

**Ko-Evolution von Aktivitäten
und synaptischen Gewichten
in assoziativen neuronalen Netzwerken**

Diplomarbeit

vorgelegt von
Christian Albers

Matrikelnummer 1368133

Institut für Theoretische Physik
der
Universität Bremen

Bremen den 02.11.2009

Erstgutachter: Klaus Pawelzik
Zweitgutachter: Helmut Schwegler

Inhaltsverzeichnis

1	Einleitung und Motivation	3
2	Biologische Grundlagen	7
2.1	Das Neuron	7
2.2	Die Synapse	9
2.3	Spike-Timing Dependent Plasticity und Lernen	10
3	Theoretische Grundlagen	12
3.1	Das Modellneuron	12
3.2	Erweiterung des Modells: Mean Field Beschreibung	14
3.2.1	Stochastische Neurone	14
3.2.2	Mean Field-Beschreibung	15
3.3	Netzstrukturen	16
3.3.1	Rekurrente neuronale Netze	16
3.3.2	Attraktoren, das Hopfield-Netzwerk und Content Adressable Memory	16
3.3.3	Das Perzeptron	18
3.4	Lernregeln in künstlichen neuronalen Netzen	20
3.4.1	Die Perzeptronlernregel	21
3.4.2	Die Perzeptronlernregel in einem rekurrentem Netz	22
3.4.3	Fehlerfunktion	23
3.5	Umrechnung in ein Ratenmodell mit rein positiven Zuständen	24

4	Die Simulationen	27
4.1	Die Modelle	27
4.2	Die Muster	27
4.3	Die Lernprotokolle	28
4.4	Bestimmung der Kapazität; Überlapp	28
5	Simulationen des diskreten Netzes	30
5.1	Batch-Lernen	30
5.1.1	Der Lernalgorithmus	30
5.1.2	Abhängigkeit der Kapazität von der Schwelle	32
6	Simulation des kontinuierliches Netzes	34
6.1	Die Lernregel aus der Fehlerfunktion	35
6.2	Batch-Lernen	36
7	Online-Lernen	38
7.1	Lösung für die Änderung der Gewichte im diskreten Netz	38
7.2	Online-Lernen im Mean-Field-Netz	40
7.3	Diffusion der Eingaben	42
8	Korrelierte Muster	44
8.1	Lokal korrelierte Muster	44
8.1.1	Online	44
8.1.2	Batch	48
8.2	Global korrelierte Muster im Batch-Lernen	49
9	Diskussion	52
9.1	Zusammenfassung der Ergebnisse	52
9.2	Ausblick	53

Kapitel 1

Einleitung und Motivation

Das Gehirn ist ein komplexes Organ, das aus ungefähr 10^{11} einzelnen Nervenzellen, Neurone genannt, aufgebaut ist. Im Mittel hat jedes Neuron 10^4 Verbindungen zu anderen Neuronen. Diese hohe Vernetzung ist die Grundlage für unsere kognitiven Fähigkeiten. Wir nehmen die Umwelt mit unseren Sinnen wahr, die Sinneseindrücke werden verarbeitet und erzeugen unsere Reaktion auf die Welt. Die Reaktion ist variabel; sie hängt davon ab, wo wir uns befinden, wie wir uns fühlen, und vor allem davon, was wir in der Vergangenheit gelernt haben. Gleichzeitig können wir auch immer neue Erinnerungen aufbauen. Es stellt sich daher die Frage, wie Erinnerungen aus unseren Sinneseindrücken erzeugt werden.

Ein Neuron kommuniziert mit anderen Neuronen über sogenannte Aktionspotentiale, die wegen ihres zeitlich-räumlichen Verlauf auch Spikes genannt werden. Diese stellen eine chemisch-elektrische Erregung dar, die die Verästelungen der Neurone entlanglaufen. Kommen sich zwei dieser Verästelungen zweier verschiedener Neurone nahe, werden Synapsen ausgebildet. Eine Synapse ist im Prinzip auf der Senderseite (“präsynaptisch”) ein Reservoir an Botenstoffen, den Neurotransmittern, die beim Eintreffen eines Aktionspotentials von der Synapse ausgeschüttet werden und die den synaptischen Spalt überwinden. Diese Neurotransmitter erregen an den zugehörigen Rezeptoren auf der Empfängerseite (“postsynaptisch”) nun einen Spike, der dann bis zum Zellkern läuft.

Durch Kenntnis der Vorgänge sind wir aber nicht in der Lage, zu erkennen, was Gedächtnis nun eigentlich ist. Ein Neuron feuert; wo ist das Gedächtnis? Hinweise darauf, was Gedächtnis sein kann, liefern sowohl experimentelle Befunde als auch Modelle und Simulationen. Miyashita [1]

untersuchte 1988 bei Primaten einen bestimmten Bereich im temporalem Cortex, indem mit Elektroden die Aktionspotentiale einzelner Neurone gemessen wurden. Den Tieren wurden Bilder gezeigt, und zu jedem Bild gehörte ein Satz gemessener Neurone, die in dieser Konstellation nur bei diesem bestimmten Bild feuerten. Ein einzelnes Neuron konnte durchaus bei verschiedenen Bildern aktiv sein, aber ein Bild erzeugte ein charakteristisches Muster in der Aktivierung. Neuere Ergebnisse wurden in [2] vorgestellt. Hier wurde die Ortsrepräsentation im Hippocampus von Ratten untersucht. Die Tiere wurden zuerst in eine quadratische und danach in eine kreisförmige "Arena" (ein nach oben offener Kasten) gesetzt, und die zugehörige Aktivierung im Hippocampus gemessen. Danach setzte man die Ratten in Arenen, die mehr dem Quadrat oder dem Kreis ähnelten, und je nachdem, welcher ursprünglich kennengelernten Arena die aktuelle ähnlich war, entsprach die gemessene neuronale Aktivierung der ähnlicheren Ursprungsform.

Diese experimentellen Befunde können mit Hilfe von theoretischen Modellen neuronaler Netzen (siehe z.B. das Hopfield-Netz, [3], das Modell in [17]) verstanden werden: Ein Gedächtnisinhalt ist eine bestimmte Kombination an Neuronen, die zusammen feuern. Wird dem Netz ein unvollständiges Bild präsentiert, indem man zum Beispiel nur einen Teil der Neurone anregt, die zu einem bestimmten Gedächtnisinhalt gehören, dann werden die aktiven Neurone durch die Art der Vernetzung die noch stillen Neurone anregen, so dass am Ende das zur vollständigen Erinnerung gehörende Aktivitätsmuster der Neurone zu sehen ist. In diesem Bild des Hopfield-Netzes sind die Erinnerungen stabile attraktive Fixpunkte der Netzwerkdynamik. Die Eigenschaften des Netzes, also die Erinnerungen, werden durch die Stärke der gegenseitigen synaptischen Anregungsstärken festgelegt.

Neben der Frage nach den molekularen Mechanismen der synaptischen Veränderlichkeit ist aber auch die Frage wichtig, wodurch die Veränderungen bewirkt werden. Eine erste Formulierung der Ursache synaptischer Verstärkung stammt von Donald Hebb (1949, [5]): Trägt ein Neuron A wiederholt und konsistent dazu bei, dass Neuron B feuert, dann müssen metabolische Prozesse an der Synapse stattfinden, so dass die Effizienz, mit der Neuron A Neuron B anregt, gesteigert wird. Dieses Postulat ist heute noch gültig und zeigt den Grund synaptischer Veränderungen. Die Synapse wird als Antwort auf Spikes, die in bestimmter Weise dort ankommen verändert. Das Hebb'sche Postulat wird häufig abgekürzt, zum Beispiel zu der Hebb'schen Lernregel im Hopfield-Netz. Dort wird die Kausalität zu einer Korrelation: Haben zwei Neurone in einem zu

speichernden Gedächtnisinhalt den gleichen Zustand, dann wird die Verbindung zwischen ihnen um einen bestimmten Betrag erhöht, andernfalls abgesenkt.

Das originale Postulat hat aber einen Nachteil: es macht nur Aussagen über die Verstärkung der Synapse zwischen zwei erregenden (exzitatorischen) Neuronen. Ein Netz aus exzitatorischen Neuronen, in dem Synapsen nur verstärkt werden können, wäre instabil. Es gibt in theoretischen Modellen einige Möglichkeiten, das Problem zu umgehen. Zum Beispiel können alle synaptischen Verbindungen eines Neurons nach jedem Lernschritt normalisiert werden, was zu einer Art Wettbewerb zwischen den Synapsen führt. Oder die Lernregel ist nicht nur vom Vorhandensein von Aktivität abhängig, sondern auch von der Feuerrate. Ein Beispiel für eine solche Lernregel ist die BCM-Regel: Eine geringe Feuerrate eines postsynaptischen Neurons bei aktivem präsynaptischen Neuron führt zu einer Abschwächung von Synapsen ([6]).

1997 wurde experimentell gefunden, dass nicht (nur) die Feuerrate Einfluß auf eine Änderung einer Synapse hat, sondern auch das "Timing", d.h. der zeitliche Abstand von prä- und postsynaptischen Spikes. Markram et al. [7] fanden heraus, dass die Kombination von präsynaptischen Spikes 10 ms *vor* postsynaptischen Spikes zu einer Verstärkung, ein präsynaptischer Spike 10 ms *nach* dem postsynaptischen Spike zu einer Abschwächung der Synapse führten. Diese Befund wurde später von anderen Gruppen bestätigt, und auch die Abhängigkeit der Veränderung von dem Intervall zwischen den Spikes wurde genauer bestimmt (für eine Übersicht siehe z.B. [8]). Dieses Verhalten der Synapse wird als Spike-Timing Dependent Plasticity (STDP) bezeichnet, und liefert eine Vorschrift, die sowohl die Verstärkung als auch die Abschwächung von synaptischen Stärken beschreibt.

Senn, Markram und Tsodyks haben in einer Arbeit versucht, die STDP als Veränderung der Öffnungswahrscheinlichkeit der präsynaptischen Vesikel zu modellieren [10]. Die Ergebnisse dieser Arbeit lassen bei Vereinfachung der Formulierung den Anschluss der STDP an schon existierende Lernregeln in neuronalen Netzen zu.

Die Fragestellung in dieser Arbeit geht aber nicht in erster Linie von biologischen Vorbildern aus. Der Ausgangspunkt ist die Perzeptronlernregel, die als ein abstrakter Algorithmus erdacht wurde. Dieser Algorithmus erlaubt es, ein Perzeptron zum Klassifizieren von Eingangsmustern einzusetzen. In einem rekurrenten neuronalen Netz eingesetzt ermöglicht die Perzeptronlernregel das Lernen von Mustern, die dann zu sogenannten Attraktoren der Dynamik werden. Die in dieser

Arbeit betrachteten Netze sind solche aus einfachen Modellneuronen, das bedeutet, dass jedes Neuron eine Verbindung zu jedem anderen Neuron besitzt. Das Lernen von Gesamtzuständen bestimmter Aktivitäten der Neurone, den Mustern, wird untersucht.

In Kapitel 2 werden die biologischen Grundlagen vorgestellt, die zum Verständnis des Themas nötig sind. Dabei wird auf das Neuron, die Synapse, und bekannte Ergebnisse aus Experimenten zur Veränderung der Synapse eingegangen. Im Kapitel 3 werden die theoretischen Grundlagen der beiden hier behandelten Modelle künstlicher neuronaler Netze gelegt. Wichtig ist hierbei natürlich das Modellneuron, seine verschiedenen Beschreibungen und das Verhältnis zum biologischen Vorbild. Außerdem werden die möglichen Netzstrukturen und verschiedene Lernvorschriften vorgestellt. Im Kapitel 4 wird die Durchführung der Simulationen beschrieben, unter anderem, wie die Kapazität eines Netzes bestimmt werden kann. Kapitel 5 widmet sich den Ergebnissen der Simulationen in einem Netz aus diskreten Neuronen, also Neuronen, die nur zwei verschiedene Zustände einnehmen können. In Kapitel 6 wird das gleiche für das zweite Netzwerkmodell mit kontinuierlichen Neuronenzuständen vorgestellt. Im Kapitel 7 werden die Ergebnisse des zweiten Lernmodus, dem sequentiellen Lernen, vorgestellt. Im Kapitel 8 geht es um das Lernen von korrelierten Mustern, und die Unterschiede zu den vorher betrachteten unkorrelierten Mustern. Im letzten Kapitel 9 werden die Ergebnisse zusammengefasst und diskutiert.

Kapitel 2

Biologische Grundlagen

2.1 Das Neuron

Die Darstellung in diesem Abschnitt lehnt sich an die Lehrbücher an (siehe z.B. [11]).

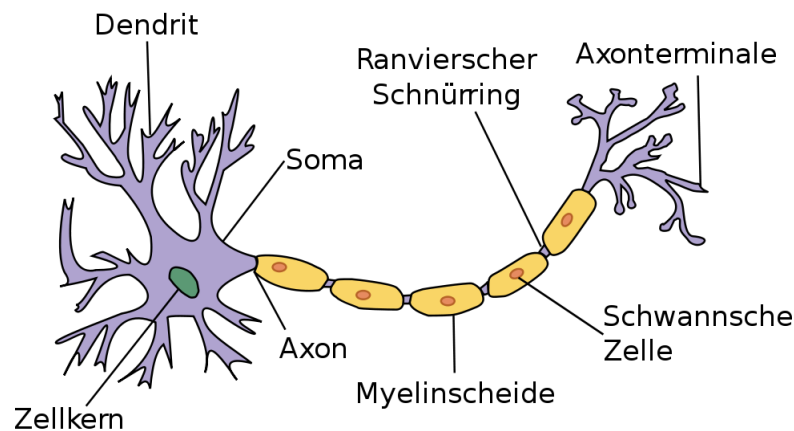


Abbildung 2.1: Schematische Darstellung eines Neurons.

Ein Neuron besteht aus mehreren Teilen: Dem Zellkörper, dem Axonhügel und davon ausgehendn Axon, und den Dendriten. Das Neuron wird durch seine Membran von dem Umgebungsmedium getrennt. In der Membran sind Ionenkanäle und Ionenpumpen eingelagert, die den Fluss von Ionen in die Zelle hinein und aus der Zelle heraus kontrollieren. Im Ruhezustand besteht dadurch ein Konzentrationsunterschied der Ionen von der Zelle zum umgebenden Medium, was einen Potentialunterschied von typischerweise 70mV erzeugt. Der Potentialunterschied

kann durch den Fluss von Ionen verändert werden. Besteht ein Nettoeinfluß von positiven Ionen, dann wird der Potentialunterschied verringert (sog. Depolarisation). Ist eine bestimmte Schwelle in der Depolarisation erreicht, wird eine positive Rückkopplung eingeleitet. Dadurch wird am Axonhügel ein sogenanntes Aktionspotential ausgelöst, eine Zone hoher Depolarisation, die das Axon entlangwandert. Der zeitliche und räumliche Verlauf dieser Depolarisation wird durch spannungsabhängige Ionenkanäle gesteuert. Bei einer leichten Depolarisation öffnen sich Na^+ -Kanäle, worauf diese Ionen in die Zelle eintreten, die Depolarisation wird verstärkt. Danach öffnen sich durch den Verlauf der Spannung die Kanäle für Ka^+ -Ionen, die aus der Zelle austreten. Das Resultat ist eine Umkehrung der Depolarisation in eine leichte Hyperpolarisation. Durch diese Diffusionsprozesse ist am Ende des zeitlichen Verlaufs des Aktionspotentials ein Konzentrationsausgleich gegeben, wodurch ein weiteres Aktionspotential für eine kurze Zeit verhindert wird. Die Zeit, in der das Neuron gezwungenermaßen inaktiv ist, wird absolute Refraktärzeit genannt. Durch diesen Prozess wird auch ein Zurücklaufen des Spikes verhindert.

Am Axon befinden sich an den Stellen, wo Dendriten anderer Neurone nahe sind, sogenannte Synapsen. Trifft ein Aktionspotential (Spike) auf eine Synapse, werden dort Vesikel geöffnet, kleine Bläschen, die Botenstoffe enthalten. Diese Botenstoffe oder auch Neurotransmitter überwinden den synaptischen Spalt und können auf der Seite des Dendriten durch die Aktivierung passender Rezeptoren ein weiteres Aktionspotential auslösen, welches den Dendritenbaum bis zum Zellkörper des Neurons entlangwandert. Dort wird das Aktionspotential das Membranpotential des Neurons verändern.

Es gibt exzitatorische und inhibitorische Synapsen. Der Unterschied liegt darin, dass exzitatorische Synapsen mit den übermittelten Spikes das Membranpotential der Zielzelle erhöhen (also depolarisierend wirken), was die Zelle in Richtung Auslösen eines eigenen Spikes treibt. Inhibitorische Synapsen unterdrücken die Aktivität des Zielneurons. Im Allgemeinen wirkt eine Zelle entweder nur exzitatorisch oder nur inhibitorisch. Zusätzlich unterscheiden sich Spikes verschiedener exzitatorischer Synapsen voneinander in der Größe der verursachten Änderung.

Die Zelle macht also so etwas eine Summierung der Spikes, die von anderen Neuronen kommen, und wird selber einen Spike abgeben, wenn genügend Spikes angekommen sind. Das stimmt natürlich nicht ganz, da jedes Membranpotential, dass sich vom Gleichgewichtszustand unterscheidet, in Richtung Gleichgewichtsspannung zerfallen wird. Wenn also die Abstände zwischen

den Spikes zu groß sind, stimmt das Bild der linearen Summierung nicht mehr.

