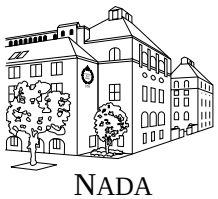
KTH Numerical Analysis
and Computer Science

Memory Consolidation in Artificial Neural Networks

Axel Liljencrantz

TRITA-NA-E03148



NADA

Numerisk analys och datalogi
KTH
100 44 Stockholm

Department of Numerical Analysis
and Computer Science
Royal Institute of Technology
SE-100 44 Stockholm, Sweden

Memory Consolidation in Artificial Neural Networks

Axel Liljencrantz

TRITA-NA-E03148

Master's Thesis in Biomedical Engineering (20 credits)
at the School of Engineering Physics,
Royal Institute of Technology year 2003
Supervisor at Nada was Anders Sandberg
Examiner was Anders Lansner

Abstract

Memory consolidation is a well known experimental phenomenon which has been shown to exist in several kinds of animals as well as humans. This thesis treats a model for learning new memories which is based on memory consolidation. The model has been implemented using artificial neural networks of a type called Bayesian Confidence Propagation Neural Networks, and its functionality has been tested using a large set of simulations. The results show that the model works and predicts results similar to experimental results.

Key words: Memory consolidation, artificial neural networks, Bayesian Confidence Propagation Neural Networks, adaptation, the medial temporal lobe.

Minneskonsolidering i artificiella neuronnät

Sammanfattning

Minneskonsolidering är en välkänt experimentellt fenomen som har påvisats hos flera djurarter liksom hos människor. Detta examensarbete behandlar en modell för inlärning av minnen baserad på minneskonsolidering. Modellen har implementerats med hjälp av artificiella neuronnät av typen Bayesian Confidence Propagation Neural Networks, och dess funktionalitet har testats med hjälp av ett stort antal simuleringar. Resultaten visar att modellen fungerar och förutsäger resultat som liknar experimentella resultat.

Nyckelord: Minneskonsolidering, artificiella neuronnät, Bayesian Confidence Propagation Neural Networks, adaptation, mediala temporala loben.

Acknowledgments

I would like to thank my adviser and roommate Anders Sandberg as well as my roommate Lotta Eriksson for making this thesis a pleasure to write. I would also like to thank all my friends, especially Sara Abrahamsson and Martin Johansson who have made sure I've spent as little time as possible doing anything even remotely related to this thesis, like sleeping. I am also grateful to my girlfriend Anna and my parents Bożena and Gustaf Liljencrantz for their never ending support.

Contents

Chapter 1

Introduction

1.1 Overview

The objective of this thesis is to examine the theory about the workings of the memory system of higher animals, and how this theory can be applied using a set of computer simulations. This theory, known as memory consolidation, is applied using a type of neural networks called Bayesian Confidence Propagation Neural Networks.

1.2 The human brain

> "So far then this much is plain, that all animals must necessarily have a certain amount of heat. But as all influences require to be counter-balanced, so that they may be reduced to moderation and brought to the mean (for in the mean, and not in either extreme, lies the true and rational position), nature has contrived the brain as a counter-poise to the region of the heart with its contained heat, and has given it to animals to moderate the latter, combining in it the properties of earth and water." [2]

The main function of the brain is to control the body. The human brain is a device consisting of roughly 1000 billion individual cells, about 100 billion of these cells are of a type called neurons, the remaining cells are called neuroglia [43]. Neurons have several properties which make them unique [37]:

- Neurons are connected to each other through synapses; each neuron can have many thousand synapses.
- Neurons can quickly and efficiently signal each other by firing cascades of chemicals, known as transmitters, through their synapses.
- Neurons can respond to combinations of signals from several neurons simultaneously, so called spatial summation, as well as signals at different points in time, so called temporal summation.

- Neurons can respond to signals in different ways. The most common response is adjusting its own firing frequency, other responses include altering its response to signals from other neurons and creating or removing synapses.

These properties allow the neurons to form a collective network for processing input from the sensory organs and choosing and shaping a suitable action for the body to perform in response. This is called the cognitive process. An important part of the cognitive process is the ability to filter out and store important data in order to retrieve and use it at a later point in time. This is called memory.

1.3 An overview of memory research

The subject of memory has been discussed by philosophers since long before the field of psychology came into existence. Aristotle likened the memory to a piece of wax, in which a signet ring can leave an imprint. A memory is seen as a sensible object without its matter, and recall is seen as the imagination of an earlier impression without the actual sensation. Recollection is, according to Aristotle, governed by laws which regulate association between a sensation and previous memories based on similarity, continuity and other factors [1, 3]. Though Aristotle understood that memory should not be viewed as a passive warehouse for previous experiences, his theories tell us very little of how the brain works, how memories are stored and retrieved.

The first researcher to study learning as it occurred in research subjects was Hermann Ebbinghaus. One of his most famous results is that students forget about 90% of what they learn in class in thirty days.

1.3.1 Memory classifications

At the end of the 19$^{\text{th}}$century, William James suggested dividing memory into primary (short term, STM) and secondary (long term, LTM) memory [21]. His definition of primary memory is the memory which is held for only a moment, while secondary memory is the memory which can be made unconscious and later retrieved.

The existence of a short term memory has been demonstrated trough multiple experiments [10, 30], where test subjects are given information which is rapidly forgotten if active rehearsal is prevented. Evidence also exists in neuropsychological form in cases of impaired LTM and preserved STM [33] and impaired STM and preserved LTM [34].

Experiments show that the long term memory does not consist of a single system. Cohen and Squire divide the memory into declarative and procedural memory [12, 38]. They define declarative memory as memories which are consciously recollected, and procedural memory as memory content which remains inaccessible (i.e. you cannot verbally explain how) even when causing behavioral changes, such as knowing how to ride a bicycle. Studies on amnesia patients reveal intact learning abilities for

procedural skills such as motor skills and classical conditioning in patients with no conscious recollection of learning the task [12].

Further subdivisions within declarative memory have been suggested by Tulving [42]. Episodic memories are memories of actual events, while semantic memory is memory of abstract knowledge, such as knowing the capital of Assyria. It has been suggested that semantic memory is simply the merging of multiple related episodic memories based on their common attributes [4]. The relation between these types of memories are described in figure 1.1.

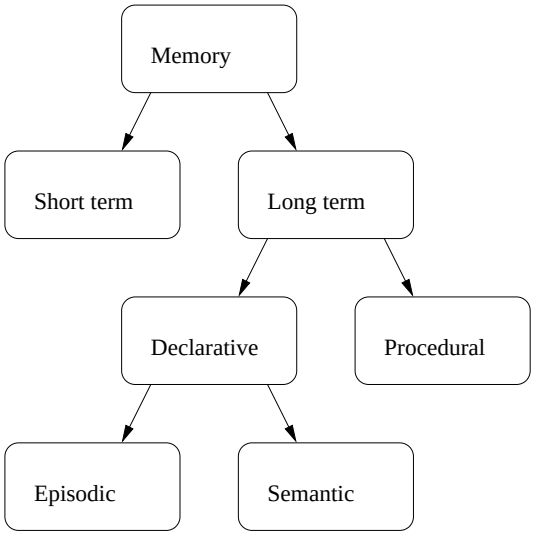


Figure 1.1. Some of the proposed classes of memories and their internal relation.

