs-Schleife auf dem Weg zu einer ganzheitlichen Beschreibung biologischer Systeme, wird sie in Kapitel 6. nochmals diskutiert.

Zweitens folgt aus Abbildung 5.12, dass die ohnehin schon hervorragenden Konvergenzeigenschaften neuronaler Netze, welche mit Lernregel 4.b) trainiert werden, nochmals durch Aktions-Selektions-Mechanismus II. gesteigert werden können. Dies gilt bis hin zu kleinen Werten von p_{rw}, bei

[9] Die mikroskopische Ebene besteht aus der Betrachtung von Ionenkanälen und der Dynamik einzelner Ionen.

denen es immer schwieriger wird, die vorgegebene Abbildung zu lernen, da immer weniger abkürzende Pfade existieren, um Input- mit Outputneuronen, innerhalb von vier Zeitschritten zu verknüpfen.
Tatsächlich wurde ASM II. nicht nur ausgewählt, weil es die gleiche Präsentationsstatistik wie ASM I. für die Muster hat, sondern weil untersucht werden sollte, ob sich mit einfachsten Mitteln eine Art *Vollständige Induktion des Lernens* realisieren lässt. Hierunter verstehe ich das systematische Aufbauen von Wissen auf bereits Gelerntem.
Dies erscheint zunächst nicht besonders interessant, sondern eher selbstverständlich. Jedoch besitzt ein biologisches neuronales Netz keinen übergeordneten Lehrer, der auf einer globalen Ebene entscheidet, wann welche Synapse wie geändert werden soll, um einerseits neues Wissen zu erwerben und andererseits altes Wissen nicht zu vergessen. Aus diesem Grund wurden in dieser Arbeit die Eigenschaften von neuronalen Netzen untersucht, welche eine Synapsenänderung durch eine lokale Regel [Hebb, 1949] in Kombination mit einem globalen Reinforcementsignal [Strassinopoulos und Bak, 1995] erfahren. Hierdurch besteht natürlich immer die Gefahr, dass eine Synapsenänderung Auswirkungen auf bereits richtig gelernte Pfade hat[10]. Es stellt sich also die Frage, wie man in solch einem neuronalen Netz in möglichst einfacher Weise so etwas wie eine Hierarchie einführen kann. Diese Frage lässt sich natürlich nicht unabhängig von den beteiligten Komponenten des Gesamtsystems 5.1 beantworten, da nur in deren Komposition alle Eigenschaften erkennbar werden.
Der Ansatz, um solch eine Hierarchie einzuführen, besteht meiner Ansicht nach im geschickten Ausnützen des Neuronengedächtnisses in Kombination mit dem Reinforcementsignal. Durch ein gezieltes Herabsetzen des Neuronenzählers kann die Wahrscheinlichkeit für eine Änderung der Synapsenstärken sehr klein werden, wodurch bereits gelernte neuronale Pfade geschützt werden. Dies wird indirekt durch Aktions-Selektions-Mechanismus II. bewirkt, denn das Muster $\mathfrak{m}+1$ wird erst hinzugenommen, wenn $\mathfrak{m}$ Muster richtig gelernt wurden. Wenn aber das neue Muster hinzugenommen wird, das mit einer hohen Wahrscheinlichkeit zuerst falsch zugeordnet wird, dann folgen hierauf $\mathfrak{m}$ Muster, welche die Neuronenzähler, infolge des positiven Reinforcementsignals, in diesen Pfaden reduzieren, was sie unsensitiver gegenüber einer Änderung der Synapsenstärke macht. Diese Eigenschaft ist in Lernregel 4.a), aufgrund des fehlenden Gedächtnisses, wie oben bereits diskutiert, nicht vorhanden.
Somit kann man mit Lernregel 4.b) Hierarchie in einer sehr einfachen Form, unter Verwendung von ASM II. und dessen Eigenschaften, in ein neuronales Netz einführen, ohne dass man einen globalen "Lehrer" braucht.

[10] Durch den Spezialfall der extremalen neuronalen Dynamik ist diese Metapher für die bisher untersuchten neuronalen Netze exakt.

5.3.5 Zusammenfassung

In diesem Kapitel wurden weitere Eigenschaften der Lernregel mit Neuronengedächtnis anhand spezieller Simulationen untersucht und Unterschiede zu der von [Chialvo und Bak, 1999; Bak und Chialvo, 2001] vorgeschlagenen Lernregel herausgearbeitet.

Hierbei ergab sich, dass die Lernregel mit Neuronengedächtnis für die gesamte Netzwerkklasse, welche durch den Konstruktionsalgorithmus 5.1 definiert ist, bis hinunter zu niedrigen Werten des Umverknüpfungsparameters p_{rw} ein sehr gutes Konvergenzverhalten zeigte. Somit können also insbesondere auch *small world* Netzwerke mit dieser Lernregel trainiert werden, die aus biologischer Sicht von besonderem Interesse sind.
Verglichen, mit der von [Chialvo und Bak, 1999; Bak und Chialvo, 2001] vorgeschlagenen Lernregel, sind die erhaltenen Resultate immer deutlich besser.

Ein weiterer Unterschied zwischen diesen beiden Lernregeln konnte durch die Einführung eines zusätzlichen Parameters in Form eines Aktions-Selektions-Mechanismus (ASM) herausgearbeitet werden. Hierbei konnte auf Lernregel 4.a) kein Einfluss der unterschiedlichen ASM festgestellt werden, wohingegen die Lernregel mit Neuronengedächtnis deutlich davon beeinflusst wurde. Somit konnte gezeigt werden, dass unter bestimmten Voraussetzungen das Verhalten auf makroskopischer Ebene die neuronale Dynamik auf mesoskopischer Ebene beeinflussen kann. Generell wurde durch die Einführung eines Aktions-Selektions-Mechanismus ein Zusammenhang zur Wahrnehmungs-Handlungs-Schleife, in einer sehr einfachen Form, hergestellt.

Darüberhinaus gab es nicht nur eine Beeinflussung durch die beiden unterschiedlichen Aktions-Selektions-Mechanismen auf die Lernregel mit Neuronengedächtnis, es konnte auch gezeigt werden, dass der triviale ASM I. schlechtere Konvergenzeigenschaften bewirkt, als ASM II., der sich als rudimentäre Form von hierarchischem Lernen interpretieren ließ.

Zusammenfassend kann man festhalten, dass die Untersuchungen in diesem Kapitel die generischen Eigenschaften der Lernregel mit Neuronengedächtnis unterstreichen und darüberhinaus die Notwendigkeit eines Gedächtnisses für eine biologisch plausible Lernregel demonstrieren, da nur infolge eines Gedächtnisses verschiedene Aktions-Selektions-Mechanismen unterschieden und somit vorteilhaft ausgenutzt werden können.

Kapitel 6

Aktive Computation in offenen Systemen

6.1 Einleitung

Eines der Hauptresultate des letzten Kapitels bestand darin, dass das Konvergenzverhalten der untersuchten neuronalen Netze wesentlich davon abhing, welcher Aktions-Selektions-Mechanismus in Kombination mit einer Lernregel verwendet wurde. So konnte gezeigt werden, dass aufgrund eines bestimmten Verhaltens, also eines bestimmten Aktions-Selektions-Mechanismus, die Dynamik der synaptischen Änderung so beeinflusst werden kann, dass die Konvergenz des neuronalen Netzes schneller erfolgt, verglichen mit einem anderen Aktions-Selektions-Mechanismus. Hierdurch wurde klar, dass es sich zwischen neuronaler Dynamik einerseits und Verhalten andererseits um eine geschlossene wechselseitige Beeinflussung handelt, welche als Wahrnehmung-Handlungs-Schleife (*Action-Perception-Cycle*) bezeichnet wird.
Um den Einfluss verschiedener Aktions-Selektions-Mechanismen auf das Gesamtsystem, also die Wahrnehmung-Handlungs-Schleife, zu untersuchen, werden die jeweils im Gesamtsystem 3.1 beschriebenen Komponenten, welche in den bisherigen Kapiteln zum Teil eher metaphorisch verstandenen wurden, durch der Anschauung zugängige ersetzt. Hierdurch gelangt man zu einer Beschreibungsebene, die ein einfaches biologisches System, welches im folgenden als autonomer Agent bezeichnet wird, in einer raumartigen Umgebung beschreibt. Da die Auswirkung eines Aktions-Selektions-Mechanismus auf die Wahrnehmung-Handlungs-Schleife nicht in allgemeiner Form behandelt werden kann, wird im folgenden als spezielles Problem das der Selbstlokalisation eines autonomen Agenten in einer Umgebung gewählt.

6.2 Überblick experimenteller und theoretischer Arbeiten

Die erste Quelle einer Arbeit, welche sich mit der Wahrnehmungs-Handlungs-Schleife im weiteren Sinne beschäftigt, stammt von dem italienischen Philosophen Thomas von Aquin[1] [Aquinas, 1272]. Bis heute beschäftigen sich mit der Wirkungsweise und dem Verständnis der Wahrnehmungs-Handlungs-Schleife viele Forscher aus so unterschiedlichen Bereichen wie Philosophie, Kognitive Psychologie [Hommel et al., 2002; Franz et al., 2000], Neuroinformatik [Weng et al., 2001; Arleo und Gerstner, 2000; Nolfi und Floreano, 1999] und Neurobiologie, jedoch ohne das Problem zu lösen. Hierbei verstehe ich unter der "Lösung des Problems" eine vollständige formale, mathematische Beschreibung des Phänomens, welche es gestattet, Vorhersagen zu machen, die sich experimentell verwerfen lassen.
Sogar in technisch angewandten Gebieten wie der Robotik [Brooks, 1999; Thrun, 1998; Oore et al., 1997] oder agentenbasierter Suchmaschinen für das Internet wird versucht das Problem der Wahrnehmungs-Handlungs-Schleife, welches in diesen Gebieten nur noch im übertragenen Sinn gilt, da die zugrundeliegenden Systeme keine biologischen mehr sind, in den Griff zu bekommen, bzw. durch dem jeweiligen Kontext angemessener Zusatzannahmen, handhabbar zu machen. Da sich beide Bereiche nicht mit Grundlagenfragen im Bereich der Naturwissenschaften befassen[2], wird im folgenden hierauf nicht weiter eingegangen. Jedoch soll nicht unerwähnt bleiben, dass es einen Ideen- und Methodentransfer zwischen Grundlagenforschung und technisch angewandter Forschung gibt, so dass der Ursprung von Ideen nicht immer eindeutig zurückverfolgt werden kann.

Wie in der Einleitung bereits kurz erwähnt ist es nicht möglich, die Wahrnehmungs-Handlungs-Schleife in allgemeiner Form zu untersuchen. Dies liegt daran, dass die Wahrnehmungs-Handlungs-Schleife an allen sensomotorischen Integrationen beteiligt ist, welche einem Lebewesen zur Verfügung stehen. Insbesondere ist diese Integration also multimodal, d.h. sie betrifft alle Sinnesorgane, was zu einer nichtlinearen Überlagerung bzw. Verarbeitung im neuronalen System führt, da das Gehirn nichtlinear aus einzelnen Modulen aufgebaut ist, die jeweils auf bestimmte Modalitäten spezialisiert sind und in Abhängigkeit von allen anderen Modulen auf die Motorik einwirken. Deshalb muss immer ein spezielles Beispiel ausgewählt werden, an dem dann spezielle Fragen beantwortet werden können. Aus diesem Grund konzentriere ich mich im folgenden auf Arbeiten, die sich mit

[1] Diese Quelle wird von W. Freeman in [Freeman, 2000] angeführt.

[2] Dies liegt an der Verwendung von Annahmen und Nebenbedingungen, welche nicht notwendigerweise biologisch plausibel sein müssen.

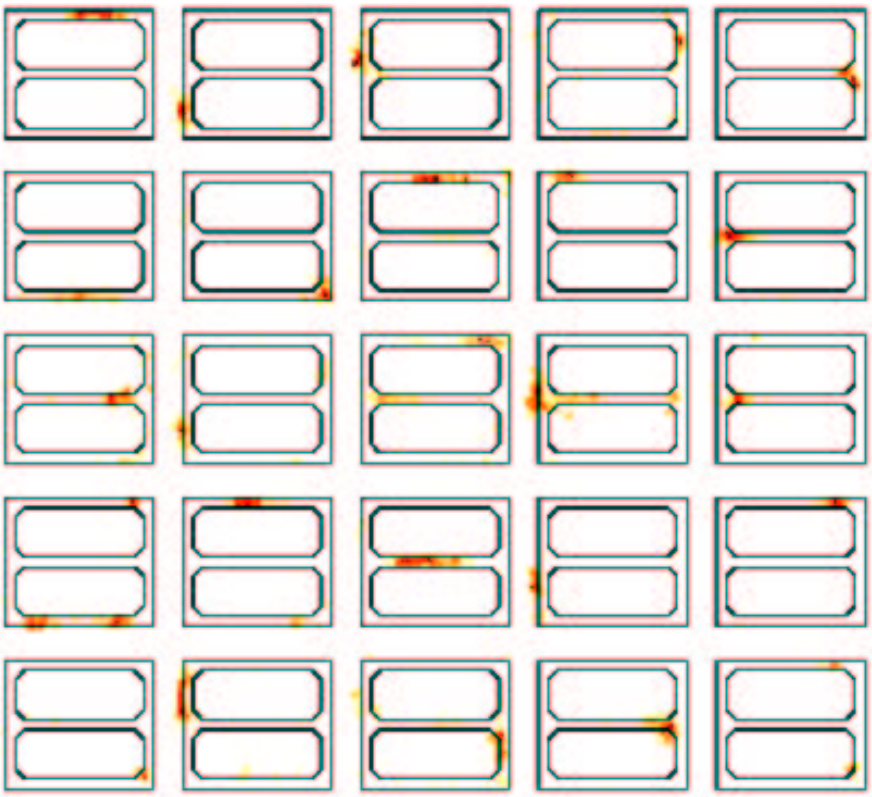


Abbildung 6.1: Darstellung von 25 gleichzeitig abgeleiteten *place cells* im Hippocampus einer Ratte, die sich in einem acht-förmigen Labyrinth bewegt. Die Ratte läuft eine geschlossene Acht, wobei sie den oberen Kreis im Uhrzeigersinn durchläuft. Die *place fields* der Ratte wurden durch unterschiedliche Farben kenntlich gemacht, wobei rot einer hohen und gelb einer niedrigen Feuerrate der jeweiligen *place cell* entspricht. (Abbildung ist aus [Zhang et al., 1998] entnommen.)

der Lokalisation von Ratten in einem Labyrinth und der zugrundeliegenden neuronalen Aktivität im Hippocampus beschäftigen.

An freilaufenden Ratten wurden im Hippocampus sogenannte Platz Zellen (*place cells*) gefunden. Dabei handelt es sich um Pyramidalzellen in CA1 und CA3, die immer dann verstärkt feuern, wenn sich die Ratte an einem bestimmten Ort (*place field*) in ihrer Umgebung befindet, unabhängig von nur einem bestimmten lokalen sensorischen Input [O'Keefe und Dostrovsky, 1971; O'Keefe, 1976]. Dies deutet darauf hin, dass die Gesamtheit des sensorischen Inputs aus verschiedenen Modalitäten, die für den jeweiligen Ort charakteristisch sind, das Herausbilden der *place cells* und deren Unabhängigkeit von einzelnen Stimuli, bewirkt. Da verschiedene *place cells* an verschiedenen

place fields in der Umgebung feuern, kodieren sie zusammengenommen die momentane Position der Ratte [Wilson und McNaughton, 1993].
In Abbildung 6.1 ist dies für ein spezielles Experiment von [Zhang et al., 1998] zur Veranschaulichung dargestellt. Abgebildet ist das Feuerverhalten von 25 gleichzeitig abgeleiteten Neuronen (*place cells*) und deren zugehörigen *place fields* in einem acht-förmigen Labyrinth. Die unterschiedlichen Farben kodieren die unterschiedlichen Feuerraten der jeweiligen Neuronen. Hierbei steht rot für eine höhere Feuerrate als gelb.

Desweiteren wurden sogenannte *Head-direction* Zellen im dorsalen Post Subiculum gefunden [Taube et al., 1990]. Wie der Name schon erahnen lässt, kodieren diese Zellen die Richtung des Kopfes der Ratte, und zwar unabhängig von der Stellung des Körpers und vom momentanen Ort innerhalb der Umgebung.
Solche Zellen, welche sensitiv auf die Orientierung des Kopfes antworten, wurden von [O'Keefe und Nadel, 1978] vorhergesagt, da sie schon damals vermuteten, dass der Hippocampus als kognitive Karte bei der räumlichen Navigation von Nagetieren eine wichtige Rolle spielen könnte.

Es soll nicht unerwähnt bleiben, dass der Hippocampus nicht nur Relationen zwischen räumlichen, sondern auch zwischen nicht-räumlichen Größen herstellen kann. So führten [Wood et al., 1999] ein Experiment durch, bei dem Ratten an bestimmten Stellen, die mit einem bestimmten Geruch belegt waren, nach Futter graben mussten. Die Ratten wurden nur dann belohnt, wenn sie an einer Stelle gruben, die anders roch als die letzte, an der sie gruben. Die Aufgabe bestand also darin, sich an die Relation bestimmter Gerüche zu erinnern.
Als Resultat fanden sie unterschiedliches Feuerverhalten von Neuronen. Einige feuerten nur bei bestimmten Gerüchen. Andere antworteten nur, wenn der gegenwärtige Geruch wie der vorhergehenden war, ansonsten blieben sie ruhig. Wieder andere Zellen reagierten auf Kombinationen von Eigenschaften. Die Behauptung, dass der Hippocampus nicht nur bei der räumlichen, sondern auch bei der nicht-räumlichen Repräsentation von Relationen eine Rolle spielt, wird schon lange vermutet und durch das Experiment von [Wood et al., 1999] erhärtet und in besonders schöner Weise demonstriert. Für einen Überblick siehe z.B. [Eichenbau, 1996].

Die oben geschilderten Resultate werden verständlicher, wenn man sich die neuroanatomische Struktur der Hippocampusformation und deren Lage im Gesamtsystem, also im Gehirn, ansieht. Der Hippocampus, der aus dem Gyrus dentatus, dem Ammonshorn (*Cornu ammonis*, CA) und dem Subiculum besteht, erhält hauptsächlich über den Neocortex vorverarbeitete und multimodale Sensorinformation. Diese tritt über den entorhinalen Cortex in die

Hippocampusformation ein. Von dort aus gelangt sie über den Gyrus dentatus zu CA3 und zu CA1 im Ammonshorn. Von CA1 geht es über das Subiculum zurück zu tieferen Schichten des entorhinalen Cortex und von dort wieder in den Neocortex. Diese Beziehungen sind in Abbildung 6.2 anschaulich dargestellt [Cohen und Eichenbaum, 1993]. Besonders hervorgehoben seien die rekurrenten Strukturen im Gyrus dentatus und CA3.

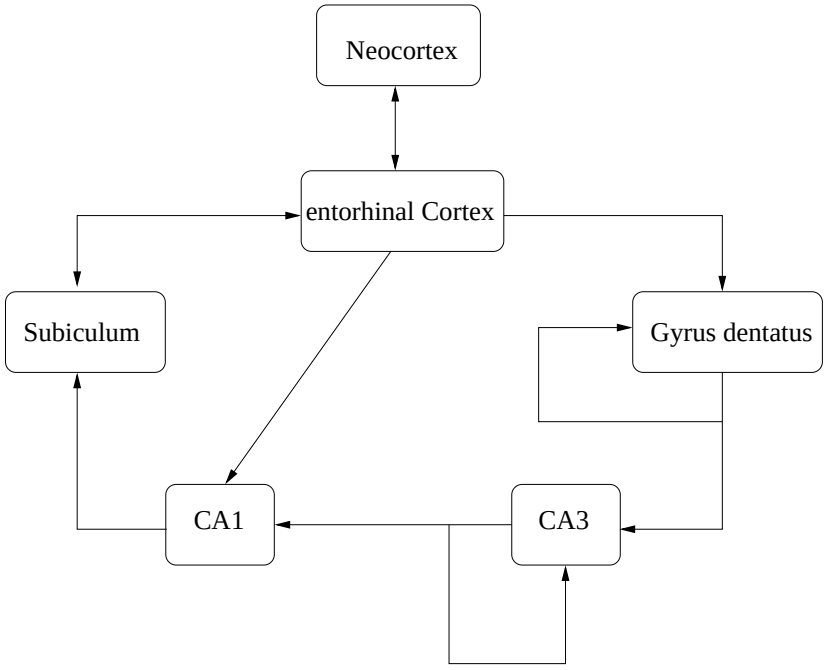


Abbildung 6.2: Schematisch Darstellung der Hippocampus Formation, [Cohen und Eichenbaum, 1993]. Die Pfeile zeige den vereinfachten Informationsfluss bzw. seine Richtung an.

Aufgrund all dieser Resultate geht man davon aus, dass der Hippocampus eine mitentscheidende Rolle beim sogenannten deklarativen Gedächtnis spielt. Unter deklarativem Gedächtnis versteht man all das Wissen, das bewusst verfügbar ist und geäußert werden kann und das somit im Gegensatz zum prozeduralen Gedächtnis steht, das im wesentlichen für die Ausführung sensomotorischer Fertigkeiten, die nicht notwendigerweise bewusst ablaufen müssen, verantwortlich ist [Squire, 1987]. Aufgrund von psychologischen Studien von Patienten mit Läsionen, und Läsionsexperimenten am Hippocampus von Versuchstieren, geht man davon aus, dass der Hippocampus bei der Organisation von Lernen und (deklarativem) Gedächtnis eine entscheidende Rolle spielt. Eine Schädigung des Hippocampus führt zu Beeinträchtigungen bzw. dem vollständigen Verlust, neues Wissen erwerben zu können, wohingegen Wissen, das vor der Läsion erworben wurde, noch vorhanden und abrufbar

ist. Hieraus schloss man, dass der Hippocampus an der Konsolidierung von Wissen maßgeblichen Anteil hat, jedoch der Hippocampus selbst kein Speicher für gelerntes Wissen darstellt, siehe [Eichenbaum, 1997] für einen Überblick.

Insgesamt kann man festhalten, dass Untersuchungen, welche sich mit der Lokalisation einer Ratte in einem Labyrinth beschäftigen, einerseits zwar eine spezielle Fragestellung bearbeiten, andererseits ist der an dieser Lokalisation beteiligte Teil des Gehirns, nämlich der Hippocampus, ein zentrales Bindeglied, zwischen verschiedenen corticalen Arealen und insbesondere an der ganz allgemeinen Organisation von Lernen und Gedächtnis beteiligt, so dass sich hieraus Resultate ergeben, die durchaus für ein allgemeines Verständnis der Wahrnehmungs-Handlungs-Schleife von Bedeutung sein können. Außerdem ist die Unabhängigkeit der *place cells* von einer bestimmten Modalität hervorragend dazu geeignet, die detailierte Sensorik und Motorik nur abstrakt aufzufassen, da die ankommenden Sinnesreize schon soweit vorverarbeitet und verschränkt sind, dass eine einzige Modalität praktisch für den Hippocampus nicht mehr auflösbar ist. Für die Ausgänge des Hippocampus gilt das gleiche, da sie nicht direkt in den Motorcortex gelangen, sondern erst weiterverarbeitet werden.