2.2 Die Synapse

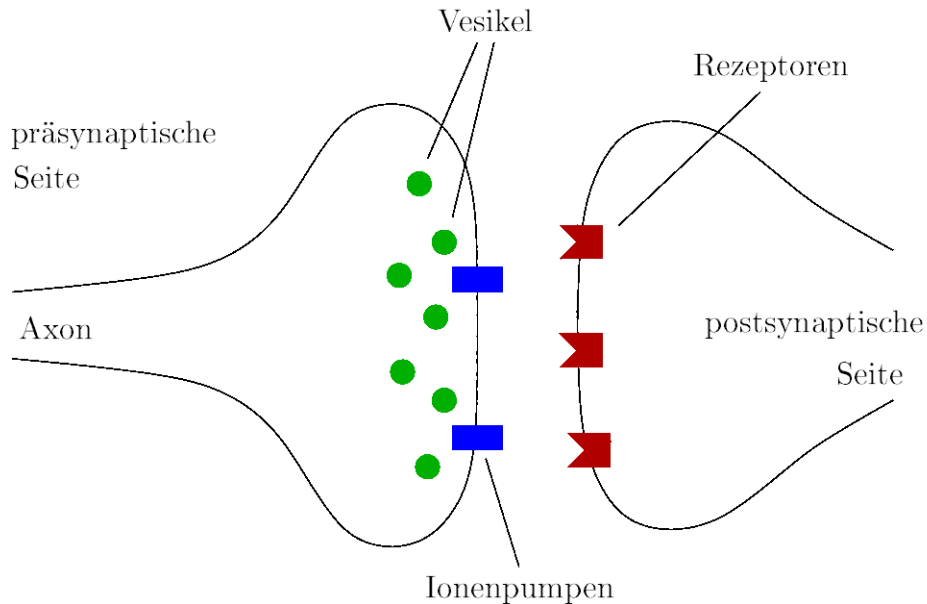


Abbildung 2.2: Schematischer Aufbau einer Synapse

Die Synapse ist der Teil des Neurons, welcher den Spalt zwischen Axon und Dendrit überwindet. Auf der Seite des Axons (“präsynaptisch”) befinden sich die Vesikel, die sich als Antwort auf einen Spike öffnen können. Trifft ein Spike ein, schütten sie ihre Neurotransmitter aus, die dann über den Spalt hinweg diffundieren. Trifft ein Molekül eines Neurotransmitters auf einen passenden Rezeptor, wird ein Ionenkanal geöffnet. Ein geöffneter Ionenkanal lässt entweder nur Anionen oder Kationen durch oder ist ganz auf bestimmte Ionen, z.B. Ca^{2+} oder Na^+ , spezialisiert. Die eingelassenen Ionen verändern den Potentialunterschied auf der Empfängerseite und es wird dann wieder ein Spike ausgelöst, der den Dendrit weiterwandert. Die Vesikel füllen sich nach dem Entlassen der enthaltenen Neurotransmitter langsam wieder auf, ein schnell folgender zweiter Spike wird daher auf weniger aktive Vesikel treffen als der erste. Dies ist Teil eines Vorgangs, der Short-Time Depression genannt wird, bei dem schnell aufeinanderfolgende präsynaptische Spikes immer mehr an Wirkung verlieren. Es geht bis zu dem Punkt, an dem die Synapse gar

nicht mehr auszulösen vermag. Es gibt allerdings auch einen Anbahnungsprozess (Facilitation), der im Zeitraum von Sekunden nach einem Spike die Anregbarkeit einer Synapse erhöht. Was man an der Synapse antrifft, hängt von den Arten der beteiligten Neurone ab.

Es wird angenommen, dass diese schnellen Prozesse auf Sekundenskala Teil des “Working Memory”, des Arbeitsgedächtnisses sind, bei dem gewisse Gedächtnisinhalte für eine kurze Zeit zwischengespeichert werden müssen, damit das Gehirn weiter mit ihnen arbeiten kann [12].

2.3 Spike-Timing Dependent Plasticity und Lernen

Neben diesen schnellen Prozessen, die die Stärke der Synapse beeinflussen, gibt es auch Veränderungen auf langer Zeitskala, von Stunden bis Tage und länger. Frühere Beobachtungen von exzitatorischen Synapsen wurden unter Long-Term Potentiation (LTP) und Long-Term Depression (LTD) zusammengefasst. Die LTP ist die Verstärkung der Synapse, die relativ spät (einige Stunden) nach dem veränderndem Ereignis auftritt, LTD die Abschwächung. Ab einer gewissen Feuerrate des präsynaptischen Neurons führt eine starke postsynaptischen Depolarisation zu LTP, und niedrige präsynaptische Feuerraten, entweder kombiniert mit einer geringen postsynaptischen Depolarisation oder gar keiner postsynaptischer Veränderung führen zu LTD. Zusammengefasst sind LTP und LTD also zwei Mechanismen, die eine Synapse in beide Richtungen verändern können, und die dabei von der Aktivität der beteiligten Neurone abhängen.

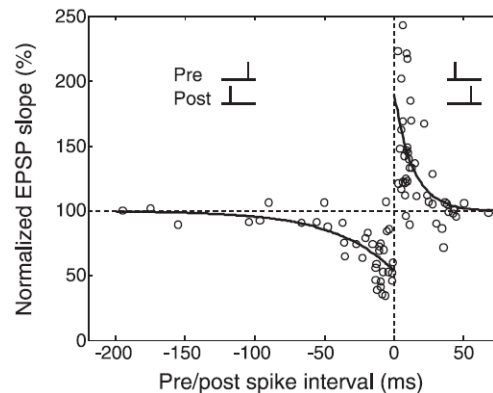


Abbildung 2.3: Die Änderung der Synapse in Abhängigkeit vom Intervall zwischen den Spikes prä- und postsynaptisch. Negative Zeit entspricht einem Spike post- vor präsynaptisch. Aus [8]

Spike-Timing Dependent Plasticity (STDP) ist ein neueres Paradigma für die Veränderung von Synapsen. Im Experiment werden Spikes prä- und postsynaptisch gepaart, indem von außen die Zellen zu bestimmten Zeitpunkten zum Feuern angeregt werden. Diese Spikepaarungen werden mit einem festen zeitlichen Abstand zwischen prä- und postsynaptischem Spike mit einer Frequenz von einigen Hertz für eine kurze Zeit induziert. Nach einer Zeit, die im Bereich von Stunden liegt, wird dann die Stärke der Synapse mit ihrer Stärke vor der Anregung verglichen (Für eine Übersicht über die experimentellen Befunde, siehe [13],[8]). Die Änderung des synaptischen Gewichts hängt dann von dem genauen zeitlichen Abstand zwischen den Spikes prä- und postsynaptisch ab. Kommt zuerst der präsynaptische Spike und danach der postsynaptische, steigt das synaptische Gewicht, und zwar um so mehr, je kürzer der Abstand zwischen ihnen ist. Feuert zuerst das postsynaptische Neuron und danach das präsynaptische, wird die Synapse abgeschwächt, und auch um so mehr, je kürzer der Zeitraum zwischen den Spikes ist. Dieser Zusammenhang wird in Abb. 2.3 gezeigt.

Dass die Synapse den Zeitunterschied messen kann, liegt an den sogenannten Back-Propagating Action Potentials (siehe z.B. [9]). Wenn das Neuron ein Aktionspotential auslöst, dann wandert ein Spike das Axon entlang, aber zusätzlich laufen auch Spikes den Dendritenbaum “verkehrt herum” zurück. Das Eintreffen eines Back-Propagating Action Potentials bewirkt auf der postsynaptischen Seite der Synapse, dass sich bestimmte spannungsabhängige Ionenkanäle öffnen. Der resultierende Fluss von Ionen kommt als Detektor an der Synapse für postsynaptische Spikes in Frage.

Kapitel 3

Theoretische Grundlagen

Es existiert eine Vielzahl von Modellen, um die Funktion von Neuronen und die Vorgänge in Netzen aus Neuronen nachzubilden. Dabei gilt typischerweise, dass eine Vereinfachung des Modells und damit schnelle Simulationen der zugehörigen Dynamik mit einer Entfernung vom biologischen Vorbild erkaufte werden. Auf der einen Seite gibt es das Hodgkin-Huxley-Modell, welches aus einer Anzahl von Differentialgleichungen mit einigen Parametern besteht (siehe z.B. [11]). Dieses Modell beschreibt das Verhalten eines Neurons sehr gut, und es war ein wichtiger Schritt bei der Aufklärung der Vorgänge in einem Neuron. Es eignet sich aber nicht für die Simulation von größeren Netzen, da die Simulation eines Hodgkin-Huxley-Neurons relativ rechenintensiv ist. Auf der anderen Seite steht das McCulloch-Pitts-Neuron, welches einfach eine lineare Summation der Zustände anderer Neurone, multipliziert mit den synaptischen Gewichten ist. Dies Modell ist besonders einfach, aber sehr weit vom biologischen Vorbild entfernt. Trotzdem bietet es Vorteile, da Simulationen von neuronalen Netzen sehr leicht und schnell durchgeführt werden können, und trotzdem bestimmte Eigenschaften des Vorbilds "Gehirn" erhalten bleiben. Die Fragestellung dieser Arbeit ist auf die Auswirkungen einer perzeptronartigen Lernregel auf künstliche neuronale Netze beschränkt, daher ist das McCulloch-Pitts-Neuron das hier verwendete Modell.

3.1 Das Modellneuron

Das McCulloch-Pitts-Neuron ist eine starke Vereinfachung eines realen Neurons. Die Eingabe in das Neuron erfolgt nicht in Form von Spikes, sondern ist die mit den synaptischen Kopplungen

gewichtete Summe der Zustände aller anderen Neurone, die auf das betrachtete Neuron wirken. Dabei kennt das Neuron nur zwei mögliche Zustände, 0 und 1. 1 bedeutet, das Neuron feuert (ist aktiv), 0 bedeutet, es feuert nicht ist inaktiv. Unterschiede in der Aktivität, zum Beispiel unterschiedliche Feuerraten, werden nicht berücksichtigt. Die Summation der Eingaben ist linear. Alle Effekte von möglichen Wechselwirkungen zwischen Spikes wie auch der Zerfall einer Auslenkung des biologischen Neurons aus dem Gleichgewichtszustand werden nicht berücksichtigt. Die synaptische Kopplung ist eine einzige Zahl, ebenso die Schwelle des Neurons, wenn sie nicht gleich auf Null gesetzt wird. Daraus ergibt sich das Schaltbild in Abb. 3.1. Die Entwicklung in der Zeit ist diskret; die Dynamik verläuft in Schritten. Der Zustand eines Neurons zum Zeitpunkt $t + 1$ berechnet sich daher aus den Zuständen der anderen Neurone zum Zeitpunkt t :

$$n_i(t + 1) = \Theta(h_i) = \Theta\left(\sum_j w_{ij}n_j(t) + I_i\right) \quad (3.1)$$

$n_i(t)$ bezeichnet den Zustand des Neurons i zum Zeitpunkt t , h_i ist die Gesamteingabe, w_{ij} ist das synaptische Gewicht von Neuron j auf Neuron i , und I_i bezeichnet eine feste, vom Netzzustand unabhängige Eingabe auf das Neuron i . In einigen Einführungen wird diese feste Eingabe als Schwelle bezeichnet, da beides äquivalent ist. Die Übertragungsfunktion Θ (auch Gainfunktion genannt) ist die Stufenfunktion:

$$\Theta(x) = \begin{cases} 1 & \text{falls } x \geq 0 \\ 0 & \text{sonst} \end{cases} \quad (3.2)$$

Die Dynamik eines Netzes aus McCulloch-Pitts-Neuronen ist zeitdiskret. Dabei gibt es zwei verschiedene Möglichkeiten, wie man die Zustände eines Neurons zu einem Zeitpunkt t berechnet. Entweder werden zu jedem Zeitschritt alle Neuronen neu berechnet; das ist eine synchrone Dynamik. Im Gegensatz dazu steht die asynchrone Dynamik, wo in jedem Zeitschritt ein Neuron zufällig ausgewählt wird, dessen Zustand neu berechnet wird.

Um bestimmte Rechnungen zu vereinfachen, kann man die möglichen Zustände des Modellneurons mit -1 (inaktiv) und $+1$ (aktiv) bezeichnen. Die Simulationen in dieser Arbeit wurden sämtlich mit diesem Neuron durchgeführt. Die Umrechnung ist einfach:

$$S_i = 2 \cdot n_i - 1 \quad (3.3)$$

S_i ist der zu n_i gehörige Zustand eines Neurons mit Zuständen $-1/ +1$.

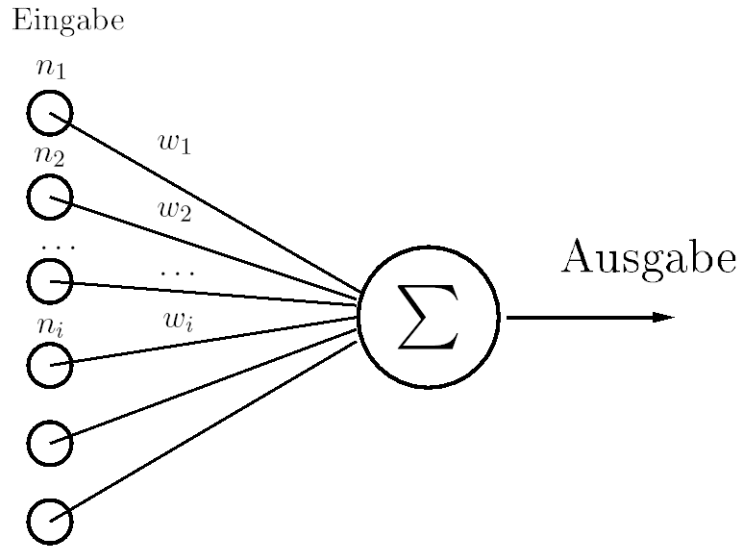


Abbildung 3.1: Schematisches Schaltbild eines McCulloch-Pitts-Neurons

3.2 Erweiterung des Modells: Mean Field Beschreibung

3.2.1 Stochastische Neurone

Um ein Zufallselement in das System einzuführen, kann man von der streng deterministischen Dynamik weggehen und den möglichen Zuständen der Neurone Wahrscheinlichkeiten zuweisen. Wenn ein Neuron ein Update erfährt, dann hat jeder der beiden möglichen Zustände eine gewisse Wahrscheinlichkeit, die von der Gesamteingabe h_i abhängen wird. Eine geläufige Wahl ist die logistische Funktion:

$$P(S_i = \pm 1) = f_\beta(\pm h_i) = \frac{1}{1 + \exp(\mp \beta h_i)} \quad (3.4)$$

Stochastische Neurone reflektieren eine gewisse Unsicherheit bei der Aktivität von realen Neuronen, d.h. sie sind in einem gewissen Sinne realistischer. Sie haben auch Vorteile für bestimmte Berechnungen. Zum einen führt in einem Hopfield-Netz eine stochastische Dynamik der Neuronen dazu, dass die gewünschten Zustände des Netzes häufiger erreicht werden (siehe auch Abschnitt 3.3.2). Zum anderen bietet die Stochastizität die Möglichkeit, zu einer mittleren Beschreibung der Neurone überzugehen, wobei man den mittleren Zustand eines Neurons mit einer Feuerrate identifizieren kann. Diesen Schritt kann man gehen, um sich wieder dem biologischen Vorbild anzunähern.

Der Parameter β beschreibt das Rauschen im Netz. Kleine Werte für β bedeuten eine hohe Unsicherheit für den Zustand des Neurons. Die Idee für die Einführung von stochastischen Neuronen kommt aus der Bearbeitung des Problems der Spingläser, dem Ising-Modell. In diesem Modell gilt

$$\beta \propto \frac{1}{T} \quad (3.5)$$

womit das Rauschen anschaulich mit einer Art Temperatur und temperaturbedingten Fluktuationen der Spinzustände in Verbindung gebracht wird. Diese Temperatur hat im Bild des neuronalen Netzes natürlich nichts mit der Umgebungstemperatur zu tun, sondern ist ein frei wählbarer Systemparameter.

3.2.2 Mean Field-Beschreibung

Die Beschreibung im Mean Field folgt ebenfalls aus dem Ising-Modell. Die Eingabe auf ein Neuron ist

$$\langle h_i \rangle = \sum_j w_{ij} \langle S_j \rangle \quad (3.6)$$

$\langle S_j \rangle$ ist die mittlere Aktivität des Neurons j . Durch die Summation der mittleren Aktivitäten ist die Eingabe auf das Neuron i ebenfalls eine Mittelung, die dabei einer Mittelung der Zustände über ein Ensemble von gleichen Neuronen mit Eingabe h_i entspricht. Im Ising-Modell ist die mittlere Eingabe $\langle h_i \rangle$ das mittlere Feld am Ort des Spins I , daher Mean Field Beschreibung. Die mittlere Aktivität eines Neurons kann man nun mit dem mittleren Feld leicht berechnen:

$$\langle S_i \rangle = (+1) \cdot P(S_i = +1 | \langle h_i \rangle) + (-1) \cdot P(S_i = -1 | \langle h_i \rangle) \quad (3.7)$$

Setzt man für die Wahrscheinlichkeit die logistische Funktion (3.4) ein, so erhält man

$$\langle S_i \rangle = \tanh(\beta \langle h_i \rangle) \quad (3.8)$$

Im Folgenden wird zur Abkürzung $\langle S_i \rangle = x_i$ gesetzt. Die Gleichung (3.8) ist die zu Gleichung (3.1) äquivalente Vorschrift, um die Dynamik der Neuronen in dem Netz zu beschreiben.