Schachter and Tulving define different memory systems from psychological characteristics [32]. Each memory system performs multiple tasks with similar functional features, such as working memory or skill learning. This classification has been shown to have a correspondence in test subjects [12] [28].

1.3.2 The biological basis for memory

Near the end of the 19$^{\text{th}}$century, it was independently suggested by both Tanzi and Cajal that memory is stored by changing the synaptic connections between neurons [14]. Modern research has mainly focused on Long Term Potentiation (LTP) as the biological basis for memories. LTP is a phenomenon where stimulation of neurons causes steeper rise times in the potential of postsynaptic neurons for several hours, which suggests enhanced synaptic strength. The phenomenon was originally discovered when stimulating neurons with afferent connections to the hippocampus with high frequency tetanic stimulation [9] [8]. There is much evidence that supports the importance of LTP in the formation of memories [26] [13].

Another common phenomenon which could be important to memory storage is Spike Timing Dependant Plasticity (STDP), where the timing relationship between the presynaptic and the postsynaptic neuron can cause either potentiation (presynaptic neuron fires first) or depression (postsynaptic neuron fires first) [23] [5].

1.4 An overview of research on the Medial temporal lobe

The Medial Temporal Lobe (MTL) is an area of the brain which is located on the inside of the temporal lobe (see figure 1.2). It has been known for several decades that the MTL plays an important role in creating new memories. The first important piece of evidence of the MTLs importance for memory was the patient HM. In 1953 HM underwent surgery to treat his intractable epilepsy, bilaterally removing his MTL. It was discovered that HMs ability to form declarative long-term memories was severely degraded, so called anterograde amnesia. His memories of his life previous to his surgery was mostly unaffected, but memories of the period shortly before the procedure where degraded, a phenomenon called temporally-graded retrograde amnesia. It is important to note that HMs non-declarative memory seemed intact, suggesting that the MTL is not important for the formation of these memories. Other patients with lesions to the MTL exhibit similar behavior, but the number of cases is limited due to the highly specific nature of the lesion.

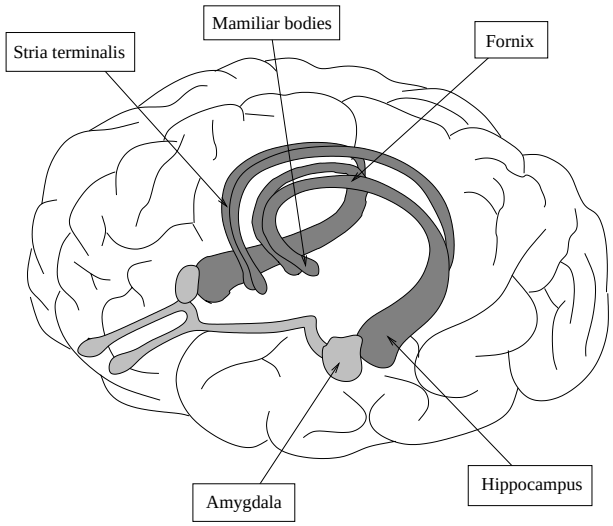


Figure 1.2. The location of the MTL. The dark areas are critical to declarative memory formation.

Squire has reviewed several studies on monkeys as well as rats, testing the effects of lesions to different parts of the MTL in order to narrow down the brain regions needed for memory storage [41, 39]. These tests suggest that a large part of the MTL, including the hippocampus, entorhinal and perirhinal cortex and the diencephalon,

is important in forming new long-term memories. Several neighboring areas, such as the amygdala, have been shown to have little importance in the formation of memories.

Squire and Alvarez have also modeled the phenomenon of retrograde amnesia after MTL damage [40]. They show that temporally graded retrograde amnesia is common in both humans and experimental animals with damage to the MTL. They conclude that as time passes after learning, the importance of the MTL diminishes.

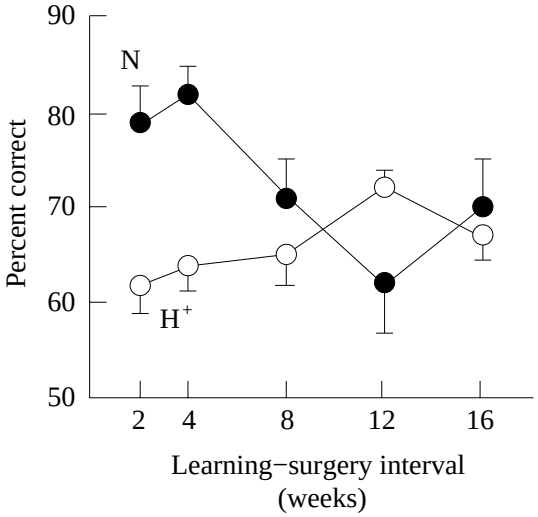


Figure 1.3. Performance of normal monkeys (N) and monkeys with removal of hippocampus and surrounding areas (H^+) on retention of 100 object discrimination problems learned at different times before surgery. From Zola-Morgan and Squire[44].

Another set of key experiments for understanding the role of the MTL in long term memory consolidation were performed on rats by Skaggs and McNaughton [36]. These experiments showed that the MTL is active in rats when memorizing a new environment, such as a maze. More importantly, these experiments show that certain cells in the MTL, dubbed "place cells", fire more rapidly in a specific area of the maze, suggesting that the rats actually form some form of spatial representation of the maze in the MTL while learning to navigate it. This lends itself very well to the theory that the MTL stores memories, but another important observation has also been made in these experiments, which may give an indication to the nature of the MTL. If one measures the activity in a sleeping rat who has memorized a new maze during the day, the place cells activate in patterns, and these patterns will be consistent with the labyrinths geometry. In other words, it can be argued that the rat is dreaming of running through the maze while it is sleeping.

Hoffman and McNaughton have performed similar experiments on monkeys [17]. By implanting several hundred electrodes into four different cortical areas of macaque monkeys, they were able to show that neurons in the neocortex and in the MTL that co-fired during task performance also co-fired during the following rest-period, even

though no task was performed. They also showed that this co-activity was a result of the task performance by measuring co-activity during a rest period before task performance. Much like in Skaggs and McNaughtons experiments, it was found that during the rest-period the neurons genrally fired in the same order as they had originally fired during task performance.

1.5 Memory consolidation

It seems reasonable to conclude from these experiments that some form of memory storage takes place in the MTL. The observation that rats seem to spontaneously recall memories while sleeping suggests that these memories are somehow processed long after the original event. This hypothetical processing of a memory from labile state to a more stable one is called memory consolidation. Memory consolidation was first proposed by Müller and Pilzecker in 1900 as an explanation for why memory performance is degraded by presenting new material shortly after exposure to the original memories.

One of the most influential MTL theories, which was introduced by Marr, is that the MTL is in fact a fast learning memory, which forgets after a time on the order of a few days [24]. During this time, the MTL projects its contents to the cortex, so that by the time the memory fades from the MTL, it has been permanently stored in the cortex. Marr suggested that the reason for this additional step in learning was to build new associations among concepts. It seems reasonable that a change in the syntactical connectivity which will last for many years may take a while to form. It has also been found in some types of artificial neural networks that a fast learning memory is also fast at forgetting [31]. If this is the case for the brain, this temporary storage is needed in order not to overwrite earlier memories while permanently storing a new memory. The time from when a memory is inserted into the MTL until it is stored in the cortex may also be used by various cognitive processes to filter out unimportant information.