Aufbauend auf diesen und anderen experimentellen Befunden wurden eine Reihe theoretischer Arbeiten erstellt, welche diese Resultate benutzen, um biologisch plausible Modelle der Navigation zu erstellen [Blum und Abbott, 1996; Gerstner und Abbott, 1997; Herrmann et al., 1999b,a; Arleo und Gerstner, 2000; Trullier und Meyer, 2000]. In der vorliegenden Arbeit werde ich ein abgewandeltes Modell von [Herrmann et al., 1999b] verwenden, um das Problem der Selbstlokalisation eines autonomen Agenten zu untersuchen, und damit Aussagen über die prinzipielle Arbeitsweise der Wahrnehmungs-Handlungs-Schleife zu erhalten.
Bevor ich den in [Herrmann et al., 1999b] vorgestellten, speziellen mathematischen Formalismus in abgewandelter Form präsentiere, werde ich eine kurze, anschaulich abstrakte Darstellung der Wahrnehmungs-Handlungs-Schleife geben, damit die in Abschnitt 6.4 angeführten Formulierungen bereits mit einer Visualisierung belegt sind. Durch diese Art der Darstellung werden allgemeine Bezüge zu komplexen adaptiven Systemen, siehe [Schuster, 2002] für einen Überblick, deutlich.

6.3 Anschauliche Darstellung der Wahrnehmungs-Handlungs-Schleife

Eine anschauliche Darstellung der Wahrnehmungs-Handlungs-Schleife ist durch Abbildung 6.3 gegeben. Hier befindet sich das Tier, welches im folgenden abstrakt als autonomer Agent bezeichnet wird, auf einem eindimensionalen Gitter mit reflektierenden Randbedingungen. Der sensorische Input besteht in der Farbe der Zelle, auf der sich der Agent gerade befindet, und ist entweder weiß oder schwarz. Die Handlungen oder Aktionen, welche von dem Agenten ausgeführt werden können, bestehen in Bewegungen auf dem Gitter nach oben bzw. unten. Der Agent wird punktförmig angenommen, d.h. es gibt keine Orientierung des Agenten auf den Zellen.

Auf der rechten Seite in Abbildung 6.3 ist schematisch das Innenleben des Agenten aufgelistet. Der Agent soll über ein Gedächtnis einer gewissen Länge verfügen, da ohne Gedächtnis keine zielgerichteten Aktionen möglich sind. Des weiteren soll der Agent über mindestens ein Belohnungssystem verfügen, durch welches ein internes Wertesystem eingeführt wird. Der Agent

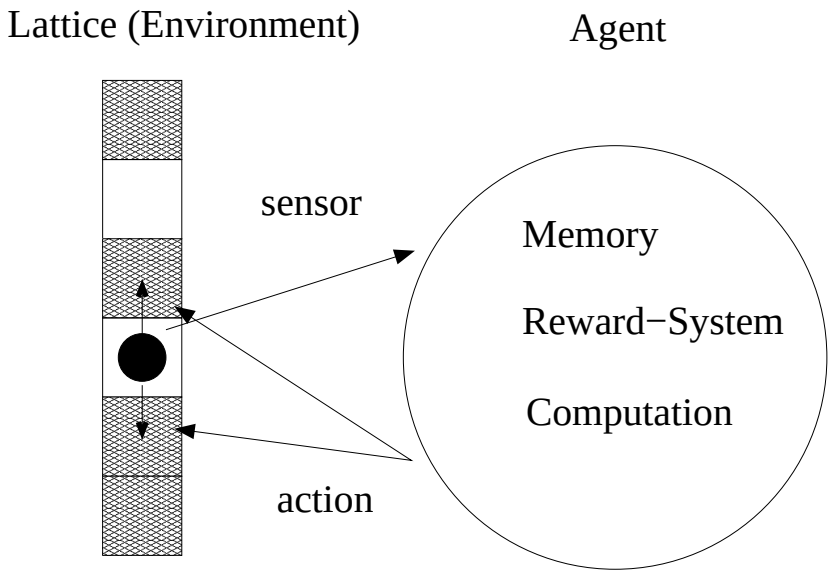


Abbildung 6.3: Schematische Beschreibung der Wahrnehmungs-Handlungs-Schleife. Der Agent ist auf dem eindimensionalen Gitter mit reflektierenden Randbedingungen positioniert und erhält die Farbe der Zellen als sensorischen Input. Dieser wird in der internen Struktur des Agenten, welche ein Gedächtnis und ein Belohnungssystem besitzt, verarbeitet. Hieraus resultiert dann die Wahl einer Aktion des Agenten, welche eine Positionsänderung auf dem Gitter zur Folge hat.

kann über unterschiedliche Wertesysteme verfügen, die daraus resultieren, dass es unterschiedliche interne Prozesse gibt, die auch unterschiedlich bewertet werden können. Der dritte Bestandteil stellt die Befähigung dar, aufgrund der im Gedächtnis, durch das Belohnungssystem und des sensorischen Inputs zur Verfügung stehenden Information, Berechnungen durchzuführen, welche als konkretes Resultat die nächste auszuführende Aktion des Agenten oder eine Antizipation von zu erwartenden Stimuli liefert.

Neben einer räumlichen Diskretisierung der Umgebung, nämlich dem eindimensionalen Gitter, soll auch die Zeit als diskret angenommen werden.

Die bisher vorgenommene Charakterisierung der Wahrnehmungs-Handlungs-Schleife ist mit der allgemeinen Beschreibung von komplexen adaptiven Systemen, wie sie z.B. in [Schuster, 2002] vorgenommen wird, identisch. Es sei angemerkt, dass die Wahrnehmungs-Handlungs-Schleife als komplexes adaptives Systeme angesehen werden kann. Die Umkehrung gilt jedoch nicht immer, da es auch adaptive Mechanismen z.B. auf einer evolutionären Zeitskala gibt, welche mit neuronalen Prozessen nicht direkt etwas zu tun haben.

6.3.1 Prinzipielle Arbeitsweise

Die Beschreibung der prinzipiellen Arbeitsweise der Wahrnehmungs-Handlungs-Schleife ist nicht ganz einfach und erst recht nicht eindeutig. Dies liegt daran, dass die Wahrnehmungs-Handlungs-Schleife einen geschlossenen Zyklus darstellt und eine Beschreibung diesen Zyklus notwendigerweise an einer Stelle "aufschneiden" muss, um mit der Darstellung zu beginnen.

Aus diesem Grund wähle ich subjektiv als Startpunkt dieser Beschreibung den Zeitpunkt t, an dem sich der Agent auf dem Gitter an der Position $\hat{x}_t$ befindet und von dort einen sensorischen Input $\hat{s}_t = f_6(\hat{x}_t)$ erhält. Dieser sensorische Input $\hat{s}_t = f_6(\hat{x}_t)$ kann aufgrund von äußeren Einflüssen, wie z.B. mangelnder Beleuchtung, fehlerbehaftet sein. Dies hat zur Folge, dass nicht der tatsächlich vorhandene physikalische Reiz $\hat{s}_t = f_6(\hat{x}_t)$, sondern der fehlerbehaftete Input $s_t = f_7(\hat{s}_t, \hat{x}_t)$ dem Agent zur Verfügung steht und somit wahrgenommen wird.
Dieser Input s_t wird dann vom Agent dazu verwendet, verschiedene Berechnungen unter Berücksichtigung des Gedächtnisses W^1 und W^2 und der inneren Zustände i_t, welche z.B. wie in Kapitel 6.4. Wahrscheinlichkeitsverteilungen sein können, durchzuführen, die als Resultat eine neue Aktion des Agenten a_{t+1} zum Zeitpunkt $t+1$ ergibt, sowie eine Vorhersage für den nächsten zu erwartenden Input $\overline{s}_{t+1}$ liefert. Die Vorhersage des

nächsten zu erwartenden Inputs $\overline{s}_{t+1}$ ist deshalb notwendig, weil hierdurch ein Vergleich zwischen tatsächlich erhaltenem s_{t+1} und antizipiertem Input $\overline{s}_{t+1}$ durchgeführt und mit Hilfe des Bewertungssystems des Agenten zum selbstüberwachten Lernen des Agenten verwendet werden kann. Somit ist kein äußeres Reinforcementsignal durch einen äußeren Beobachter mehr notwendig, um ein Lernen des Agenten zu ermöglichen. Das Resultat des Belohungssystems des Agenten wird dann dazu verwendet, die inneren Freiheitsgrade durch 6.8 und 6.9 so zu adaptieren, dass die Vorhersage des zu erwartenden Inputs im Laufe der Zeit immer besser wird. Mit inneren Freiheitsgraden des Agenten ist allgemein dessen Gedächtnis, und im speziellen die Bildung einer internen Repräsentation der äußeren Umgebung, gemeint.
Zu dem selbstüberwachten Lernmechanismus muss gesagt werden, dass dieser auf der Voraussetzung beruht, dass die interne Struktur des Agenten dazu geeignet ist, eine Repräsentation der Umgebung bilden zu können. Dies setzt zum einen die Kenntnis der Größe (Anzahl der Zellen) der äußeren Umgebung voraus und zum anderen topologische Zusammenhänge der äußeren Umgebung, die ebenfalls in der internen Struktur des Agenten verankert sein müssen.
Die Idee des Self-Reinforcement geht meiner Ansicht nach auf [Samuel, 1959] zurück, der mit solch einem Mechanismus Algorithmen adaptierte, welche dadurch Dame spielen lernten. Damit konnte er das Problem des verzögerten Reinforcements umgehen, da beim Damespiel erst ganz zum Schluss feststeht, wer gewonnen hat. Das Problem besteht hierbei darin, dass künstlich eingeführte Reinforcementsignale für jeden Zwischenzug, basierend beispielsweise auf einer Wertetabelle, die den jeweiligen Spielstand bewertet, aufgrund der Komplexität des Problems nicht berechenbar sind. Dies mag zunächst etwas erstaunen, da ein Damespiel nur endlich lange Zeit dauert und die in dieser Zeit angenommene Anzahl der Zustände ebenfalls endlich ist. Doch das zugrunde liegende mathematische Problem hat tiefere Ursachen, wie man z.B. an Untersuchungen von eindimensionalen deterministischen Cellulären Automaten mit zwei Zuständen sehen kann [Wolfram, 1984].

Aufgrund der ausgewählten Aktion a_{t+1} des Agenten findet eine Positionsänderung $\hat{x}_{t+1} = f_4(\hat{x}_t, a_{t+1})$ auf dem Gitter statt. Diese Positionsänderung kann wieder, z.B. durch Reibung, fehlerbehaftet sein und von der beabsichtigten Positionsänderung $x_{t+1} = f_5(\hat{x}_t, a_{t+1})$ abweichen.
Neben all diesen dynamischen Prozessen gibt es auch noch eine zeitliche Entwicklung der inneren Zustände, die durch Gleichung 6.2 beschrieben wird und in Abhängigkeit der Aktion erfolgt. Dies ist deshalb Notwendig, da die so erhaltenen inneren Zustände i_{t+1} unter anderem zu einer Vorhersage von s_{t+1} verwendet werden sollen. Die Vorhersage kann jedoch nur unter Einbeziehung der vorher durchgeführten Aktion zu einer vernünftigen Vorhersage führen, da ansonsten keine Synchronisation mit der Umgebung erfolgen kann.

Abstrakt kann man folgendes Funktionenschema angeben, welches die oben beschriebene Arbeitsweise zusammenfasst.

$$
\begin{aligned}
a_{t+1} &= f_1(i_t, W_t^1) && (6.1)\\
i_{t+1} &= f_2(i_t, a_{t+1}, W_t^2) && (6.2)\\
\overline{s}_{t+1} &= f_3(i_{t+1}, W_t^2) && (6.3)\\
\hat{x}_{t+1} &= f_4(\hat{x}_t, a_{t+1}) && (6.4)\\
x_{t+1} &= f_5(\hat{x}_t, a_{t+1}) && (6.5)\\
\hat{s}_t &= f_6(\hat{x}_t) && (6.6)\\
s_t &= f_7(\hat{s}_t, \hat{x}_t) && (6.7)\\
\triangle W_t^1 &= f_8(s_{t+1}, \overline{s}_{t+1}, W_t^1) && (6.8)\\
\triangle W_t^2 &= f_9(s_{t+1}, \overline{s}_{t+1}, W_t^2) && (6.9)
\end{aligned}
$$

Hier bezeichnen Variablen, die mit einem $\hat{}$ versehen sind, Variablen auf dem äußeren Gitter, die dem Agenten nicht direkt zugänglich sind, wohingegen die anderen Variablen vom Agenten verwendet werden können, um seine Berechnungen durchzuführen. Die angegebenen Funktionen f_1 bis f_9 können deterministisch oder stochastisch, wie in Kapitel 6.4., sein. Der adaptive Teil des Agenten wird mit W_t^1 und W_t^2 bezeichnet. Die W^i's stellen sozusagen das Gedächtnis, beispielsweise die anzupassenden Gewichte eines neuronalen Netzes, und somit die Gesamtheit aller Erfahrungen des Agenten dar, das im Laufe der Zeit durch den Vergleich von antizipiertem $\overline{s}_{t+1}$ und tatsächlich beobachtetem Input s_{t+1} adaptiert wird. Diese Adaptation der zwei verschiedenen Gedächtnisse $\triangle W^1$ und $\triangle W^2$ wird durch die Gleichungen 6.8 und 6.9 zum Ausdruck gebracht.
Es sei ausdrücklich darauf hingewiesen, dass für die Zeit $t = \hat{t}$ angenommen wurde, was im Falle einer statischen Umgebung, $\hat{s}_t = f_{t,1}(\hat{x}_t) = f_1(\hat{x}_t)$ für alle t, keine wesentliche Einschränkung darstellt.

Das Funktionensystem 6.1 bis 6.9 stellt ein einfaches[3], abstraktes Funktionensystem zur Beschreibung der Wahrnehmungs-Handlungs-Schleife, unter Berücksichtigung von Fehlern in der Sensorik und Positionsänderungen in der äußeren Umgebung, sowie einem adaptiven Mechanismus für den Aktions-Selektions-Mechanismus und der Vorhersage des nächsten zu erwartenden Inputs, dar.

[3]Das einfachste adaptive Funktionensystem besitzt nur ein einziges Gedächtnis W, das für alle inneren Zustände $a, i, \overline{s}$ verwendet wird.

6.4 Formale Beschreibung der Wahrnehmungs-Handlungs-Schleife für einen Spezialfall

Um die Wahrnehmungs-Handlungs-Schleife untersuchen zu können, ist es notwendig deren Bestandteile konkret zu definieren. Dazu muss die allgemeine Formulierung durch die Gleichungen 6.1 bis 6.9 mit Leben gefüllt werden. Da ich keine Möglichkeit sehe, dies für den allgemeinen Fall durchzuführen, gebe ich einen Spezialfall an, der später biologisch plausibel begründet werden soll.

6.4.1 Umgebung

Die Umgebung soll durch ein eindimensionales Gitter der Größe N mit reflektierenden Randbedingungen geben sein. Jeder Zelle des Gitters $\hat{x} \in \{1, \ldots, N\}$ wird ein Symbol $\hat{s} \in \{0, 1\}$ zugeordnet, gemäß Gleichung 6.6, welches eine beobachtbare Größe der Umgebung darstellt. Die Position $\hat{x}$ auf dem Gitter sei nicht direkt beobachtbar.
Der Beobachtungsfehler eines Symbols $\hat{s}$ auf einer Gitterzelle $\hat{x}$ aufgrund von äußeren Einflüssen sei durch den Markov Prozess

$$P(s|\hat{s}) = \begin{pmatrix} 1 - p^{oe}_{s=1} & p^{oe}_{s=2} \\ p^{oe}_{s=1} & 1 - p^{oe}_{s=2} \end{pmatrix} \tag{6.10}$$

ortsunabhängig[4] definiert. Das durch den Agent beobachtete Symbol s wird dann aus der Verteilung

$$P(s) = \sum_{\hat{s}} P(s|\hat{s})P(\hat{s}) \tag{6.11}$$

gezogen, wobei natürlich $P(\hat{s}) = \delta_{\hat{s},\hat{s}=f_6(\hat{x})}$ gesetzt wird. Somit wird aus Gleichung 6.7 nun $s_t = f_7(\hat{s}_t)$, die durch 6.10 und 6.11 definiert wird. Im folgenden wird nur der Fall des symmetrischen Beobachtungsfehlers $p^{oe}_{s=1} = p^{oe}_{s=2}$ (*observation error*) untersucht.

Da die äußere Umgebung des Agenten ein eindimensionales Gitter ist, soll der Agent nur über zwei unterschiedliche Aktionen $a_t \in \{-1, 1\}$ verfügen. Der Fehler p^{pe} (*position error*) in der Gitterpositionsänderung, aufgrund von äußeren Einflüssen, sei in Abhängigkeit der beiden möglichen Aktionen

[4] Dies sei aus Gründen der Einfachheit angenommen, denn im allgemeine Fall gilt $P(s|\hat{s}, \hat{x})$ für den stationären Markov Prozess des Beobachtungsfehlers.

durch

$$P(\hat{x}_{t+1}|\hat{x}_t, a_{t+1} = 1) =$$

$$\begin{pmatrix} p^{pe} & p^{pe} & 0 & 0 & \cdots & \cdots & 0 \\ 1-p^{pe} & 0 & p^{pe} & 0 & \cdots & \cdots & 0 \\ 0 & 1-p^{pe} & 0 & \ddots & \ddots & \cdots & \vdots \\ 0 & 0 & 1-p^{pe} & \ddots & \ddots & \cdots & 0 \\ \vdots & \vdots & \cdots & \ddots & 0 & p^{pe} & 0 \\ 0 & \vdots & \cdots & 0 & 1-p^{pe} & 0 & p^{pe} \\ 0 & 0 & \cdots & 0 & 0 & 1-p^{pe} & 1-p^{pe} \end{pmatrix} \tag{6.12}$$

und

$$P(\hat{x}_{t+1}|\hat{x}_t, a_{t+1} = -1) =$$

$$= \begin{pmatrix} 1-p^{pe} & 1-p^{pe} & 0 & 0 & \cdots & \cdots & 0 \\ p^{pe} & 0 & 1-p^{pe} & 0 & \cdots & \cdots & 0 \\ 0 & p^{pe} & 0 & \ddots & \ddots & \cdots & \vdots \\ 0 & 0 & p^{pe} & \ddots & \ddots & \cdots & 0 \\ \vdots & \vdots & \cdots & \ddots & 0 & 1-p^{pe} & 0 \\ 0 & \vdots & \cdots & 0 & p^{pe} & 0 & 1-p^{pe} \\ 0 & 0 & \cdots & 0 & 0 & p^{pe} & p^{pe} \end{pmatrix} \tag{6.13}$$

gegeben. Hierbei wurde schon aus Gründen der Einfachheit eine homogene Umgebung angenommen, die zudem noch stationär ist.
Die fehlerbehaftete Gitterposition $\hat{x}_{t+1}$ ergibt sich dann durch Ziehen aus der Verteilung

$$P(\hat{x}_{t+1}) = \sum_{\hat{x}'_t} P(\hat{x}_{t+1}|\hat{x}'_t, a_{t+1}) P(\hat{x}'_t) \tag{6.14}$$

Hierbei ist $P(\hat{x}'_t) = \delta_{\hat{x}'_t, \hat{x}_t}$, wobei $\hat{x}_t$ den tatsächlichen physikalischen Ort des Agenten auf dem Gitter beschreibt. Man beachte, dass die reflektierenden Randbedingungen des Gitters durch die Gleichungen 6.12 und 6.13 berücksichtigt werden. Somit ist Gleichung 6.4 definiert.

6.4.2 Dynamik innerer Zustände 1.

Die inneren Zustände des Agenten werden durch eine diskrete Wahrscheinlichkeitsverteilung P^1_i beschrieben, wobei $i \in I$. Als Spezialfall wird $I = N$

gewählt, wodurch die Größe der äußere Umgebung als bekannt vorausgesetzt wird.
Die Dynamik der inneren Zustände soll durch einen Markov Prozess definiert werden [Herrmann et al., 1999b],

$$P^1_{t+1}(i) = \sum_{i'}^{I} T_{a_t}(i|i')P^1_t(i') \tag{6.15}$$

so dass das System in der Wahrscheinlichkeitsverteilung P^1 ein Gedächtnis der Länge eins hat. Als Markov Prozess T_{a_t} wird

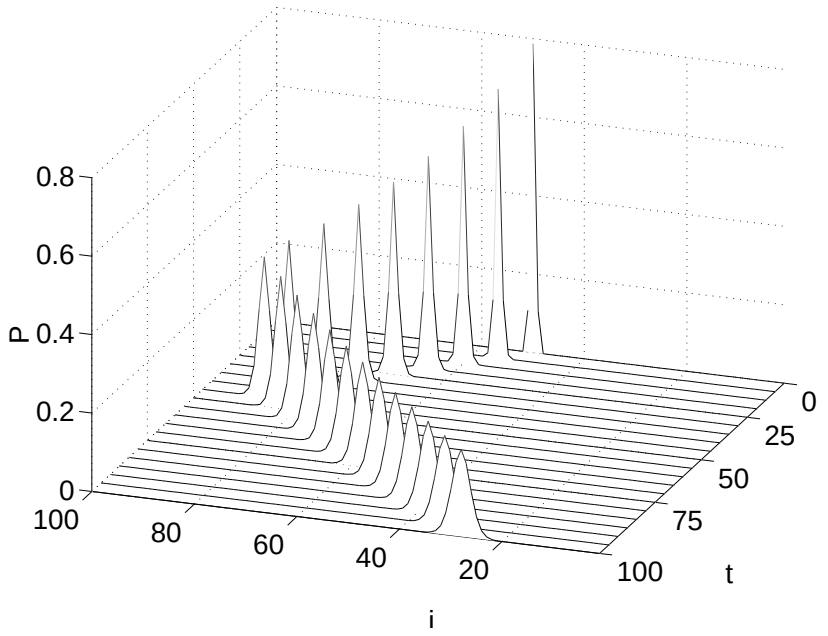


Abbildung 6.4: Veranschaulichung des Markov Prozesses 6.15. Dargestellt ist die zeitliche Entwicklung einer Gaußfunktion, die anfänglich mit $\mu(t = 0) = 50$ und $\sigma(t = 0) = 0.5$ initialisiert wurde. Für die Übergangswahrscheinlichkeit 6.16 wurde $\sigma_1 = 0.35$ angenommen. Für die Zeitschritte $0 \leq t \leq 39$ wurde $a_t = 1$ gesetzt und für $40 \leq t \leq 100$ war die angenommene Aktion $a_t = -1$.

$$T_{a_t}(i|i') = N_1 \exp(-\frac{(i - i' + a_t)^2}{2\sigma_1^2}) \tag{6.16}$$

verwendet. N_1 ist hierbei die Normierung. Die Indizierung durch die Aktion a_t soll die Beeinflussung der Driftrichtung von $T_{a_t}(i|i')$ symbolisieren.

In Abbildung 6.4 ist der Markov Prozesses 6.15 anhand eines speziellen Beispiels veranschaulicht. Dort ist der zeitliche Verlauf einer Gaußfunktion

dargestellt, die durch $\mu(t = 0) = 50$ und $\sigma(t = 0) = 0.5$ initialisiert wurde. Für die Übergangswahrscheinlichkeit 6.16 wurde $\sigma_1 = 0.35$ angenommen. Anfänglich wurde für die Aktion $a_t = 1$ angenommen, wodurch sich der Mittelwert des Gaußes hin zu höheren Werten verschiebt. Ab $t = 40$ wurde $a_t = -1$ gesetzt, was eine Umkehrung der Drift bewirkt. Man erkennt deutlich, wie die Drift durch die vorgegebene Aktion gesteuert wird. Weiterhin bewirkt die iterierte Anwendung von 6.15 eine Verbreiterung des Gaußes.
Durch diese Eigenschaften ist der durch 6.16 definierte Markov Prozess zu einem Ornstein-Uhlenbeck Prozesses ähnlich [Risken, 1996].

Die Dynamik der internen Zustände ist hierdurch nur teilweise definiert. Dies wird durch Abschnitt 6.4.4 vervollständigt.