In der Mean Field Beschreibung können die Neurone also Zustände in einem kontinuierlichen Intervall $[-1, +1]$ einnehmen. Dies erlaubt, den Zustand eines Neurons mit einer Feuerrate zu identifizieren. Dabei ist ein maximaler Wert für die Feuerrate auch in einem biologischen Neuron

zu finden. Wie in Kapitel 2 beschrieben, kann ein Spike nur weitergeleitet werden, wenn zwischen Zelle und äußerem Medium ein Konzentrationsgefälle herrscht. Gleichzeitig sorgt der Durchlauf eines Spikes für einen Konzentrationsausgleich, und verhindert so einen kurz darauf folgenden Spike.

3.3 Netzstrukturen

3.3.1 Rekurrente neuronale Netze

Ein rekurrentes neuronales Netz ist der allgemeinste Fall eines neuronalen Netzes. Zwischen jedem Paar von Neuronen i und j bestehen Verbindungen in Form von synaptischen Gewichten w_{ij} beziehungsweise w_{ji} , wobei im Allgemeinen kein Zusammenhang zwischen den Gewichten in beide Richtungen besteht: $w_{ij} \neq w_{ji}$. Der Zustand eines Neurons zur Zeit $t + 1$ ist eine Funktion der Zustände aller anderen Neurone zur Zeit t :

$$S_i(t + 1) = \text{sign}\left(\sum_j w_{ij} S_j(t)\right) \quad (3.9)$$

sign ist die Vorzeichenfunktion. Der feste Input I_i wird in dieser Beschreibung weggelassen. Wenn man einen festen Term braucht, kann man ihn durch ein Neuron ersetzen, welches für alle Zeiten auf einem unveränderlichen Zustand belassen wird.

3.3.2 Attraktoren, das Hopfield-Netzwerk und Content Adressable Memory

Ein Fixpunkt in einem Netzwerk ist ein Zustand des Netzes, der einmal erreicht sich nicht mehr verändert. Wenn ein Netz darüberhinaus aus Zuständen, die in der "Nähe" (im Zustandsraum betrachtet) des Fixpunktes liegen immer wieder in diesen relaxiert, dann hat man es mit einem Attraktor der Dynamik zu tun.

Neuronale Netze als Systeme mit Attraktoren der Dynamik sind aus zwei Gründen interessant. Zum einen verhalten sich einige Hirnareale so, als ob sie Systeme seien, die Attraktoren der Dynamik besitzen (siehe z.B. die Befunde von Miyashita (1988), [1]). Man zeigt einen Stimulus, z.B. ein bestimmtes Bild, und das untersuchte Hirnareal zeigt daraufhin ein bestimmtes Aktivitätsmuster, das auch nach dem Entfernen des Stimulus für eine gewisse Zeit erhalten bleibt. Ein

künstliches Netz, welches unter einem Stimulus in einen Attraktor fällt, kann daher als Beschreibung dieses Verhaltens dienen. Zum anderen sind Modellnetze mit Attraktoren als sogenannter Content Adressable Memory (CAM) einsetzbar. In einem CAM sind Muster gespeichert, und bei Präsentation eines verrauschten oder unvollständigen Musters wird das System in den Zustand laufen, der dem präsentierten Muster am nächsten liegt. Man vergleiche dieses Verhalten mit dem eines elektronischen Computers, in dem jeder Speicherinhalt mit einer Adresse versehen ist, über den er geschrieben und ausgelesen wird.

Das Hopfield-Netzwerk ist ein Spezialfall eines rekurrenten Netzwerks. Seine Gewichte sind symmetrisch, d.h. das synaptische Gewicht zwischen zwei Neuronen ist in beide Richtungen gleich groß. Um das Hopfield-Netz als CAM verwenden zu können, werden die Muster ξ^μ nach der Hebb'schen Lernregel in die Gewichte geprägt:

$$\begin{aligned} w_{ij} &= \alpha \xi_i \xi_j \text{ für ein Muster} \\ w_{ij} &= \alpha \sum_{\mu} \xi_i^{\mu} \xi_j^{\mu} \text{ für mehrere Muster} \end{aligned} \tag{3.10}$$

Der Faktor α wird im Allgemeinen auf $1/N$ gesetzt, wobei N die Anzahl der Neurone ist. Man sieht, dass durch die Lernregel die Gewichte w_{ij} symmetrisch werden. Das ist biologisch unplausibel, bringt jedoch Vorteile, wenn man die Eigenschaften des Netzes bestimmen möchte. Insbesondere folgt aus der Symmetrie der Gewichte die Existenz einer Energiefunktion:

$$H = -\frac{1}{2} \sum_i \sum_j w_{ij} S_i S_j \tag{3.11}$$

Diese Energiefunktion folgt aus dem Ising-Modell der Spingläser. Im Hopfield-Netz stellen die nach der Hebb-Regel gespeicherten Muster Minima der Energielandschaft dar. Man kann zeigen, dass bei der deterministischen Dynamik nur Zustandsänderungen erlaubt sind, bei denen die Energie des Netzes sinkt [14]. Daher stellt ein gespeichertes Muster einen stabilen attraktiven Fixpunkt der Dynamik dar.

Es gibt in diesem Zusammenhang zwei Probleme beim Hopfield-Netz. Zum einen ist die Kapazität des Netzes beschränkt auf $0,138N$ Muster (N ist die Anzahl der Neurone). Versucht man, mehr Muster einzuspeichern, dann werden die neuen Muster keine Attraktoren, und die schon gelernten Muster "brechen zusammen", d.h. verlieren ihre Stabilität. Zum anderen erzeugt die Hebb-Regel neben den gelernten Mustern im Allgemeinen weitere lokale Minima in

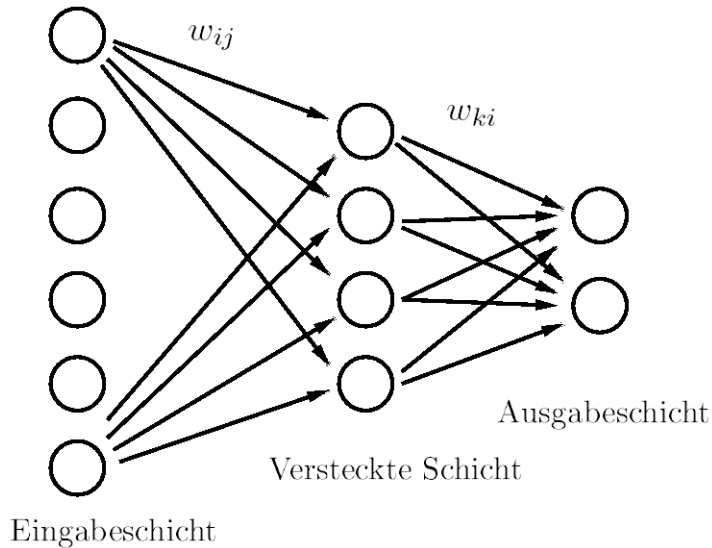


Abbildung 3.2: Ein Beispiel für ein Zweischichtenperzeptron

der Energielandschaft ([14]). Es kann auch passieren, dass zwei ähnliche Muster zu einem Überlagerungsmuster verschmelzen, selbst wenn man unter der Kapazitätsgrenze bleibt. Das erste Problem kann man in gewissem Ausmaß vermeiden, indem man stochastische Neurone einführt (siehe Abschnitt 3.2.1). Die Stochastizität führt dazu, dass dem Netz zufällig Energie hinzugefügt wird. Dies hat den Effekt, dass flache (lokale) Minima geglättet werden können.

3.3.3 Das Perzeptron

Das Perzeptron gehört zu den Feed-Forward-Netzen und ist im Vergleich zu rekurrenten Netzen stark in den Verbindungen der Neurone untereinander eingeschränkt. Die Neurone sind in Schichten organisiert, und jedes Neuron einer Schicht erhält Eingaben von allen anderen Neuronen der Schicht davor. Das gilt natürlich nicht für die Eingabeschicht, deren Zustand von außen eingestellt wird. Darüberhinaus gibt es keine Verbindungen, d.h. insbesondere dass die Neurone einer Schicht untereinander nicht verbunden sind, und ein Neuron auch nicht ein Signal wieder zurück zu einem Neuron in der vorhergehenden Schicht schicken kann.

Die Benennung der Perzeptrone richtet sich nach den Schichten von Verbindungen; ein Perzeptron mit nur einer Eingabe- und einer Ausgabeschicht wird Einschichtenperzeptron genannt. Die Schichten, die nicht Eingabe oder Ausgabe sind, werden versteckte Schichten genannt (siehe

Abb. 3.2).

Im Falle eines Einschichtenperzeptrons wird auf die k Eingabeneurone ein Muster ξ_k gegeben. Es erzeugt ein Muster O_i auf der Ausgabeschicht nach der Vorschrift

$$O_i = g(h_i) = g\left(\sum_k w_{ik}\xi_k\right) \quad (3.12)$$

Die Funktion $g(h)$ ist die Übertragungsfunktion, für die es verschiedene Wahlen gibt. Sie ist im Allgemeinen eine sigmoide Funktion wie der Tangens Hyperbolicus oder eine Stufenfunktion. Ein Perzeptron kann auch aus linearen Neuronen aufgebaut werden, dann ist die Übertragungsfunktion die Identität $g(h) = h$, diese werden in der vorliegenden Arbeit jedoch nicht betrachtet.

Funktion eines Einschichtenperzeptrons

Ein Perzeptron erhält eine Eingabe und erzeugt daraus eine Ausgabe. Wenn man die Gewichte richtig wählt, kann das Perzeptron dazu dienen, Eingabemuster ξ_k^μ zu unterscheiden, indem man sie auf das gewünschte Ausgabemuster ζ_i^μ abbildet:

$$O_i^\mu = \zeta_i^\mu$$

für alle i und μ . Die Aufgabe ist daher ebenfalls eine Assoziation von zwei Mustern, wobei die Paare aber vorgegeben sind.

Die Frage ist weiterhin, was die Grenzen dieser Assoziation sind. Wir betrachten den einfachsten Fall, ein Einschichtenperzeptron mit deterministischen Neuronen und der Vorzeichenfunktion als $g(h)$. Die Ausgabeneurone kann man isoliert betrachten, da sie untereinander nicht verbunden sind. Der Index i für das Ausgabeneuron kann dann fallengelassen werden und die Berechnung der Ausgabe wird zu

$$O = \text{sign}\left(\sum_k w_k \xi_k\right) = \text{sign}(\vec{w} \cdot \vec{\xi}) \quad (3.13)$$

Man kann die Gewichte auf das Ausgabeneuron und das Eingabemuster als Vektoren auffassen, da die Summation $\sum_k w_k \xi_k$ dem Skalarprodukt zweier N -dimensionaler Vektoren entspricht. Der Vektor der Gewichte steht im Raum der Eingabemuster senkrecht auf einer Hyperebene, die den Ursprung als Element enthält. Eingabemuster, die auf der gleichen Seite der Ebene wie der Gewichtsvektor liegen, liefern im Skalarprodukt mit dem Gewichtsvektor eine positive Zahl,

das Ergebnis wird also $O = +1$ sein. Liegen die Eingabemuster auf der abgewandten Seite, ist das Skalarprodukt negativ und $O = -1$. Wenn man also verschiedene Eingabemuster mit einer bestimmten Ausgabe des Ausgabeneurons verbinden möchte, dann muss es eine Hyperebene geben, die man im Raum der Muster so legen kann, dass alle diejenigen mit geforderter Ausgabe $\zeta = +1$ auf der einen Seite, und die mit Ausgabe $\zeta = -1$ auf der anderen Seite der Hyperebene liegen. Diese Eigenschaft ist die lineare Separabilität der Muster.

3.4 Lernregeln in künstlichen neuronalen Netzen

Für das Lernen in künstlichen neuronalen Netzen existieren eine Reihe von Vorschriften und Lernregeln (siehe z.B. [14]). Eine Klasse von Regeln werden unter “überwachtem Lernen” zusammengefasst. Hier gibt es eine äußere Instanz, die die Ausgabe des Netzes überwacht und beurteilt. Nach der Beurteilung werden die Gewichte verändert. Die zweite Klasse von Lernregeln werden als “unüberwachtes Lernen” bezeichnet. Hierbei hängt die Änderung der synaptischen Gewichte im Allgemeinen von der Aktivität der Neurone ab. Man muß außerdem unterscheiden, ob die Änderungen des synaptischen Gewichts nur von den beiden beteiligten Neuronen abhängt (Lokalität), oder von einem Signal, welches vom gesamten Netz erzeugt wird.

Das originale Postulat von Donald Hebb für die Veränderung von Synapsen macht eine Aussage über Kausalität: Ein Neuron A treibt wiederholt ein Neuron B zum Feuern, also muss das synaptische Gewicht von A nach B verstärkt werden. In künstlichen Netzen wird dieses Postulat häufig zu der Hebb’schen Lernregel abgekürzt, die besagt: Wenn zwei Neurone gleichzeitig unter dem Einfluß eines externen Stimulus aktiv sind, verstärke das Gewicht zwischen ihnen. Dabei geht die Kausalität verloren, und die Hebb’sche Lernregel wird zu einem Maß für Korrelationen. Ein Beispiel für die Anwendung der Lernregel ist in Abschnitt 3.3.2 gegeben. Allgemein werden aber Terme der Art $\Delta w_{ij} \propto S_i S_j$ als Hebbterme bezeichnet, also Änderungen der synaptischen Gewichte, die das Produkt der Aktivitäten prä- und postsynaptisch sind. Dementsprechend werden Terme mit umgekehrtem Vorzeichen als Anti-Hebbterme bezeichnet.

Um in theoretischen Modellen das Wachstum der Gewichte zu begrenzen, wurden verschiedene Methoden erdacht. Eine Möglichkeit ist die Normalisierung der Gewichte auf einen konstanten Wert (siehe z.B. [15]). Dies ist ein Beispiel für eine nicht-lokale Lernregel. Eine andere Möglich-

keit zeigt die BCM-Lernregel. Sie wurde in [6] vorgestellt, und vereint Hebb- und Antihebbterme in einer aktivitätsabhängigen Formulierung. Hierbei ist die Änderung der synaptischen Gewichte $\delta w_{ij} \propto \Phi(c_i)d_j$. c_i ist die postsynaptische, d_j die präsynaptische Aktivität. In die Funktion Φ geht zusätzlich eine Schwelle Θ_M ein, die ein gleitendes Mittel darstellt. Unterhalb der Schwelle Θ_M ist $\Phi(S_i)$ negativ, ansonsten positiv. Der Zusammenhang wird in Abb. 3.3 gezeigt.

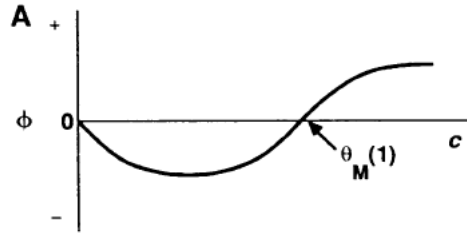


Abbildung 3.3: Die Funktion Φ in der BCM-Lernregel. Aus [6]

Die BCM-Regel besitzt neben der Lokalität noch eine weitere wünschenswerte Eigenschaft. Die Änderung der Synapse geht für eine konstante Aktivität gegen Null, da sich das gleitende Mittel bei einer konstanten Aktivität dieser annähert.

3.4.1 Die Perzeptronlernregel

Wenn eine Anzahl von linear separablen Mustern vorliegt, dann gibt es einen einfachen Lernalgorithmus, der für ein Einschichtenperzeptron inkrementell den trennenden Gewichtsvektor erzeugt. Die Muster ξ_k^μ werden der Reihe nach ins Perzeptron eingegeben. Ist die Ausgabe $O_i^\mu = \zeta_i^\mu$, dann ist das Ausgabe richtig und die Gewichte auf das Ausgabeneuron werden durch dieses Muster nicht geändert. Ist die Ausgabe jedoch nicht die gewünschte, werden die Gewichte in Richtung Ausgabeneuron geändert. Diese Vorschrift fasst man in der folgenden Regel zusammen:

$$\Delta w_k = \begin{cases} 2\eta\zeta^\mu\xi_k^\mu & \text{falls } O^\mu \neq \zeta^\mu \\ 0 & \text{sonst} \end{cases} \quad (3.14)$$

η ist die Lernrate, die klein gewählt wird. Wenn man sich auf diskrete Neuronen mit Werten $+1$ und -1 beschränkt, dann kann man die Lernregel alternativ formulieren:

$$\begin{aligned}\Delta w_k &= \eta(1 - \zeta^\mu O^\mu) \zeta^\mu \xi_k^\mu \quad \text{oder} \\ \Delta w_k &= \eta(\zeta^\mu - O^\mu) \xi_k^\mu\end{aligned}\tag{3.15}$$

Das Perzeptron kann aber noch mehr als Paare von Eingabe- und Ausgabemustern zu assoziieren. Man kann zum Beispiel aus der Gesamtmenge der zu assoziierenden Muster eine Trainingsmenge entnehmen und das Perzeptron nach dem Lernen zum Klassifizieren der Eingabemuster verwenden. Um die Klassifizierung zuverlässiger zu machen, fügt man in die Lernregel dann eine Schwelle ein:

$$\Delta w_k = \eta \Theta(N\kappa - h^\mu \zeta^\mu) \zeta^\mu \xi_k^\mu\tag{3.16}$$

wobei $h^\mu = \sum_k w_k \xi_k^\mu$ die Eingabe auf das betrachtete Ausgabeneuron, und Θ wieder die Stufenfunktion ist. In der Lernregel (3.14) wird nur das Vorzeichen des Skalarprodukts mit der gewünschten Ausgabe verglichen. In der erweiterten Lernregel muss das gerichtete Skalarprodukt $h_i \xi_i$ dazu noch um den Betrag $N\kappa$ größer sein als der Abstand des Musters von der trennenden Hyperebene. Das bedeutet folgendes: Wenn man das Muster verwechselt, dann wird die Eingabe h'_i unter dem verwechselten Muster um den Wert h_i (unverwechselt) schwanken. Durch den Mindestabstand des Musters von der Hyperebene $N\kappa$ ist die Klassifikation jedoch resistent gegen ein bestimmtes Ausmaß an Abweichung. Die Schwelle erhöht also die Zuverlässigkeit der Klassifikation.