1.5.1 Simulating memory consolidation

Marr did not do any simulations of his theory, but several interesting computational models have been created based on it.

- Alvarez and Squire used a simple artificial neural network of 12 nodes to create a computer simulation which showed retrograde amnesia curves similar to those of test subjects [40].
- Cartling simulated a system consisting of a MTL that repeatedly activates neocortical patterns and thus achives memory consolidation by long term potentiation [11].
- Murre created a memory model called Tracelink based on memory consolidation. The idea behind Tracelink is that when creating new long term memories,

the cortex needs to create new synapses between distant parts of the cortex, a process which takes time. During this time, the MTL acts as a temporary link between different regions of cortex. According to Murre, the hippocampus also has a second role as a control center for plasticity [27].

- Bibbig et al. has simulated a system where every sensory pathway stores memories associated to that particular form of sensory input, and the hippocampus acts as a link between these separate memory storage areas, creating associations between different aspects of memories such as smell, sound and images [7].

- Bibbig et al. has also investigated the possibility that the hippocampus might in fact learn memories in two stages. The first stage would consist of weak heterosynaptic potentiation and the second stage would consist of strong arythmic bursting during sleep, which would cause long term potentiation. Thus, memory consolitation would be performed in multiple steps, both within the hippocampus and in the cortex [6].

1.6 An introduction to artificial neural networks

> My CPU is a nooral netwoak processoh - a learning computah
> — *Arnold Schwarzenegger, Terminator 2.*

Every neuron in the brain is a separate signal processing unit, capable of performing simple computations in parallel with every other neuron in the brain. The digital computer is in comparison a much more sequential system. Early processors only performed one single computation at any point in time. While modern processors can perform hundreds of instructions simultaneously, and clusters of computers can perform many thousand instructions simultaneously, this is still a far cry from the enormous parallelism of the brain. Despite these fundamental differences, making computers simulate the behavior of a neural network has provided useful results. A computer simulation of a neural network, an artificial neural network, could potentially solve some of the problems that have been encountered in computational science. These problems mostly boil down to the explicit nature of a computer: writing a computer program for making an educated guess has proven much more difficult than writing a computer program for making an exact calculation. Artificial neural networks have been touted as the magical silver bullet of computer science – self-modifying machines that are able to think for themselves, understand the world around them and perhaps even become aware of their own existance. Given the difference in complexity between a human brain and a CPU, as well as the difference in timescale between the evolution of the human species and the evolution of computers, it should come as no surprise that this has not yet happened.

1.6.1 The internals of an artificial neural network

Artificial neural networks consist of a population of nodes, corresponding to the neurons of a biological neural network. These nodes interact through projections (a set of connections between nodes), corresponding to the synapses of a biological neural network. Every node is associated with an activity corresponding to the activity of a neuron. Every connection from one node to another is associated with a weight, signifying the strength of the connection between the two nodes. When a node is updated, the activity of every node with a connection to the node being updated is multiplied with the weight associated with the relevant projection. All these weighted activities are then summed into a single value, which is inserted into an arbitrary function, usually referred to as the squashing function or the threshold function. The output of this threshold function is the new activity for the node.

There is a wide variety of ways to complicate and extend a neural network. Every node can have an additional state beside the activity, usually referred to as the bias. The bias is added to the input sum before inserting it into the threshold function. The threshold function may use more input variables than only the weighted activity of the input neurons, for instance the previous activity of the node. Not all networks use a single sum for turning all the weighted activities into a single value. More complex operations are sometimes performed. As an example, Sigma Pi networks calculate the sum of partial products of the weighted averages [35].

1.6.2 Network topology

Neural networks use one of two main network topologies [16]:

- The connections between nodes in a recurrent networks are not structured in any way. Any node may project to any other node. This means that recurrent networks can contain loops.
- The connections between nodes of a feed-forward network are structured so that the nodes may be divided into layers. A node may not have any connections to other nodes in the same or previous layers. There is an input layer, an output layer, and an arbitrary number of intermediate (hidden) layers.

Feed-forward networks are obviously a subcategory of recurrent networks.

1.6.3 Weight updating

So far the question of how the weights in the network are calculated has not been touched. There are a great many algorithms for updating the weights, resulting in completely different behaviors. These update rules vary greatly in both function and biological plausibility.

Many update rules are influenced by the theories presented by Donald Hebb. In his 1949 seminal, *The organization of behaviour*, he wrote: "When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing

it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased."[15] Any update rule where the weight of the projection between two nodes changes in proportion to their co-firing rate is called a Hebbian learning rule. Two neural networks, including their weight update function are described in the method chapter.

1.6.4 Tasks performed by artificial neural networks

Artificial neural networks (ANN) can perform a large number of different functions. These include pattern completion, pattern completion, input classification and function approximation. These tasks are to some degree equivalent. They all deal with taking an incomplete, noisy or otherwise obscured input and making a guess, based on previous experience, as to what the best output should be.

1.6.5 Auto-associative learning

One of the tasks that can be performed by an ANN is known as auto-associative learning. Auto-associative learning is a form of pattern recognition. Given a noisy or partial pattern, the task for an auto-associative memory is to recall the original version. It can be argued that declarative memory is a form of auto-associative memory. When we try to remember something, like the name of the capital of Assyria[1], we know roughly what it is we try to remember, namely a city which is the capital of the country of Assyria. All we need to do is fill in the blanks, in this case the name of the city. For examples of associations, see figure 1.4.

1.7 The objective of this thesis

The goal of this thesis is to evaluate the memory model based on the ideas in section 1.5. This is done by constructing a computer model based on a type of ANN called Bayesian Confidence Propagation Neural Networks (BCPNN). The performance of this computer model is thoroughly evaluated. This is done by comparing the results obtained from simulations with actual data, analyzing areas of interest for future research, as well as predicting the outcome of possible experiments.

[1] The capital of Assyria is Nineveh.

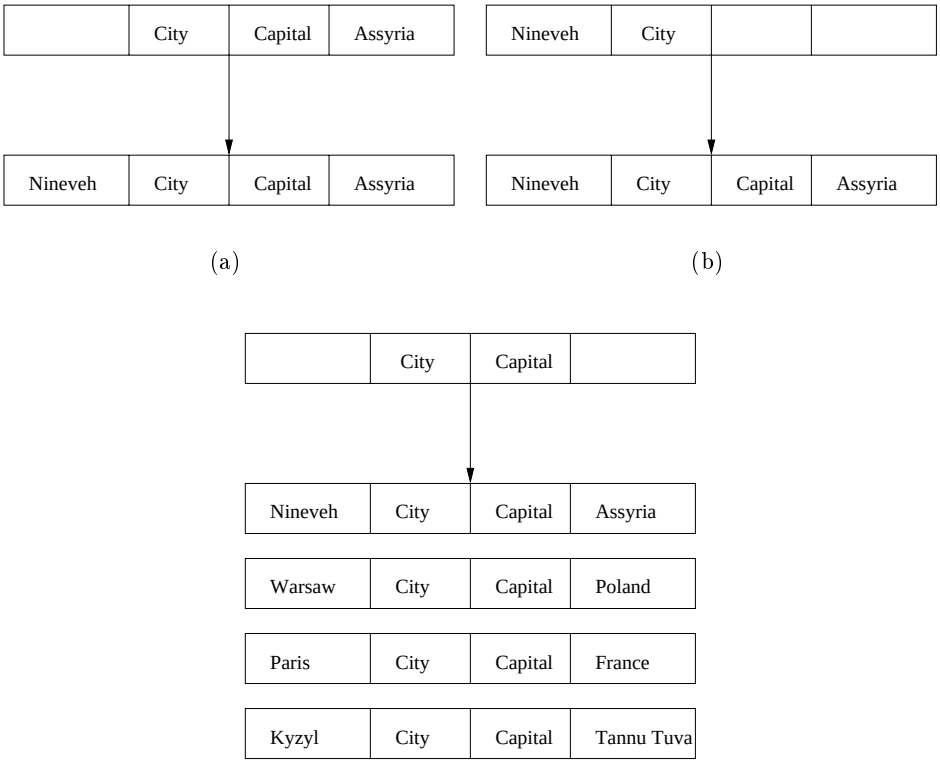


Figure 1.4. (a) and (b) show two examples of auto-association. Some amount of information about an object (The city of Nineveh) is inserted, and a more complete set of information is obtained. (c) shows an example of free recall. The subject thinks of objects with a property (Cities which are capitals), and a number of associations are made(to capitals of different countries).