6.4.3 Interner Evaluierungsmechanismus

Da der Agent intern eine Repräsentation der äußeren Umgebung bilden soll, benötigt er ein Kriterium, um beurteilen zu können, ob dieses Ziel auch erreicht wurde. Hierfür wird die Vorhersage des Agenten für das nächste zu erwartende Beobachtungssymbol verwendet [Herrmann et al., 1999b,a].

$$P^2_{t+1}(s) = \sum_i P^S_t(s|i) P^1_{t+1}(i) \tag{6.17}$$

$$\overline{s}_{t+1} = \operatorname*{argmax}_s \left\{ P^2_{t+1}(s) \right\} \tag{6.18}$$

Gleichung 6.17 stellt den einfachsten Vorhersagemechanismus dar, den man wählen kann, nämlich einen linearen Schätzer. Dieser wichtet die inneren Zustände P^1 mit einer bedingten Wahrscheinlichkeit $P^S(s|i)$, welche die inneren Zustände P^1 mit dem nächsten zu erwartenden Beobachtungssymbol verknüpft. Da P^2 wieder eine Wahrscheinlichkeitsverteilung darstellt, muss man noch ein zusätzliches Kriterium einführen, welches das vorhergesagte Symbol $\overline{s}_{t+1}$ liefert. Das einfachste, was man hierfür wählen kann, ist die Argmax-Funktion 6.18, welche die Komponente der Verteilung P^2 auswählt, die der größten Wahrscheinlichkeit entspricht. Die Argmax-Funktion ist nur ein mathematischer Ausdruck für einen winner-take-all Mechanismus, von dem in den bisherigen Kapiteln bereits ausgiebig Gebrauch gemacht wurde.
Hierdurch ist Gleichung 6.3 definiert.

Die bedingte Wahrscheinlichkeitsverteilung $P^S(s|i)$ stellt den Teil des inneren Systems des Agenten dar, welcher gelernt werden muss. Dies geschieht on-line, d.h. jeder eintreffende sensorische Input wird verwendet, um $P^S(s|i)$ um einen geringen Betrag zu verändern.

$$P^S_{t+1}(s|i) = P^S_t(s|i) + \delta P^S_t(s|i) \tag{6.19}$$

Um die Änderung $\delta P^S_{t+1}(s|i)$ zu bestimmen, wird von der Kostenfunktion

$$E_{cost} = \frac{1}{2} \sum_s (\delta_{s_t,s} - P^S_t(s))^2 \tag{6.20}$$

der Gradient nach den zu lernenden bedingten Wahrscheinlichkeiten $P^S(s|i)$ gebildet [Herrmann et al., 1999b].

$$\delta P^S_{t+1}(s|i) = \frac{\partial E_{cost}}{\partial P^S(s|i)} = \epsilon(\delta_{s_t,s} - P^S_t(s))P^1_t(i) \tag{6.21}$$

Hierbei ist ϵ eine Lernrate, die angemessen klein zu wählen ist.
Der Grund für die Wahl der Kostenfunktion 6.20 besteht darin, dass man annimmt, eine gute Vorhersage liege genau dann vor, wenn die Wahrscheinlichkeitsverteilung $P^S_t(s)$ zu einer Deltadistribution entartet, so dass das Beobachtungssymbol s_t exakt vorhergesagt wird [Herrmann et al., 1999b]. Diese Annahme ist einleuchtend und der Ansatz 6.20 sehr einfach, jedoch natürlich nicht eindeutig.
Aus Gründen der Einfachheit, und weil in diesem Kapitel die Untersuchung von Aktions-Selektions-Mechanismen im Vordergrund stehen soll, wird 6.21 als Lernregel für die bedingten Wahrscheinlichkeiten $P^S(s|i)$ verwendet. Gleichung 6.21 ist nur eine zu Hebb ähnliche Lernregel, dies ist jedoch nicht so schlimm, denn die dem Agent zugrunde liegende interne Struktur ist auch kein neuronales Netz im Sinn der vorangegangenen Kapitel. Vielmehr handelt es sich um eine gemittelte Beschreibungsform eines neuronalen Netzes, bei dem die einzelnen Neuronen nicht mehr aufgelöst werden. Hierauf wird bei der Diskussion nochmals ausführlicher eingegangen.

6.4.4 Dynamik innerer Zustände 2.

In Sektion 6.4.2 wurden die inneren Zustände P^1 zwar von der Aktion beeinflusst, jedoch ansonsten von keinen weiteren Größen. Es erscheint nicht unvernünftig, wenn P^1 noch von der Qualität der Vorhersage abhängen würde, also Feedback erhielte. Die Qualität der Vorhersage des Agenten steht diesem jedoch nur indirekt zur Verfügung, da er von außen keine Mitteilung darüber, z.B. in Form eines Reinforcementsignals, erhalten soll. Der gesamte Lernvorgang soll selbstorganisiert ablaufen, so dass der Agent nur Größen benutzen kann, die entweder über den Sensor beobachtet werden können oder intern im Laufe der Zeit adaptiert wurden.
Aus diesem Grund erscheint die bedingten Wahrscheinlichkeit $P^S(s|i)$ als geeignet, da sie einerseits im Laufe der Zeit, infolge sensorischen Inputs, durch Gleichung 6.19 an die Umgebung adaptiert wird und andererseits maßgeblich an der Vorhersage des Agenten für das nächste zu beobachtende Symbol durch Gleichung 6.17 beteiligt ist.

Formal kann man diese Rückkopplung mit Hilfe der Bayesschen Formel erreichen [Herrmann et al., 1999b].

$$P^1_{t+2}(i) \sim P^1_{t+2}(i|s_t) = N^{-1} P^S_{t+1}(s_t|i) P^1_{t+1}(i) \tag{6.22}$$

Hierbei stellt N die Normierung dar.

$$N = \sum_i^I P^S_{t+1}(s_t|i) P^1_{t+1}(i) \tag{6.23}$$

Die neuen inneren Zustände $P^1_{t+2}(i)$ werden also durch $P^1_{t+2}(i|s_t)$ approximiert. Es sei ausdrücklich darauf hingewiesen, dass die bedingte Wahrscheinlichkeit nur an der Stelle s_t verwendet wird, da dies der einzige Einfluss einer äußeren Größe ist, der dem Agent zum Zeitpunkt t zur Verfügung steht.
Nun wird auch verständlich, warum die zweistufige Dynamik der inneren Zustände nicht schon vollständig in Abschnitt 6.4.2 definiert wurden konnte, ohne den kausalen Zusammenhang des ablaufenden Zyklus zu durchbrechen. Zuerst erfolgt also eine Propagation der inneren Zustände in Abhängigkeit der auszuführenden Aktion 6.15. Danach wird, basierend auf den momentanen inneren Zuständen 6.15, eine Vorhersage über das nächste zu erwartende Beobachtungssymbol durch 6.17 und 6.18 berechnet, die mit dem danach tatsächlich beobachteten Symbol dazu verwendet wird, die bedingte Wahrscheinlichkeit $P^S(s|i)$ durch Gleichung 6.19 zu adaptieren. Erst jetzt kann die aktualisierte bedingte Wahrscheinlichkeit $P^S(s_t|i)$ dazu verwendet werden, um auf die inneren Zustände zurückzuwirken. Dieser Feedback lässt sich als Selbstkonsistenztest interpretieren, denn die inneren Zustände werden durch Gleichung 6.22 um so stärker beeinflusst, je schlechter die Vorhersage war. Dies macht man sich durch Betrachtung der relativen Größenordnungen von $P^S(s_t|i)$ und P^1 klar. Denn für ein richtig vorhergesagtes Beobachtungssymbol sind $P^S_{t+1}(s_t|x_t)$ und $P^1_{t+1}(x_t)$ größer als alle anderen Komponenten $P^S_{t+1}(s_t|i)$ und $P^1_{t+1}(i)$ für $i \neq x_t$. Somit wird aber auch $P^1_{t+2}(x_t)$ größer sein, als alle anderen Komponenten $P^1_{t+2}(i)$ für $i \neq x_t$ und somit selbstkonsistent mit P^1_{t+1}. Den Fall einer schlechten Vorhersage macht man sich analog klar.

Hierdurch ist Gleichung 6.2 vollständig definiert.

6.4.5 Belohnungssystem vs. Aktions-Selektions-Mechanismus

Ein Aktions-Selektions-Mechanismus ist im allgemeinen Fall ein dynamischer Prozess, der aufgrund bisher gesammelter Informationen und der momentanen internen Zustände des Agenten eine Entscheidung darüber trifft, welche Aktion vom Agenten im nächsten Zeitschritt ausgeführt werden

soll. Für diese abstrakte Beschreibung durch Gleichung 6.1 sollen im folgenden drei mögliche Realisierungen angegeben werden. Da die Klasse aller möglichen Aktions-Selektions-Mechanismen unendlich groß ist, werden die drei konkreten Realisierungen so gewählt, dass sie von ihrer Struktur her sehr einfach sind und dennoch, aus theoretischer Sicht, interessante Eigenschaften besitzen.
Bevor die drei unterschiedlichen Aktions-Selektions-Mechanismen angegeben werden, soll ihre Wahl motiviert werden.

Wenn man sich ein biologisches, neuronales System ansieht, so lässt sich dieses grob in drei Beschreibungsebenen einteilen. Die erste, mikroskopische Beschreibungsebene, besteht z.B. aus der Ionenkanaldynamik der Zellmembranen und der Diffusion der Neurotransmitter im synaptischen Spalt chemischer Synapsen. Die zweite, mesoskopische Ebene, beschreibt z.B. das Entstehen bzw. Auslösen der Aktionspotentiale von Neuronen und die dritte, makroskopische Ebene, das Verhalten des gesamten Organismus in seiner Umgebung.
Die Frage, die man sich bei der Wahl eines Aktions-Selektions-Mechanismus stellen muss, lautet:

> *Wie findet eine selbstorganisierte top-down Kommunikation zwischen diesen drei Beschreibungsebenen statt?*

Es ist einleuchtend, dass eine niedrige Beschreibungsebene immer Auswirkungen auf die darüberliegenden Ebenen hat, die Umkehrung ist jedoch nicht trivial. In diesem Zusammenhang ergaben die Untersuchungen in Kapitel 5.3.4.4., dass ein Gedächtnis in einer Hebb-artigen Lernregel (mesoskopische Ebene) unterschiedliche Verhalten, welche durch unterschiedliche Aktions-Selektions-Mechanismen bewirkt werden, detektieren kann. Somit ist es nicht unplausibel, dass ein sinnvoller Aktions-Selektions-Mechanismus über ein Gedächtnis verfügt, das basierend auf der bisher gelernten internen Repräsentation der äußeren Umgebung die Entscheidung über die nächste, auszuführende Aktion trifft. Diese Entscheidung ist deshalb ortsabhängig, weil durch den Aktions-Selektions-Mechanismus eine Exploration der gesamten äußeren Umgebung durchgeführt werden soll, da für einen Bereich der äußeren Umgebung, an dem der Agent nie gewesen ist, auch keine sinnvolle interne Repräsentation gebildet werden kann.
Die Verknüpfung der Aktionsselektion mit der bisher gelernten internen Repräsentation ergibt nun, neben dem Gedächtnis, per se zwei weitere Grundbestandteile der einfachsten Aktions-Selektions-Mechanismen. Diese Grundbestandteile sind Diffusion und Drift der Aktionsselektion, bezüglich der bisher gelernten internen Repräsentation. Denn diese kann, für endliche Zeiten, notwendigerweise nur eine approximierte Beschreibung der äußeren Umgebung zur Verfügung stellen, die mit einer gewissen Unsicherheit

versehen ist. Aus diesem Grund kann eine Selektion der Aktion, basierend auf der internen Repräsentation, nicht deterministisch sein, sondern sie muss durch einen stochastischen Prozess realisiert werden. Unter Einbeziehung der Nebenbedingung, nämlich Exploration der äußeren Umgebung, ergibt sich darüberhinaus eine Ergodizitätsforderung für diesen stochastischen Prozess im Raum der internen Repräsentation. Aus diesen Überlegungen folgt nun als einfachster Aktions-Selektions-Mechanismus ein stochastischer Prozess mit Diffusion oder Drift.

Mathematische Bemerkung
Eine formale Begründung für die heuristische Motivation der Bestandteile des stochastischen Prozesses für den Aktions-Selektions-Mechanismus kann unter Verwendung der Kramers-Moyal (vorwärts) Expansion für Markov Prozesse erster Ordnung gegeben werden. Hierbei berechnet sich die zeitliche Entwicklung einer Wahrscheinlichkeitsdichte $P(x,t)$ für einen Markov Prozess $T(x,t+\tau|x',t)$ durch

$$\frac{\partial P(x,t)}{\partial t} = \sum_{n=1}^{\infty} (-\frac{\partial}{\partial x})^n D^{(n)}(x,t) P(x,t) \tag{6.24}$$

Hierbei sind $D^{(n)}(x,t)$ die Kramers-Moyal Koeffizienten der Entwicklung, die sich durch

$$D^{(n)}(x',t) = \lim_{\tau \to 0} \frac{1}{\tau n!} \int (x-x')^n T(x,t+\tau|x',t) dx \tag{6.25}$$

berechnen. Aus dieser Beziehung liest man ab, dass die ersten beiden Koeffizienten einer solchen Expansion Drift $D^{(1)}$ und Diffusion $D^{(2)}$ sind [Risken, 1996]. Im Falle verschwindender (oder abgeschnittener) Terme $D^{(n)} = 0$ für $n \geq 3$ erhält man aus Gleichung 6.24 die Fokker-Planck Gleichung.

Die drei zu untersuchenden Aktions-Selektions-Mechanismen (ASM) werden wie folgt speziell gewählt :

1. (symmetrischer) Random Walk	Diffusion
2. Linear Walk	Drift
3. Active Walk	Diffusion + Drift

Die ersten beiden ASM werden nur deshalb gewählt, um einen Bezugspunkt für den Active Walk zu erhalten, denn ein Random und Linear Walk besitzen nur ein Gedächtnis der Länge Null bzw. eins und sind vollständig von den internen Zuständen des Agenten unabhängig. Deshalb werden diese beiden Aktions-Selektions-Mechanismen und alle anderen ASM, die gleichfalls

unabhängig von den internen Zuständen des Agenten sind, als passiv bezeichnet. Interessanterweise lässt sich der Linear Walk als eindimensionale Version des Braitenbergalgorithmus [Braitenberg, 1986] interpretieren, der in [Herrmann et al., 1999b] als Aktions-Selektions-Mechanismus verwendet wurde.

Definition 6.1 (Random Walk (symmetrisch))
Die auszuführende Aktion a_t *wird in jedem Zeitschritt aus der Verteilung* $P_a^{RW} = (p_{rw}, 1 - p_{rw})$ *mit* $p_{rw} = \frac{1}{2}$ *(symmetrisch) gezogen.*

Definition 6.2 (Linear Walk)

1. *initialisiere* $a_{t=0}$ *zufällig und gleichverteilt aus* $(0.5, 0.5)$

2.a *falls* $1 < \lfloor <i_{t-1}> \rceil < I$ *und* $a_{t-1} = 1 \Longrightarrow a_t = 1$

2.b *falls* $1 < \lfloor <i_{t-1}> \rceil < I$ *und* $a_{t-1} = -1 \Longrightarrow a_t = -1$

2.c *falls* $\lfloor <i_{t-1}> \rceil = 1$ *und* $a_{t-1} = -1 \Longrightarrow a_t = 1$

2.d *falls* $\lfloor <i_{t-1}> \rceil = I$ *und* $a_{t-1} = 1 \Longrightarrow a_t = -1$

Hier bezeichnet $\lfloor < i_{t-1} > \rceil = \lfloor \sum_i^I i P_{t-1}^1(i) \rfloor$ den auf- bzw. abgerundeten Mittelwert der inneren Zustände i.

$$\lfloor n.x \rceil = \begin{cases} n & , \text{if } x < 5 \\ (n+1) & , \text{if } x \geq 5 \end{cases} \tag{6.26}$$

für alle $n \in \mathbb{N}$.

Der Active Walk wird nun wie folgt definiert.

Definition 6.3 (Active Walk)

1. *initialisiere* $P_{sr,L}^{rel}(i, t = 0)$ *zufällig aus der Gleichverteilung* $[0, 1]$ *für alle* $i \in I$

2.a *ändere* $P_{sr,L}^{rel}(i, t)$ *in jedem Zeitschritt* t *nur an der Stelle* $\lfloor < i_t > \rceil = \lfloor \sum_i^I i P_t^1(i) \rceil$ *durch* $\triangle P_{sr,L}^{rel}(\lfloor < i_t > \rceil, t) = -\frac{r_t}{L}$*, wobei* r_t *ein self-reinforcement Signal und* L *die Gedächtnislänge der self-reward map* $P_{sr,L}^{rel}$

$$\begin{aligned} &\text{falls} \quad P_{sr,L}^{rel}(\lfloor < i_t > \rceil, t) > 1 \Longrightarrow P_{sr,L}^{rel}(\lfloor < i_t > \rceil, t) = 1 \\ &\text{falls} \quad P_{sr,L}^{rel}(\lfloor < i_t > \rceil, t) < 0 \Longrightarrow P_{sr,L}^{rel}(\lfloor < i_t > \rceil, t) = 0 \\ &\text{falls} \quad P_{sr,L}^{rel}(i, t) \equiv 0 \; \forall \; i \in \{1, \ldots, I\} \Longrightarrow \text{reinitialisiere} \;\; P_{sr,L}^{rel}(i, t) \\ &\qquad\qquad \text{zufällig aus der Gleichverteilung } [0, 1] \;\; \forall \; i \in I \end{aligned}$$

2.b die Aktion a_t *ergibt sich nun aus* $i_{max} = \text{argmax}_i\, P^{rel}_{sr,L}(i,t)$ *durch*

$$\begin{aligned}
&\text{falls} \quad i_{max} > \lfloor <i_t> \rceil \Longrightarrow a_t = 1 \\
&\text{falls} \quad i_{max} < \lfloor <i_t> \rceil \Longrightarrow a_t = -1 \\
&\text{falls} \quad i_{max} = \lfloor <i_t> \rceil \Longrightarrow \text{wähle } a_t \text{ zufällig und gleichverteilt}
\end{aligned}$$

Bei der Definition des Active Walk wurde versucht, die Auswirkungen einer Diffusion (Random Walk) und einer Drift (Linear Walk) gleichermaßen zu berücksichtigen. Durch die Änderung der *self-reward map* $P^{rel}_{sr,L}$ an der Stelle des gerundeten Mittelwerts $\lfloor < i_t > \rceil$ durch das self-reinforcement Signal r_t, bildet sich im Laufe der Zeit im Raum der internen Zustände I eine Art Belohnungs/Bestrafungs-Landschaft heraus. Hierbei wird das self-reinforcement Signal r_t durch Vergleich des vorhergesagten $\overline{s}_t$ mit dem tatsächlich wahrgenommenen Input s_t durch

$$r_t = \begin{cases} 1, \text{if} & \overline{s}_t = s_t \\ -1, \text{if} & \overline{s}_t \neq s_t \end{cases} \tag{6.27}$$

ermittelt. Somit kann man $r_t = 1$ als Belohnung (richtige Vorhersage) und $r_t = -1$ als Bestrafung (falsche Vorhersage) interpretieren.

Im Falle $P^{rel}_{sr,L} \not\equiv 0$ stellt die Aktionsauswahl durch 2.b) in der Definition des Active Walk eine gerichtete Selektion dar, die sich zu der Stelle der internen Repräsentation hinbewegt, die im bisherigen Verlauf des Lernprozesses im Mittel am meisten bestraft wurde. Hierbei erstreckt sich die Mittelung an jeder Stelle der internen Repräsentation I über einen Zeitraum von jeweils L Zeitschritten.
Wenn man L gegen unendlich gehen lassen würde, so erhielte man an jeder Stelle der internen Repräsentation eine Wahrscheinlichkeitsverteilung $P_{sr,L}(i)$ über den self-reinforcement Signalen r_t. Dieser Grenzübergang in L stellt jedoch aus praktischer Sicht eines realen biologischen Systems ein Problem dar, da dessen Lebenszeit immer endlich ist. Um in einer realen Umgebung bestehen zu können, ist es deshalb notwendig, die Entscheidung über die nächste zu wählende Aktion, auf der Basis der bis zu diesem Zeitpunkt zur Verfügung stehenden Information zu treffen. Aus diesem Grund stellt die self-reward map $P^{rel}_{sr,L}$ auch keine Wahrscheinlichkeitsverteilung an jeder Stelle der internen Zustände I über den self-reinforcement Signalen r_t dar, sondern nur relative Häufigkeiten, was durch das Symbol ^{rel} explizit angezeigt wird.

Falls nun $P^{rel}_{sr,L} \equiv 0$ eintrit, was aufgrund einer willkürlich gewählten Gedächtnislänge $L \ll \infty$ passieren kann, ohne dass der Lernprozess schon vollständig abgeschlossen ist, wird durch die zufällige Reinitialisierung der

self-reward map $P^{rel}_{sr,L}$ erneut[5] eine diffuse Auswahl der Aktion induziert, die verhindert, dass bestimmte Bereiche im Raum der internen Repräsentationen nicht erreicht werden.

Durch Zusammenspiel zwischen belohnungs/bestrafungsabhängiger Änderung der *self-reward map* und Reinitialisierung im Falle $P^{rel}_{sr,L} \equiv 0$, erhält der durch 6.3 definierte Aktions-Selektions-Mechanismus Komponenten, die sowohl Drift wie Diffusion enthalten.

6.4.6 Evaluierung des Agenten durch einen äußeren Beobachter

Die bisher in den Kapiteln 6.4.1. bis 6.4.5. angegebenen Komponenten definieren die internen Strukturen des Agenten sowie dessen Wechselwirkung mit seiner Umgebung. Um die hierdurch definierte Wahrnehmungs-Handlungs-Schleife des Agenten quantifizieren zu können, benötigt man noch ein geeignetes Fehlermaß, welches im folgenden angegeben und diskutiert werden soll.

Um die Problematik zu verstehen, welche die Konstruktion eines solchen Fehlermaßes mit sich bringt, soll zuvor eine einfachere Problemklasse diskutiert werden. Diese Problemklasse besteht darin, eine Abbildung, bestehend aus einer diskreten Anzahl von M Mustern, richtig zu lernen. Hierfür lässt sich, wie in Kapitel 5.3.4. geschehen, zu jedem Zeitpunkt der individuelle[6] Absolutfehler

$$E(t)_{iabs} = \frac{\#\text{der bereits gelernten Muster zum Zeitpunkt } t}{M} \tag{6.28}$$

angeben. Hierbei bezeichnet M die Anzahl aller zu lernenden Muster.
Bei der Berechnung des Fehlers $E(t)_{iabs}$ ist zu beachten, dass nach jedem Lernschritt t, alle zu lernenden Muster M dem (neuronalen) System präsentiert werden und verglichen wird, ob die jeweilige Abbildung auch richtig gelernt wurde, ohne dem (neuronalen) System zu gestatten, seine freien Parameter (synaptischen Gewichte) zu adaptieren. Bei dieser Evaluierung handelt es sich also um einen reinen Test des (neuronalen) Systems. Dies lässt sich, etwas abstrakter, so interpretieren, als bilde man zu jedem Zeitpunkt t eine identische Kopie des zu untersuchenden (neuronalen) Systems, bei der entweder alle vorhandenen Lernregeln entfernt oder alle Lernraten identisch Null gesetzt werden. Mit dieser voll funktionsfähigen[7], jedoch lernunfähigen Kopie des

[5] Da zu Beginn des Lernprozesses $t = 0$ die self-reward map $P^{rel}_{sr,L}$ bereits zufällig initialisiert wurde.

[6] Dies soll darauf hindeuten, dass hier keine Mittelung über ein Ensemble vorgenommen wird.

[7] Bezüglich des zu kopierenden Systems.