3.4.2 Die Perzeptronlernregel in einem rekurrentem Netz

In den Überlegungen, die zu der Lernregel (3.16) geführt haben, wurde von den Ausgabeneuronen eines Perzeptrons nur ein einziges betrachtet, die Ergebnisse sind jedoch allgemein gültig. Nun kann man fragen, was passiert, wenn man diese Lernregel in einem rekurrenten Netzwerk aus N Neuronen anwendet. Für das Lernen wird ein Neuron i des Netzes als Ausgabeneuron eines Perzeptrons mit $N - 1$ Eingabeneuronen betrachtet. Das Eingabemuster ist ξ_j^μ für alle $j \neq i$, und der zugehörige O_i Zustand des Neurons i ist ξ_i^μ . Mit diesen Zuordnungen kann man leicht eine Lernregel formulieren:

$$\Delta w_{ij}^\mu = \eta \Theta(N\kappa - h_i^\mu \xi_i^\mu) \xi_i^\mu \xi_j^\mu\tag{3.17}$$

mit $h_i^\mu = \sum_j w_{ij} \xi_j^\mu$. Um ein Muster $\vec{\xi}$ zu Lernen, berechnet man für alle Neurone h_i und vergleicht $h_i \xi_i$ mit der Schwelle $N\kappa$. Ist $h_i \xi_i$ größer als die Schwelle, bleiben die Gewichte auf das Neuron i gleich. Im anderen Fall wird auf jedes Gewicht ein inkrementeller Hebbterm ($\eta \xi_i \xi_j$) aufaddiert. Diese Vorschrift wird für alle Muster so oft durchgeführt, bis alle Eingaben die Schwelle erreicht oder überschritten haben. Die Gewichte werden durch die Lernregel im Allgemeinen assymetrisch werden, da die Bedingung in der Stufenfunktion nicht symmetrisch ist, daher $\Delta w_{ij} \neq \Delta w_{ji}$.

Das so gelernte Muster ist ein Fixpunkt der Dynamik des Netzes. Die Anhaltebedingung für das Lernen ist

$$h_i \xi_i \geq N\kappa \quad (3.18)$$

Es gilt also mindestens (bei Gleichheit) für jedes Neuron im Muster $\vec{\xi}$

$$S_i = \text{sign}(h_i) = \text{sign}(N\kappa \xi_i) = \text{sign}\left(\sum_j w_{ij} \xi_j\right) \quad (3.19)$$

Damit ist das Muster stabil. Die Schwelle in der Lernregel sorgt dafür, dass das Muster auch ein attraktiver Fixpunkt ist, denn sie gewährleistet, dass solange weniger als eine gewisse Anzahl Neurone “umgeklappt” sind das Netz beim Updaten wieder in das Muster hineinfließt. Es seien j Neurone im Zustand des Musters, und j' Neurone mit umgekehrtem Vorzeichen, $S_j = \xi_j$ und $S_{j'} = -\xi_{j'}$. Dann ist die Eingabe

$$\begin{aligned} h_i &= \sum_j w_{ij} S_j + \sum_{j'} w_{ij'} S_{j'} = \sum_j w_{ij} \xi_j - \sum_{j'} w_{ij'} \xi_{j'} \\ &= \sum_k w_{ik} \xi_k - 2 \sum_{j'} w_{ij'} \xi_{j'} = N\kappa \xi_i - 2 \sum_{j'} w_{ij'} \xi_{j'} \end{aligned} \quad (3.20)$$

Solange also $N\kappa \xi_i > 2 \sum_{j'} w_{ij'} \xi_{j'}$ gegeben ist, bleibt $\text{sign}(h_i) = \xi_i$.

Diese Verwendung der Perzeptronlernregel wurde 1987 von Diederich und Opper vorgeschlagen (siehe [16]). Dabei beschränkten sie sich nicht auf ein Muster, sondern zeigten, dass man eine Anzahl von Mustern gleichzeitig mit dem Algorithmus lernen kann und dass dieser in einer endlichen Anzahl von Schritten aufhört.

3.4.3 Fehlerfunktion

Um eine Lernregel in Netzen mit kontinuierlichen Neuronenwerten herzuleiten, kann man eine Fehlerfunktion (oder auch Objective Function) definieren, die minimiert werden soll. Die richtigen

Gewichte werden dann über einen Gradientenabstieg in den Gewichten bestimmt. Eine mögliche Wahl für eine solche Fehlerfunktion ist

$$E = \frac{1}{2} \sum_{\mu} (\xi_i^{\mu} - x_i^{\mu})^2 \quad (3.21)$$

wobei $x_i^{\mu} = g(\sum_j w_{ij} \xi_j^{\mu})$ der Zustand des Neurons i nach einem Schritt der Dynamik bei Eingabe des Musters $\vec{\xi}^{\mu}$ ist. Diesen Ausdruck leitet man nach w_{ij} ab:

$$\frac{\partial E}{\partial w_{ij}} = \sum_{\mu} (\xi_i - x_i) \cdot g'(\sum_j w_{ij} \xi_j^{\mu}) \cdot \xi_j^{\mu} \quad (3.22)$$

$g'(h)$ ist die Ableitung der Übertragungsfunktion g . Man setzt nun $\partial E / \partial w_{ij} = \Delta w_{ij}$ und bekommt so eine Vorschrift für die Änderung der Gewichte. Die genaue Form der Gewichtsänderung hängt natürlich von der Übertragungsfunktion g ab.

3.5 Umrechnung in ein Ratenmodell mit rein positiven Zuständen

Die bisher betrachteten Neuronen haben den Nachteil, dass sie schlecht mit dem biologischen Neuron in Bezug zu setzen sind. Andere Neuronen im Gehirn sehen nur die Aktionspotentiale, kontinuierlich belegbare Zustände werden daher normalerweise mit Raten identifiziert. Die bisher betrachteten Neuronen können Zustände im Intervall $[-1, +1]$ besetzen, allerdings ist nicht klar, was es bedeutet, wenn ein Neuron im Zustand -1 ist. Eine Möglichkeit das Problem zu beheben ist es, den Wert Null als Spontanaktivität zu bezeichnen, und Zustände kleiner als Null eine Unterdrückung dieser spontanen Spikes bedeuten zu lassen.

Eine weitere Abweichung vom biologischen Vorbild ist die Tatsache, dass ein Neuron zu anderen Neuronen sowohl positive als auch negative synaptische Gewichte besitzen kann. Im Gehirn wirkt ein Neuron entweder nur exitatorisch oder inhibitorisch auf andere Neurone. Diese beiden Mängel kann man beheben, indem man das Modell auf eines mit einem rein exitatorischen Netz und einer globalen Inhibition umrechnet. Das entspricht durchaus vorhandenen Strukturen im Gehirn. Schematisch wird das Netz in Abb. 3.4 gezeigt.

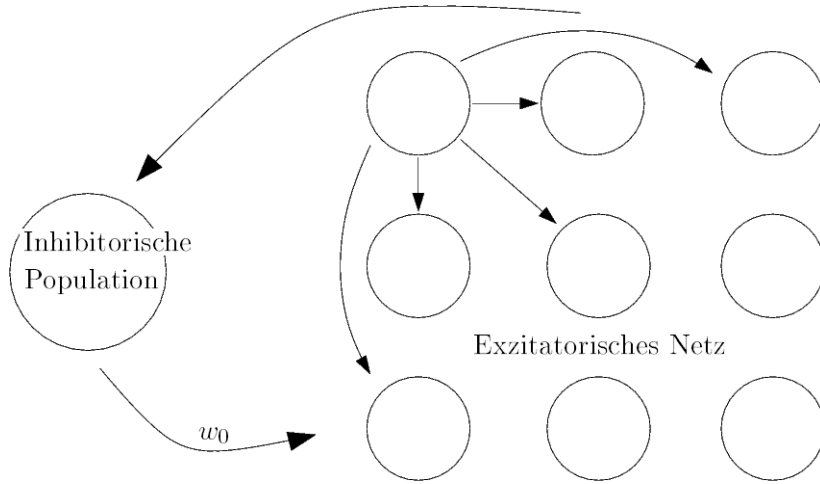


Abbildung 3.4: Die Struktur des Netzes aus exzitatorischen Neuronen mit globaler Inhibition

Die exzitatorischen Neurone regen sich gegenseitig an. Außerdem treiben sie gleichmäßig die inhibitorische Population, welche durch ein lineares Neuron repräsentiert wird. Das inhibitorische Neuron hemmt außerdem alle exzitatorischen Neurone gleich mit Gewicht w_0 . Es resultiert die Dynamik

$$\sigma_i = g^+(h_i^+) = g^+ \left(\sum_j w_{ij}^+ \sigma_j - w_0 \sum_j \sigma_j \right) + \Theta_i = g^+ \left(\sum_j (w_{ij}^+ - w_0) \sigma_j \right) \quad (3.23)$$

Θ_i ist hierbei eine Schwelle. Zur Unterscheidung wird der Zustand eines exzitatorischen Neurons i mit σ_i , die rein positiven Gewichte mit w_{ij}^+ und die Übertragungsfunktion mit g^+ bezeichnet. Die Übertragungsfunktion g^+ kann zum Beispiel die Stufenfunktion Θ (für diskrete Neurone) oder die logistische Funktion sein (für kontinuierliche Neurone). Die Umrechnung der Zustände vom einen ins andere Modell sind dann

$$\begin{aligned} \sigma_i &= \frac{1}{2}(x_i + 1) , & x_i &= 2\sigma_i - 1 \\ g^+ &= \frac{1}{2}(g + 1) , & g &= 2g^+ - 1 \end{aligned} \quad (3.24)$$

Um die gleichen Attraktoren in den beiden Netzen zu erhalten, wird $h_i^+ = h_i$ gefordert, d.h. die Eingaben im Modell mit rein positiven Neuronen sollen den Eingaben im Modell mit $-1/+1$ -

Neuronen entsprechen:

$$\begin{aligned}
\sum_j (w_{ij}^+ - w_0) \sigma_j + \Theta_i &= \sum_j w_{ij} x_j \\
&= \sum_j w_{ij} (2\sigma_j - 1) \\
&= \sum_j 2w_{ij} \sigma_j - \sum_j w_{ji}
\end{aligned} \tag{3.25}$$

Wenn man das absolute Minimum der Matrix w mit $\min(w)$ bezeichnet, dann erfüllen die Zuweisungen

$$\begin{aligned}
\Theta_i &= - \sum_j w_{ij} \\
w_{ij}^+ &= 2w_{ij} - 2\min(w) \\
w_0 &= -2\min(w)
\end{aligned} \tag{3.26}$$

die Forderung nach Gleichheit. Es ist also möglich, das in dieser Arbeit verwendete Modell auf eines mit rein positiven Neuronen abzubilden.

Eine direkte Umrechnung der Lernregel $\Delta w_{ij}^\mu \propto (\xi_i^\mu - x_i^\mu) \xi_j^\mu$ auf positive Neurone ergibt

$$\Delta w_{ij}^+ \propto (\xi_i^{\mu'} - \sigma_i^{\mu'}) (2\xi_i^{\mu'} - 1) \tag{3.27}$$

Die Umrechnung führt also eine Schwelle für die präsynaptische Aktivität ein.

Kapitel 4

Die Simulationen

4.1 Die Modelle

In dieser Arbeit wird das Lernen in zwei unterschiedlichen, aber verwandten Typen von künstlichen neuronalen Netzen untersucht. Das erste Netz ist das diskrete Netz mit Neuronen, die die beiden Zustände -1 und $+1$ einnehmen können. Der verwendete Lernalgorithmus ist die in Abschnitt 3.4.2 vorgestellte Gleichung (3.17). Das zweite Netz ist die Mean Field-Version des diskreten Netzes (Abschnitt 3.2.2). Prinzipiell führt die Verwendung der Perzeptronlernregel in diesem Netz dazu, dass die Muster auch unter der gemittelten Dynamik stabil bleiben. Die Motivation ist aber, die aus einer Fehlerfunktion gewonnene Vorschrift (3.22) zu untersuchen, die durch die Wahl der Übertragungsfunktion der Perzeptronlernregel entspricht.

4.2 Die Muster

Mit "Muster" werden Zustände des Netzes bezeichnet, in die das Netz von außen gesetzt wird. Sie haben keine weitere Bedeutung im Sinne von Bildern. Die Musterelemente sind immer entweder -1 oder $+1$, auch für das kontinuierliche Netz, in dem die Neurone das gesamte Intervall besetzen können. Für die meisten Simulationen sind die Muster unkorreliert, d.h. die Wahrscheinlichkeit, dass ein Musterelement ξ_i^μ einen Wert hat, ist $1/2$ und hängt nicht von anderen Musterelementen des gleichen oder anderer Muster ab. Korrelierte Muster werden in Kapitel 8 betrachtet.

4.3 Die Lernprotokolle

Die Muster wurden in beiden Netzen nach zwei unterschiedlichen Protokollen gelernt. Das erste Protokoll ist das Batch-Lernen. Die zu Lernenden Muster werden nacheinander präsentiert, das bedeutet, dass die Zustände der einzelnen Neuronen auf die zum Muster gehörenden Werte gesetzt werden. Bei jeder Präsentation wird festgestellt, ob die Gewichte nach der jeweiligen Vorschrift angepasst werden müssen. Dies ist dann der Fall, wenn nach einer Iteration ein Neuron sich nicht mehr auf dem zum Muster gehörigen Zustand befindet. Wenn nach einem Durchlauf aller Muster kein Gewicht mehr verändert werden muss, bricht der Algorithmus ab.

Beim Batch-Lernen ist es möglich, eine Kapazitätsgrenze zu erreichen. Es kann gezeigt werden [14], dass die maximale Kapazität eines rekurrenten Netzes für unkorrelierte Muster bei $2N$ liegt. Eine Motivation dieser Arbeit ist es, zu bestimmen, ob der Lernalgorithmus dieser Grenze nahekommt, und ob es Parameter gibt, die die Kapazität beeinflussen.

Das zweite Protokoll ist das Online-Lernen. Hierbei wird nur ein Muster präsentiert, und zwar immer wieder, bis es stabil ist. Wenn das der Fall ist, wird zum nächsten Muster übergegangen. Hierbei gibt es keine "Obergrenze" der Kapazität wie im Batch-Fall, da ein einzelnes Muster immer stabilisiert werden kann, egal wie die Gewichte vorher aussehen. Die Erwartung ist hierbei, dass ein einmal gelerntes Muster nach einer bestimmten Anzahl darauf eingepprägter Muster kein Attraktor mehr ist.

4.4 Bestimmung der Kapazität; Überlapp

Um zu bestimmen, ob ein Muster gelernt wurde oder nicht, wird der Überlapp des Musters mit dem zugehörigen mittleren Zustand des Netzes bestimmt. Das heißt, man gibt das Muster in das Netz ein und startet die Dynamik. Nach ausreichend vielen Schritten wird sich ein stationärer mittlerer Zustand des Netzes eingestellt haben. Ist das Muster ein Attraktor, bleibt das Netz natürlich für alle Zeit in dem gleichen Zustand. Andernfalls wird der Netzzustand im Allgemeinen in jedem Zeitschritt ein anderer sein, falls es nicht in einen anderen Attraktor läuft.

Der Überlapp m zweier beliebiger Zustände ξ^μ und ξ^ν ist ihr Korrelationskoeffizient:

$$\rho(\xi^\mu, \xi^\nu) = \frac{\langle \xi^\mu \xi^\nu \rangle - \langle \xi^\mu \rangle \langle \xi^\nu \rangle}{\sigma_{\xi^\mu} \sigma_{\xi^\nu}} \quad (4.1)$$

Für die betrachteten Muster nimmt er eine besonders einfache Form an. Die Mittelwerte $\langle \xi^\mu \rangle$ und $\langle x(t) \rangle$ sind gleich Null, da $P(\xi_i^\mu) = 0.5$ ist und nicht von anderen Elementen abhängt. Das gleiche wird für $x(t)$ angenommen. Die Standardabweichungen σ_ξ und $\sigma_{x(t)}$ werden damit zu Eins. Wenn man nun den Überlapp über die Anzahl der Schritte K mittelt, dann bleibt übrig

$$\langle m \rangle^\mu = \frac{1}{KN} \sum_{t=1}^K \sum_{i=1}^N \xi_i x_i(t) \quad (4.2)$$

Obwohl es nicht ganz richtig ist, wurde diese Gleichung auch für das kontinuierliche Netz übernommen.

Um die Kapazität zu bestimmen, muss nach dem Durchlauf des Lernalgorithmus überprüft werden, ob die gelernten Muster auch stabile Fixpunkte, oder noch besser Attraktoren sind. Im Batch-Protokoll ist es im Prinzip einfach, dies nachzuweisen: Ist der Algorithmus fertig, sind alle Muster gelernt. Tatsächlich aber wird eine Überschreitung der Kapazität dazu führen, dass die Gewichte nicht konvergieren, und in diesem Fall wird der Algorithmus nie abbrechen. Daher werden die Schritte gezählt und nach einer willkürlich festgelegten Grenze abgebrochen. Danach werden die Muster der Reihe nach in das Netzwerk eingegeben, und die Dynamik gestartet. Das Netz läuft eine bestimmte Anzahl von Schritten ohne Aufzeichnung. Danach wird eine weitere feste Anzahl von Schritten iteriert und bei jedem Schritt der Überlapp des eingegebenen Musters mit dem derzeitigen Zustand des Netzes bestimmt. Der endgültige Überlapp ist die Mittelung über die Schritte. Im Batch-Fall interessiert es allerdings nicht, ob ein bestimmtes Muster gelernt wurde, daher wurde ein weiteres Mal über alle μ gemittelt: $\langle m \rangle = 1/P \sum_\mu \langle m \rangle^\mu$.