Chapter 2

Model and Method

This chapter begins by giving an overview of two types of ANNs, the latter of which is the type used in this thesis. It contains a large number of equations and may prove difficult to read. Readers without a mathematical background may skip sections 2.1 and 2.2. The rest of the chapter deals with the models and methods used in this thesis and should be readable to anyone.

2.1 Hopfield networks

One of the most well-known and thorougly studied types of ANNs is the Hopfield network. It is a recurrent, fully connected network working as an auto-associative memory. It has also been used for combinatorial optimization, for instance for approximatly solving the traveling salesman problem [19]. Every node in the Hopfield network can be in one of two states, and a simple Hebbian update rule is used. The simplicity of the Hopfield network makes it possible to analyse it using methods from statistical mechanics.

Input patterns in Hopfield networks consist of a string of the alphabet $A \in (-1, 1)$. The weight of the projection from node i to node j of the Hopfield network are calculated as

$$w_{ij} = \frac{1}{N} \sum_{k=1}^{p} x_i^k x_j^p,$$

where x_i^k is the i^{th}element of the hk^{th}pattern. It can easily be seen that every correlated bit-pair will increase the weight by one, and every anti-correlated bit-pair will decrease the weight by one, i.e. a simple Hebbian update rule. It is important to note that the matrix w will always be symmetric.

Because the Hopfield network is recurrent, and its input must be a binary string, the threshold function of the Hopfield network must always output ± 1. The definition of the Hopfield networks threshold function is

$$\mathrm{f}(x) = \begin{cases} 1 & x > 0 \\ -1 & \text{otherwise} \end{cases}.$$

This makes the update rule for a node x_i

$$x_i = \mathrm{f}(\sum_j w_{ij} x_j). \tag{2.1}$$

When retrieving a pattern from a Hopfield network, the weights are initially set to the input pattern. Then all the activities are sequentially updated using equation 2.1 until a stable state is reached.

Some properties of Hopfield networks are summarized below [18]:

- A Hopfield network always reaches a stable state, a so called attractor.
- A Hopfield network will always reach an attractor in $O(\log N)$ complete updates or less.
- A Hopfield network can store up to $0.14N$ patterns as attractors.

The set of states of a given Hopfield network which converge to the same attractor are called the basin of attraction for the given attractor. The basin of attraction is usually a set of patterns which are similar to the attractor.

2.1.1 Limitations of Hopfield Networks

- Hopfield networks are inept at handling interdependent data. When two attractors are very similar, their pools of attraction will be very small.
- Because the weights in Hopfield networks can grow arbitrarily large, and because the firing frequency is usually at 50%, Hopfield networks do not seem biologically plausible.
- The learning speed can not be adjusted in a Hopfield network, which makes it unsuitable for simulations of multiple networks with different learning speeds.

2.2 Bayesian confidence propagation neural networks

Like Hopfield networks, Bayesian confidence propagation neural networks (BCPNN) can be used as auto-associative, recurrent networks. When used this way, they function very similarly, and aside from the adjustable parameters of the BCPNN, one can be used as a black-box replacement for the other.

2.2.1 A simple classifier

The BCPNN can be heuristically derived from Bayes rule, which can be expressed as

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}.$$

We start by attempting to calculate the probabilities π_j of the attributes y_j given a set x of observed binary attributes x_i. Both are discrete and x_i are independent. By using Bayes rule, we get

$$\pi_j = P(y_j|x) = P(y_j)\Pi_{i=1}^{n}\frac{P(x_i|y_j)}{P(x_i)} = P(y_j)\Pi_{i=1}^{n}\frac{P(x_i,y_j)}{P(y_j)P(x_i)}$$

If we want to use this formula in a neural network, we can take the logarithm of the entire expression and obtain

$$\log \pi_j = \log P(y_j) + \sum_{i=1}^{n} \log\Big(\frac{P(x_i,y_j)}{P(y_j)P(x_i)}\Big),$$

replacing the product with a sum and using a logarithm for a squashing function. Suppose now that only the attributes x_i such that $i \in A \subseteq \{1,\ldots,n\}$ are known. If the attributes are independent, we can change the expression to

$$\log \pi_j = \log P(y_j) + \sum_{i=1}^{n} o_i \log\Big(\frac{P(x_i,y_j)}{P(y_j)P(x_i)}\Big),$$

where

$$o_i = \begin{cases} 1 & i \in A \\ 0 & \text{otherwise} \end{cases}.$$

By doing this, we obtain a formula that can be implemented as a single-layered feed-forward neural network. The node activity is o_i, the weights are $log\frac{P(x_i,y_j)}{P(y_j)P(x_i)}$ and the bias is $\log P(y_j)$.

2.2.2 Hypercolumns

If not all attributes x_i are independent, Bayes rule does not hold. But two dependent binary attributes can be replaced by a single attribute with four states, representing the four possible combinations. Similarly, N dependent binary attributes can be

replaced by one attribute with 2^N states. To extend our formula to this case, we introduce a double index notation, where the first index indicates the attribute and the second index indicates the particular value. We will also need the variable M_i, signifying the number of states for the ith attribute. The probabilities $\pi_{jj'}$ for the attributes $y_{jj'}$ can be calculated in the same way as in 2.2.1. We obtain the expression

$$\log \pi_{jj'} = \log P(y_{jj'}) + \sum_{i=1}^{n} o_{ii'} \log \Big(\sum_{i'=1}^{M_i} \frac{P(y_{jj'}, x_{ii'})}{P(y_{jj'})P(x_{ii'})} \Big)$$

We note here that each node coding for the same attribute is linked to the other nodes for the same attribute through the inner sum. This modular structure, commonly referred to as hypercolumns, has also been observed in the brain. The most well known example is edge orientation coding in the visual cortex [20].

We will now replace the binary input $o_{ii'}$ with a probability $P_{X_i}(x_{ii'})$. This extension allows the network to deal with guesses and estimates as input, instead of only dealing with known data. This change means that we are calculating the expected value of the probability of the attribute $y_{jj'}$, not the actual value $\pi_{jj'}$. We will call this estimate $\hat{\pi}_{jj'}$.

The output probabilities $\pi_{jj'}$ have a normalized sum, because the probability of being in any state is always 1. This does not have to be true for the estimates $\hat{\pi}_{jj'}$. Because of this it makes sense to normalize the output probabilities. In practice, the input attributes $x_{ii'}$ are usually only approximately independent, which further increases the need to normalize the output probabilities.