Originalsystems werden dann alle Fragen nacheinander durchgegangen (alle Muster werden präsentiert) und die, vom (neuronalen) System gegebenen Antworten mit den gewünschten Antworten verglichen. Hierbei ist die Reihenfolge, mit der die Fragen gestellt (Muster präsentiert) werden, ohne Bedeutung, da es keine Relationen zwischen den Mustern gibt. Die einzige Schwierigkeit, die aus praktischer Sicht auftreten kann, besteht in dem immer größer werdenden Zeitaufwand dieser Testabfrage für den Fall $M \to \infty$.

Im Unterschied zu dem oben beschriebenen Problem, bei dem die zu lernenden Muster M voneinander unabhängig waren, stehen die auf einem eindimensionalen Gitter angeordneten Muster in Form von zu beobachtenden Symbolen s untereinander in einer topologischen Beziehung.
Ein weiterer Unterschied zwischen einer zu lernenden Abbildung und dem Agenten, der sich auf seinem diskreten Gitter bewegen und einzelne Zellsymbole beobachten kann, besteht darin, dass der Agent über interne Zustände P^1 verfügt, die von vorangegangenen Zeitpunkten durch Gleichung 6.15 und 6.22 abhängen und die Vorhersage über das nächste zu erwartende Beobachtungssymbol $\overline{s}$ in Abhängigkeit von diesen internen Zuständen P^1 erfolgt. Konkret besteht die Abhängigkeit in der Vorhersage von der zurückliegenden Sequenz der vorangegangenen Aktionen a_t und der tatsächlich beobachteten Gittersymbole s_t. Abstrakt kann man sagen, dass die Vorhersage des nächsten zu erwartenden Beobachtungssymbols $\hat{s}$ von dem vorangegangenen raumzeitlichen Pfad des Agenten abhängt.
Somit ist es nicht möglich, den Agenten, der bereits eine interne Repräsentation seiner Umgebung gelernt hat, einfach willkürlich von einer Stelle des Gitters an irgendeine andere Stelle zu setzen, und zu erwarten, dass der Agent eine richtige Vorhersage über das nächste zu erwartende Beobachtungssymbol berechnen kann, ohne die internen Zustände P^1 für die jeweilige Gitterposition geeignet anzupassen. Aus biologischer Sicht kann man die internen Zustände P^1 als *on-going activity* interpretieren.

Individueller Gesamtfehler E_{iabs}

Ein Fehlermaß, das zu 6.28 analog ist und die zusätzlichen Unterschiede berücksichtigt, lässt sich nun wie folgt definieren. Zuerst erstelle man eine identische Kopie des Agenten und entferne alle adaptiven Mechanismen. Danach setze man den Agenten in eine Testumgebung, die identisch zu der Umgebung ist, in welcher der Agent bisher "gelebt" hat. Nun lässt man den Agent eine vorgegebene Zeit T durch einen vorgegebenen Aktions-Selektions-Mechanismus mit der Testumgebung interagieren. Hierdurch erhält man eine zeitgeordnete Sequenz

$$\underline{e}_{T(P_k)} = (e_{T(P_k),1}, \ldots, e_{T(P_k),T}) \tag{6.29}$$

der individuellen Vorhersagefehler $e_{T(P_k),t}$ zu jedem Zeitpunkt t, die sich als Vektor schreiben lässt, wobei die t'te Komponente dem Zeitpunkt t entspricht. Hierbei wurde der spezielle Pfad P_k der Länge T in Abhängigkeit des vorgegebenen Aktions-Selektions-Mechanismus erzeugt, was durch das Symbol $T(P_k)$ zum Ausdruck gebracht wird. Die Sequenz der individuellen Vorhersagefehler 6.29 determiniert zugleich die Menge der individuellen Vorhersagefehler $M_{T(P_k)}$, die sich durch

$$M_{T(P_k)} = \{\underline{e}_{T(P_k)}\} = \{\underline{e}^{T_1}_{T(P_k)}\} \cup \ldots \cup \{\underline{e}^{T_E}_{T(P_k)}\} \tag{6.30}$$

beschreiben lässt. Hier bezeichnet $\underline{e}^{T_i}_{T(P_k)}$ einen Vektor der Länge T_i, dessen Komponenten die zeitgeordneten individuellen Vorhersagefehler an der Gitterposition i sind. Insgesamt gilt natürlich $T = \sum_i^E T_i$. Gleichung 6.30 gibt also eine Zerlegung der Sequenz 6.29 bzgl. der Gitterposition[8].
Durch 6.30 ist man nun in der Lage, den gemittelten Vorhersagefehler auf der jeweiligen Gitterposition angeben zu können.

$$E^i = \frac{1}{T_i} \sum_{j=1}^{T_i} \underline{e}^{T_i}_{T(P_k),j} \quad , i \in \{1, \ldots, E\} \tag{6.31}$$

Der individuelle Gesamtfehler $E_{iabs}(T(P_k))$ in der Testumgebung ergibt sich dann aus der Mittelung aller Fehler auf den jeweiligen Gitterpositionen.

$$E_{iabs}(T(P_k)) = \frac{1}{E} \sum_{i=1}^{E} E^i = \frac{1}{E} \sum_{i=1}^{E} \frac{1}{T_i} \sum_{j=1}^{T_i} \underline{e}^{T_i}_{T(P_k),j} \tag{6.32}$$

Gleichung 6.32 beschreibt zwar den individuellen Gesamtfehler, jedoch in Abhängigkeit einer willkürlich gewählten Pfadlänge T, deren zugehöriger Pfad P_k mit einem willkürlichen Aktions-Selektions-Mechanismus erzeugt wurde. Um Gleichung 6.32 unabhängig von T und P_k zu machen, muss man noch über alle möglichen Pfade k mitteln und die Pfadlänge gegen unendlich gehen lassen, wodurch man zu

$$E_{iabs} = \lim_{k\to\infty} \frac{1}{kE} \sum_{k'}^{k} \sum_{i=1}^{E} \lim_{T_i\to\infty} \frac{1}{T_i} \sum_{j=1}^{T_i} \underline{e}^{T_i}_{T(P'_k),j} \tag{6.33}$$

gelangt.

[8]Um nicht zu viele neue Symbole einführen zu müssen, wurde $\{\underline{e}^{T_i}_{T(P_k)}\}$ verwendet, um damit eine Menge zu bezeichnen, die aus den Komponenten des Vektors $\underline{e}^{T_i}_{T(P_k)}$ besteht. Die j'te Komponenten des Vektors wird durch $\underline{e}^{T_i}_{T(P_k),j}$ bezeichnet.

Approximierter individueller Gesamtfehler $\tilde{E}_{iabs}$

Durch die Ableitung des letzten Abschnitts ergab sich ein Maß für den individuelle Gesamtfehler E_{iabs}. Die Frage, die sich nun stellt, lautet: Wie lässt sich der Ausdruck 6.33 auswerten?
Zunächst muss man festhalten, dass Gleichung 6.33 nicht in allgemeiner Form ausgewertet werden kann, da bisher kein allgemeiner Ansatz für das oben beschriebene Problem der Wahrnehmungs-Handlungs-Schleife[9] existiert. Aus diesem Grund muss man Gleichung 6.33 speziell für den hier angegebenen Ansatz näherungsweise auswerten.

Der in Kapitel 6.4.1. bis 6.4.5. definierte Ansatz für die Beschreibung der Wahrnehmungs-Handlungs-Schleife eines autonomen Agenten stellt eine Komposition der jeweiligen Einzelbestandteile dar, die durch kausal festgelegte Zusammenhänge in zeitlicher Abfolge ablaufen. Aus diesem Grund ist es nur möglich, unter Berücksichtigung der in Kapitel 6.4.1. bis 6.4.5. definierten Struktur einen approximierten individuellen Gesamtfehler E_{iabs} anzugeben, der unter bestimmten Voraussetzungen ähnlich zu 6.33 sein wird[10].
Der approximierte individuellen Gesamtfehler $\tilde{E}_{iabs}$ ergibt sich nun durch folgende spezielle Eigenschaften des in Kapitel 6.4.1. bis 6.4.5. definierten Gesamtsystems.

1. P^1 ist im stationären Zustand scharf gepeakt (glockenförmig)
2. P^2 berechnet sich aus dem linearen Schätzer 6.17
3. das vorhergesagte Symbol $\overline{s}$ ergibt sich aus der nichtlinearen $argmax$ Funktion 6.18

Die Punkte 2. und 3. sind klar, da sie so definiert sind. Punkt 3. folgt aus dem Feedback, den die inneren Zustände P^1 durch die Bayessche Formel 6.22 erhalten[11], und somit einem Auseinanderlaufen der Wahrscheinlichkeitsverteilung, die durch 6.15 bewirkt wird, entgegenwirkt. Die Forderung, dass P^1 nur im stationären Zustand scharf gepeakt ist bedeutet einfach, dass durch eine unvollständig gelernte innere Repräsentation $P^S(s|i)$, infolge des Bayesschen Feedback auf P^1, die Form von P^1 stark gestört werden kann.
Setzt man alle drei Punkte voraus, so kann man folgern, dass der Agent eine richtige Vorhersage $\overline{s}$ macht, sobald die Relation

$$P^S(s_t|\lfloor < i > \rceil) \; > \; P^S(s \neq s_t|\lfloor < i > \rceil) \tag{6.34}$$

[9] Auch nicht für den Spezialfall einer eindimensionalen äußeren Umgebung mit den in 6.4.1. angenommenen Einschränkungen.
[10] Da beispielsweise nicht bekannt ist, wie man aus der Gesamtstruktur in Kapitel 6.4.1. bis 6.4.5. eine Liapunovfunktion bestimmt, die als Kostenfunktion verwendet werden könnte
[11] Siehe Erklärung in Kapitel 6.4.4.

richtig gelernt wurde[12]. Hier bezeichnet $\lfloor < i > \rceil = \lfloor \sum_i^I i P^1(i) \rceil$ den auf- bzw. abgerundeten Mittelwert der inneren Zustände, der mit der Gitterposition des Agenten $\hat{x}$ korrespondieren muss. Es ist also nicht notwendig, die tatsächlich vorhandene, äußere bedingte Wahrscheinlichkeit $\hat{P}^S(\hat{s}|\hat{x})$ exakt zu lernen, um eine richtige Vorhersage über die zu beobachtenden Symbole s zu erhalten.
Den approximierten individuellen Gesamtfehler $\tilde{E}_{iabs}$ erhält man somit aus

$$\tilde{E}_{iabs}(t) = \frac{\#\text{der bereits gelernten Relationen zum Zeitpunkt } t}{E} \tag{6.35}$$

Hier bezeichnet E die Gesamtzahl aller Relationen, welche für zwei Beobachtungssymbole mit der Größe des Gitters übereinstimmt.

Nun bleibt noch die Frage zu klären, wann 6.35 eine gute Approximation für 6.33 darstellt. Da bei Punkt 2 die Stationarität von P^1 gefordert wurde, die im allgemeinen erst erreicht wird, wenn nahezu die gesamte Repräsentation gelernt wurde, wird 6.35 mit 6.33 immer besser übereinstimmen, wenn $\tilde{E}_{iabs}$ gegen Null geht. Dies ist aber genau der Fall, der hier von Interesse ist, da bei den folgenden Untersuchungen nur der Zeitpunkt bestimmt werden soll, an dem der approximierte individuelle Gesamtfehler zum ersten Mal Null wird. Es wird also, ähnlich wie in Kapitel 5., die *first-passage* Zeit T_{FPT} bestimmt, wann $\tilde{E}_{iabs}$ das erste Mal die Schwelle Null erreicht.

Somit kann zusammenfassend festgehalten werden, dass der approximierte individuelle Gesamtfehler $\tilde{E}_{iabs}(t)$ durch 6.35 zu jedem Zeitpunkt berechnet, jedoch nur $\tilde{E}_{iabs}(T_{FPT}) = 0$ verwendet wird, wodurch der Fehler zu E_{iabs} in Gleichung 6.33 minimal wird und die *first-passage* Zeit T_{FPT} folgt. Aus T_{FPT} und der dazugehörigen Wahrscheinlichkeitsverteilung p^{FPT} lassen sich dann, wie in Kapitel 5.3.4. angegeben, verschiedene Größen berechnen, wie die Verteilungsfunktion P^E und die *mean first-passage time* $< T_{FPT} >$.

Es soll nicht unerwähnt bleiben, dass die bisherigen Ausführungen eine gewisse Analogie zu der Arbeit von [Turing, 1950] aufweisen, in der die Entscheidung darüber, ob ein System denken kann und somit als intelligent bezeichnet wird, auf ein Imitationspiel zwischen einem Frager und zwei Befragten (Mensch oder Maschine) übertragen wurde. Der Frager hat die Möglichkeit, mit den Befragten eine gewisse Zeit lang zu kommunizieren, und muss dann, aufgrund seines Eindrucks, den er durch die Kommunikation mit den Befragten gewonnen hat, entscheiden, welcher der Befragten ein Mensch (Mann oder Frau) und welcher eine Maschine ist. In abgewandelter Form ist dieses Imitationsspiel als Turing-Test bekannt geworden [French, 2000].
Die Analogie zwischen dem Turing-Test und dem individuellen Gesamtfehler

[12] Dies macht man sich am besten klar, indem man $P^1(i) = \delta_{i,\hat{x}}$ setzt.

E_{iabs} besteht nicht darin, dass hier entschieden werden soll, ob der in Kapitel 6.4.1. bis 6.4.5. definierte autonome Agent intelligent mit seiner Umgebung wechselwirkt, sondern darin, dass der Agent in einer Testumgebung eine gewisse Zeit lang befragt wird. Die einzigen Fragen, die der Agent in seiner Umgebung beantworten kann und die für einen äußeren Beobachter (Frager) einen Sinn ergeben, sind die nach dem nächsten zu erwartenden Beobachtungssymbol $\overline{s}$.

6.5 Ergebnisse

Im folgenden wird der in Kapitel 6.4. vorgestellte mathematische Formalismus zur Beschreibung eines autonomen Agenten in einer raumartigen Umgebung untersucht. Hierbei steht die Auswirkung verschiedener Aktions-Selektions-Mechanismen auf das Lernen einer internen Repräsentation einer äußeren Umgebung im Vordergrund. Alle Simulationen wurden, sofern keine anderen Angaben gemacht werden, mit einem Ensemble der Größe $N = 5000$ durchgeführt.

In Abbildung 6.5 ist die *mean first-passage time* $< T_{FPT} >$ für Beobachtungs- und Positionsfehler von $p^{oe} = p^{pe} = 0$ (durch $+$ gekennzeichnet) und $p^{oe} = p^{pe} = 0.02$ (durch $\Diamond$ gekennzeichnet) in Abhängigkeit verschiedener Gittergrößen E doppellogarithmisch dargestellt. Die blauen Kurven wurden mit dem Active Walk, die grünen mit dem Linear Walk und die roten mit dem Random Walk als Aktions-Selektions-Mechanismus ermittelt.
Man erkennt, dass im Fall verschwindender Beobachtungs- und Positionsfehler der Verlauf der *mean first-passage time* $< T_{FPT} >$ für den Active, Linear und Random Walk einem Potenzgesetz folgt, wohingegen $< T_{FPT} >$ für $p^{oe} = p^{pe} = 0.02$ für alle drei Aktions-Selektions-Mechanismen exponentiell ansteigt.
Hieraus folgt, dass der in Kapitel 6.4. vorgestellte mathematische Formalismus grundsätzlich in der Lage ist, sowohl im idealen Fall von verschwindendem Beobachtungs- und Positionsfehler als auch unter realistischeren Bedingungen für $p^{oe} = p^{pe} = 0.02$, eine interne Repräsentation zu bilden. Das exponentielle Ansteigen der *mean first-passage time* $< T_{FPT} >$ deutet jedoch darauf hin, dass die Bildung der internen Repräsentation für Beobachtungs- und Positionsfehler, die ungleich Null sind, sehr zeitaufwendig sein kann.

In Abbildung 6.6 ist das zu Abbildung 6.5 analoge Ergebnis für Beobachtungs- und Positionsfehler von $p^{oe} = 0$, $p^{pe} = 0.02$ (durch $+$ gekennzeichnet) und $p^{oe} = 0.1$, $p^{pe} = 0$ (durch $\Diamond$ gekennzeichnet) in Abhängigkeit verschiedener Gittergrößen E doppellogarithmisch dargestellt.
Auffällig ist hier, dass alle mittleren *first-passage* Zeiten $< T_{FPT} >$ exponentiell

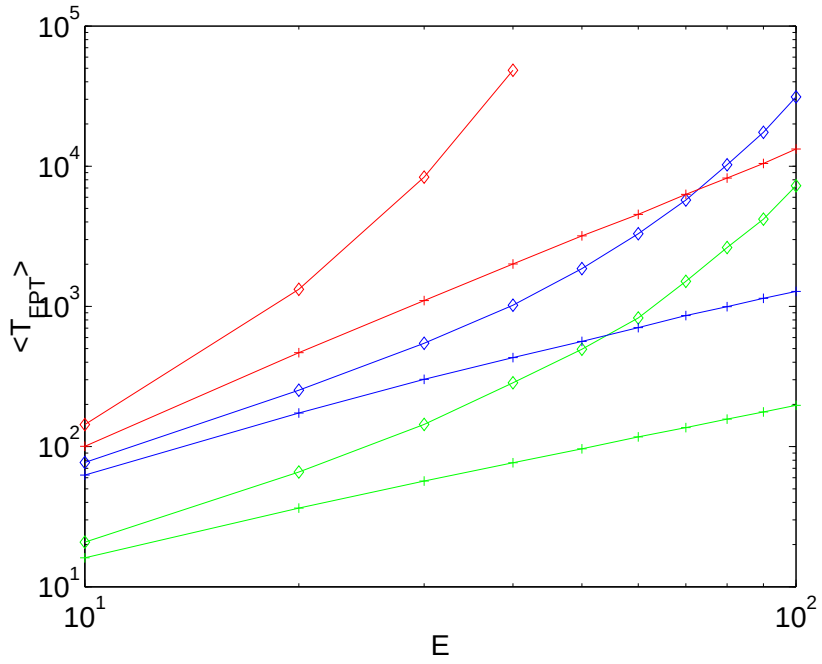


Abbildung 6.5: Darstellung der *mean first-passage time* $< T_{FPT} >$ in Abhängigkeit der Gittergröße E. Die blauen Kurven wurden mit dem Active Walk, die grünen mit dem Linear Walk und die roten mit dem Random Walk als Aktions-Selektions-Mechanismus ermittelt. Die Kreuze $+$ symbolisieren den Fall $p^{oe} = p^{pe} = 0$ und die Diamanten $\Diamond$ stehen für $p^{oe} = p^{pe} = 0.02$. Die Gedächtnislänge der *self-reward map* betrug $L = 10$.

ansteigen, bis auf den Active Walk für $p^{oe} = 0.1$, $p^{pe} = 0$. Für diesen Fall ist es sogar so, dass die Werte von $< T_{FPT} >$ ab einer Gittergröße von $E = 50$ kleiner werden als für den Linear Walk.

Um diesen Sachverhalt zu analysieren, betrachte man die Verteilungsfunktion $p^{E=0}$ in Abbildung 6.7. Dort ist $p^{E=0}$ für den Active (blaue Kurven) und Linear (rote Kurven) Walk für zwei unterschiedliche Gittergrößen $E = 30$ und $E = 100$ und alle vier bisher untersuchten Paare von Beobachtungs- und Positionsfehler dargestellt.
Für den Fall $p^{oe} = 0.1$, $p^{pe} = 0$ fällt auf, dass die Verteilungsfunktion des Active Walk für $t \sim 10^3$ einen deutlichen Knick aufweist. Dieser Knick ist ein erster Hinweis auf die Auswirkung der Reinitialisierung der *self-reward map*, falls $P^{rel}_{sr,L} = 0$ eintrit. Alle anderen Verteilungsfunktion des Active Walk lassen solch einen geknickten Verlauf höchstens erahnen, jedoch nicht wirklich erkennen. Diese Frage lässt sich mit Hilfe von Abbildung 6.9 klären, welche die Histogramme der Reinitialisierungszeiten zeigt, die weiter unten diskutiert werden wird.

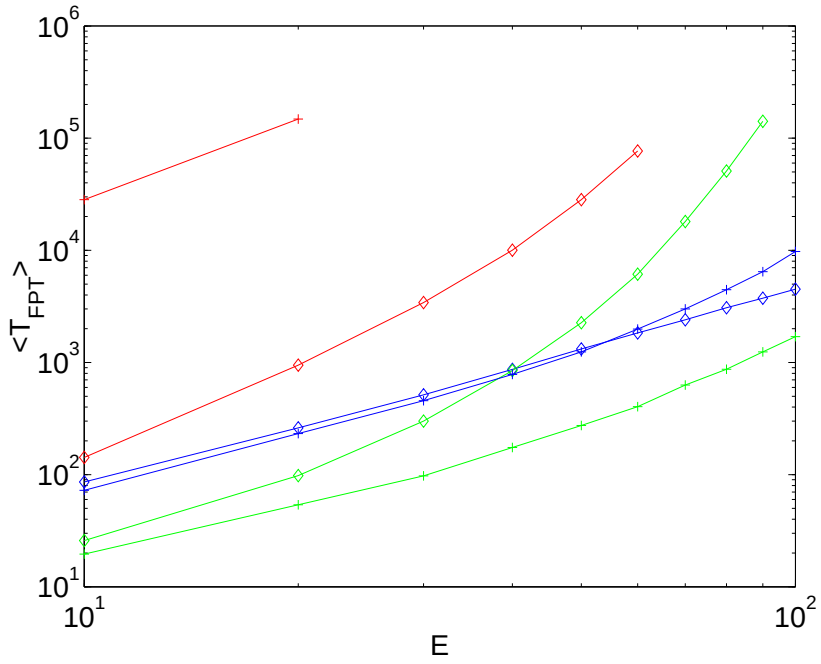


Abbildung 6.6: Darstellung der *mean first-passage time* $< T_{FPT} >$ in Abhängigkeit der Gittergröße E. Die blauen Kurven wurden mit dem Active Walk, die grünen mit dem Linear Walk und die roten mit dem Random Walk als Aktions-Selektions-Mechanismus ermittelt. Die Kreuze $+$ symbolisieren den Fall $p^{oe} = 0$, $p^{pe} = 0.02$ und die Diamanten $\Diamond$ stehen für $p^{oe} = 0.1$, $p^{pe} = 0$. Die Gedächtnislänge der *self-reward map* betrug $L = 10$.

Für den Linear Walk erkennt man in Abbildung 6.7 den sehr steilen Anstieg in p^{FPT} für $p^{oe} = p^{pe} = 0$. Dies liegt daran, dass im fehlerfreien Fall eine Gitterzelle im Mittel nur zweimal besucht werden muss, bevor eine korrekte Vorhersage über das Zellsymbol getroffen werden kann, und der Linear Walk im wahrsten Sinn des Wortes "geradlinig" über das Gitter läuft. Aus diesem Grund ist der Linear Walk für diesen Fall auch besser geeignet als der Active und insbesondere der Random Walk. Dies liegt daran, dass der Active Walk insbesondere durch die Reinitialisierung der *self-reward map* einen Diffusionsanteil enthält und der Random Walk ohnehin ein reiner Diffusionsprozess ist. Im Falle $p^{oe} = 0$, $p^{pe} = 0.02$ sieht man sehr schön, wie aus der steilen Flanke in p^{FPT} ein treppenartiger Anstieg wird. Dies liegt daran, dass jetzt einige wenige Beobachtungssymbole nach im Mittel 2 Überquerungen des gesamten Gitters noch nicht gelernt wurden, der Linear Walk aber den Agenten zwingt, immer das gesamte Gitter zu durchlaufen, bevor erneut eine Beobachtung an den noch zu lernenden Zellen vorgenommen werden kann. Für größer werdende Gitter sieht man, wie die Höhe der Treppenstufen immer niedriger wird und besonders der Anfangssprung immer kleiner wird.