Im Falle des Online-Lernens geht man ähnlich vor: Die Muster werden nacheinander gelernt. Nach dem letzten Muster werden sie in umgekehrter Reihenfolge ins Netz gegeben und jeweils der mittlere Überlapp bestimmt. Es interessiert anders als beim Batch-Lernen nicht die Mittelung über alle Muster einer Simulation, vielmehr ist der mittlere Überlapp des Musters, das n Lernschritte vor dem letzten Muster gelernt wurde wichtig. Hierzu wird über eine Anzahl von Simulationen der Überlapp des n -ten Musters gemittelt.

Kapitel 5

Simulationen des diskreten Netzes

Das diskrete Netz besteht aus N Neuronen mit möglichen Zuständen -1 und $+1$. Das Netz ist rekurrent mit asymmetrischen Verbindungsgewichten, das bedeutet dass jedes Neuron Verbindungen von und zu allen anderen Neuronen besitzt, wobei im Allgemeinen gilt $w_{ij} \neq w_{ji}$. Die Schwellen aller Neurone sind auf den Wert Null gesetzt, so dass das Netz der Dynamik

$$S(t+1) = \text{sign}(h_i) = \text{sign}\left(\sum_j w_{ij} S_j(t)\right)$$

gehört.

5.1 Batch-Lernen

Um eine eventuelle Abhängigkeit der Kapazität zu erkennen, wurden Netze aus 50, 100, 200 und 300 Neuronen betrachtet. Für jedes Netz wurden $P = N \dots 2N$ Muster erzeugt, wobei für jedes Element eines Musters $\vec{\xi}^\mu$ gilt: $P(\xi_i^\mu = 1) = 0.5$. Die Muster sind daher unkorreliert, so dass für zwei beliebige Muster μ und ν gilt: $\sum_i \xi_i^\mu \xi_i^\nu \approx 0$.

5.1.1 Der Lernalgorithmus

Die P Muster werden dem Netzwerk nacheinander präsentiert. Zu jedem Muster werden die gerichteten Eingaben $\xi_i^\mu h_i^\mu = \xi_i^\mu \sum_j w_{ij} \xi_j^\mu$ bestimmt, und mit der Schwelle $\kappa(N-1)$ verglichen. Ist $h_i^\mu \xi_i^\mu < \kappa(N-1)$, wird auf die Gewichte w_{ij} ein Hebbterm $\eta \xi_i^\mu \xi_j^\mu$ aufaddiert, wobei η die Lernrate ist. Sie ist für alle Simulationen auf $\eta = 0,01$ gesetzt. Dieser Algorithmus entspricht der Lernregel

(3.17). Die Konvergenz der Lernregel wird mit der Summe der Gewichtsänderungen in jedem Schritt überprüft. Ist die Summe $\sum_{\mu} \sum_{ij} |\Delta w_{ij}^{\mu}| = 0$, dann bleiben die Gewichte unverändert und die Muster sind alle stabile Fixpunkte der Dynamik. Da die Möglichkeit besteht, dass die Gewichte bei Kapazitätsüberschreitung nicht konvergieren, werden die Schritte gezählt und bei zu langem Lauf abgebrochen.

Ergebnis der Simulation

Es wurde für jedes N über 20 Läufe gemittelt, der Schwellenparameter ist für alle Netze $\kappa = 1$. Die Kapazität im Batch-Protokoll wird in Abb. 5.1 gezeigt. In der Graphik sind auf der Abzisse die Anzahl der Muster, normiert mit der Anzahl der Neuronen des Netzes abgetragen. Zu jeder Anzahl Muster $[P/N]$ wird der zugehörige gemittelte Überlapp dargestellt.

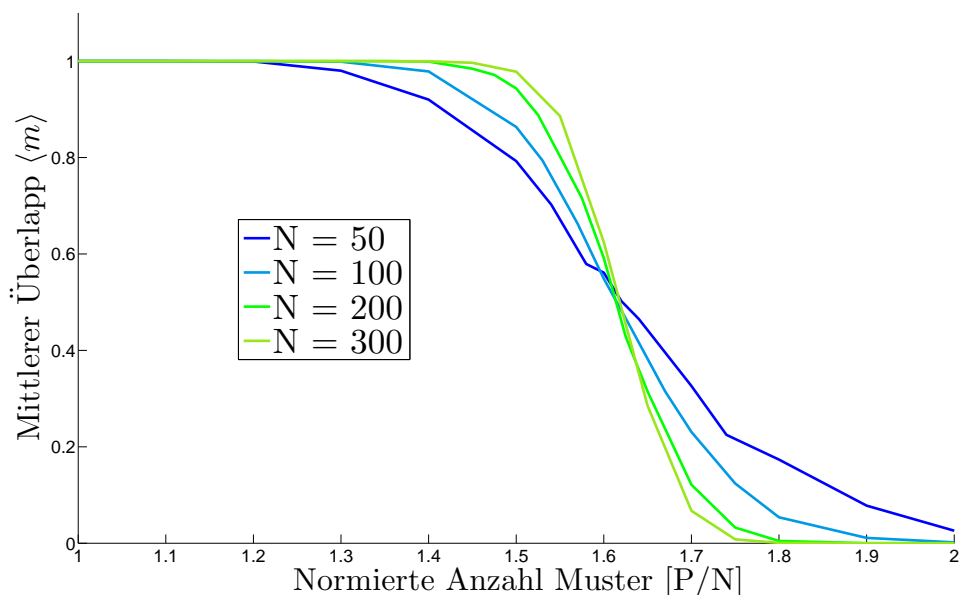


Abbildung 5.1: Die Kapazität des diskreten Netzes mit unkorrelierten Mustern im Batch-Fall

Die Kapazität des Netzes liegt bei $1,5N$ bis $1,6N$, je nachdem, ob man "Überreste" der Muster noch als Gedächtnis bezeichnen will. Ein mittlerer Überlapp von 0,5 legt nahe, dass zumindest ein Teil der Muster stabil ist oder sich Attraktoren gebildet haben, die den zu speichernden Mustern ähnlich sind. Die Kurven verlaufen mit steigender Anzahl der Neurone des

Netzes immer steiler, was nahelegt, dass das Fehlen der scharfen Grenze der Kapazität durch Randeffekte, bedingt durch die endliche (kleine) Anzahl der Neuronen bewirkt wird. Interessanterweise schneiden sich alle Kurven bei $P/N \approx 1,6$, so dass die Kapazität eines sehr großen Netzes diesem Wert nahekommen sollte.

5.1.2 Abhängigkeit der Kapazität von der Schwelle

Die Schwelle ist ein Mittel, um den Attraktionsbereich um das Muster herum zu vergrößern. Im Bild des Perzeptrons ist die Schwelle der minimale Abstand aller gelernten Muster von der trennenden Hyperebene, sie ermöglicht es also, auch verrauschte Muster richtig zu erkennen. In einem rekurrenten Netz mit nach der Perzeptronregel gelernten Muster beeinflusst die Schwelle die Stabilität des Musters (siehe Abschnitt 3.4.2), und damit die Größe des Attraktionsbereichs im Zustandsraum. Man kann daher vermuten, dass sie einen Einfluß auf die Kapazität des Netzes hat.

Ergebnis der Simulation

Um den Einfluß der Schwelle zu bestimmen, wurde für $N = 300$ die Schwelle κ von 0 bis 0,8 variiert. In der Graphik auf der Abzisse wieder die mit N normierte Anzahl der zu speichernden Muster abgetragen, und die dazugehörige mittlere Überlapp $\langle m \rangle$ auf der Ordinate. Die Mittelung fand über 10 Simulationen je N statt.

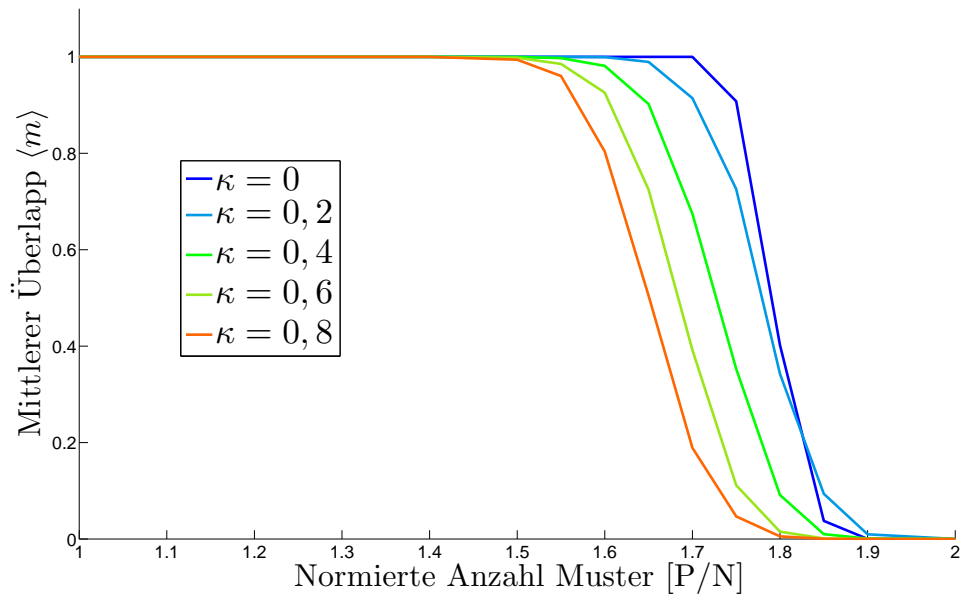


Abbildung 5.2: Kapazität im diskreten Netz in Abhängigkeit von der Schwelle κ

Die Schwelle hat einen Einfluß auf die Kapazität. Je kleiner die Schwelle, desto weiter kommt die Kapazität an einen Wert heran, der etwas höher als $P/N = 1,8$ liegt ($\kappa = 0$).

Kapitel 6

Simulation des kontinuierlichen Netzes

Das kontinuierliche Netz ist eine Mean Field-Version des diskreten Netzes, d.h. die Neurone können Zustände im Intervall $[-1, +1]$ einnehmen. Die Werte in diesem Intervall werden mit Feuerraten identifiziert, wobei -1 keine Aktivität, und $+1$ maximale Feuerrate ist. Die Annahme ist biologisch begründet; durch die absolute Refraktärzeit (siehe Kap. 2) wird eine Mindestzeit zwischen den Spikes eingeführt. Die wesentliche Näherung in diesem speziellen Modell ist die Wahl der Übertragungsfunktion. Legt man für die Wahrscheinlichkeit eines Zustandes die logistische Funktion (3.4) zugrunde, dann ergibt die Mean Field-Näherung den Tangens Hyperbolicus als Übertragungsfunktion. Diesen kann man durch eine stückweise lineare Funktion annähern, wie in Abb. 6.1 gezeigt.

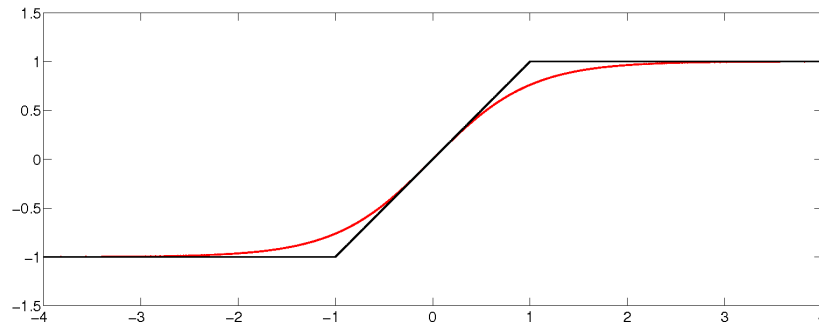


Abbildung 6.1: Der Tangens Hyperbolicus (rot) und die lineare Näherung (schwarz) für $\beta = 1$

Die Funktion ist dann definiert durch

$$g(h) = \begin{cases} +1 & \text{für } h > \frac{1}{\beta} \\ \beta h & \text{für } |h| < \frac{1}{\beta} \\ -1 & \text{für } h < -\frac{1}{\beta} \end{cases} \quad (6.1)$$

β ist der gleiche Parameter wie in Gln. (3.4). Diese spezielle Wahl der Übertragungsfunktion hat den Vorteil, dass eine Gewichtsänderung, wie sie die Einführung der Fehlerfunktion wie in Abschnitt 3.4.3 beschrieben erzeugt, der Perzeptronlernregel entspricht. Für den Lernalgorithmus ergibt diese Wahl zusätzlich den Vorteil, dass es eine definierte Abbruchbedingung für das Lernen gibt.

Die entsprechende Updateregeln ergibt sich dann aus Gln. (3.8), wenn der Tangens Hyperbolicus durch $g(h)$ ersetzt wird.

6.1 Die Lernregel aus der Fehlerfunktion

Nach Gln. (3.22) muss die Ableitung der Übertragungsfunktion in der Änderung der Gewichte verwendet werden. Die Ableitung der speziellen Wahl für g ergibt

$$\frac{\partial g(h)}{\partial h} = \begin{cases} 0 & \text{für } h > \frac{1}{\beta} \\ \beta & \text{für } |h| < \frac{1}{\beta} \\ 0 & \text{für } h < -\frac{1}{\beta} \end{cases} \quad (6.2)$$

Das bedeutet für die Gewichte, dass es nur für Eingaben, die zwischen $-1/\beta$ und $1/\beta$ liegen eine Änderung gibt:

$$\Delta w_{ij}^{\mu} = \eta' \beta (\xi_i^{\mu} - x_i^{\mu}) \xi_j^{\mu}, \quad \text{für } |h| < 1/\beta \quad (6.3)$$

mit der Lernrate η' . Dieser Ausdruck entspricht der Perzeptronlernregel (3.15), für kontinuierliche Neurone, wenn man die Lernrate $\eta = \eta' \beta$ setzt.

Im diskreten Netz ist die Bedingung, damit das Lernen aufhört, dass die gerichtete Eingabe $\xi_i h_i$ größer oder gleich der Schwelle $(N - 1)\kappa$ ist. Diese Wahl der Übertragungsfunktion führt ebenfalls eine "harte" Anhaltebedingung ein. Wenn die Ableitung der Übertragungsfunktion gleich Null ist, werden die Gewichte nicht geändert. Das hat die Funktion einer Schwelle, weil die

Ableitung der Übertragungsfunktion nur im Intervall $[-1/\beta, 1/\beta]$ ungleich von Null ist. Auch aufgrund dieser Eigenschaft steht zu erwarten, dass die Eigenschaften des Mean-Field-Netzes ähnlich denen des diskreten Netzes sind.

6.2 Batch-Lernen

Der Lernalgorithmus wurde analog zu dem diskreten Netz umgesetzt. Es wurden Netze aus 50, 100, 200 und 300 Neuronen erstellt, für die jeweils in verschiedenen Durchgängen P Muster erzeugt wurden, $P = N \dots 2N$. Diese P Muster werden dem Netz nacheinander präsentiert, und die Gewichte nach Vorschrift (6.3) geändert. Der Algorithmus bricht ab, wenn entweder die kumulierte Gewichtsänderung $\sum_{\mu, i, j} |\Delta w_{ij}^{\mu}|$ für einen Durchgang aller Muster kleiner als ein bestimmter Wert wird, oder die Anzahl der Schritte einen bestimmten Wert erreicht. In dem Fall wird angenommen, dass die Gewichte nicht konvergieren, also die Kapazität überschritten ist.

Es zeigte sich, dass der Lernalgorithmus im Mean-Field-Modell wesentlich länger brauchte als im diskreten Netz. Das liegt daran, dass in der Lernregel (6.3) der Term $(\xi_i^{\mu} - x_i^{\mu})$ immer kleiner wird, je näher die Neuronenzustände nach einer Iteration am Muster liegen. Daher war die Abbruchbedingung des Algorithmus, dass die kumulierte Gewichtsänderung nur kleiner als eine kleine Zahl ($\approx 10^{-7}$) wird. Um die Stabilität zu zeigen, wurde für die Bestimmung des Überlapp eine andere Steigung des nichtkonstanten Abschnitts von g benutzt: $\beta' = 2 \cdot \beta$.

Ergebnis der Simulation

Die Simulationen wurden ähnlich wie in Kapitel 5 durchgeführt. Es wurden zu jedem N $P = N \dots 2N$ unkorrelierte Muster erzeugt, die mit dem Algorithmus Gln. (6.3) inkrementell gelernt wurden. Die Lernrate η wurde gleich 0,01 gesetzt, die Steigung $\beta = 1/(N - 1)$. Die Netze sind 50, 100, 200 und 300 Neuronen groß, und die gezeigten Kurven sind Mittelungen über je 20 Läufe für jedes N .