2.2.3 Recurrence

By replacing the discrete inputs of the networks with probabilities, the output of the network has taken the same form as the input. If we connect the network onto itself, we have a recurrent network instead of a feed-forward network. When used in a recurrent fashion, the network is initially presented with an external stimuli, working as a starting guess, which will be refined in every iteration of the network and eventually reaching a stable state. This obviously corresponds closely to how a Hopfield network works.

2.2.4 Incremental learning

If all patterns are available simultaneously, it is quite easy to count the occurrences and co-occurrences of all attributes, and using these counts to approximate the probabilities $P(x_{ii'})$, $P(y_{jj'})$ and $P(x_{ii'}, y_{jj'})$. If we wish for our network to operate continuously we need an approximation of our probabilities that does not require all patterns to be known at once. An obvious candidate is exponentially smoothed

running averages of $\hat{\pi}_{ii'}$ for approximating $P(x_{ii'})$ and $P(y_{ii'})$ and using the coincident activity $\hat{\pi}_{ii'}\hat{\pi}_{jj'}$ for approximating $P(x_{ii'}, y_{jj'})$. Calling our approximated probabilities $\Lambda_{ii'}(t)$ and $\Lambda_{ii'jj'}(t)$, we get the following definitions:

$$\begin{aligned}
\frac{\mathrm{d}\Lambda_{ii'}(t)}{\mathrm{d}t} &= \frac{1}{\tau_L}(\hat{\pi}_{ii'}(t) - \Lambda_{ii'}(t)) \\
\frac{\mathrm{d}\Lambda_{ii'jj'}(t)}{\mathrm{d}t} &= \frac{1}{\tau_L}(\hat{\pi}_{ii'}(t) - \Lambda_{ii'jj'}(t)),
\end{aligned}$$

where τ_L is the learning time constant. In the continuous case, we also need a parameter regulating the update rate for the node states, which we will call τ_c.

2.2.5 Multiple projections

Sometimes it may be desirable to have multiple sets of projections in the same populations. In these cases, it is often preferable to have different strengths for the different projections, so that their relative importance to the state of the population can be regulated. This is accomplished by introducing the notion of a gain variable, g. The learning rate of each projection may be different as well. When using multiple projections it is also sometimes advantageous to project from one population of nodes to another, i.e. to use a feed-forward network.

2.2.6 The BCPNN equation

By now, the BCPNN equation is by far to long to write down on a single line. Here is the equation for a recurrent node with a single projection, with the calculation of the weights, estimated probabilities and normalization separated into separate equations.

$$\begin{aligned}
\tau_c\frac{\mathrm{d}h_{ii'}(t)}{\mathrm{d}t} &= g\left[\log\Lambda_{ii'}(t) + \sum_j^N \log\Big(\sum_{j'}^{M_i} w_{ii'jj'}(t)\hat{\pi}_{jj'}(t)\Big)\right] - h_{ii'}(t) \\
\hat{\pi}_{ii'}(t) &= \frac{e^{h_{ii'}}}{\sum_j^N e^{h_{ij}}} \\
\frac{\mathrm{d}\Lambda_{ii'}(t)}{\mathrm{d}t} &= \frac{1}{\tau_L}(\hat{\pi}_{ii'}(t) - \Lambda_{ii'}(t)) \\
\frac{\mathrm{d}\Lambda_{ii'jj'}(t)}{\mathrm{d}t} &= \frac{1}{\tau_L}(\hat{\pi}_{ii'}(t) - \Lambda_{ii'jj'}(t)). \\
w_{ii'jj'}(t) &= \frac{\Lambda_{ii'jj'}(t)}{\Lambda_{ii'}(t)\Lambda_{jj'}(t)}
\end{aligned}$$

There is a number of possible BCPNN extensions which have been used in this thesis, but will not be more thoroughly described here. These are:

- $\Lambda_{ii'}$ must be kept from reaching zero, to avoid division by zero. This is done by introducing a minimum network activity, λ_0.

- Both the node activity and the estimated probabilities can be further smoothed through multiple passes of exponential smoothing.

- The node weights at start-up time need to be determined. They are usually set by calculating the expected weights after a short time of inactivity.

2.3 Overview of the MTL and cortex model

The memory model simulated in this thesis consisted of two populations of nodes, one representing the MTL and one representing the cortex. Both of these contained a set of internal projections using a Bayesian learning rule, thus turning them into auto-associative memories.

Each simulation consisted of a number of cycles representing days. Each day consisted of a learning phase where memories were inserted into the MTL and a consolidation phase where the MTL teaches these memories to the cortex.

Because the learning rate of a BCPNN can be adjusted, it was possible to simulate a cortex which needed a significantly longer time to form a memory than the MTL, and which also retained memories for significantly longer.

The size of the networks was limited by the performance of modern computers and the patience of the user. Unless otherwise stated, the simulations were performed using a MTL consisting of 80 neurons and a cortex consisting of 200 neurons. The size was a trade off between running time and a large network.

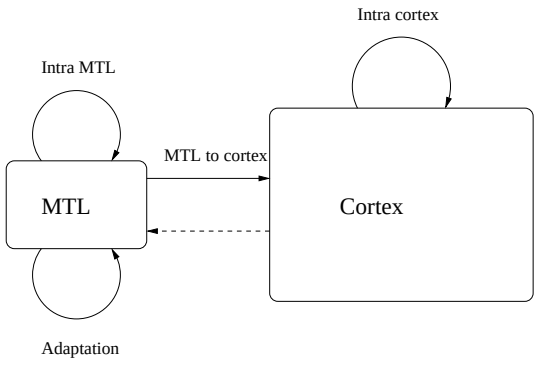


Figure 2.1. The LTM memory model as implemented with BCPNNs. The cortex to MTL projection is dashed because it is not used in this thesis.

2.4 Network parameters

A BCPNN network has several adjustable parameters. Each population has an update rate, governing the speed at which a node changes from one value to another. Each projection has the following parameters:

- A gain, which is a measure of how strongly this projection affects the activity of the target population.
- A learning rate, governing the time it takes to learn a new pattern.

The default values for the network parameters are shown in table 2.1.

Table 2.1. The default values of the networks parameters.

Name	Explanation	Value
τ_L^{MTL}	MTL time constant	200 ms
g^{MTL}	MTL gain	1.0
τ_L^{adapt}	Adaptation time constant	50 ms
g^{adapt}	Adaptation gain	-0.8
τ_L^{CTX}	Cortex time constant	4000 ms
g^{CTX}	Cortex gain	1.0
dt	Time step	10 ms
Λ_0	Base activity	dt/τ_L
N_{MTL}	MTL size	80 Nodes
N_{CTX}	Cortex size	200 Nodes
	Hypercolumn size	10 Nodes
	Day length	800 ms
	Night length	16 000 ms

2.5 Input

The input patterns are strings of random bits, chosen randomly. In real life, the input is obviously not always independent, however there is strong evidence that memories are decorrelated by lower order networks before being inserted into memory, thus random independent bitpatterns should be suitable as input for memory storage. Because of the hypercolumn structure of the BCPNN, exactly one bit is set in each column of the bitset. For most of the simulations, a BCPNN with 200 units arranged in hypercolumns of ten units were used. For such networks, input consists of 200 bits, where one bit in every column of ten bits is set. A new set of inputs were created for every simulation, and the results were averaged over several runs. Unless otherwise stated, the input set consisted of 200 patterns divided into ten days.