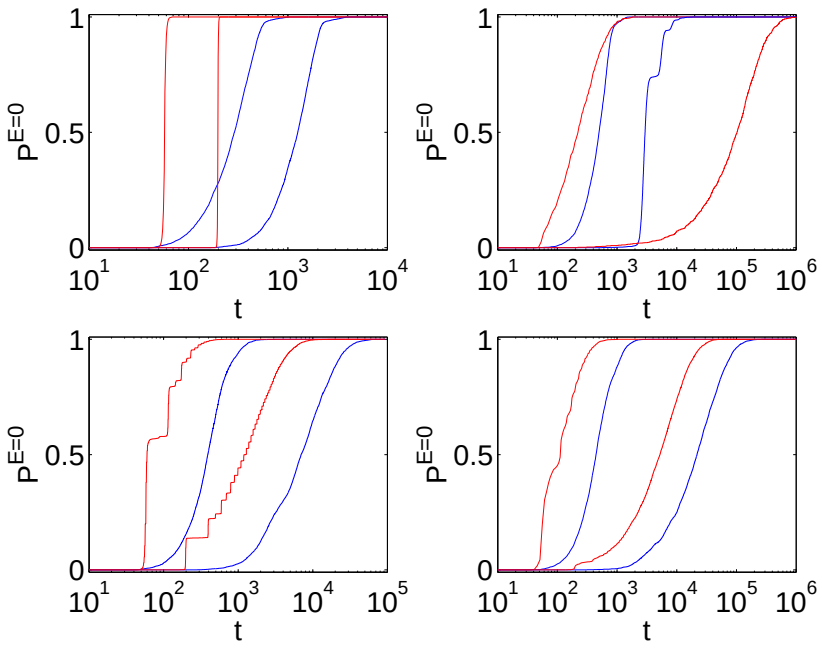


Abbildung 6.7: Darstellung der Verteilungsfunktion $P^{E=0}$ des Linear (rote Kurve) und Active (blaue Kurve) Walks für Gitter der Größe $E = 30$ und $E = 100$. Die früher ansteigende Kurve entspricht jeweils der Gittergröße $E = 30$. Oben links: $p^{oe} = p^{pe} = 0$. Oben rechts: $p^{oe} = 0.1$, $p^{pe} = 0$. Unten links: $p^{oe} = 0$, $p^{pe} = 0.02$. Unten rechts: $p^{oe} = p^{pe} = 0.02$. Die Gedächtnislänge der *self-reward map* betrug $L = 10$.

Dieses Szenario spitzt sich durch den Fall $p^{oe} = 0.1$, $p^{pe} = 0$ noch weiter zu. Hier lässt sich die Stufenform gar nicht mehr erkennen und die Konvergenzzeiten steigen für größer werdende Gitter so stark an, dass schließlich ab $E = 50$ der Active Walk den besseren Aktions-Selektions-Mechanismus darstellt.

Um den Einfluss der Reinitialisierungen der *self-reward map* des Active Walk genauer zu untersuchen, wurden in Abbildung 6.8 die Wahrscheinlichkeitsverteilungen p^{FPT} für den Fall $p^{oe} = 0.1$, $p^{pe} = 0$ für verschiedene Gittergrößen dargestellt. Man erkennt, dass sich ab einer Gittergröße von $E = 50$ aus der glockenförmigen Verteilung eine mehrfach gepeakte Verteilung deutlich herausbildet. Hierdurch wird der Verlauf der Verteilungsfunktion $P^{E=0}$ in Abbildung 6.7, die sich ja aus der Wahrscheinlichkeitsverteilung p^{FPT} berechnet, klar. Eine Erklärung für die Lage der Maxima der auftretenden Peaks wird durch Abbildung 6.9 gegeben, in der die Histogramme der Reinitialisierungszeiten dargestellt sind, die weiter unten diskutiert werden wird.
Im Vergleich hierzu ist die Wahrscheinlichkeitsverteilung p^{FPT} für den Linear

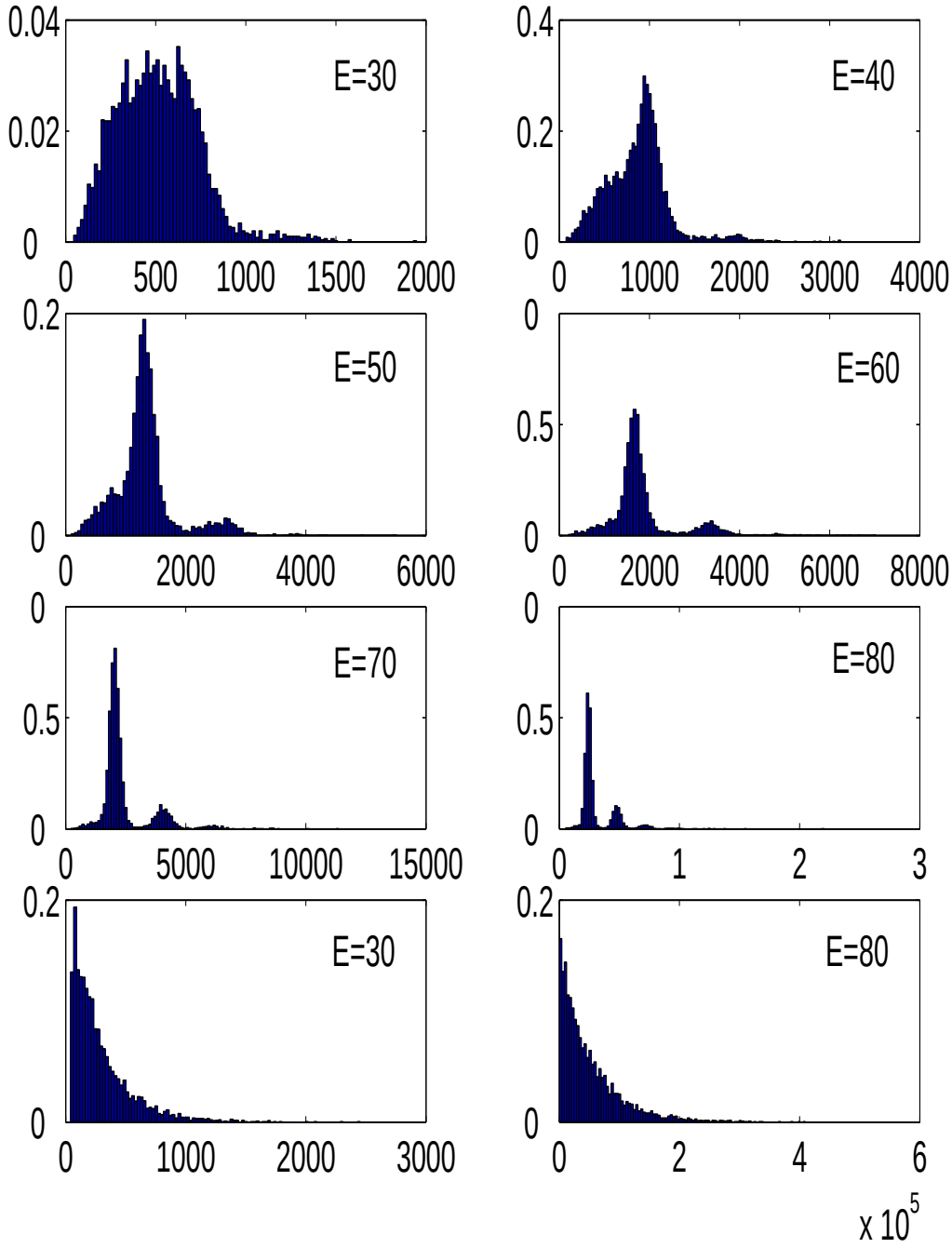


Abbildung 6.8: Darstellung der Wahrscheinlichkeitsverteilung p^{FPT} für den Fall $p^{oe} = 0.1$, $p^{pe} = 0$. Die ersten sechs Verteilungen wurden mit dem Active Walk und die letzten beiden mit dem Linear Walk als Aktions-Selektions-Mechanismus ermittelt. Die Abbildungstitel geben die jeweils verwendete Gittergröße an. Die Gedächtnislänge der *self-reward map* betrug $L = 10$.

Walk in den unteren beiden Graphen in 6.7 dargestellt. Die Verteilung ändert sich in ihrer Form für verschieden große Gitter nicht, was auch zu erwarten war.

Zum besseren Verständnis der Arbeitsweise des Active Walk wurden in Abbildung 6.9 die Histogramme der Reinitialisierungszeiten H^r für ein Gitter der Größe $E = 80$ für alle vier bisher untersuchten Paare von Beobachtungs- und Positionsfehler dargestellt. Man erkennt, dass die Peaks in der Wahrscheinlichkeitsverteilung p^{FPT}, bzw. die Knicke in der Verteilungsfunktion $P^{E=0}$, ihren Ursprung in den Reinitialisierungen der *self-reward map* haben und die Zeitpunkte der jeweiligen Maxima der Peaks etwas früher liegen, als die Maxima der Peaks in den entsprechenden Wahrscheinlichkeitsverteilungen p^{FPT}. Dies lässt sich direkt durch einen Vergleich der oberen, rechten Abbildung in 6.9 mit der Abbildung in der dritten Zeile und rechten Spalte in 6.8 ablesen.
Weiterhin kann erklärt werden, warum in der Verteilungsfunktion $P^{E=0}$ in Abbildung 6.7 für $p^{oe} = p^{pe} = 0$ und $p^{oe} = 0$, $p^{pe} = 0.02$ und $p^{oe} = p^{pe} = 0.02$

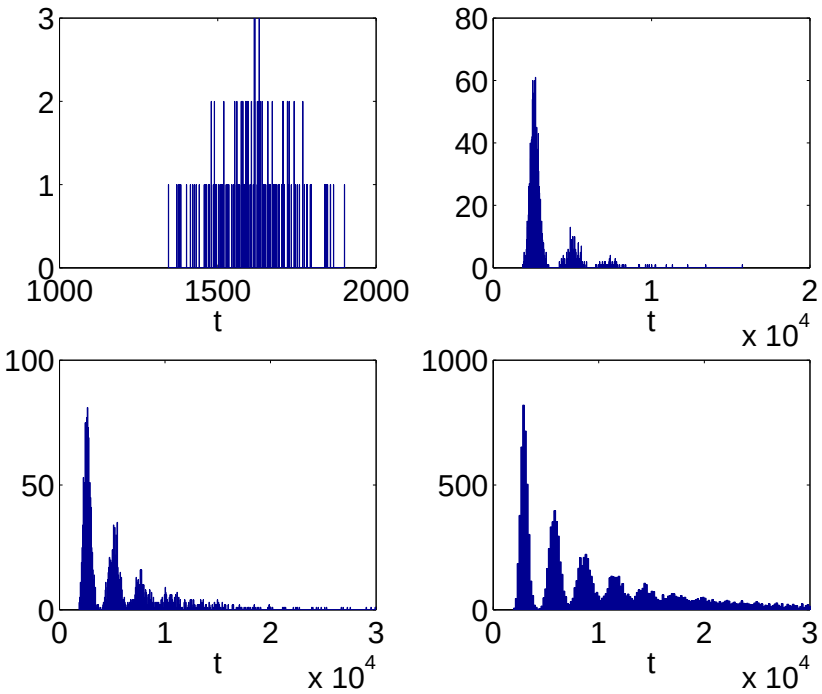


Abbildung 6.9: Darstellung der Histogramme der Reinitialisierungszeiten für den Active Walk in einem Gitter der Größe $E = 80$. Die Gedächtnislänge der *self-reward map* betrug für alle Simulationen $L = 10$. Oben links: $p^{oe} = p^{pe} = 0$. Oben rechts: $p^{oe} = 0.1$, $p^{pe} = 0$. Unten links: $p^{oe} = 0$, $p^{pe} = 0.02$. Unten rechts: $p^{oe} = p^{pe} = 0.02$.

keine Knicke beobachtet werden konnten. Für $p^{oe} = p^{pe} = 0$ ist dies klar, da das zugehörige Histogramm der Reinitialisierungszeiten H^r in 6.9 glockenförmig ist und keine periodischen Peaks aufweist. Die beiden anderen Fälle sind nicht so trivial, da die beiden unteren Histogramme H^r in Abbildung 6.9 eben gerade periodische Peaks aufweisen. Jedoch sind diese Peaks, im Vergleich zu H^r für $p^{oe} = 0.1$, $p^{pe} = 0$, für zunehmende Rangordnung der Peaks in einen immer größer werdenden Untergrund eingebettet. Hierdurch verschmieren die Peaks in den Wahrscheinlichkeitsverteilungen und die Knicke in den Verteilungsfunktionen.

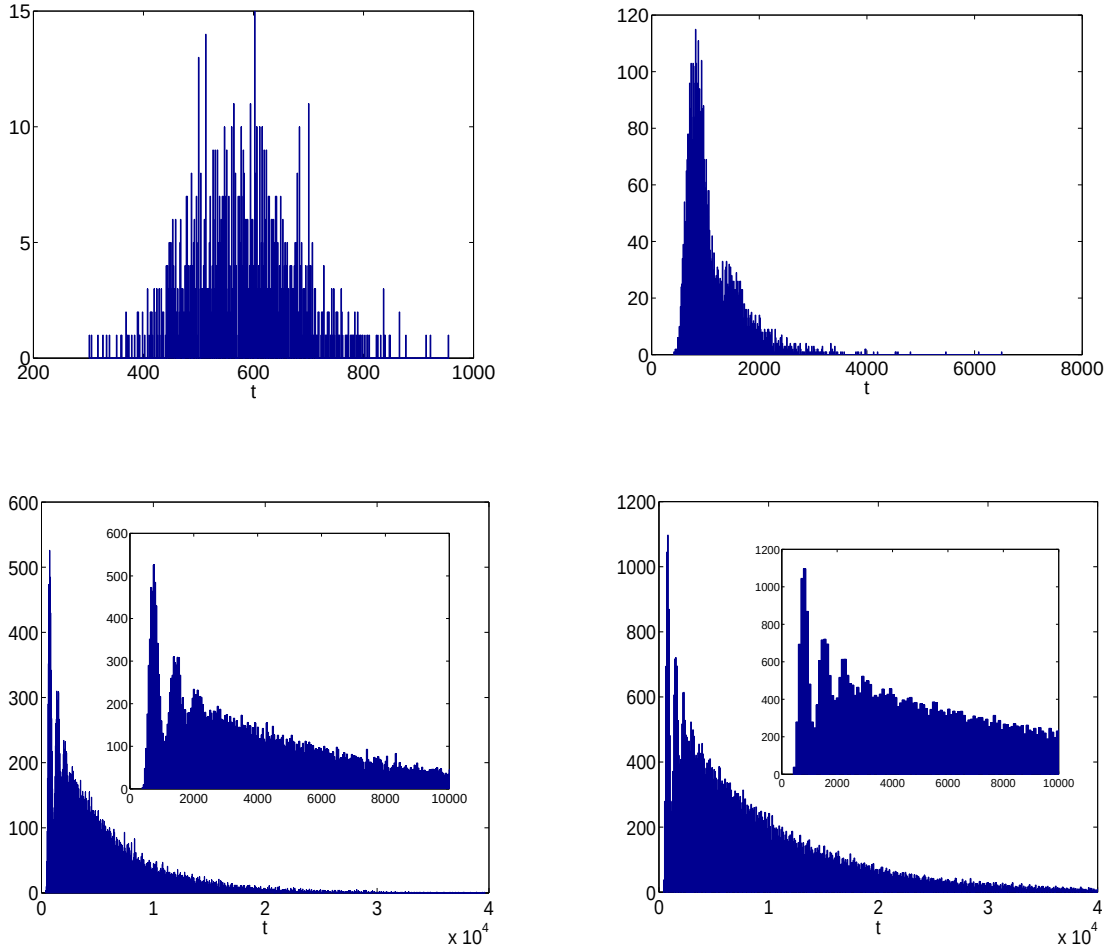


Abbildung 6.10: Darstellung der Histogramme der Reinitialisierungszeiten H^r für den Active Walk in einem Gitter der Größe $E = 80$. Die Gedächtnislänge der *self-reward map* betrug für alle Simulationen $L = 2$. Oben links: $p^{oe} = p^{pe} = 0$. Oben rechts: $p^{oe} = 0.1$, $p^{pe} = 0$. Unten links: $p^{oe} = 0$, $p^{pe} = 0.02$. Unten rechts: $p^{oe} = p^{pe} = 0.02$.

Nachdem nun die Arbeitsweise des Active Walk und dessen antreibenden Mechanismus in Form der auftretenden Reinitialisierungen der *self-reward map* geklärt ist, stellt sich die Frage, wie sich eine Veränderung der Gedächtnislänge L der *self-reward map* auf das Konvergenzverhalten auswirkt. Aufgrund bisher erhaltener Ergebnisse über die Histogramme der Reinitialisierungszeiten H^r in Abbildung 6.9 und den mittleren *first-passage* Zeiten $< T_{FPT} >$ in Abbildung 6.5 und 6.6 liegt es nahe, die Gedächtnislänge L zu reduzieren. Dies sieht man besonders deutlich für den Fall $p^{oe} = p^{pe} = 0$. Denn hier setzen die ersten Reinitialisierungen erst ein, nachdem der Agent, der als Aktions-Selektions-Mechanismus den Linear Walk verwendet, bereits eine interne Repräsentation der Umgebung gelernt hat. Deshalb wurde bei den folgenden Untersuchungen $L = 2$ gewählt.

In Abbildung 6.10 sind die Histogramme der Reinitialisierungszeiten H^r für ein Gitter der Größe $E = 80$ für alle vier bisher untersuchten Paare von Beobachtungs- und Positionsfehler dargestellt. Man erkennt für alle untersuchten Fälle deutlich, dass die Reinitialisierungen zu niedrigeren Zeiten hin verschoben sind. Die jeweils korrespondierende Wahrscheinlichkeitsverteilung p^{FPT} ist in Abbildung 6.11 graphisch dargestellt.

Hier sei besonders darauf hingewiesen, dass in p^{FPT} für die Fälle $p^{oe} = 0$, $p^{pe} = 0.02$ und $p^{oe} = p^{pe} = 0.02$ keine Peaks mehr aufgelöst werden können, da die jeweils zugehörigen Peaks in H^r in einen sehr großen Untergrund eingebettet sind. Die zu diesen Parameterwerten zugehörigen mittleren *first-passage* Zeiten betragen von links oben nach rechts unten $< T_{FPT} >= \{587, 1396, 5260, 9415\}$.
Bis auf den Fall $p^{oe} = 0$, $p^{pe} = 0.02$ hat sich die Verkürzung des Gedächtnisses L der *self-reward map* positiv auf die erhaltenen mittleren *first-passage* Zeiten, verglichen mit den Werten in Abbildung 6.5 und 6.6, ausgewirkt. Das kontraproduktive Verhalten für $p^{oe} = 0$, $p^{pe} = 0.02$ lässt sich durch den Feedback von Bayes in Form von 6.22 auf die inneren Zustände P^1 in Kombination mit einer noch unvollständig gelernten inneren Repräsentation, die auf den bedingten Wahrscheinlichkeiten $P^1(i|s)$ basiert, erklären. Vereinfacht ausgedrückt lässt sich das, durch ein Positionsfehler verursachte Auseinanderlaufen von P^1, am besten dadurch kompensieren, dass der Agent sich zu den Gitterrändern hin bewegt, da dort, aufgrund der periodischen Randbedingungen der inneren Zustände, die vorher auseinander gelaufene Verteilung automatisch verschärft wird. Dies wird aber gerade durch einen Prozess mit Drift, wie dem Linear Walk, und einen so niedrig wie möglichen Diffusionsanteil realisiert. Da der Diffusionsanteil des Active Walk für eine kleiner werdende Gedächtnislänge L infolge ansteigender Häufigkeiten in den Reinitialisierungen anwächst, werden die hierdurch erhaltenen Ergebnisse schlechter.

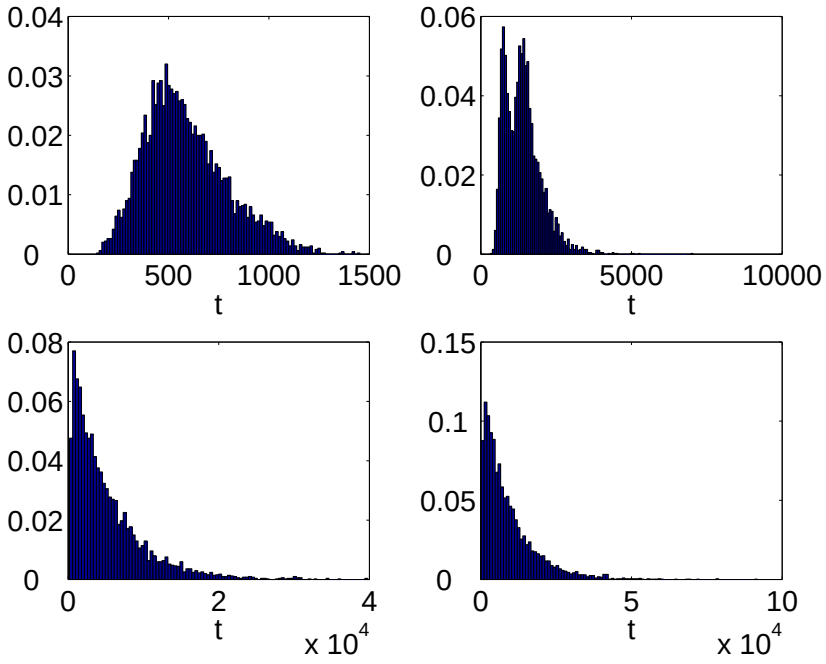


Abbildung 6.11: Darstellung der Wahrscheinlichkeitsverteilung p^{FPT} für den Active Walk in einem Gitter der Größe $E = 80$. Die Gedächtnislänge der *self-reward map* betrug für alle Simulationen $L = 2$. Oben links: $p^{oe} = p^{pe} = 0$. Oben rechts: $p^{oe} = 0.1$, $p^{pe} = 0$. Unten links: $p^{oe} = 0$, $p^{pe} = 0.02$. Unten rechts: $p^{oe} = p^{pe} = 0.02$.

Die bisher durchgeführten Untersuchungen ergaben, dass für die Fälle $p^{oe} = p^{pe} = 0$ und $p^{oe} = 0$, $p^{pe} = 0.02$ der Linear Walk die besten Resultate bzgl. der mittleren *first-passage* Zeit lieferte. Da der Grund hierfür, wie oben geschildert, auf die Drift zurückzuführen ist, kann der Active Walk für diese Parameterkonfigurationen keine besseren Resultate liefern, da der Anteil der Diffusion überwiegt. Für $p^{oe} = 0.1$, $p^{pe} = 0$ hingegen lieferte der Active Walk die besten Ergebnisse, da aufgrund des relativ hohen Beobachtungsfehlers die *self-reward map* ihre positiven Eigenschaften bei der Aktionsauswahl einbringen konnte.