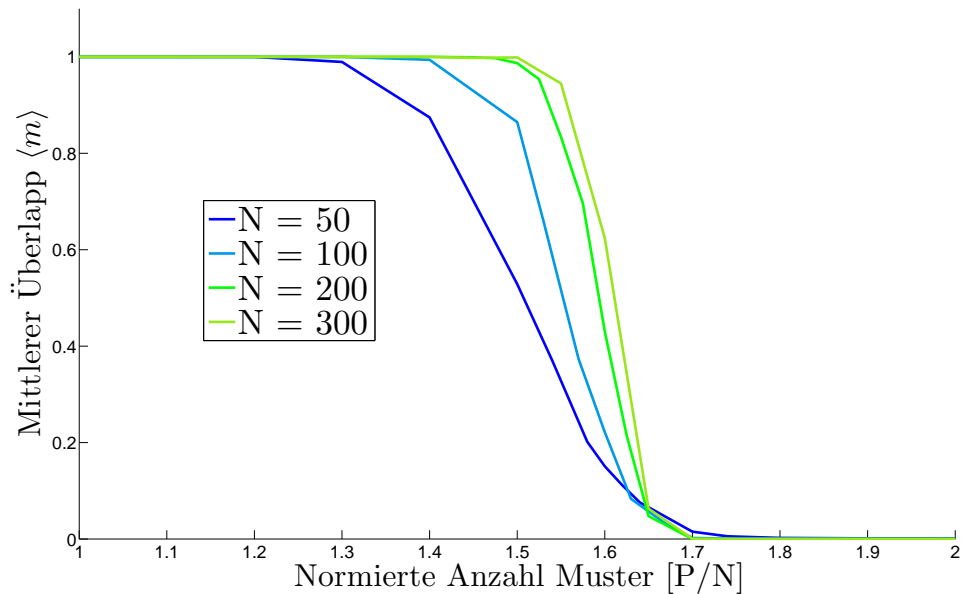


Abbildung 6.2: Die Kapazität des Mean Field-Netzes mit unkorrelierten Mustern

Auf der Ordinate ist die normierte Anzahl der Muster P/N aufgetragen, auf der Abzisse der gemittelte Überlapp für jede Anzahl Muster. Die Steigung β entspricht dem Kehrwert des im diskreten Netzes verwendeten Schwelle $(N - 1)\kappa$ mit $\kappa = 1$. Daher überrascht es nicht, dass die Kapazität auch im Bereich 1,5 - 1,6 liegt. Wie im diskreten Netz werden die Kurven mit steigendem N steiler. Auffällig ist außerdem, dass die Kapazität mit der Anzahl der Neurone N zu steigen scheint, bis sie sich einem Wert $\approx 1,65$ annähert.

Kapitel 7

Online-Lernen

Der zweite Modus des Lernens ist das Online-Lernen. Die Muster ξ^μ werden dem Netzwerk hierbei nacheinander präsentiert. Der Unterschied ist jetzt, dass das Muster solange immer wieder gezeigt wird, bis es stabil und die Eingaben $h_i \xi_i^\mu \geq \kappa(N-1)$ sind. Das bedeutet, dass der Lernalgorithmus (3.17) für nur ein μ durchgeführt und danach zum nächsten Muster übergegangen wird.

7.1 Lösung für die Änderung der Gewichte im diskreten Netz

Die Perzeptronlernregel angewendet auf ein Muster führt dazu, dass die Gewichte auf ein Neuron i hin alle um den gleichen Betrag $|\Delta w_{ij}|$ geändert werden. Variabel ist nur das Vorzeichen. In der Gln. (3.17) ist das Argument der Stufenfunktion $N\kappa - h_i \xi_i$. Die Bedingung für das Lernen ist also

$$h_i \xi_i < N\kappa$$

und hängt nur von i ab.

Wir betrachten jetzt ein Neuron, welches noch instabil ist. Es gibt nun für das Neuron eine Eingabe h'_i , für welche gilt

$$\xi_i h'_i = N\kappa \tag{7.1}$$

h'_i ist also die gesuchte Eingabe, bei der das Neuron stabil wird. Es gilt außerdem

$$h'_i = h_i + \Delta h_i$$

Damit erhalten wir den Ausdruck

$$\Delta h_i = N\kappa\xi_i - h_i \quad (7.2)$$

Weiterhin gilt für h'_i

$$h'_i = \sum_j (w'_{ij})\xi_j = \sum_j (w_{ij} + \Delta w_{ij}^{ges})\xi_j$$

wobei Δw_{ij}^{ges} die kumulierte Gewichtsänderung bis zum Erfüllen der Bedingung ist. Sie ergibt sich zu

$$\Delta w_{ij}^{ges} = \sum_{m=1}^M \Delta w_{ij}^m = \sum_{m=1}^M \eta \xi_i \xi_j = \xi_i \xi_j \sum_{m=1}^M \eta$$

wobei M die Anzahl der Lernschritte ist, bis das Neuron i unter dem Muster die Zieleingabe h'_i erreicht hat. Das Vorzeichen der Gewichtsänderung wird durch $\xi_i \xi_j$ festgelegt, und ihr Betrag durch $\sum_m^M \eta = |\Delta w_{ij}^{ges}|$. Damit haben wir die Möglichkeit, die Änderung der Eingabe Δh_i zu bestimmen.

$$\Delta h_i = \sum_j \Delta w_{ij}^{ges} \xi_j = |\Delta w_{ij}^{ges}| \sum_j \xi_i \xi_j \xi_j = |\Delta w_{ij}^{ges}| \xi_i \sum_j 1 = |\Delta w_{ij}^{ges}| \xi_i (N - 1)$$

Und der Betrag der gesamten Gewichtsänderung wird zu

$$|\Delta w_{ij}^{ges}| = \frac{N\kappa\xi_i - h_i}{\xi_i(N - 1)} \quad (7.3)$$

Das Ergebnis ist interessant, weil für die spezielle Wahl der Übertragungsfunktion in dieser Arbeit die Änderungen der Gewichte im diskreten Netz und dem Mean Field-Netz gleich sind (siehe Abschnitt 7.2).

Ergebnis der Simulation

Die Simulation wurde durchgeführt wie im vorherigen Kapitel beschrieben. Die Schwelle κ lag für alle Durchläufe bei 1, und es wurde über 200 Läufe gemittelt. Die Muster wurden wie in Kapitel 4 beschrieben nach dem Lernen in umgekehrter Reihenfolge ins Netz gegeben, und dann ihr Überlapp mit der mittleren Netzaktivität bestimmt. Daher ist in der Graphik das zuletzt gelernte Muster ganz links, dann kommt das vorletzte gelernte Muster usw. Es ist wieder die normierte Anzahl Muster P/N abgetragen, um etwaige Effekte der Anzahl der Neuronen sehen zu können.

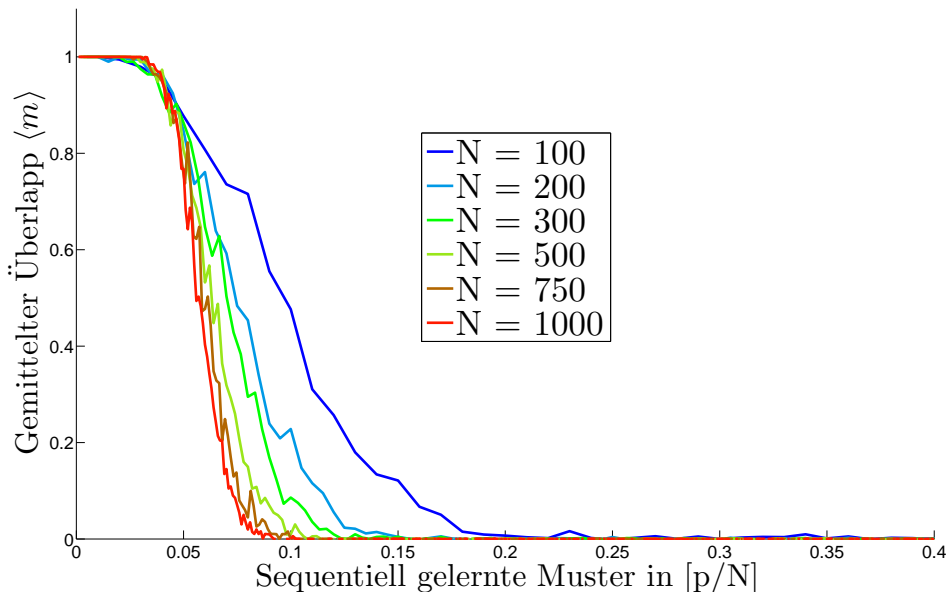


Abbildung 7.1: Die Kapazität des Netzes mit unkorrelierten Mustern im Online-Fall

Die Kapazität liegt bei 0,05, also wesentlich geringer als im Batch-Fall.

Wichtig ist dabei folgendes zu bemerken. Die Kapazität hing in den Simulationen auch davon ab, wieviele Muster schon gelernt wurden. Wenn die Kapazität nach wenigen (ca. $P \approx 1N$ Muster) überprüft wurde, lag sie bei 0,1, also höher als zu späteren Zeiten (nach $P \approx 10N$ Mustern). Die Gewichte wurden am Anfang gleichverteilt in dem Intervall $[-1, +1]$ initialisiert. Diese Verteilung ändert sich natürlich mit dem Lernen, und ist wahrscheinlich für die Veränderung der Kapazität verantwortlich.

7.2 Online-Lernen im Mean-Field-Netz

Auch beim Online-Lernen ist der Algorithmus bis auf die Lernregel identisch mit dem beim diskreten Netz verwendeten. Durch die spezielle Wahl der Übertragungsfunktion ergibt sich außerdem, dass auch die Änderung der Gewichte identisch ist bei beiden Modellen.

Um zu sehen, dass diese Gainfunktion zu den gleichen Gewichtsänderungen wie im Falle der Perzeptronlernregel führt, führt man die gleiche Rechnung durch wie im Abschnitt 7.1. Nach Gleichung 6.3 entscheidet die Differenz $\xi_i - \beta h_i$ darüber, ob die Gewichte geändert werden.

Betrachten wir nun ein Neuron i , das unter dem Muster $\vec{\xi}$ noch nicht stabil ist. Dann wird es eine Eingabe h'_i geben, bei der das Lernen aufhört.

$$\xi_i - \beta h'_i = 0 \quad (7.4)$$

h'_i ist wieder die Summe aus der Eingabe vor dem Lernen und der Änderung:

$$h'_i = h_i + \Delta h_i = \sum_j w_{ij} \xi_j + \sum_j \Delta w_{ij}^{ges} \xi_j \quad (7.5)$$

Die gesamte Änderung der Gewichte ist die Summe aus den inkrementellen Termen:

$$\Delta w_{ij}^{ges} = \sum_{m=1}^M \Delta w_{ij}^m \quad (7.6)$$

Solange das Neuron noch nicht stabil ist, ist $|\xi_i| > |\beta h_i|$, also

$$\text{sign}(\xi_i) = \text{sign}(\xi_i - \beta h_i) \quad (7.7)$$

Daher gilt

$$|\Delta w_{ij}| = \xi_i \cdot \eta(\xi_i - \beta h_i) \quad (7.8)$$

Damit kann man die Gesamtänderung der Gewichte berechnen.

$$\begin{aligned} \xi_i - \beta h_i &= \beta \Delta h_i \\ \frac{1}{\beta} - \xi_i h_i &= \xi_i \sum_j \sum_m \Delta w_{ij}^m \xi_j = \sum_j \sum_m |\Delta w_{ij}^m| \xi_j \xi_j \end{aligned} \quad (7.9)$$

Der Betrag der inkrementellen Gewichtsänderung auf ein Neuron i ist aber für alle j gleich, daher

$$\sum_j \sum_m |\Delta w_{ij}^m| \xi_j \xi_j = \sum_j |\Delta w_{ij}^{ges}| = (N-1) |\Delta w_{ij}^{ges}| \quad (7.10)$$

Und es ergibt sich für die Gesamtänderung

$$|\Delta w_{ij}^{ges}| = \frac{1/\beta - \xi_i h_i}{N-1} = \frac{\frac{1}{\beta} \xi_i - h_i}{\xi_i (N-1)} \quad (7.11)$$

Mit der Korrespondenz $N\kappa = 1/\beta$ ist das der gleiche Ausdruck wie Gln. (7.3). Daher wurde für den Mean-Field-Fall keine weitere Simulation durchgeführt, die Kapazität in Abschnitt 7.1 ist auch für dies Netz gültig.

7.3 Diffusion der Eingaben

Die Kapazität bei dem Online-Lernen ist im Vergleich zum Batch-Lernen gering. Es stellt sich die Frage nach dem Grund. Die Lernregel führt im Online-Lernen dazu, dass für das zuletzt gelernte Muster ξ^ν gilt:

$$\left(\sum_j w_{ij}\xi_j^\nu\right)\xi_i^\nu = h_i^\nu \xi_i^\nu \geq (N-1)\kappa \quad \text{für alle } i$$

Die Gewichte werden also soweit geändert, bis alle gerichteten Eingaben mindestens einem bestimmten Wert entsprechen. Im Batch-Modus führt die Lernregel zu einem Ausgleich, bis die Bedingung für alle Muster gilt, aber im Online-Modus verändern später gelernte Muster μ die Gewichte und damit die Eingaben, wenn das Netz in den Zustand ξ^ν gesetzt wird.

Die Gewichtsänderungen hängen nicht voneinander ab, da die Muster unkorreliert sind. Es wird jetzt die Eingabe h_i' auf das Neuron i betrachtet, nachdem das Muster ξ^ν und danach ein weiteres Muster ξ^μ gelernt wurden.

$$h_i' = \sum_j w_{ij}\xi_j^\nu + \sum_j \Delta w_{ij}\xi_j^\nu = h_i + \sum_j |\Delta w_{ij}|\xi_i^\mu \xi_j^\mu \xi_j^\nu \quad (7.12)$$

Der Ausdruck rechts ist das Produkt aus drei Musterelementen, die mit einer Wahrscheinlichkeit 0,5 +1 oder -1 sind, so dass die Änderung der Eingabe auf Neuron i bis auf die Größe $|\Delta w_{ij}|$ einer Art Random Walk unterliegt. Daher ist die Erwartung, dass die Eingaben beim Lernen weiterer Muster ungerichtet diffundieren.

Um das zu überprüfen, wurde ein Netz aus 1000 Neuronen simuliert, und über 200 Simulationen gemittelt. Nach $10N$ gelernten Mustern wurde eines festgehalten, und die Entwicklung der Eingaben mit der Anzahl der gelernten Muster festgehalten. Daraus ergeben sich die in Abb. 7.2 und 7.3 gezeigten Verteilungsdichten der Eingaben.

Die Verteilung wird breiter und nähert sich einer Normalverteilung an, während ihr Maximum sich zu kleineren Werten von $h_i^\nu \xi_i^\nu$ hin verschiebt.

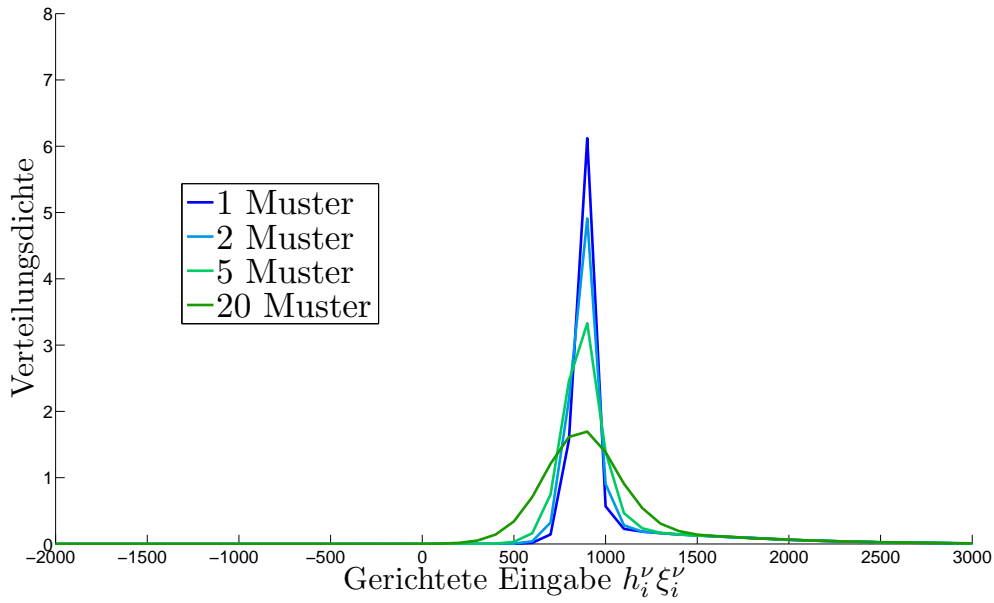


Abbildung 7.2: Die Verteilungsdichte der Eingaben $h_i^\nu \xi_i^\nu$ nach 1,2,5 und 20 nach ξ^ν gelernten Mustern

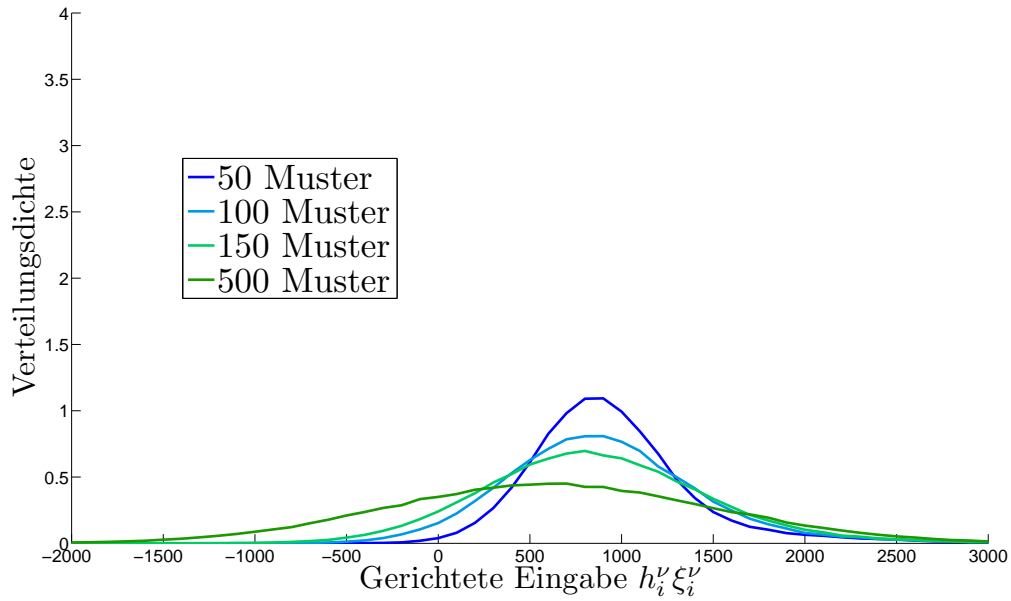


Abbildung 7.3: Die Verteilungsdichte der Eingaben $h_i^\nu \xi_i^\nu$ nach 50,100,150 und 500 nach ξ^ν gelernten Mustern

Kapitel 8

Korrelierte Muster

Korrelierte Muster sind solche, in denen zwischen den Musterelementen ein Zusammenhang besteht. Bisher wurden nur unkorrelierte Muster betrachtet. Hierbei ist die Wahrscheinlichkeit, dass ein Element ξ_i^μ eines Musters -1 oder $+1$ ist, unabhängig von den anderen Elementen i und anderen Mustern μ gleich $0,5$.