2.6 The adaptation projection

In order for a transfer of patterns from the MTL to the cortex to occur, the MTL has to display all of the patterns that have been stored in it, one at a time. To do this, a method for iterating over the currently stored MTL patterns needs to be devised. The method used in this thesis was described by Sandberg [31]. It has been shown that synapse depression occurs in neurons after longer periods of firing, this leads to short term adaptation. This can be simulated in BCPNNs by using a second projection inside the MTL. This second projection, called the adaptation projection, has a negative gain. The negative gain has the effect of creating a negative bias against the current state. Whenever the MTL enters a stable state (i.e. a previously stored pattern) this bias will accumulate, and in time the negative bias from the adaptation projection will be larger than the positive bias from the pattern completion projection. Since the BCPNN slowly forgets older inputs, the bias against a given state from the adaptation projection will slowly decrease toward zero when the MTL is not in that state. Therefore, when left to itself, the MTL will continually cycle between different memory patterns that have been previously stored.

The adaptation process resembles spike frequency adaptation in pyramidal cells as described by McCormick et al. [25]. Additionally, many synapses exhibit depression and facilitation depending on the frequency of presynaptic spikes and neuron classes, possibly as a result of transmitter depletion.

Adaptation simulates an actual process in the neurons, which is present both in the cortex and in the MTL. In this thesis, no adaptation projection was used inside the cortex, because this would only increase the running time of the program and make it slightly more difficult to test whether a memory had been successfully stored in the cortex. This should not affect the results from the simulations in any way, because the timescale involved in the adaptation projection is longer than the one used when recalling stored memories.

The distribution of visiting times of different patterns is dependent on differences in learning rate and exposure time between patterns and the pattern uniqueness [31].

2.7 The intra MTL projection

The MTL consisted of a population of neurons with two internal sets of projections. The first was a normal BCPNN with a time constant of 200 ms, the second projection was an adaptation projection using a time constant of 50 ms.

2.8 The MTL to cortex projection

There are several possible ways to connect the MTL and cortex to each other. The weights in the MTL to cortex projections can be decided beforehand, and remain

static during the simulation. Another possibility is using dynamic weight updating to continually change the MTL to cortex connection.

2.8.1 Static projections

Since the cortex has many more neurons than the MTL, a simple 1 to 1 mapping is not possible. Several different static connection patterns were evaluated. For the sake of simplicity, all static projections that were tested contained a 1 to 1 projection from the MTL to a subset of the cortex population equal in size to the MTL. Because the node order is arbitrary, it can be argued that this 1 to 1 subset does not seem impossible.

The projections to the remaining neurons can be chosen in many different ways. The simplest is setting the rest of the projections to zero. Another possibility is connecting each neuron in the MTL to a number of randomly chosen cortical nodes. This type of projection is shown in figure 2.2. The pure 1 to 1 subset represents a very sparse projection, which is not entirely biologically plausible, as the MTL is rather densely connected with the rest of the cortex.

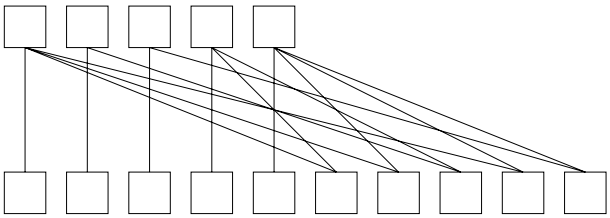


Figure 2.2. An example projection with 5 MTL neurons, 10 cortex neurons and a projection density of two.

2.8.2 Plastic connections

When using a plastic MTL to cortex projection, the input patterns must be simultaneously presented to both the MTL and to the cortex, so the MTL to cortex projection will learn to associate the two patterns with each other. In this case, it is not necessary for the MTL input pattern to have any real relation to the pattern to be stored in the cortex. Because of this, it is possible to associate any pattern in the MTL with the cortex pattern.

An additional advantage of using plastic connections is that the pattern size is equal to the cortex size instead of the MTL size. Because of the tremendous number of neurons in the cortex in comparison to the amount of perceived information in a memory, this may seem like a minor point, but given the large size difference between the MTL and the cortex, it should not be overlooked.

When using a plastic MTL to cortex projection, there is no need for the MTL pattern to hold any relation to the cortex pattern. So long as every cortex pattern maps to a unique MTL pattern, the MTL to cortex projections will learn this new

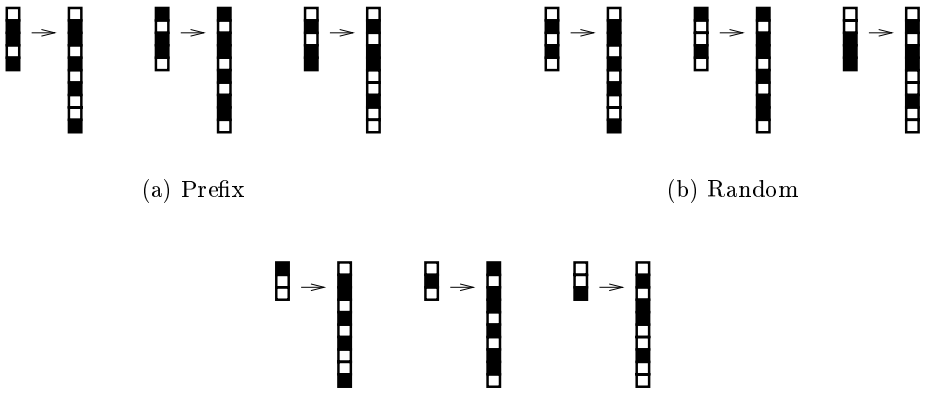
relationship and project the correct pattern accordingly. This introduces a new degree of freedom to the system: Choosing a MTL pattern. Three different approaches where tested, and are described below:

1. Prefix patterns, where the MTL pattern is a prefix of the cortex pattern.

2. Random patterns, where the MTL patterns independant of the cortex patterns and randomly chosen.

3. Orthogonal patterns, where the n^{th}pattern in the MTL has only the n^{th}node active, so that no patterns ever overlap. In this approach, the MTL has the same number of nodes as the number of patterns per day and only one hypercolumn.

Examples of these MTL patterns are shown in figure 2.3.



(a) Prefix

(b) Random

(c) Orthogonal

Figure 2.3. Different plastic MTL to cortex projections. Subfigures (a), (b) and (c) show prefix, random and orthogonal MTL patterns respectively. The smaller pattern represents MTL and the larger one represents the cortex.

2.9 The cortex to MTL projection

In biological systems, there are often a large number of connections going "backwards", i.e. against the flow of information. The most well known example of this is in the visual system, where the number of connections from the primary visual cortex to the lateral geniculate nucleus is larger than the number of connections in the opposite direction, though much evidence indicates that the actual flow of information is in the other direction [22].

2.10 Intra-cortical projections

The intra-cortical projection consisted of a BCPNN projection with a time constant several orders of magnitude larger than that of the MTL. The cortex did not have an adaptation projection, because it would only increase the running time of the simulation without providing any useful information.