Nun stellt sich die Frage, ob es möglich ist, für den realistischsten Fall von nicht verschwindenden Beobachtungs- und Positionsfehler, eine geeignete Parametereinstellung des Active Walk zu finden, für die der Active Walk bessere Resultate erzielt als der Linear Walk.
Um diesen Punkt zu untersuchen, wurde für die folgenden Simulationen ein Beobachtungs- und Positionsfehler von $p^{oe} = 0.07$, $p^{pe} = 0.02$ angenommen. Den Grund hierfür kann man aus Abbildung 6.5 ablesen, denn die mittlere

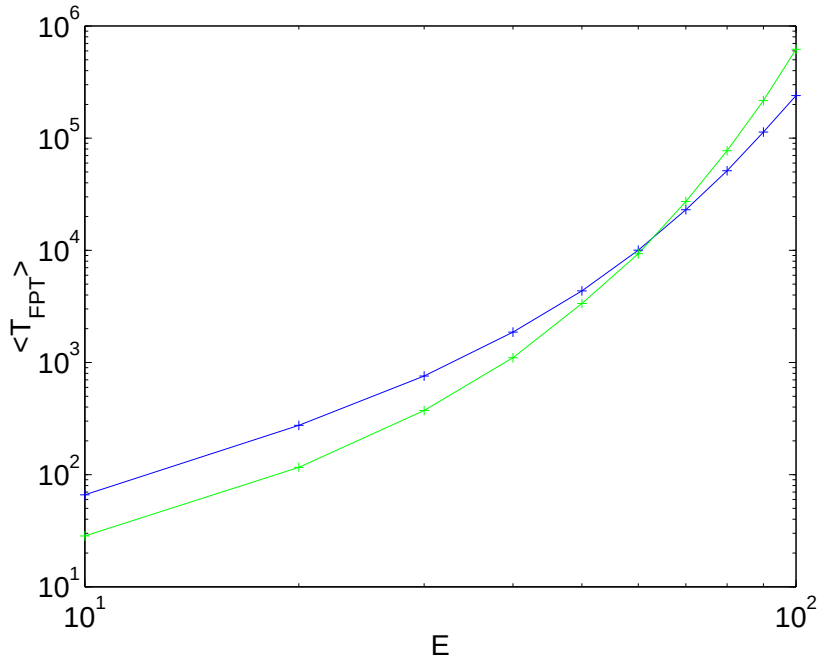


Abbildung 6.12: Darstellung der *mean first-passage time* $< T_{FPT} >$ in Abhängigkeit der Gittergröße E für den Active (blaue Kurve) Walk und den Linear (grüne Kurve) Walk. Die Gedächtnislänge der *self-reward map* betrug $L = 2$ und die Beobachtungs- und Positionsfehler $p^{oe} = 0.07$ und $p^{pe} = 0.02$.

first-passage Zeit liegt für den Linear Walk im Fall $p^{oe} = p^{pe} = 0.02$ und $E = 100$ im Bereich $< T_{FPT} >\sim 10^4$. Diese zeitliche Größenordnung ist jedoch zu niedrig, um einen auf Diffusion basierenden Mechanismus wie den Active Walk, effektiv einsetzen zu können. Deshalb wird die Schwierigkeit des zu lernenden Problems dadurch erhöht, dass der Beobachtungsfehler auf $p^{oe} = 0.07$ erhöht wird.
In Abbildung 6.12 sind die mittleren *first-passage* Zeiten für den Active (blaue Kurve) und Linear (grüne Kurve) Walk in Abhängigkeit der Gittergröße dargestellt[13]. Als Gedächtnislänge der *self-reward map* wurde $L = 2$ gewählt. Man sieht, wie ab einer Gittergröße von $E = 70$ die mittleren *first-passage* Zeiten für den Linear Walk größer werden als für den Active Walk, wodurch gezeigt ist, dass die Verwendung der *self-reward map* für ein schwerer zu lernendes Problem einen funktionellen Vorteil liefert.
Leider ist der Anstieg beider Kurven exponentiell, was den Erfolg des Active Walk drastisch schmälert, da ein sehr guter Aktions-Selektions-Mechanismus nach einem Potenzgesetz ansteigen müsste, um ein Explodieren in der Lernzeit für zunehmende Gittergrößen zu verhindern.

[13]Die Ensemblegröße betrug für den Linear Walk $N = 4000/800$ für $E = 90/100$ und für den Active Walk $N = 2000$ für $E = 100$. Ansonsten wie bisher immer $N = 5000$.

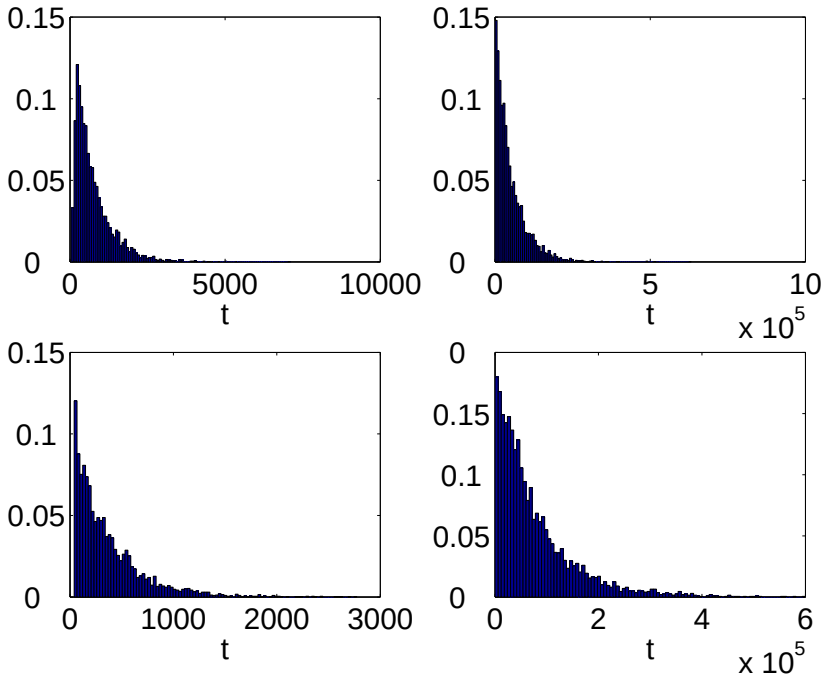


Abbildung 6.13: Darstellung der Wahrscheinlichkeitsverteilung p^{FPT} für den Active (obere Abbildungen) und Linear (untere Abbildungen) Walk für Beobachtungs- und Positionsfehler von $p^{oe} = 0.07$ und $p^{pe} = 0.02$. Die Gedächtnislänge der *self-reward map* betrug $L = 2$. Linke Seite: Gittergröße $E = 30$. Rechte Seite: Gittergröße $E = 80$.

In Abbildung 6.13 ist die Wahrscheinlichkeitsverteilung p^{FPT} für den Active (oben) und Linear (unten) Walk für den Fall $p^{oe} = 0.07$ und $p^{pe} = 0.02$ für Gittergrößen von $E = 30$ und $E = 80$ dargestellt. Die Verteilungen sehen sich für gleiche Gittergrößen recht ähnlich, da nun auch die Peaks für große Gitterwerte in p^{FPT} nicht mehr aufgelöst werden. Dies lässt sich wiederum durch das Histogramm H^r der Reinitialisierungszeiten erklären, dass in der oberen Abbildung in 6.14 dargestellt ist.
Für dieses Histogramm H^r ist in der mittleren Abbildung 6.14 das Histogramm der Inter-Reinitialisierungs-Intervalle H^{IRI} aufgetragen, also die Differenz der Reinitialisierungszeiten $\Delta t_r = t_{k+1} - t_k$ für zwei aufeinanderfolgende Reinitialisierungen[14]. Man erkennt hieran sehr schön den periodischen Charakter der Reinitialisierungen, der im Mittel nach $< t >_r = 860$ Zeitschritten erfolgt[15].
In der unteren Abbildung 6.14 ist die Wahrscheinlichkeitsverteilung p_k^{peak} für das Eintreten der k'ten Reinitialisierung aufgetragen. Durch die halblogarith-

[14]Hier gibt t_k den Zeitpunkt der k'ten Reinitialisierung an.
[15]In H^{IRI} sind die Differenzen Δt_r aus allen Ensemblesimulationen enthalten.

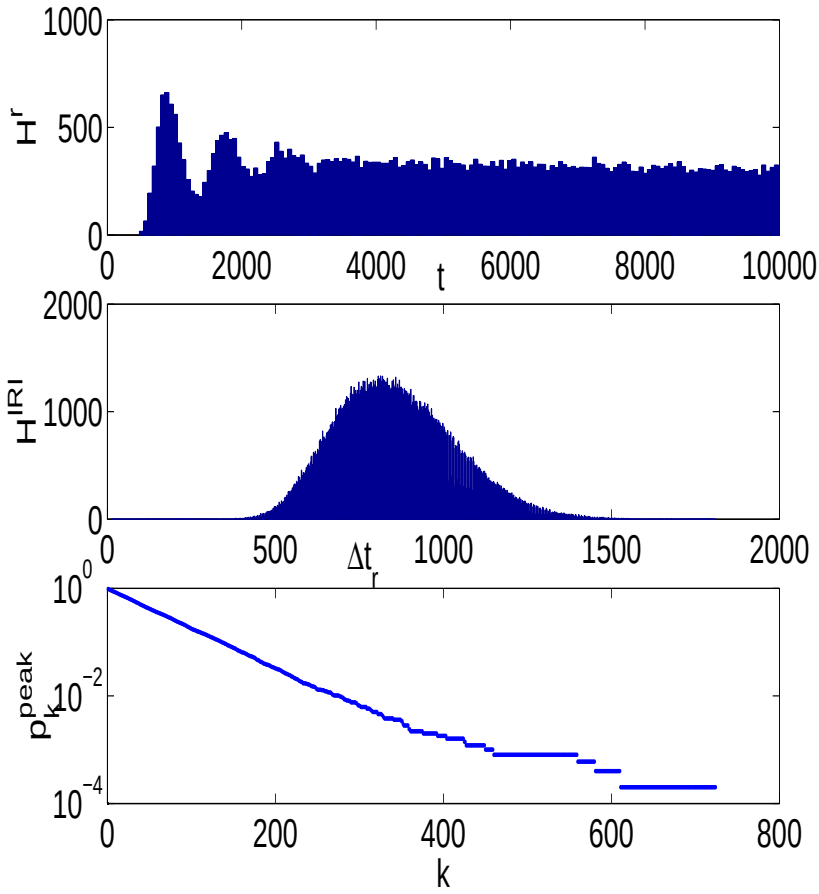


Abbildung 6.14: Alle Darstellungen wurden für einen Active Walk in einem Gitter der Größe $E = 80$ ermittelt. Die Gedächtnislänge der *self-reward map* war $L = 2$ und der Beobachtungs- und Positionsfehler betrug $p^{oe} = 0.07$ und $p^{pe} = 0.02$. Die Ensemblegröße betrug $N = 5000$. Oben: Histogramm H^r der Reinitialisierungszeiten. Mitte: Histogramm H^{IRI} der Differenz der Reinitialisierungszeiten Δt_r für zwei aufeinanderfolgende Reinitialisierungen (Inter-Reinitialisierungs-Intervalle). Unten: Wahrscheinlichkeitsverteilung p_k^{peak} für das Eintreten der k'ten Reinitialisierung.

mische Skalierung erkennt man den exponentiellen Charakter der Verteilung, der sich bis zu 600 Peaks erstreckt[16].

[16]Der Fall $p^{oe} = 0.07$ und $p^{pe} = 0.02$ eignet sich hierfür besser als alle anderen, da die auftretenden Lernzeiten sehr groß sind und deshalb sehr oft reinitialisiert wird.

6.6 Diskussion

Die erhaltenen Ergebnisse der durchgeführten Simulationen haben gezeigt, dass die in Kapitel 6.4. vorgestellte interne Struktur des autonomen Agenten in der Lage ist, eine Repräsentation eines äußeren eindimensionalen Gitters zu lernen. Genauer gesagt lag die *first-passage* Zeit für alle durchgeführten und in Kapitel 6.5. präsentierten Resultate immer unterhalb der maximalen Simulationszeit.

Der Vergleich von drei unterschiedlichen Aktions-Selektions-Mechanismen ergab enorme Unterschiede in den resultierenden mittleren *first-passage* Zeiten $< T_{FPT} >$. Dies ist deshalb von besonderem Interesse, da man hierbei beachten muss, dass die Simulationen mit drei identischen Agenten in drei identischen Umgebungen durchgeführt wurden, und die drei Agenten sich einzig in dem verwendeten Aktions-Selektions-Mechanismus unterschieden, der verwendet wurde, um die nächste Aktion auszuwählen. Darüberhinaus stellt jeder dieser drei Aktions-Selektions-Mechanismen eine vernünftige Wahl für einen Aktions-Selektions-Mechanismus dar, denn die interne Repräsentation konnte jeweils gelernt werden.
Dadurch kam klar zum Ausdruck, dass die Lerndynamik eines Gesamtsystems nicht nur durch die verwendete Lernregel maßgeblich bestimmt wird, sondern durch die Gesamtheit aller Komponenten in 3.1, insbesondere durch den verwendeten Aktions-Selektions-Mechanismus. Hieraus folgt, dass es keine guten oder schlechten Lernregeln bzw. Aktions-Selektions-Mechanismen in dem Sinn gibt, sondern nur schlecht oder gut zusammenpassende Komponenten des Gesamtsystems. Aus diesem Grund wird ein Verständnis von Lerndynamiken natürlicher Systeme nur erreicht werden, wenn die Wahrnehmungs-Handlungs-Schleife des Organismus untersucht wird.

Von besonderer Einfachheit war die Struktur des Active Walk, die auf einem Wechselspiel zwischen Drift und Diffusion in Form einer zu adaptierenden *self-reward map* und deren Reinitialisierung beruhte. Dieser rein regelbasierte Mechanismus kam ohne die Verwendung von global definierten Begriffen wie Fehler oder Information aus, die nur einem äußeren Beobachter, der über die Information des Gesamtsystems verfügt, zur Verfügung stehen. Interessanterweise konnte der Agent mit dem Active Walk als Aktions-Selektions-Mechanismus nicht nur eine interne Repräsentation der Umgebung bilden, er konnte dies im Falle $p^{oe} = 0.1$, $p^{pe} = 0$ sogar in polynomieller Zeit, was für den Random und Linear Walk nicht möglich war.
Durch die Verringerung der Gedächtnislänge der *self-reward map* auf $L = 2$ konnte nicht nur die Lernzeit stark gesenkt werden, wodurch der Active Walk für den Fall $p^{oe} = 0.07$ und $p^{pe} = 0.02$ bessere Resultate als der Linear Walk erzielen konnte, sondern auch klar gemacht werden, dass die *self-reward*

map, auch nicht unbeabsichtigt, versucht, globale Größen zu approximieren[17]. Somit beschreibt der Active Walk in erster Linie einen stochastischen Prozess, dessen Auswirkungen in Form von ausgeführte Handlungen des Agenten, gut mit dessen Umgebung zusammenpassen, und für einen äußeren Beobachter, der die *first-passage* Zeit $< T_{FPT} >$ misst, Sinn ergeben.

Der zuletzt diskutierte Punkt ist ähnlich zu der Arbeit von [Wakeling und Bak, 2001], in der vorgeschlagen wird, dass Intelligenz nur im Kontext einer Umgebung (*embodiment*) existiert. In diesem Sinn kann die in Kapitel 6.4. definierte Wahrnehmungs-Handlungs-Schleife eines autonomen Agenten als notwendige Voraussetzung angesehen werden, um das Phänomen Intelligenz zu untersuchen. Darüberhinaus wird die Frage nach der Quantität der Intelligenz dadurch beantwortet, dass vorgeschlagen wird zu untersuchen, wie gut das System darin ist, Probleme in seiner Umgebung zu lösen. Bezogen auf dieses Kapitel besteht die Analogie darin, dass eine interne Repräsentation eines eindimensionalen Gitters gebildet wird, und darauf basierend eine Vorhersage des nächsten zu erwartenden Beobachtungssymbols berechnet wird.
Um keine Missverständnisse aufkommen zu lassen, soll ausdrücklich angemerkt werden, dass hier nicht behauptet wird, der in Kapitel 6.4. definierte Agent zeige intelligentes Verhalten in seiner Umgebung. Es soll nur darauf hingewiesen werden, dass alle bekannten, notwendigen Bedingungen erfüllt sind, um intelligente Systeme in ihrer einfachsten Form zu untersuchen.
Aus Gründen der Vollständigkeit sei noch angemerkt[18], dass die Ideen in [Wakeling und Bak, 2001] natürlich nicht originär sind, sondern auf [Varela et al., 1991] und [Brooks, 1991] zurück gehen[19] und insbesondere in der Philosophie eine lange Tradition haben.

Bei der Ableitung des approximierten individuellen Gesamtfehlers $\tilde{E}_{iabs}(t)$, wurde von einer Relation 6.34 für die bedingten Wahrscheinlichkeiten $P^S(s|i)$ Gebrauch gemacht, deren Begründung im folgenden genauer ausgeführt werden soll, da sich hieraus interessante Konsequenzen ergeben.
Bei der Definition der Umgebung und des Beobachtungsfehlers in Kapitel 6.4.1. wurde vorausgesetzt, dass jede Zelle des Gitters genau ein Symbol $\hat{s}$ enthält und als $P(s) = \delta_{\hat{s},s}$ geschrieben werden kann, welches fehlerbehaftet mit der Wahrscheinlichkeit $P(s|\hat{s})$ beobachtet wird. Die Annahme, dass jede Zelle nur genau ein Symbol enthält spiegelt die Tatsache wieder, dass ein Apfel immer ein Apfel und nie eine Orange ist, auch wenn man, z.B. infolge

[17] Da die zur Verfügung stehende Zeitspanne L, über die gemittelt werden kann, keine statistische Mittelwertbildung mehr erlaubt.

[18] Dies ist deshalb notwendig, da die Referenzenliste in [Wakeling und Bak, 2001] unvollständig ist.

[19] Siehe auch Referenzen darin.

einer unzureichenden Beleuchtung, beide verwechselt. Hieraus folgt aber, dass für $0 \leq p^{oe} < 0.5$, was ebenfalls eine plausible Annahme darstellt, immer

$$\hat{s} = \underset{s}{\operatorname{argmax}}\, P(s|\hat{s}) \tag{6.36}$$

gilt, woraus $P(s) = \delta_{\hat{s},s}$ für jede Gitterzelle folgt, was wieder die Ausgangsverteilung der Symbole auf dem Gitter darstellt. Die Frage lautet nun, welche Wahrscheinlichkeitsverteilung der beobachtbaren Symbole der Agent vorhersagen soll: $P(s) = \delta_{\hat{s},s}$, die Verteilung der tatsächlichen Symbole, oder $P(s|\hat{s})$, die Verteilung der fehlerbehafteten Symbole?
Die Frage wurde absichtlich irreführend gestellt, denn der Agent muss natürlich keine Wahrscheinlichkeitsverteilung vorhersagen, sondern lediglich das Symbol, welches als nächstes beobachtet wird, da jede Zelle des Gitters genau ein Symbol enthält. Die Antwort lautet also

$$\hat{s} \stackrel{!}{=} \underset{s}{\operatorname{argmax}}\, P^2(s) = \underset{s}{\operatorname{argmax}} \sum_i P^S(s|i) P^1(i) \tag{6.37}$$

Die Verteilung P^2 muss also lediglich so beschaffen sein, dass

$$P^2(\hat{s}) > P^2(s \neq \hat{s}) \tag{6.38}$$

erfüllt ist. Dies stellt eine Analogie zu 6.36 dar.
Nun kann man als Spezialfall $P^1(i) = \delta_{i,k}$ annehmen, wobei k fest gewählt wird, und in 6.37 einsetzen. Dann folgt hieraus

$$\hat{s} \stackrel{!}{=} \underset{s}{\operatorname{argmax}}\, P^S(s|k) \tag{6.39}$$

Hieraus folgt aber Relation 6.34. In Worten ausgedrückt heißt das, es existiert gar keine eindeutig bestimmte bedingte Wahrscheinlichkeit $P^S(s|i)$, die gelernt werden muss, um eine richtige Vorhersage zu machen.
Dies ist ein sehr schönes Resultat, da es aus praktischer Sicht enorm zeitaufwendig sein kann, Wahrscheinlichkeitsverteilungen möglichst genau zu schätzen. Deshalb halte ich es auch für unplausibel, dass biologische Systeme versuchen, Wahrscheinlichkeitsverteilungen der äußeren Umgebung zu schätzen. Dieser Ansatz zeigt, dass es ausreicht, Relationen zwischen Komponenten einer Wahrscheinlichkeitsverteilung zu lernen.

6.6.1 Biologische Plausibilität

Wie in [Herrmann et al., 1999b] angegeben, kann die interne Struktur des Agenten als Hippocampusmodell interpretiert werden. Hierbei entsprechen die inneren Zustände P^1 einer neuronalen Aktivität in CA3 und die Verteilung P^2 einer Aktivität in CA1. Denn wie in Kapitel 6.2. beschrieben, gibt es

in CA3 viele rekurrente Verknüpfungen, was sich durch den Markov Prozess 6.15 interpretieren lassen, da 6.15 iteriert angewendet wird. Von CA3 gibt es Verbindungen nach CA1, welche sich als 6.17 interpretieren lassen.
Ergänzend zu [Herrmann et al., 1999b] soll noch angeführt werden, dass die Komponenten der Verteilung P^1 nicht der Aktivität einzelner Neuronen entsprechen, sondern die Aktivität der Neuronen, welche auf die jeweiligen Orte sensitiv sind. Konkret stelle man sich das so vor, dass sich das Tier in seiner Umgebung bewegt. Zu jedem Zeitpunkt t feuern an jedem Ort $\hat{x}$ bestimmte Zellen, nämlich die *place cells*. Teilt man nun die Anzahl der *place cells* durch die Anzahl aller Zellen, die an irgendeinem Ort feuern, so ergibt sich die Komponente $P^1(\hat{x})$ der Verteilung P^1 an den jeweiligen Orten $\hat{x}$. Durch diese Interpretation erhält man erstens immer eine Wahrscheinlichkeitsverteilung und zweitens berücksichtigt man, dass *place cells* benachbarter, räumlicher Orte im neuronalen Netz weit voneinander entfernt liegen können.
Bei der Definition der Umgebung des Agenten wurde davon ausgegangen, dass jede Zelle ein skalares Symbol enthält. Dies spiegelt die Tatsache wider, dass *place cells* nicht von einen speziellen Reiz abhängen, sondern auf die Gesamtheit aller, an einem bestimmten Ort vorhandenen Reize, reagieren [O'Keefe und Dostrovsky, 1971; O'Keefe, 1976]. Dies und die Tatsache, dass der Hippocampus bereits vorverarbeitete Sensorinformation bekommt, rechtfertigen die Simplifikation der Sensorik auf einen skalaren Wert.

Kapitel 7

Zusammenfassung

In der vorliegenden Dissertation "Aktive Computation in offenen Systemen" wurde die Wahrnehmungs-Handlungs-Schleife (*action-perception-cycle*), die bei allen biologischen neuronalen Systemen zu finden ist, untersucht. Dies wurde dadurch realisiert, dass die Lerndynamik eines neuronalen Systems im Bezug auf ein zu lernendes Problem in Abhängigkeit von Parametern des Gesamtsystems analysiert wurde. Hierbei wurden die mathematischen Modelle, welche die Wahrnehmungs-Handlungs-Schleife beschrieben, so einfach wie möglich gewählt, um gerade noch die interessierenden signifikanten Eigenschaften aufzulösen.
Die durchgeführten Untersuchungen lassen sich in zwei aufeinander aufbauende Teile zerlegen. Im ersten Teil dieser Untersuchungen, der die Kapitel 3. bis einschließlich 5. umfasst, wird die Wahrnehmungs-Handlungs-Schleife von neuronalen Netzwerken untersucht, die in eine abstrakte Umgebung eingebettet sind. Im zweiten Teil, der Kapitel 6. beinhaltet, wird diese metaphorische Interpretation der Umgebung dadurch ersetzt, dass das neuronale System in einer räumlichen Umgebung untersucht wird, in der man das neuronale System als Agent bezeichnet, um die Analogie zu biologischen Organismen, die in einer Umgebung leben, deutlich zum Ausdruck zu bringen. Allgemeiner ausgedrückt befasst sich der erste Teil mit einer mesoskopischen und der zweite Teil mit einer makroskopischen Beschreibung der Wahrnehmungs-Handlungs-Schleife und deren Abhängigkeit von bestimmten Systemparametern auf und zwischen den jeweiligen Beschreibungsebenen.