8.1 Lokal korrelierte Muster

Es gibt verschiedene Arten, wie man Eingabemuster miteinander korreliert. Lokal korrelierte Muster sind solche, bei denen zwischen den Mustern mit ähnlichem μ Korrelationen bestehen, die aber mit dem Abstand der Muster abfallen. Konkret wurden die Muster nacheinander erzeugt, in dem von einem Muster ν zum nächsten $\nu + 1$ im Mittel 10% der Elemente das Vorzeichen wechselten.

8.1.1 Online

Das Lernprotokoll ist wie in Abschnitt 7 beschrieben. Die Muster werden erzeugt, indem von einem Muster zum nächsten im Mittel 10% der Elemente das Vorzeichen wechseln.

Ergebnis der Simulation

Es wurde über 200 Simulationen gemittelt. Die Schwelle lag bei $N - 1$.

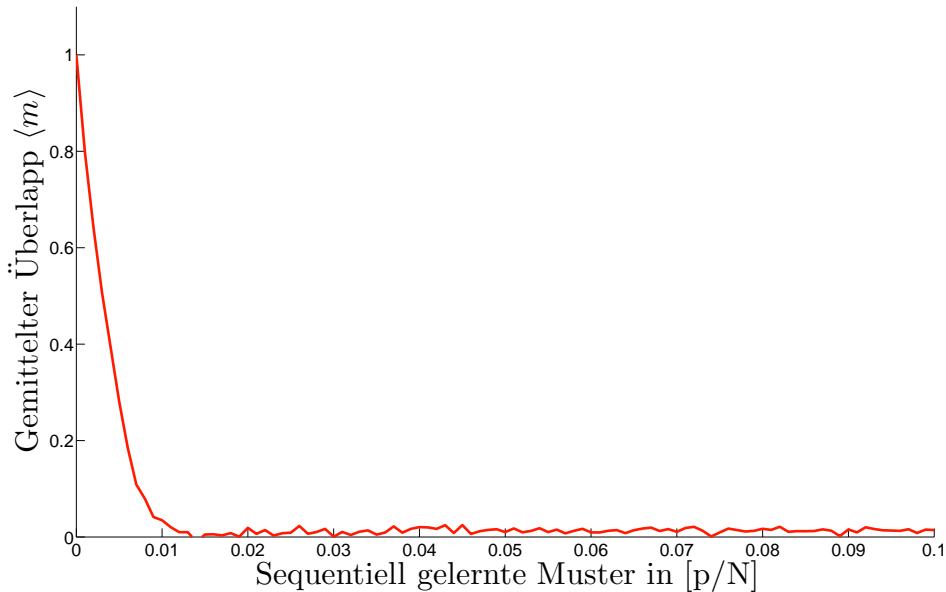


Abbildung 8.1: Die Kapazität im Online-Lernen mit stark korrelierten Mustern

Die Kapazität ist praktisch Null. Nur das zuletzt gelernte Muster ist stabil, schon das nächste Muster ist kein Fixpunkt mehr.

Diffusion der Eingaben

Auch in diesem Fall wurde die Diffusion der Eingaben $h_i^y \xi_i^y$ betrachtet, auf die gleiche Weise wie in Abschnitt 7.3 beschrieben. Das Netz besteht wieder aus 1000 Neuronen, und der Schwellparameter ist $\kappa = 1$. Die Ergebnisse der Simulationen werden in Abb. 8.2 und 8.3 gezeigt. Auffällig ist, dass schon nach zwei Mustern im Bereich $h_i^y \xi_i^y \approx -500$ eine Häufung zu erkennen ist. Dies wird im Ausschnitt Abb. 8.4 im Detail gezeigt.

Es gibt zwei Unterschiede zu den Ergebnissen bei unkorrelierten Mustern. Erstens sind die Eingaben bei korrelierten Mustern breiter verteilt (Skala der Abzisse), und zweitens bleibt das Maximum der Verteilung relativ konstant.

Die geringe Kapazität läßt diesen Befund erwarten. Denn ein negativer Wert für $h_i^y \xi_i^y$ bedeutet, dass das Vorzeichen der Eingabe entgegengesetzt zum Vorzeichen des Musterelements i ist, so dass dieses Element instabil geworden ist.

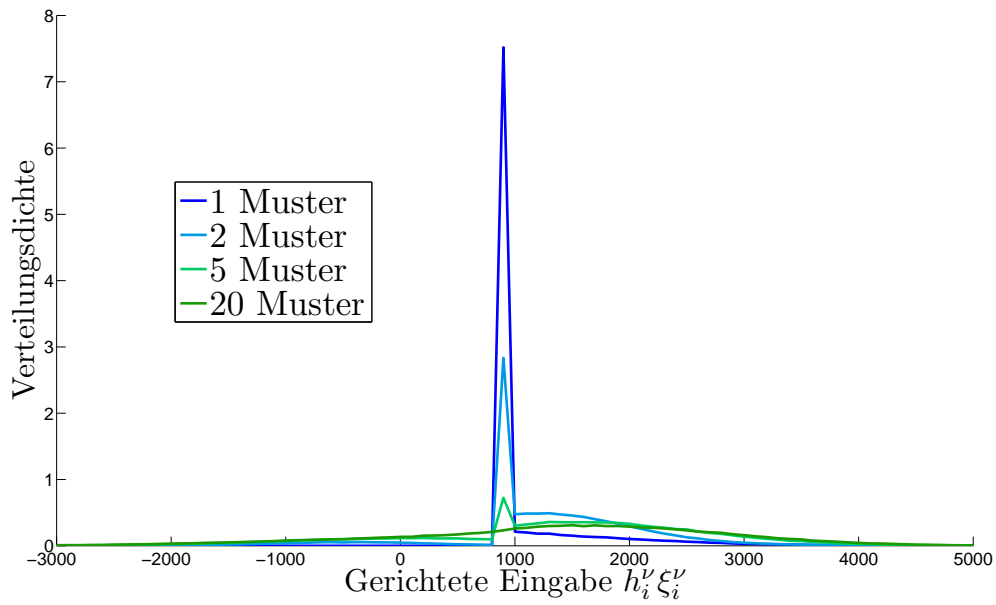


Abbildung 8.2: Die Verteilungsdichte der Eingaben $h_i^\nu \xi_i^\nu$ nach 1,2,5 und 20 nach ξ^ν gelernten Mustern

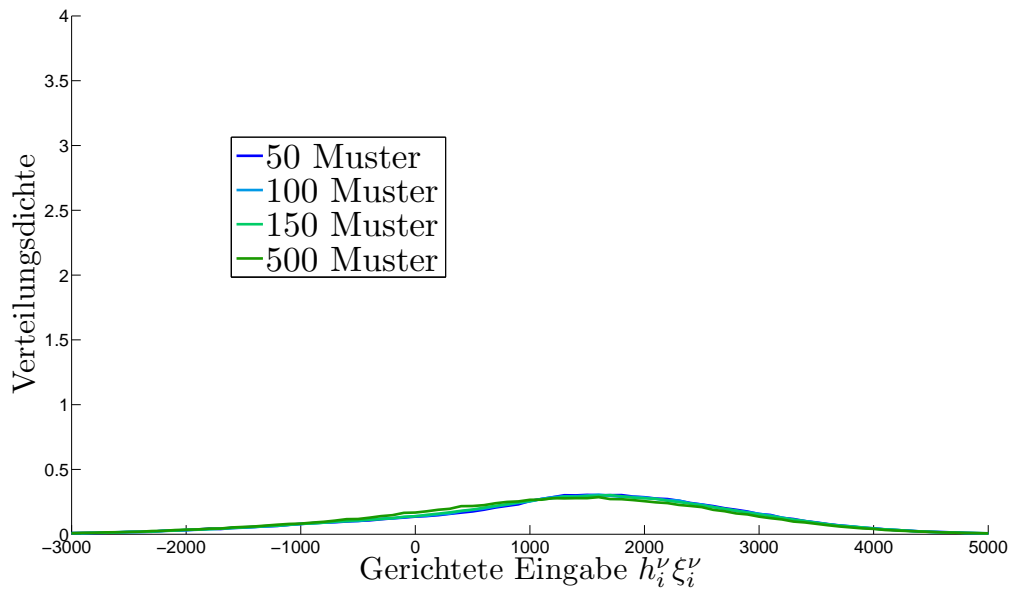


Abbildung 8.3: Die Verteilungsdichte der Eingaben $h_i^\nu \xi_i^\nu$ nach 50,100,150 und 500 nach ξ^ν gelernten Mustern

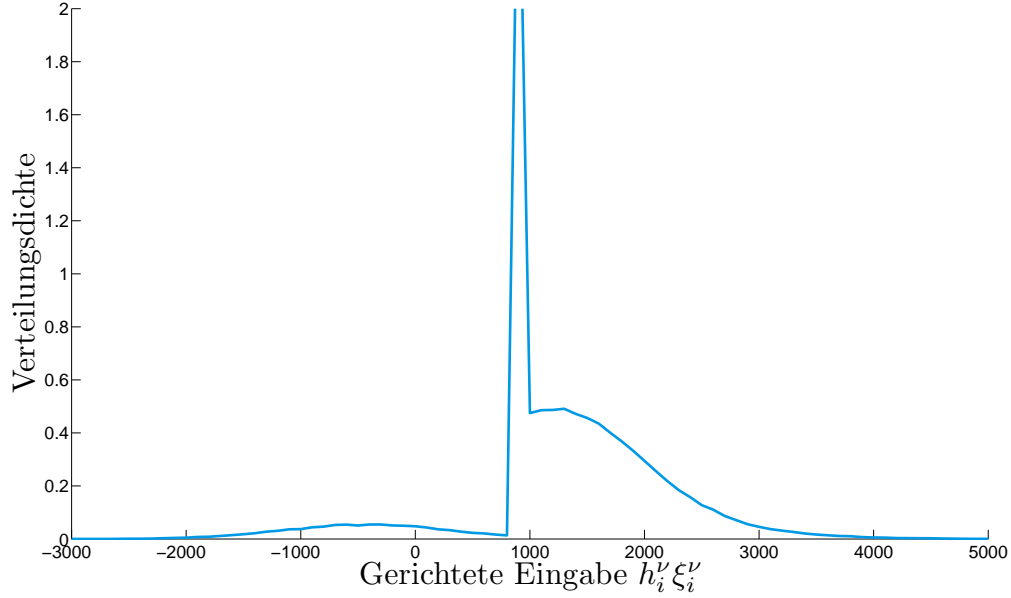


Abbildung 8.4: Die Verteilungsdichte der Eingaben $h_i^\nu \xi_i^\nu$ aus Abb. 8.2 für nach 2 gelernten Mustern in der Vergrößerung

Man kann zeigen, warum die Diffusion der Eingaben verstärkt ist. Betrachten wir zwei Muster ξ^ν und ξ^μ . Zuerst wird das Muster ν gelernt, so dass die Eingaben

$$h_i^\nu = \sum_j w_{ij} \xi_j^\nu$$

sind. Danach wird das Muster ξ^μ gelernt. Die Gewichte werden sich um einen Beitrag

$$\Delta w_{ij} = |\Delta w_{ij}| \xi_i^\mu \xi_j^\mu$$

ändern, wobei der Betrag der Gewichtsänderung durch Gln. (7.3) gegeben ist:

$$|\Delta w_{ij}| = \frac{N \kappa \xi_i^\mu - h_i^\nu}{\xi_i^\mu (N - 1)}$$

Nun betrachtet man die Eingabe h_i' auf das Neuron i nach dem Lernen von Muster ξ^μ :

$$h_i' = h_i^\nu + \sum_j \Delta w_{ij} \xi_j^\nu = h_i^\nu + |\Delta w_{ij}| \xi_i^\mu \sum_j \xi_j^\nu \xi_j^\mu$$

Wenn die beiden Muster korreliert sind, dann gilt folgende Bedingung: $\sum_i \xi_i^\nu \xi_i^\mu > 0$. Das bedeutet, dass die Änderung der Gewichte im Mittel gerichtet ist. Das erkennt man, indem beide

Seiten mit ξ_i^ν multipliziert werden:

$$h_i' \xi_i^\nu = h_i \xi_i^\nu + \xi_i^\mu \xi_i^\nu |\Delta w_{ij}| \sum_j \xi_j^\nu \xi_j^\mu$$

Bei den Elementen, die von einem zum nächsten Muster gleich geblieben sind, wird sich die gerichtete Eingabe unter dem vorher gelernten Muster zu positiven Werten hin verschieben, bei den Elementen mit verschiedenen Vorzeichen verschiebt sich die Eingabe im Mittel zu negativen Werten hin. Durch diese Richtung ergibt sich eine verstärkte Diffusion der Eingaben.

8.1.2 Batch

Auch hier wurden die Simulationen analog zu dem in Abschnitt 6.2 (also im Mean-Field-Fall) beschriebenen Algorithmus durchgeführt, die Muster sind wie oben beschrieben lokal korreliert. Die Anzahl der Muster lag hier aber höher, wie zu erkennen wurden Muster bis zu einer Anzahl $4N$ versucht zu Lernen. Das Ergebnis ist in Abb. 8.5 zu Sehen.

Ergebnis der Simulation

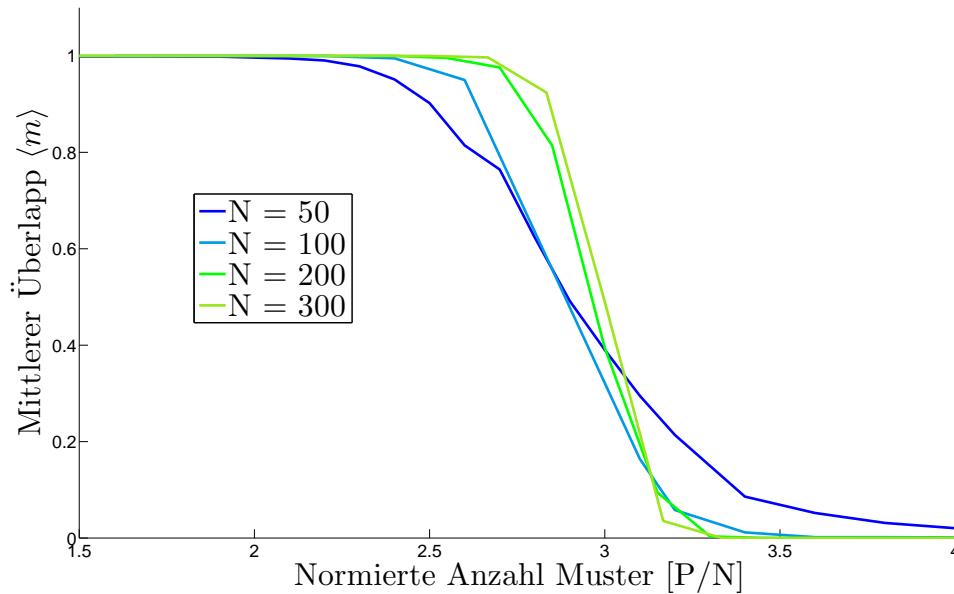


Abbildung 8.5: Die Kapazität im Batch-Lernen mit korrelierten Mustern

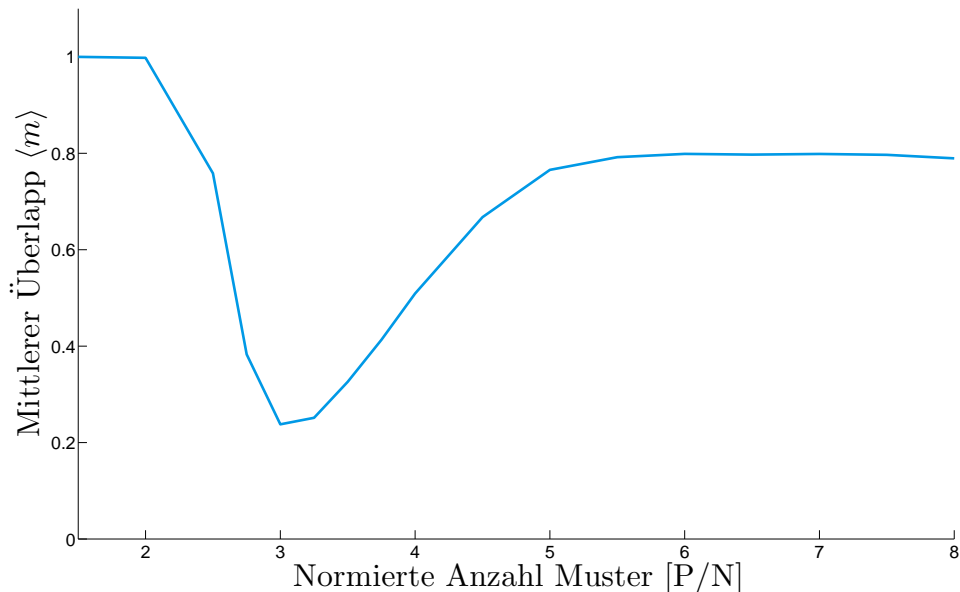


Abbildung 8.6: Batch-Lernen mit im Mittel 10% Abweichung von einem Vorlagenmuster

Man erkennt, dass die Kapazität des Netzes höher ist als bei unkorrelierten Mustern. Ähnlich wie im Falle unkorrelierter Muster wird der Abfall des Überlapp steiler, je höher die Anzahl der Neurone ist.