2.11 Learning cycle

Most of the simulations used the same learning cycle. The cycle consists of two phases which are repeated ten times. The two phases are:

1. The waking phase. In this phase, the MTL was presented with 20 patterns. The adaptation and the MTL to cortex projection was disabled, and the learning rate of the cortex was set to zero. The MTL learned the new patterns, but the cortex was completely nonplastic. This phase only lasts for 80 timesteps, a small fraction of the sleep time. This does not seem reasonable at first glance, as most animals are asleep and awake for roughly the same amount of time. The reason for this is that the day time only represents the time when the MTL is plastic, i.e. when the MTL is recording new memory.

2. The sleeping phase. In this phase, no external stimuli were present. All projections were enabled, but the learning rate of the MTL was set to zero. The MTL used the adaptation projections to iterate between the patterns that it had learned, so that they where reinstated into the cortex. This phase lasts for about 1 600 timesteps, or 16 000 ms.

Figure 2.5 shows these two phases. The structure of a complete simulation is shown in figure 2.4. Biologically it does not make sense to turn off learning in the cortex during daytime. But by doing so, it is assured direct learning of the cortex is not a factor and only memory consolidation performance is measured.

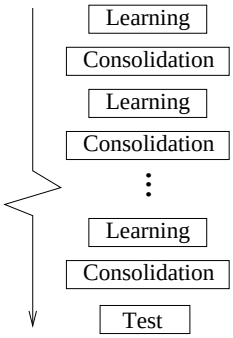
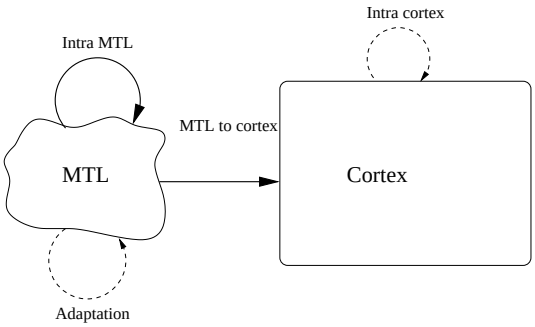
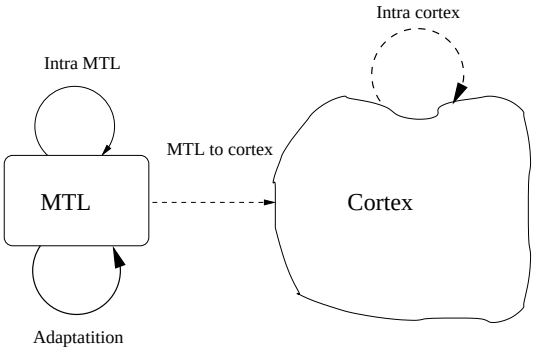


Figure 2.4. A complete simulation consisting of multiple learning and consolidation sequences and a final memory test.



(a)



(b)

Figure 2.5. The learning and consolidation phase of the network. Dashed lines denote a projection which is inactive, i.e. not adding to the network activity. Thick lines denote a projection which is being updated. Figure (a) shows the network during daytime. The MTL is learning new patterns and the MTL to cortex projections are learning to translate these to cortex patterns. Figure (b) shows the network during nighttime. The MTL is iterating through old patterns which are learned by the cortex.

2.12 Testing the memory

After each simulation a test was needed to determine the number of correctly learned patterns. This test needed to be done differently depending on the type of connection between the MTL and the cortex. The reason for this was because in the case of static projections from the MTL to the cortex, the original pattern was the same size as the MTL. Thus, the MTL is needed to act as a translator between the original pattern and it's representation in the MTL.

With static connections, the test consisted of the following steps:

1. Turn off all projections inside the MTL and set the learning rate of the cortex to zero.

2. Create a distorted version of the memory, where three of the eight hypercolumns are scrambled.

3. Project the distorted memory into the MTL.

4. Turn off the projection from the MTL to the cortex.

5. Let the cortex associate freely.

6. Examine the cortex to see if it has entered the state corresponding to the undistorted memory.

Using plastic connections, the test consisted of the following steps:

1. Turn of all projections inside the MTL, the MTL to cortex projections, and set the learning rate of the cortex to zero.

2. Create a distorted version of the memory, where three of the eight hypercolumns are scrambled.

3. Project the distorted memory into the cortex.

4. Let the cortex associate freely.

5. Examine the cortex to see if it has entered the state corresponding to the undistorted memory.

2.13 The maximum capacity of the cortex

In order to correctly evaluate the MTLs performance, it was important to know how well the network would perform with an idealized "perfect" MTL. Such an MTL would simply display all the patterns in order, for an equal amount of time.

2.14 Retrograde amnesia

The phenomenon of time-graded retrograde amnesia is well studied, and it is important to verify that this effect is indeed noticeable when using the network described in this thesis.

When testing for the existence of retrograde amnesia, two sets of simulations where run.

The first set, simulating a damaged MTL, consisted of a normal learning cycle as describe in 2.11. Because the last days memories have only been consolidated for one night, it should be expected that they are not yet completely stable.

The second set, simulating an undamaged MTL, consisted of a normal learning cycle continued by a longer period of sleep. This additional sleep provides additional time for the network to consolidate the memories of the last days.

Chapter 3

Results

3.1 The maximum capacity of the cortex

When evaluating the performance of the adaptation learning, it is important to know the storage capacity of the cortex. Because of this, a simple test of how many out of 200 random patterns were correctly recalled by a network consisting of 200 nodes when using normal learning instead of a MTL. Each of the 200 patterns were shown once in order. Figure 3.8 (see page 29) shows how the number of stored patterns varied with the total length of the sleep period in this "perfect" network and in several networks that use a MTL circuit to teach the cortex. Roughly 80 patterns were recalled using regular learning with an optimal sleep time.

3.2 Adaptation performance

In the case of perfectly orthogonal memory patterns, a correctly tuned adaptation projection cycles through all input patterns, see figure 3.1. But if the input patterns are randomly chosen, some patterns will be more similar than others, which will affect the iteration. This can be seen in figure 3.2 and 3.3. The distribution of exposure times during a typical simulation using random patterns can be seen in figures 3.4 and 3.5. Beside the fact that more than half of the patterns were never visited, it is also worth noting that the total visiting time varies greatly between the patterns. The visiting time of each pattern is strongly related to uniqueness of the pattern, as shown by Sandberg [31]. This tells us it is important to choose an MTL encoding where input patterns are as orthogonal as possible.

3.3 MTL size scaling

In the previous sections, it has been noted that the display time varies greatly between patterns during adaptation, and that these differences are correlated to the uniqueness of the patterns. This is taken to the extreme in the case of completly orthogonal patterns, which as can be seen in figure 3.1 are vgery evenly distributed.

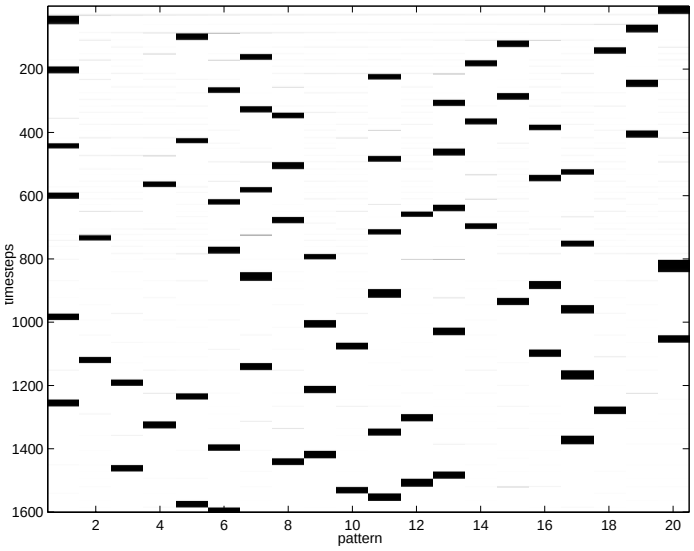


Figure 3.1. Adaptation in the MTL using orthogonal input patterns. The figure shows how close the current activity is to every pattern at different timesteps. As can be seen, the MTL converges completely to a given pattern for about 20 timesteps, after which it quickly converges to a different pattern. All patterns are visited multiple times.