Der ersten Teil der Untersuchungen beginnt mit Kapitel 3., in dem eine neue Hebb-artige Lernregel für neuronale Netze vorgestellt wird. Diese neue Hebb-artige Lernregel basiert auf der Einführung zusätzlicher Freiheitsgrade, in Form von Neuronenzählern, und somit auf einer Idee aus der Arbeit von [Klemm et al., 2000], in der eine Hebb-artige Lernregel vorgestellt wurde, die zusätzliche Freiheitsgrade in Form von Synapsenzählern besitzt. Bei dem

Versuch die Synapsen- durch die Neuronenzähler zu approximieren, um einerseits die Anzahl der vorhandenen Freiheitsgrade zu reduzieren und andererseits die Grundidee einer Mittelung über einen Teil der Inputmuster, bevor eine synaptische Änderung erfolgt, nicht zu verlieren, musste diese Approximation stochastisch erfolgen. Als stochastisches Optimierungsverfahren wurde die Methode der Extremalen Optimierung (EO) [Boettcher und Percus, 2000, 2001] verwendet. In Kombination mit der Methode des negativen Reinforcements [Chialvo und Bak, 1999; Bak und Chialvo, 2001], die nur gestattet, Synapsen, welche an einer falschen Berechnung des Netzwerks beteiligt waren, zu erniedrigen, ergibt sich hierdurch eine Hebb-artige Lernregel, die ein selektives stochastisches Updaten der synaptischen Gewichte bewirkt.

Nach der Einführung der Hebb-artigen Lernregel mit Neuronengedächtnis werden deren Eigenschaften untersucht. Dies geschieht anhand der Lerndynamik eines dreischichtigen neuronalen Netzes mit dem das XOR-Problem gelernt werden soll. Hierbei wird die Netzwerkdynamik des neuronalen Netzes als Parameter betrachtet und drei unterschiedliche Arten von Netzwerkdynamiken, nämlich winner-take-all, softmax und verrauschten winner-take-all Mechanismus, untersucht. Darüberhinaus wird der Einfluss der Gedächtnislänge Θ des Neuronengedächtnisses, des Exponenten τ der Rangordnungsverteilung und des Exponenten α der (Münz-)Verteilung auf die Lerndynamik des zu lernenden XOR-Problems untersucht.
Als Ergebnis erhält man nicht nur vergleichbar gute Konvergenzeigenschaften wie für die Hebb-artigen Lernregel mit Synapsengedächtnis [Klemm et al., 2000], sondern auch eine größere Flexibilität in den Parametern Θ, τ und α, die zu ähnlichen Konvergenzeigenschaften des neuronalen Netzes führen. Dies liegt an dem inhärent stochastischen Charakter der Lernregel, die ein kooperatives Zusammenspiel der drei Parameter Θ, τ und α bewirkt.

Ein weiteres Ergebnis besteht darin, dass die Hebb-artige Lernregel mit Neuronengedächtnis qualitativ als heterosynaptische Plastizität interpretiert werden kann [Lynch et al., 1977; Fitzsimonds et al., 1997]. Bei dieser Form von Plastizität sind nicht nur die Synapsen am Lernvorgang beteiligt, die direkt an der Signalübertragung beteiligt sind, sondern auch andere Synapsen von prä- und postsynaptischem Neuron. Darüberhinaus sind sogar Synapsen des präsynaptischen Neurons in dessen Dendritenbaum davon betroffen. Durch die Anwendung der Lernregel mit Neuronengedächtnis ergibt sich jedoch auch eine Auswirkung auf Synapsen des postsynaptischen Neurons, das über sein Axon mit weiteren Neuronen verbunden ist. Solch eine Form von vorwärts propagierter *long-term depression* (LTD) wurde jedoch experimentell bisher nicht gefunden [Fitzsimonds et al., 1997; Tao et al., 2000]. Somit können die von mir durchgeführten Untersuchungen bzgl. einer vorwärts propagierter *long-term depression* (LTD) als Hypothese hierfür angesehen werden, da

sich die experimentellen Arbeiten auf diesem Gebiet noch im Anfangsstadium befinden.

Zum Abschluss des Kapitels wird der Einfluss der Rangordnungsverteilung und der (Münz-)Verteilung der Lernregel mit Neuronengedächtnis untersucht, indem für die Rangordnungsverteilung eine exponentielle Verteilung und für die (Münz-)Verteilung eine Gleichverteilung angenommen wird. Die durchgeführten Simulationen zeigen eindeutig, dass sich diese Kombination von Verteilungen negativ auf die Grundidee von [Klemm et al., 2000] auswirkt, denn die Konvergenzeigenschaften des neuronalen Netzes werden für eine zunehmende Gedächtnislänge Θ deutlich schlechter.

In Kapitel 4. wird analytisch gezeigt, dass softmax und verrauschter winner-take-all Mechanismus nicht identisch sind und deren Differenz im allgemeinen Fall eine Funktion der inneren Felder h_i der Neuronen, der Temperatur β und des Rauschens η ist. Weiter wurde eine Methode vorgeschlagen, mit der man einem verrauschten winner-take-all Mechanismus eine approximierte Temperatur zuordnen kann. Diese approximierte Temperatur ist eine Funktion der inneren Felder h_i der Neuronen und des Rauschens η des verrauschten winner-take-all Mechanismus.
Aufbauend auf diese vorgeschlagene Methode, wird jeder Schicht eines dreischichtigen neuronalen Netzes während des Lernens des XOR-Problems eine approximierte Temperatur zugeordnet. Als Lerndynamik wird die Lernregel mit Synapsen- bzw. Neuronengedächtnis verwendet.
Das interessante Resultat besteht darin, dass man für beide Lernregeln in jeder Schicht des neuronalen Netzes einen Abkühlvorgang (*annealing*) in der approximierten Temperatur beobachten kann. Dieses Abkühlen korrespondiert mit den Konvergenzeigenschaften des mittleren Fehlers und tritt selbstorganisiert ohne äußeren Einfluss infolge der Lernregel auf. Dieses Verhalten unterscheidet sich insbesondere von *simulated annealing*[Kirkpatrick et al., 1983] einem stochastischen Optimierungsverfahren, bei dem explizit von außen der Verlauf der Temperatur vorgegeben werden muss, um ein bestimmtes Problem zu lösen.
Weiterhin ist aufgrund des Abkühlens während des Lernvorgangs insbesondere die approximierte Temperatur zeitlich nicht konstant. Daraus folgt, dass bei Verwendung eines verrauschten winner-take-all Mechanismus für die Netzwerkdynamik die einzelnen Schichten eines vorwärtsgerichteten Netzes sich thermisch nicht im Gleichgewicht befinden. Dies steht im Gegensatz zum softmax Mechanismus, der definitionsgemäß während des gesamten Lernvorgangs die konstante Temperatur β^{-1} vorgegeben bekommt, da als Größe der synaptischen Änderung $\delta = 1 = \text{const.}$ verwendet wird.

Als mögliche Anwendung im Bereich der Neurobiologie und kognitiven

Psychologie könnte man die in Kapitel 4. vorgestellte Methode der approximativen Zuordnung einer Temperatur dazu verwenden, die physiologische mit der behavioristischen Ebene zu koppeln. Dies könnte dadurch erreicht werden, dass man die Korrespondenz zwischen (beobachtetem) mittleren Fehler und Abkühlverhalten, in jeder Schicht eines lateral inhibierten neuronalen Netzes, ausnutzt. Denn in meinem abstrakten, simplifizierten Modell kodiert der zeitliche Verlauf der approximierten Temperatur den zeitlichen Verlauf der Güte des Verhaltens. Anders ausgedrückt: die Richtung der Temperaturänderung gibt die Richtung des Lernvorgangs an. Somit ließe sich physiologischen Messungen an Neuronen, die sich lateral inhibieren (winner-take-all Mechanismus), eine approximierte Temperatur zuordnen, deren zeitlicher Verlauf dann mit dem zeitlichen Verlauf der Güte des Verhaltens verglichen werden könnte.
Die aus diesem Kapitel resultierende Hypothese lautet, dass die Richtung der Temperaturänderung eines lateral inhibierten neuronalen Netzes, das signifikant an der Lösung eines zu lernenden Problems beteiligt ist, und die Richtung der Güteänderung des Verhaltens übereinstimmen.

Kapitel 5. stellt das Ende des ersten und zugleich die Schnittstelle zum zweiten Teil dar. In diesem Kapitel wird der generische Charakter der Lernregel mit Neuronengedächtnis dadurch unterstrichen, dass als zugrundeliegende Netztopologie eine Klasse von Netzwerken gewählt wird, die als *small world* [Watts und Strogatz, 1998] bezeichnet wird. Innerhalb dieser Netzwerktopologie wird untersucht, wie das Problem des Timings, also die sinnvolle Verknüpfung von Input- zu Outputneuronen nach exakt einer vorgegebenen Zeit, gelernt werden kann. Hierfür wird die Lernregel mit Neuronengedächtnis mit der von [Chialvo und Bak, 1999; Bak und Chialvo, 2001] verwendeten Lernregel in Abhängigkeit verschiedener Parameter untersucht und verglichen.

Als Ergebnis erhält man, dass die Lernregel mit Neuronengedächtnis für die gesamte Netzwerkklasse der *small world* Netze bis hinunter zu niedrigen Werten des Umverknüpfungsparameters p_{rw} ein sehr gutes Konvergenzverhalten zeigt. Somit können also insbesondere auch *small world* Netzwerke mit dieser Lernregel trainiert werden, die aus biologischer Sicht von besonderem Interesse sind [Watts und Strogatz, 1998; Latora und Marchiori, 2001, 2002].
Verglichen mit der von [Chialvo und Bak, 1999; Bak und Chialvo, 2001] vorgeschlagenen Lernregel sind die erhaltenen Resultate immer wesentlich besser. Dies bezieht sich nicht nur auf die erhaltenen Simulationszeiten, sondern auch darauf, dass die neuronalen Netze, welche mit der Lernregel von [Chialvo und Bak, 1999; Bak und Chialvo, 2001] trainiert wurden, für kleine Werte des Umverknüpfungsparameters p_{rw} gar nicht mehr innerhalb der vorgegebenen Simulationszeit das Problem lernen konnten.

Ein weiterer Unterschied zwischen diesen beiden Lernregeln kann durch die Einführung eines zusätzlichen Parameters in Form eines Aktions-Selektions-Mechanismus herausgearbeitet werden. Hierbei kann für die Lernregel von [Chialvo und Bak, 1999; Bak und Chialvo, 2001] kein Einfluss der unterschiedlichen Aktions-Selektions-Mechanismen festgestellt werden, wohingegen die Lernregel mit Neuronengedächtnis deutlich davon beeinflusst wird. Somit wird gezeigt, dass unter bestimmten Voraussetzungen das Verhalten auf makroskopischer Ebene die neuronale Dynamik auf mesoskopischer Ebene beeinflussen kann. Generell wird durch die Einführung eines Aktions-Selektions-Mechanismus ein Zusammenhang zur Wahrnehmungs-Handlungs-Schleife, in einer sehr einfachen Form, hergestellt.

Darüberhinaus gibt es nicht nur eine Beeinflussung durch die beiden unterschiedlichen Aktions-Selektions-Mechanismen auf die Lernregel mit Neuronengedächtnis es konnte auch gezeigt werden, dass der triviale Aktions-Selektions-Mechanismus I. schlechtere Konvergenzeigenschaften bewirkt als Aktions-Selektions-Mechanismus II., der sich als rudimentäre Form von hierarchischem Lernen interpretieren lässt.

Zusammenfassend kann man festhalten, dass die Untersuchungen in diesem Kapitel die generischen Eigenschaften der Lernregel mit Neuronengedächtnis unterstreichen und darüberhinaus die Notwendigkeit eines Gedächtnisses für eine biologisch plausible Lernregel demonstrieren, da nur infolge eines Gedächtnisses verschiedene Aktions-Selektions-Mechanismen unterschieden und somit vorteilhaft ausgenutzt werden können.

Im zweiten Teil der Dissertation, der aus Kapitel 6. besteht, wird das abstrakte durch ein neuronales Netze zu lernende Problem des ersten Teils, das metaphorisch als Umgebung bezeichnet wurde, durch eine räumliche Umgebung ersetzt. Hierdurch ist es gerechtfertigt das neuronale Netz, das mit einem Problem wechselwirkt, als Agent in einer Umgebung aufzufassen.

Die erhaltenen Ergebnisse der durchgeführten Simulationen zeigen, dass das von [Herrmann et al., 1999b] vorgestellte und hier verwendete mathematische Modell der internen Struktur eines autonomen Agenten in der Lage ist eine Repräsentation eines äußeren eindimensionalen Gitters zu lernen. Genauer gesagt liegt die *first-passage* Zeit für alle durchgeführten Simulationen immer unterhalb der maximalen Simulationszeit.

Ein weiteres Resultat wird aus einem Vergleich von drei unterschiedlichen Aktions-Selektions-Mechanismen gewonnen, wodurch sich deutliche Unterschiede in den resultierenden mittleren *first-passage* Zeiten $< T_{FPT} >$ für

den zugrundeliegenden stochastischen Prozess des approximierten individuellen Gesamtfehlers $\tilde{E}_{iabs}$ beim Erreichen der Schwelle $\tilde{E}_{iabs} = 0$ ergeben. Dies ist deshalb von besonderem Interesse, da alle Simulationen mit drei identischen Agenten in drei identischen Umgebungen durchgeführt werden, und die drei Agenten sich einzig in dem verwendeten Aktions-Selektions-Mechanismus unterscheiden, wobei jeder dieser drei unterschiedlichen Aktions-Selektions-Mechanismen eine vernünftige Wahl für einen Aktions-Selektions-Mechanismus darstellt, der eine interne Repräsentation der Umgebung lernen kann.
Somit konnte gezeigt werden, dass die Lerndynamik eines Gesamtsystems nicht nur durch die verwendete Lernregel bestimmt wird, sondern durch die Gesamtheit aller in 3.1 definierten Komponenten, also auch insbesondere durch den verwendeten Aktions-Selektions-Mechanismus. Hieraus folgt, dass es keine guten oder schlechten Lernregeln bzw. Aktions-Selektions-Mechanismen gibt, sondern nur schlecht oder gut zusammenpassende Komponenten des Gesamtsystems 3.1. Deshalb ist ein Verständnis von Lerndynamiken natürlicher Systeme nur durch Betrachtung der Wahrnehmungs-Handlungs-Schleife des Organismus zu erzielen.

Ein weiterer Punkt, der in Kapitel 6. untersucht wird, ist die Struktur der Aktions-Selektions-Mechanismen und insbesondere die Frage, wie sich die Entscheidung der Aktionsauswahl in einfacher Weise an die interne Struktur des Agenten ankoppeln lässt. Um einerseits die guten Eigenschaften der Aktions-Selektions-Mechanismen des Random und Linear Walk zu übernehmen und andererseits eine Ankopplung an die interne Struktur des Agenten zu erzielen, wird ein Active Walk vorgeschlagen, der auf einem Wechselspiel zwischen Drift und Diffusion in Form einer zu adaptierenden *self-reward map* und deren Reinitialisierung beruht.
Simulationen zeigen, dass der Agent mit dem Active Walk als Aktions-Selektions-Mechanismus, nicht nur eine interne Repräsentation der Umgebung bilden kann, sondern dies im Fall von Beobachtungs- und Positionsfehler $p^{oe} = 0.1$, $p^{pe} = 0$ sogar in polynomieller Zeit schafft, was für den Random und Linear Walk nicht möglich ist. Durch eine Verringerung der Gedächtnislänge der *self-reward map* auf $L = 2$, kann nicht nur die Lernzeit stark gesenkt werden, wodurch der Active Walk für den Fall $p^{oe} = 0.07$ und $p^{pe} = 0.02$ bessere Resultate als der Linear Walk erzielt, sondern auch klar gemacht werden, dass die *self-reward map*, auch nicht unbeabsichtigt, versucht globale Größen statistisch zu approximieren.
Der Active Walk beschreibt also in erster Linie einen stochastischen Prozess, dessen Auswirkungen in Form von ausgeführte Handlungen des Agenten gut mit dessen Umgebung zusammenpassen und für einen äußeren Beobachter, der die *first-passage* Zeit $< T_{FPT} >$ misst, Sinn ergeben.

Neben all diesen konkreten Ergebnissen, die sich aus Simulationen ergeben, wird in Kapitel 5. eine Methode vorgeschlagen, durch welche die Wahrnehmungs-Handlungs-Schleife analysierbar gemacht wird, und die in Kapitel 5. und Kapitel 6. Verwendung findet. Diese Methode basiert auf der Ermittlung der *first-passage* Zeit durch einen äußeren Beobachter und erlaubt somit ein strikte Trennung der Information, die dem neuronalen Netz bzw. dem Agent und dem äußeren Beobachter zur Verfügung steht. Die Berechnung des hierfür benötigten individuellen Gesamtfehlers E_{iabs} kann zwar für ein konkret zu untersuchendes System numerisch sehr aufwendig werden, jedoch lässt sich unter Einbeziehung von zusätzlicher Information, wie in Kapitel 6. gezeigt, ein approximierter individueller Gesamtfehler angeben, der statt dessen verwendet werden kann. Dieser approximierte individuelle Gesamtfehler ist für jedes System unter Berücksichtigung dessen Arbeitsweise speziell zu ermitteln.

Zusammenfassend kann man festhalten, dass man die Wahrnehmungs-Handlungs-Schleife von biologischen Systemen dadurch untersuchen kann, dass man die Lerndynamik des Systems untersucht, die sich beim Lernen eines konkreten Problems ergibt. Hierbei ist zum einen darauf zu achten, dass man das Gesamtsystem betrachtet und zum anderen auf eine strikte Trennung der Information von System und den das System bewertenden äußeren Beobachter.

Aufbauend auf die in dieser Dissertation gewonnenen Resultate lassen sich die Komponenten des Gesamtsystems 3.1 sukzessive verändern, um weiterführende Ergebnisse zu bekommen. So könnte man beispielsweise andere Neuronenmodelle, wie z.B. ein *integrate-and-fire* Neuron, oder andere Aktions-Selektions-Mechanismen untersuchen, deren zugrundeliegenden stochastischen Prozesse ineinander verschränkt sind und somit die Reinitialisierung der *self-reward map* kontinuierlich verläuft.

Wie auch immer weiterführende Arbeiten aussehen, so hoffe ich doch, dass durch diese Dissertation ein breites Bewusstsein für das Problem der Wahrnehmungs-Handlungs-Schleife geschaffen wird. Insbesondere sollte deutlich geworden sein, dass es Ansätze gibt, die Wahrnehmungs-Handlungs-Schleife theoretisch zu beschreiben und zu untersuchen. Denn wie Per Bak [Wakeling und Bak, 2001] in einem sehr ähnlichen Zusammenhang gesagt hat

> *... we have begun to demystify the subject of intelligence and maneuver it into a position where we can ask precise and well-defined questions*

Anhang A

Mathematischer Anhang

A.1 Pseudoinverse

In Kapitel 4. wird das Inverse der Matrizen 3.18 und 3.19 benötigt. Da diese Matrizen nicht quadratisch sind, kann ihr Inverses nicht auf die übliche Art berechnet werden. Statt dessen wird ein auf [Penrose, 1955] und [Moore, 1929, 1935] basierendes Verfahren verwendet, welches einer Verallgemeinerung der Inversen einer Matrix entspricht, welche auch auf nicht-quadratische Matrizen angewendet werden kann. Diese verallgemeinerte Matrixinversion wird als Pseudo- oder Moore-Penrose-Inverse bezeichnet.

A.1.1 Definition der Pseudoinversen

Die verallgemeinerte Inverse M^* einer Matrix wird für jede beliebige Matrix M durch

$$MM^*M = M \tag{A.1}$$
$$M^*MM^* = M^* \tag{A.2}$$
$$(MM^*)^T = MM^* \tag{A.3}$$
$$(M^*M)^T = M^*M \tag{A.4}$$

eindeutig definiert [Penrose, 1955].

A.1.2 Berechnung der Pseudoinversen

Die Pseudoinverse einer Matrix M lässt sich über deren Singulärwertzerlegung (SVD) bestimmen [Penrose, 1955]. Hierbei versteht man unter Singulärwertzerlegung eine Produktdarstellung der $n \times m$ Matrix M, bestehend aus drei spezielle Matrizen.

$$M = UWV^T \tag{A.5}$$

Die drei Matrizen haben folgende Eigenschaften:

1. U ist ein $m \times n$ Matrix mit orthogonalen Spalten
2. W ist eine $n \times n$ Diagonalmatrix

$$W = \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} \tag{A.6}$$

   wobei die Ordnung von D durch den Rang der Matrix M bestimmt ist.

3. V ist eine $n \times n$ Orthogonalmatrix.

Hieraus erhält man die Pseudoinverse durch

$$M^* = VW^{-1}U^T \tag{A.7}$$

Speziell für die Matrizen 3.18 und 3.19 erhält man

$$MPP^{l_1 l_2} = \begin{pmatrix} \frac{5}{18} & -\frac{1}{18} & -\frac{1}{18} & \frac{5}{18} & -\frac{1}{18} & -\frac{1}{18} \\ \frac{5}{18} & -\frac{1}{18} & -\frac{1}{18} & -\frac{1}{18} & \frac{5}{18} & -\frac{1}{18} \\ \frac{5}{18} & -\frac{1}{18} & -\frac{1}{18} & -\frac{1}{18} & -\frac{1}{18} & \frac{5}{18} \\ -\frac{1}{18} & \frac{5}{18} & -\frac{1}{18} & \frac{5}{18} & -\frac{1}{18} & -\frac{1}{18} \\ -\frac{1}{18} & \frac{5}{18} & -\frac{1}{18} & -\frac{1}{18} & \frac{5}{18} & -\frac{1}{18} \\ -\frac{1}{18} & \frac{5}{18} & -\frac{1}{18} & -\frac{1}{18} & -\frac{1}{18} & \frac{5}{18} \\ -\frac{1}{18} & -\frac{1}{18} & \frac{5}{18} & \frac{5}{18} & -\frac{1}{18} & -\frac{1}{18} \\ -\frac{1}{18} & -\frac{1}{18} & \frac{5}{18} & -\frac{1}{18} & \frac{5}{18} & -\frac{1}{18} \\ -\frac{1}{18} & -\frac{1}{18} & \frac{5}{18} & -\frac{1}{18} & -\frac{1}{18} & \frac{5}{18} \end{pmatrix} \tag{A.8}$$

$$MPP^{l_2 l_3} = \begin{pmatrix} \frac{2}{5} & -\frac{1}{10} & -\frac{1}{10} & \frac{4}{15} & -\frac{1}{15} \\ \frac{2}{5} & -\frac{1}{10} & -\frac{1}{10} & -\frac{1}{15} & \frac{4}{15} \\ -\frac{1}{10} & \frac{2}{5} & -\frac{1}{10} & \frac{4}{15} & -\frac{1}{15} \\ -\frac{1}{10} & \frac{2}{5} & -\frac{1}{10} & -\frac{1}{15} & \frac{4}{15} \\ -\frac{1}{10} & -\frac{1}{10} & \frac{2}{5} & \frac{4}{15} & -\frac{1}{15} \\ -\frac{1}{10} & -\frac{1}{10} & \frac{2}{5} & -\frac{1}{15} & \frac{4}{15} \end{pmatrix} \tag{A.9}$$

A.2 Dijkstra Algorithmus

Der Algorithmus von Dijkstra [Dijkstra, 1959] berechnet den kürzesten Abstand L von einem Knoten u_0 in einem gewichteten Graphen G der Ordnung $p = N$ zu jedem anderen Knoten u_i in G.