8.2 Global korrelierte Muster im Batch-Lernen

Neben den lokal korrelierten Mustern wurden auch Mustersätze mit globaler Korrelation betrachtet. Die Muster wurden aus einem Ausgangsmuster erzeugt. Dabei hatte jedes Element eines Musters eine bestimmte Wahrscheinlichkeit, im Vergleich zum Ausgangsmuster das Vorzeichen zu wechseln. Es wurden drei verschiedene Größen der Korrelation betrachtet, und zwar weichen im Mittel 10%, 25% und 40% der Elemente vom Ausgangsmuster ab. Eine mittlere Abweichung von 50% würde keine Korrelation bedeuten.

Ergebnis der Simulation

Die Ergebnisse der Simulationen sind in den Abb. 8.6 bis 8.8 gezeigt. Die Netze bestanden jeweils aus 100 Neuronen.

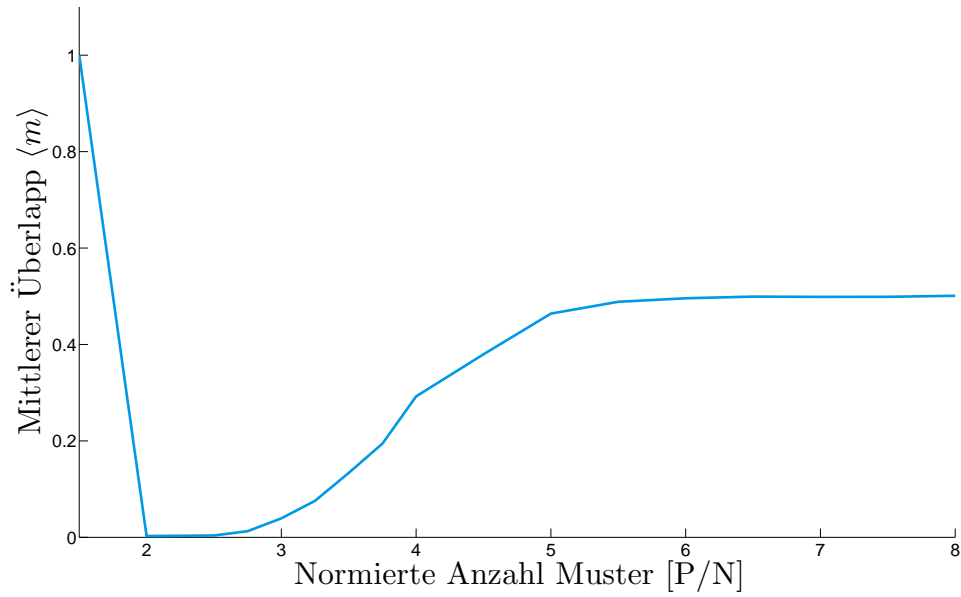


Abbildung 8.7: Batch-Lernen mit im Mittel 25% Abweichung von einem Vorlagenmuster

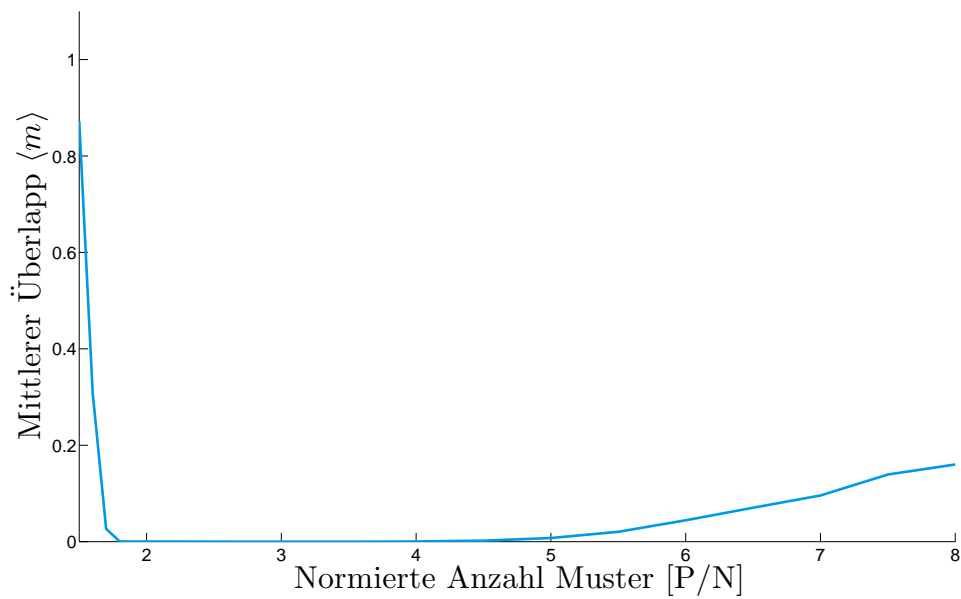


Abbildung 8.8: Batch-Lernen mit im Mittel 40% Abweichung von einem Vorlagenmuster

In dem Fall der globalen Korrelation wurde Augenmerk auf die Vorgänge gelegt, wenn die Zahl der zu Lernenden Muster weit über der Kapazität von $\approx 1,6N$ liegt. Interessanterweise ergibt dieses Protokoll keine Steigerung der Kapazität über $2N$ hinaus, wie im Falle lokal korrelierter Muster. Darüberhinaus fällt auf, dass der mittlere Überlapp für sehr hohe Anzahlen von Mustern wieder *ansteigt*. Den Hinweis darauf, was passiert, ergibt der Wert, auf den der Überlapp steigt: Er liegt nämlich bei der mittleren Korrelation der gelernten Muster mit dem Vorlagenmuster. Eine Stichprobe bei $P/N = 8$ bei 10% mittlerer Abweichung ergab, dass der Attraktor jedes der Muster das Vorlagenmuster war. Das Ergebnis ist also eine Generalisierung der verrauschten Eingaben auf das zugrundeliegende Muster.

Kapitel 9

Diskussion

9.1 Zusammenfassung der Ergebnisse

In der vorliegenden Arbeit wurde das Lernen von Mustern in künstlichen assoziativen neuronalen Netzen betrachtet. "Lernen" bedeutet dabei, dass die präsentierten Muster zu Fixpunkten oder besser Attraktoren der Dynamik werden sollen. Es zeigt sich, dass die Kapazität der Netze im Batch-Lernen für unkorrelierte Muster nahe an die theoretische Grenze der Kapazität von $2N$ gelangt, wobei die Wahl der Schwelle einen Einfluß auf die Kapazität hat. Eine Verringerung der Schwelle bringt eine Erhöhung der Kapazität des Netzes, wird aber mit einer Abnahme der Stabilität der Muster und damit der Assoziativität des Netzes erkauft. Die mit der Anzahl der Neuronen normierte Kapazität des Netzes wird außerdem immer schärfer, je höher die Anzahl der Neuronen ist.

Im Online-Lernen liegt die Kapazität hingegen sehr viel niedriger, bei $0,05N$. Dass dies so sein muss, liegt auf der Hand: Der Perzeptronalgorithmus führt inkrementell einen Ausgleich zwischen allen Mustern herbei. Wenn die Muster aber nacheinander gelernt werden, dann kann es einen solchen Ausgleich nicht geben, und die neuen Muster "zerstören" die vorher gelernten.

Die beiden betrachteten Netze, diskrete und kontinuierliche Neurone, sind sich in der Kapazität sehr ähnlich, was nicht verwundert. Ein Mean-Field-Netz stellt eine Mittelung des diskreten Falls dar. Trotzdem sind im Batch-Lernen Abweichungen zu erkennen, die relative Kapazität scheint mit der Anzahl der Neuronen anzusteigen (Vergleiche die Abb. 5.1 und 6.2). Es ist nicht ganz klar, ob dieser Unterschied ein echter Effekt ist oder der Durchführung der Simulationen

geschuldet ist. Denkbar ist ein Einfluss der Parameter, wie die Abbruchbedingung im Mean-Field-Modell oder die Anzahl der Schritte. Für das Online-Lernen konnte dagegen gezeigt werden, dass die beiden Netze die gleichen Gewichte liefern.

Etwas unerwartet ist der Kontrast in der Kapazität im Falle korrelierter Muster, die die beiden Lernvorschriften Online und Batch ergaben. Im Online-Lernen ist die Kapazität bei korrelierten Mustern fast Null. Warum das so sein muss, zeigt dann die Rechnung. Hingegen ist beim Batch-Lernen die Kapazität besser oder zumindest nicht schlechter als im Falle unkorrelierter Muster, zusätzlich besitzt das Netz unter der Perzeptronlernregel die Fähigkeit zum Generalisieren.

9.2 Ausblick

Der derzeitige Konsens in der Erforschung des Gedächtnisses ist, dass dem Formen von Gedächtnisinhalten eine Attraktordynamik zugrunde liegt [2]. Daher ist es interessant, neuronale Netzwerke zu untersuchen, die ebenfalls eine solche Dynamik aufweisen. Das hier betrachtete Modell lernt Muster, indem durch die Gewichtsänderungen die Musterzustände zu Attraktoren werden. In diesem Sinne ist das Modell und die verwendete Lernregel ein Kandidat für die Modellierung von assoziativem Gedächtnis.

Das Modell reproduziert dabei einige Eigenschaften von Gedächtnis. Zum einen ist das "Vergessen" alter Inhalte ein Resultat des Online-Lernens. Neu gelernte Muster verändern die Kopplungen zwischen den Neuronen nach und nach, vorher Gelerntes bleibt aber bis zu einem bestimmten Punkt noch ein Attraktor des Netzes. Ab diesem Punkt wird das Muster instabil. Das Vergessen ist natürlich eine wesentliche Eigenschaft des menschlichen Gedächtnisses. Zum anderen besitzt das Netz im Batch-Lernen die Fähigkeit, ähnliche Muster zu abstrahieren (Generalisierung, Abschnitt 8.2), was Überlagerung von ähnlichen Gedächtnisinhalten interpretiert werden kann.

Andererseits ist das Modell in anderen Punkten unplausibel. Im Online-Modus führen korrelierte Muster dazu, dass die Kapazität stark abnimmt. In der Simulation war nur das zuletzt gelernte Muster stabil, alle anderen Muster davor aber nicht. Im Batch-Modus mit korrelierten Mustern war die Kapazität im Gegensatz dazu sogar höher als mit unkorrelierten Mustern.

Darüberhinaus gibt es noch ein weiteres Problem. Biologische Synapsen sind in ihren Res-

sources begrenzt, d.h. sie können nur eine begrenzte Anzahl Vesikel und Rezeptoren ausbilden. Dieser Umstand wird im Modell nicht berücksichtigt, die Kopplungen können jeden beliebigen Wert annehmen. Die Diffusion der Gewichte im Online-Lernen wird zum Beispiel mit einer Diffusion der Kopplungen einhergehen; dadurch werden die Werte des maximalen Gewichts mit der Zeit zunehmen. Es wurde im Verlauf dieser Arbeit getestet (nicht gezeigt), welche Auswirkung die Beschränkung der Gewichte hat, einerseits indem die Gewichte auf ein endliches Intervall begrenzt wurden oder indem ein multiplikativer Term w_{ij}^{-1} zu der Gewichtsänderung zugefügt wurde. Beide Änderungen führten zu einer Abnahme der Kapazität des Netzes unter der Perzeptronlernregel.

Die Frage ist also, wie das vorgestellte Modell im Verhältnis zu den Vorgängen im Gehirn steht. Kann der Anschluss an experimentelle Ergebnisse erbracht werden, und wenn ja, wie? Der Ausgangspunkt der Überlegungen, die zu dieser Arbeit führten, war die Spike-Timing Dependent Plasticity. Ausgehend von der Gewichtsänderung als Reaktion auf zwei Spikes prä- und postsynaptisch (Abb. 2.3) wurde ein Modell entwickelt, welches diese Kurve wiedergibt. Dieses STDP-Modell ist bisher in zwei Konferenzbeiträgen ([18], [19]) veröffentlicht worden. Es führt die Änderung der Synapse zurück auf tiefpassgefilterte Aktivitäten der beiden beteiligten Neurone, und die Gewichtsänderung in diesem Modell ist proportional zu dem Produkt der präsynaptischen gefilterten Aktivität und der Ableitung nach der Zeit der gefilterten postsynaptischen Aktivität: $\dot{w}_{ij} \propto \dot{y}_{\text{post}} y_{\text{prä}}$. Diesen Ausdruck findet man in der Perzeptronlernregel für kontinuierliche Neurone wieder: Hier ist die Änderung des synaptischen Gewichts proportional zu der Differenz $x_{\text{post}}(t) - x_{\text{post}}(t+1)$, multipliziert mit dem Zustand des präsynaptischen Neurons. Die Differenz kann als zeitliche Änderung und damit als Ableitung verstanden werden.

Um die Ergebnisse dieser Arbeit mit dem STDP-Modell zu vergleichen sind verschiedene Ansätze denkbar. Zum einen kann man versuchen, das Lernproblem mit der STPD-Regel analytisch zu lösen oder Näherungen für den mittleren Fall zu finden, um die Ähnlichkeit zur Perzeptronlernregel zu untersuchen. Zusätzlich könnte man ein Netzwerk aus spikenden Neuronen, z.B. Integrate-and-Fire-Neuronen simulieren. Augenmerk kann man dabei auf die Eigenschaften, also Kapazität des realistischeren Netzes unter den beiden Lernprotokollen (Online und Batch) und in Abhängigkeit von der Korrelation der Muster legen. Dies sind mögliche Wege, die Ergebnisse dieser Arbeit mit dem Experiment in Verbindung zu bringen.

Literaturverzeichnis

- [1] Miyashita, M. (1988). Neuronal correlate of visual associative long-term memory in the primate temporal cortex. *Nature*, Vol. 335, 817-820
- [2] Wills, T.J., Lever, C., Cacucci, F. Burgess, N. and O'Keefe, J. (2005). Attractor dynamics in the hippocampal representation of the local environment. *Science* 308, 873
- [3] Hopfield, J.J. (1982). Neural networks and physical system with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* 79, 2554-25581
- [4] Tsodyks, M und Markram, H. (1997). The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability. *Proc. Natl. Acad. Sci. USA*, 94, 719-723
- [5] Hebb, D.O. (1949). *Organization of Behavior*. John Wiley and Sons, New York.
- [6] Bienenstock, E.L., Cooper, L.N., Munro, P.W. (1982). Theory for the Development of Neuron Selectivity: Orientation Specificity and binocular Interaction in visual Cortex. *Journal of Neuroscience*, Vol. 2, No. 1, 32-48
- [7] Henry Markram et al. (1997). Regulation of Synaptic Efficacy by Coincidence of Postsynaptic APs and EPSPs. *Science* 275, 213
- [8] Yang Dan und Mu-Ming Poo (2006) Spike Timing-Dependent Plasticity: From Synapse to Perception, *Physiol Rev* 86:1033-1048. doi:10.1152/physrev.00030.2005
- [9] Jeffrey C. Magee and Daniel Johnston (1997). A Synaptically Controlled, Associative Signal for Hebbian Plasticity in Hippocampal Neurons. *Science* 275, 209

- [10] Senn, W., Markram, H., Tsodyks, M. (2000). An Algorithm for Modifying Neurotransmitter Release Probability Based on Pre- and Postsynaptic Spike Timing. *Neural Computation* 13, 35-67
- [11] Peter Dayan and L.F. Abbott (2001) *Theoretical Neuroscience*. MIT Press
- [12] Mongillo, G., Barak, O., Tsodyks, M. (2008). Synaptic Theory of Working Memory. *Science* 319, 1543-1546
- [13] Froemke, R.C. und Dan, Y (2002). Spike-Timing-dependent synaptic modifications induced by natural spike trains. *Nature*, Vol. 416, 433-438
- [14] Hertz, J., Krogh, A. und Palmer, R.G. (1991) *Introduction to the Theory of Neural Computation*
- [15] Malsburg, Chr. v. d. (1973). Self-Organization of Orientation Sensitive Cells in the Striate Cortex. *Kybernetik* 14, 85-100
- [16] Diederich, S., Oppen, M. (1987) Learning of Correlated Patterns in Spin-Glass Networks by Local Learning Rules. *Phys. Rev. Letters*, Vol. 58 No. 9, 949-952
- [17] Barak Blumenfeld et al., Dynamics of Memory Representations in Networks with Novelty-Facilitated Synaptic Plasticity, *Neuron* (2006), doi:10.1016/j.neuron.2006.08.016
- [18] Pawelzik, K. und Albers, C. (2009) Self-organization of asymmetric associative networks. Poster auf der CNS 2009, Berlin
- [19] Pawelzik, K. und Albers, C. (2009) Adaptive spike timing dependent plasticity realises palimpsest auto-associative memories. Oral Presentation auf der BCCN 2009, Frankfurt