Algorithmus

1. Setze $i = 0$, $S = \{u_0\}$, $\bar{S} = V(G) - \{u_0\}$, $L(u_0) = 0$,
   $L(v) = \infty \; \forall \, v \in \bar{S}$.
   Falls $p = 1$ breche ab; sonst gehe zu 2.

2. Für jedes $v \in \bar{S}$ mit der Eigenschaft, dass es von u_i zu v eine Verbindung gibt prüfe, ob folgendes gilt:
   Falls $L(v) \leq L(u_i) + w_{u_i,v}$ fahre fort; sonst $L(v) = L(u_i) + w_{u_i,v}$ und $Par(v) = u_i$.

3. Bestimme $m = min\{L(v)|v \in \bar{S}\}$. Falls der Knoten $v_j \in \bar{S}$ mit der Eigenschaft $L(v_j) = m$ ausgewählt wurde, setze $L(v_j) = m$ und $u_{i+1} = v_j$.

4. Setze $S = S \cup \{u_{i+1}\}$ und $\bar{S} = \bar{S} - \{u_{i+1}\}$.

5. Setze $i = i + 1$. Falls $i = p - 1$ breche ab; sonst gehe zu 2.

Zur Vorgehensweise seien einige kurze Bemerkungen gemacht.

zu Punkt 1.:

> Die Menge S enthält alle Knoten von G, deren Abstand von u_0 bereits bestimmt wurde. Entsprechend enthält die zu S komplementäre Menge $\bar{S} = V(G) - S$ alle Knoten, deren Abstand noch nicht bestimmt wurde, wobei $V(G)$ die Menge aller Knoten des Graphen G beinhaltet.
> Der Abstand von u_0 zu einem beliebigen anderen Knoten v des Graphen wird abkürzend mit $L(v) = L(u_0, v)$ bezeichnet.

zu Punkt 2.:

> Mit $Par(v) = u_i$ wird der Knoten bezeichnet, der zu Knoten v führt (Parent).

zu Punkt 3.:

> Dieser Schritt bestimmt den nächsten Knoten u_{i+1} von $\bar{S}$ für den der minimale Abstand von u_0 nach u_{i+1} gefunden wurde.

Eine ausführliche Erklärung findet sich in [Chartrand und Oellermann, 1993].

A.3 Anmerkungen zur Simulation

Die in dieser Arbeit durchgeführten numerischen Berechnungen bzw. Simulationen wurden in Fortran 90 und Mathematica durchgeführt. Besonders für die aufwendigen Simulationen, bei denen das Verhalten eines Ensembles untersucht wurde, war die Verwendung einer effizienten Programmiersprache, wie sie Fortran 90 darstellt, unabdingbar, um zu einigermassen erträglichen Rechenzeiten zu gelangen.
Die präsentierten Graphiken der numerischen bzw. simulierten Ergebnisse wurden mit Matlab oder Mathematica und Diagramme mit xfig erstellt.
Die verwendeten Computer waren normale personal computer (PC) im Bereich von 400-1000 MHz.

Literaturverzeichnis

L. Abbott und T. Kepler (1990). Model neurons: from hodgkin huxley to hopfield. In L. Garrido, Hg., *Statistical mechanics of neural networks*, Bd. 368 von *Lecture notes in physics*, S. 5–18. Springer Verlag.

T. Achacoso und W. Yamamoto (1992). *AY's Neuroanatomy of C. elegans for Computation*. CRS Press, Boca Raton, FL.

C. Adelle und P. Alstrom (2001). Expectation and conditioning. *Physica A* **290**:251–267.

R. Albert und A. Barabasi (2002). Statistical mechanics of complex networks. *Rev. of Modern Physics* **74**:47.

P. Alstrom und D. Stassinopoulos (1995). Versatility and adaptive performance. *Phys. Rev. E* **51**:5027–5032.

T. Aquinas, St. (1272). Treatise on man. In *The Summa Theologica*. http://www.ccel.org/a/aquinas/.

A. Arleo und W. Gerstner (2000). Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity. *Biological Cybernetics* **83**:287–299.

P. Bak (1998). Physical theory in biology: Foundations and explorations. Book review in Nature 391, Book editors: C.J. Lumsden, W.A. Brandts and L.E.H. Trainor.

P. Bak und D. Chialvo (2001). Adaptive learning by extremal dynamics and negative feedback. *Phys. Rev. E* **63**.

P. Bak und K. Sneppen (1993). Punctuated equilibrium and criticality in a simple model of evolution. *Phys. Rev. Lett.* **71**:4083–4086.

P. Bak, T. Tang und K. Wiesenfeld (1987). Self-organized criticality: An explanation of the 1/f noise. *Phys. Rev. Lett.* **59**:381–384.

A. Barto, R. Sutton und C. Anderson (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transaction on Systems, Man and Cybernetic* **13**:834–846.

W. Becker, L. Kleinsmith und J. Hardin (2000). *The world of the cell*. Addison Wesley Longman.

G.-g. Bi und M.-m. Poo (1998). Synaptic modifications in cultered hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *J. Neuroscience* **18**:10464–10472.

G.-g. Bi und M.-m. Poo (2001). Synaptic modification by correlated activity: Hebb's postulate revisited. *Annual Review of Neuroscience* **24**:139–166.

T. Bliss und T. Lomo (1973). Long-lasting potentiation of synaptic transission in the dentate area of the anaesthetized rabbit following stimulation ot the perforant path. *J. Physiol.* **232**:331–356.

K. Blum und L. Abbott (1996). A model of spatial map formation in the hippocampus of the rat. *Neural Computation* **8**:85–93.

S. Boettcher (2000). Extremal optimization: Heuristics via coevolutionary avalances. *Coumputing in Science and Engeneeriung* **2**:75–82.

S. Boettcher (2002). Talk at the conference for complex adaptive systems 2002 in delmenhorst: Extremal optimization. sub. Complexity.

S. Boettcher und A. Percus (2000). Nature's way of optimizing. *Artificial Intelligence* **119**:275–286.

S. Boettcher und A. Percus (2001). Optimization with extremal dynamics. *Phys. Rev. Lett.* **86**:5211–5214.

V. Braitenberg (1986). *Vehicles*. MIT Press.

R. Brooks (1991). Intelligence without reason. In M. Kaufmann, Hg., *Proceedings of the twelfth International Joint Conference on Artificial Intelligence*, S. 569–595.

R. A. Brooks (1999). *Cambrian Intelligence*. MIT Press.

T. Brown, A. Ganong, E. Kariss und C. Keenan (1990). Hebbian synapses: biophysical mechanisms and algorithms. *Annual Review of Neuroscience* **13**:475–511.

S. Cajal (1904). *La Textura del Systema Nervioso del Hombre y los Vertebrados*. Moya, Madrid.

G. Chartrand und O. Oellermann (1993). *Applied and Algorithmic Graph Theory*. International Series in Pure and Applied Mathematics. Mc Graw-Hill.

D. Chialvo und P. Bak (1999). Learning from mistakes. *Neuroscience* **90**:1137–1148.

P. Churchland und T. Sejnowski (1992). *The Computational Brain*. MIT Press.

N. Cohen und H. Eichenbaum (1993). *Memory, Amnesia, and the Hippocampal System*. MIT Press.

F. Crick (1989). The recent excitement about neural networks. *Nature* **337**:129–132.

P. Dayan und L. Abbott (2001). *Theoretical Neuroscience: Computational and mathematical modelling of neural systems*. MIT Press.

E. Dijkstra (1959). A note on two problems in connection with graphs. *Nmerische Math.* **1**:269–271.

H. Eichenbau (1996). Is the rodent hippocampus just for 'place'? *Current Opinion in Neuroscience* **6**:187–195.

H. Eichenbaum (1997). Dclarative memory: Insights from cognitive neurobiology. *Annu. Rev. Psychol.* **48**:547–572.

P. Erdös und A. Rényi (1959). On random graphs. *Publiationes Mathematicae* **6**:290–297.

R. FitzHugh (1961). Impulses and physiological states in theoretical models of nerve membrane. *Biophys. J.* **1**:445–466.

R. Fitzsimonds, H.-j. Song und M.-m. Poo (1997). Propagation of activity-dependent synaptic depression in simple neural networks. *Nature* **388**:439–448.

V. Franz, K. Gegenfurtner, H. Bülthoff und M. Fahle (2000). Grasping visual illusions: No evidence for a dissociation between perception and action. *Psychological Science* **11**:20–25.

W. Freeman (2000). Mesoscopic neurodynamics: From neuron to brain. *Journal of Physiology; Paris* **94**:303–322.

R. French (2000). The turing test: the first 50 years. *Trends in Cognitive Sciences* **4**:115–122.

U. Frey und R. Morris (1997). Synaptic tagging and long-term potentiation. *Nature* **385**:533–536.

W. Gerstner und L. Abbott (1997). Learning navigational maps through potentiation and modulation of hippocampal place cells. *Journal of Computational Neuroscience* **4**:79–94.

D. Hebb (1949). *The Organization of Behavior*. Wiley, New York.

R. Henkel (2001). private communications.

J. Herrmann, F. Emmert-Streib und K. Pawelzik (1999a). Autonomous robots and neuroethology: Emergence of behavior from a scense of curiosity. In A. Löffel, F. Mondada und U. Rückert, Hg., *Experiments with the Mini-Robot Khepera*, Proceedings of the 1st international Khepera Workshop, S. 89–98. HNI-Verlagsschriftenreihe.

J. Herrmann, K. Pawelzik und T. Geisel (1999b). Self-localization of autonomous robots by hidden representations. *Autonomous Robots* **7**:31–40.

J. Hertz, A. Krogh und R. Palmer (1991). *Introduction to the theory of neural compuation*. Addison-Wesley.

T. Heskes und W. Wiegerinck (1996). A theoretical comparison of batch-mode, on-line, cyclic, and almost cyclic learning. *IEEE Trans. Neural Networks* **7**:919–924.

A. Hodgkin und A. Huxley (1952). A quantitative description of the membrane current and its application to conduction and excitation in nerve. *J. Physiol.* **117**:500–544.

B. Hommel, J. Müsseler, G. Aschersleben und W. Prinz (2002). The theory of event coding (tec). a framework for perception and action. *Behavioral and Brain Sciences* .

J. Honerkamp (1990). *Stochastische Dynamische Systeme*. VCH Verlagsgesellschaft mbH.

J. Hopfield (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of science USA* **79**:2554–2558.

L. Kaebeling, M. Littman und A. Moore (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* .

E. Kandel, J. Schwartz und T. Jessell, Hg. (1991). *Principles of neural science*. Prentice Hall.

B. Katz (1969). *The Release of Neural Transmitter Substances*. Liverpool University Press.

S. Kirkpatrick, C. Gellatt und M. Vecchi (1983). Optimization by simulated annealing. *Science* **220**:671–680.

K. Klemm, S. Bornhold und H. Schuster (2000). Beyond hebb: Exclusive-or and biological learning. *Phys. Rev. Lett.* **84**:3013–3016.

C. Koch (1999). *Biophysics of Computation*. Oxford Press.

V. Latora und M. Marchiori (2001). Efficient behavior of small-world networks. *Phys. Rev. Lett.* **5**:198701.

V. Latora und M. Marchiori (2002). Economic small-world behavior in weighted networks. *sub. Phys. Rev. E* .

G. Lynch, T. Dunwiddie und V. Gribkoff (1977). Heterosynaptic depression: a postsynaptic correlate of long-term potentiation. *Nature* **266**:737–739.

H. Markram, J. Lübke, M. Frotscher und B. Sakmann (1997). Regulation of synaptic efficacy by coincidence of postsynaptic aps and espsps. *Science* **275**:213–215.

W. McCulloch und W. Pitts (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* **5**:115–133.

S. Milgram (1967). The small world problem. *Psychol. Today* **2**:60–67.

M. Minsky und S. Papert (1969). *Perceptrons*. MIT Press.

E. Moore (1929). On the reciprocal of the general algebraic matrix. *Bull. Amer. Soc.* S. 394–395.

E. Moore (1935). General analysis. *Memoirs Amer. Philos. Soc.* S. 147–209.

J. Nagumo, S. Arimoto und S. Yoshizawa (1962). An active pulse transmission line simulating nerve axon. *Proc. IRE.* **50**:2061–2071.

S. Nolfi und D. Floreano (1999). Learning and evolution. *Autonomous Robots* **7**.

J. O'Keefe (1976). Place units in the hippocampus of the freely moving rat. *Exp. Neurol.* **51**:78–109.

J. O'Keefe und J. Dostrovsky (1971). The hippocampus as a spatial map: preliminary evidence from unit activity in the freely moving rat. *Brain Res.* **34**:171–175.

J. O'Keefe und L. Nadel (1978). *The hippocampus as a cognitve map*. Oxford Press.

S. Oore, G. Hinton und G. Dudek (1997). A mobile robot that learns its place. *Neural Computation* S. 683–699.

G. Orr und T. Leen (1993). Weight space probability densities in stochasitc learning: Ii. transients and basin hopping times. *Advances in Neural Information Processing Systems* S. 507–514.

N. Otmakhova und J. Lisman (1998). D1/d5 dopamine receptors inhibit depotentiation at ca1 synapses via camp-dependent mechanism. *J. Neuroscience* **18**:1270–1279.

I. Pavlov (1927). *Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex*. Oxford University Press.

R. Penrose (1955). A generalized inverse for matrices. *Proc. Cambridge Phil. Soc.* **51**:406–413.

J. Randlov und P. Alstrom (2001). Perception control. *Physica A* **289**:561–573.

A. Rapoport (1953a). Spread of information through a population with sociostructural bias. i. assumption of transitivity. *Bulletin of Mathematical Biophysics* **15**:523–533.

A. Rapoport (1953b). Spread of information through a population with sociostructural bias. ii. various models with partial transitivity. *Bulletin of Mathematical Biophysics* **15**:535–546.

S. Redner (2001). *A Guide to First-Passage Processes*. Cambridge University Press.

H. Risken (1996). *The Fokker-Planck Equation*. Springer Series in Synergetics. Springer Verlag.

F. Rosenblatt (1962). *Principles of Neurodynamics*. New York: Spartan.

D. Rumelhart, G. Hinton und R. Williams (1986a). Learning representations by back-propagating errors. *Nature* **323**:533–536.

D. Rumelhart, G. Hinton und R. Williams (1986b). Lerning internal representations by error propagation. In D. Rumelhart und J. McCelland, Hg., *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, S. 318–362. MIT Press.

A. Samuel (1959). Some studies in machine learning using the game of checkers. *IBM J. Res. Develop.* **3**:210–229.

M. Scannel (1997). *Nature* **386**:452.

M. Scannel, M. Young und B. C. (1995). *J. Neuroscience* **15**:1463.

W. Schultz, P. Dayan und P. Montague (1997). A neural substrate of prediction and reward. *Science* **275**:1593–1599.

H. Schuster (2002). *Complex Adaptive Systems*. Scator Verlag.

B. Skinner (1938). *The Behavior of Organisms: An Experimental Analysis*. Appleton-Century-Crofts, New York.

L. Squire (1987). *Memory and Brain*. Oxford University Press.

R. Stein (1967). Some models for neural variability. *Biophys. J.* **7**.

C. Stevens und Y. Wang (1994). Changes in reliability of synaptic function as a mechanism for plasticity. *Nature* **371**:704–707.

D. Strassinopoulos und P. Bak (1995). Democratic reinforcement: A principle for brain function. *Phys. Rev. E* **51**:5033–5039.

R. Sutton (1988). Learning to predict by the method of temporal differences. *Machine Learning* **3**:9–44.

R. Sutton und A. Barto (1998). *Reinforcement Learning*. MIT Press.

H. Tao, L. Zhang, G.-q. Bi und M.-m. Poo (2000). Selective presynaptic propagation of long-term potentiation in defined neural networks. *J. Neuroscience* **20**:3233–3243.

J. Taube, R. Muller und J. Ranck (1990). Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis. *J. Neuroscience* **10**:420–435.

G. Tesauro (1994). Td-gammon, a self-teaching backgammon program, achieves masterlevel play. *Neural Computation* **6**:215–219.

E. Thorndike (1911). *Animal Intelligence*. Harner, Darien, CT.

S. Thrun (1998). *Learning to Learn*. Kluwer Academic Publishers.

O. Trullier und J. Meyer (2000). Animat navigation using a cognitive graph. *Biological Cybernetics* **83**:271–285.

A. Turing (1950). Computing machinery and intelligence. *Mind* **59**:433.

J. van Hemmen (1997). Hebbian learning, its correlation catastrophe, and unlearning. *Network: Comput. Neural Systems* **8**:V1–V17.

F. Varela, E. Thompson und E. Rosch (1991). *The Embodied Mind*. MIT Press.

J. von Neumann (1958). *The Computer and the Brain*. Yale University Press.

J. Wakeling und P. Bak (2001). Intelligent systems in the context of surrounding environment. *Phys. Rev. E* **64**.

D. Watts (1999). *Small Worlds: The Dynamics of Networks between Order and Randomness*. Princeton University Press.

D. Watts und S. Strogatz (1998). Collective dynamics of 'small-world' networks. *Nature* **393**:440–442.

J. Weng, J. McCelelland, A. Pentland, O. Sporns, I. Stockman, M. Sur und E. Thelen (2001). Autonomous mental development by robots and animals. *Science* **291**:599–600.

P. Werbos (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Dissertation, Harvard University.

M. Wilson und B. McNaughton (1993). Dynamics of the hippocampus ensembles code for space. *Science* **261**:1055–1058.

S. Wolfram (1984). Universality and complexity in cellular automata. *Physica D* .

E. Wood, P. Dudchenko und H. Eichenbaum (1999). *Nature* **397**:613–616.

A. Yuille und D. Geiger (1995). Winner-take-all mechanisms. In M. Arbib, Hg., *Handbook of Brain Theory and Neural Networks*, S. 1056–1060. MIT Press.

K. Zhang, I. Ginzburg, B. McNaughton und T. Sejnowski (1998). Interpreting neuronal population activity by reconstruction: Unified framework with application to hippocampal place cells. *Journal of Neurophysiology* **79**:1017–1044.

Danksagung

Das Entstehen dieser Dissertation wäre ohne den Einfluss bzw. dem Mitwirken bestimmter Personen nicht möglich gewesen, denen ich hierfür herzlich danken möchte:

Prof. Dr. Klaus Pawelzik danke ich für das spannende Thema der Arbeit, an dem ich mich austoben durfte, und die uneingeschränkte Freiheit die mir beim Erstellen dieser Dissertation eingeräumt wurde.

Prof. Dr. Heinz Georg Schuster danke ich für die Übernahme der Begutachtung dieser Dissertation, sowie für das schreiben interessanter Artikel und Bücher, die mich inspirierten und Einfluss auf meine Arbeit hatten.

Meinem Zimmergenossen und Mitstreiter Jens Otterpohl danke ich für das Teilen guter Ideen, Musik, Witze und so manchem Bier.
Klaus Bowe, Dr. Stefan Liehr und vor allem Dr. Udo Ernst für schnelle Hilfe und Unterstützung bei Soft- und Hardwareproblemen. Dr. Christian Eurich für den Hinweis auf den Artikel von [Zhang et al., 1998]. Dr. Rolf Henkel für Interesse an meiner Arbeit und Gute Ratschläge, die nicht nur auf das wissenschaftliche begrenzt waren. Aladin Mirmohammadhosseini für inspirierende Diskussionen über biologische Neuronen, Gott und die Welt vor allem aber für charakterliche Stärke. Roland Rothenstein für Diskussionen über Spiele von, für oder mit Minderheiten, Konstruktivismus, Didaktik und für das Korrekturlesen der Dissertation. David Rothermund, ohne den einige Graphiken dieser Dissertation anders ausgesehen hätten und der mir durch seine Liebe zur Natur, unnötige Gänge in die Bibliothek erspart hat. Erich Schulzke für einen fast "perfekt" durchdachten Umzug und nächtliche Aktionen. Prof. Dr. Helmut Schwegler danke ich für die Gründung des Institutes, indem die Möglichkeit zur Forschung in interdisziplinärer Theoretischer Physik geboten wird. Speziell danke ich für seine diplomatische Art, die so manche hitzige Diskussion in der Arbeitsgruppe entschärft hat. Dr. Andreas Thiel für die Einrichtung einer LaTeX-Hotline, sowie gemeinsame Aufenthalte im Eisen und das Korrekturlesen dieser Arbeit.

Besonderer Dank gilt unserer Sekretärin Agnes Janssen, die mir viel Bürokratie abnahm.

Ich bedanke mich bei allen Mitgliedern der Arbeitsgruppen von Prof. Dr. Helmut Schwegler und Prof. Dr. Klaus Pawelzik:
Matthias Bethge, Ronald Bohrmann, Dr. Udo Ernst, Axel Etzold, Dr. Christian Eurich, Pit Hankel, Dr. Rolf Henkel, Dr. Stefan Liehr, Aladin Mirmohammadhosseini, Jens Otterpohl, Roland Rothenstein, David Rothermund, Erich Schulzke, Bailu Si, Dr. Andreas Thiel, Dennis Trenner
für eine interessante, ja ich möchte sagen einzigartige Atmosphäre.

Desweiteren bedanke ich mich bei:
Gökhan Bakir, der mir einige Kurzaufenthalte bei der Deutschen Luft- und Raumfahrtgesellschaft (DLR) ermöglichte und immer für eine Überraschung gut ist. Tom Bielefeld für trockene Kommentare und seinen Humor. Matthias Dehmer für die interessante Maus-Tastatur-Monitor Konstellation und die Rückkehr in die Wissenschaft. John D. Haynes für die Zusammenarbeit beim Taubenprojekt und interessanten Diskussionen über stochastische Oszillatoren. Hierdurch habe ich viel über interdisziplinäres Arbeiten gelernt, was mir sicherlich noch mal von Nutzen sein wird. Prof. Dr. Frank Jahnke für eine perfekt organisierte Vorlesung über Theoretische Mechanik, die mir die Arbeit in den Übungen sehr erleichtert hat. Prof. Dr. Andreas Kreiter und Prof. Dr. Klaus Pawelzik danke ich für die Veranstaltung des Seminars über “Fragen der experimentellen und theoretischen Neurobiologie”, die zur Erhellung beigetragen hat. Bei Prof. Dr. Peter Richter bedanke ich mich für die Durchführung interessanter Seminare in angenehmer Atmosphäre. Dem Fachbereich für Physik der Universität Bremen, der mir die Teilnahme am Nobelpreisträger Treffen in Lindau ermöglichte. Der Wilhelm und Else Heraeus Stiftung danke ich für die Einladung zur Sommerschule “Nichtlineare Dynamik ungeordneter Systeme” in Chemnitz, die zur Erweiterung meines Horizontes aber auch zur Einordnung wichtig war.
Seit der Einführung des Internets steht ein Medium zur Verfügung, welches es gestattet, an Information beliebiger Art, durch ein paar Mausklicks zu gelangen. Deshalb sei hier explizit den Entwicklern von Suchmaschinen, sowie all denen gedankt, die Ihre Ideen oder Arbeiten frei ins Netz stellen, und diese hierdurch zugängig machen.

Ganz besonderer Dank gebührt meinen lieben Eltern und meiner Schwester, die mich als einzige die ganze Zeit begleitet und unterstützt haben. Besonders für die finanzielle Unterstützung in den letzten Monaten meiner Arbeit bedanke ich mich herzlich, da Sie mir hierdurch ein selbstständiges und unabhängiges weiterarbeiten ermöglichten, so dass die vorliegende Arbeit meinen Stempel trägt.

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Dissertation zum Thema

Aktive Computation in offenen Systemen

ohne unerlaubte fremde Hilfe angefertigt, keine anderen als die von mir angegebenen Quellen benutzt und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Bremen, Januar 2003