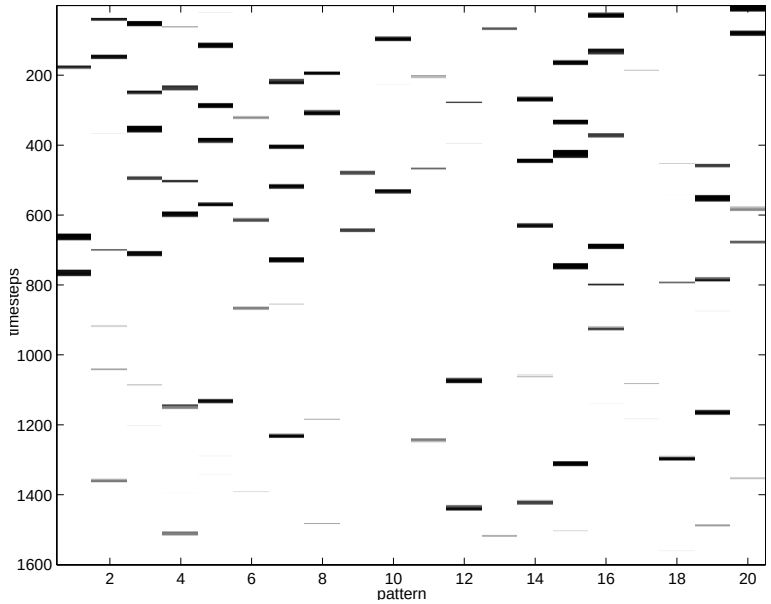


Figure 3.2. Adaptation in the MTL using random input patterns. The figure shows how close the current activity is to every pattern at different timesteps. As can be seen, the MTL converges completetly or partially to a given pattern for 1-20 timesteps, after which it converges to a different pattern. Not all patterns are visited.

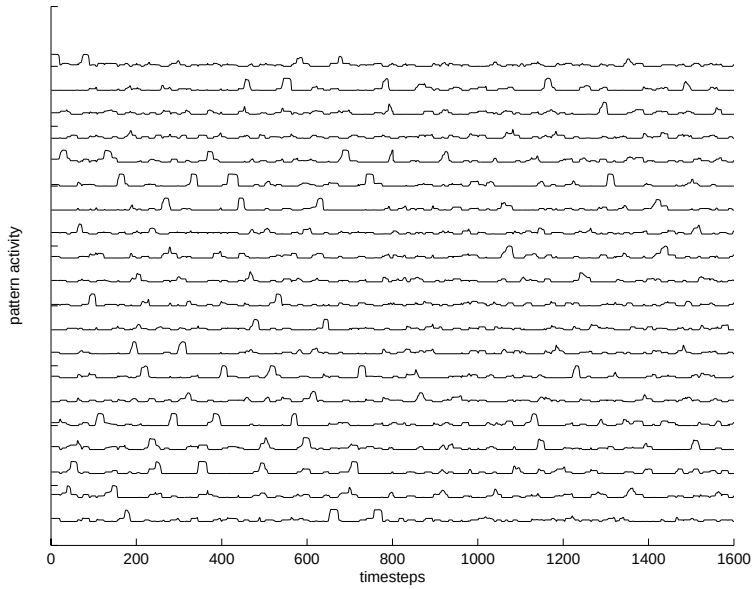


Figure 3.3. The same data as in figure 3.2, but using a different visualization. Notice the background activity. Whenever one pattern is being deactivated, several other patterns will simultaneously increase in strength, but eventually, one of them will overtake the others.

Random patterns tend to become more orthogonal as their size increase. Because of this, one can expect that increasing the MTL size would improve adaptation performance when using random patterns. Figure 3.6 shows us that there is a notable increase in performance when increasing the MTL size up to about 120 nodes.

When using orthogonal patterns, only a single neuron is needed for each memory to be stored in the cortex. Even when using such a small network, the adaptation performance is better than with a random MTL of much larger size, as can be seen in figure 3.8.

3.4 MTL to cortex projections

Many different static projections from the MTL to the cortex are possible, but the exact implementation turned out to have a limited effect on the performance of the system. For simplicity's sake, a subset of the cortex population, equal in size to the MTL, was chosen and a one to one projection between the two populations was made. The rest of the projections were chosen randomly, with a predetermined number of active connections to each cortex neuron. For an example projection, see figure 2.2. The result, as seen in figure 3.7, was that a sparse projection performs poorly when compared to a pure 1 to 1 subset projection, and when the projection density increases, the number of patters learned approached that of the baseline

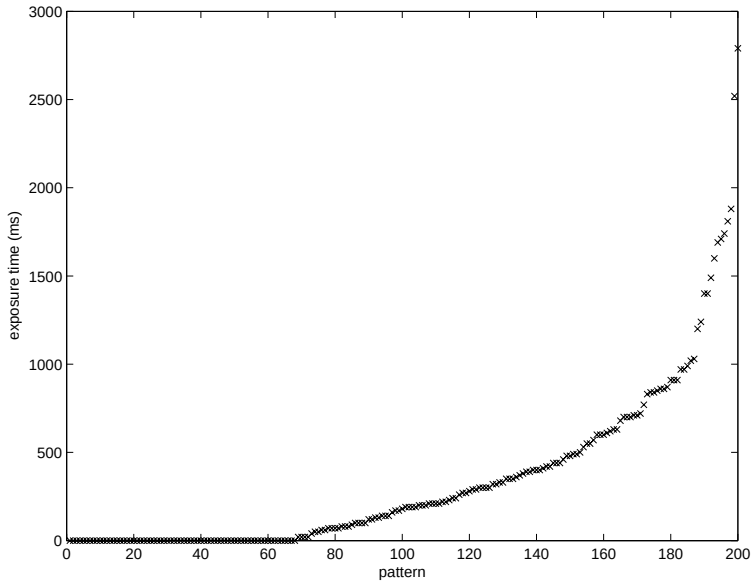


Figure 3.4. The exposure time (the time a pattern is active in the MTL and projected onto the cortex) of each pattern during a typical simulation using random input patterns. Notice that almost half of the patterns are never shown, and that among the patterns that are shown, there is a strong skew.

case. It would seem the static projections do not take advantage of the fact that the cortex is larger than the MTL. This is obviously not a desirable situation since the size difference between the real-life MTL and cortex is very large. Depending on how large a part of the MTL and the cortex respectively are assumed to be important in memory storage and consolidation, the MTL to cortex size difference is of the order 1 to 100 [29].

A more interesting situation arises when the connection from the MTL to the cortex is plastic. Figure 3.8 shows the cortex performance using plastic MTL to cortex projections and various MTL patterns. As a reference, it also shows the performance of the "perfect" MTL described in section 3.1. As can be seen, the network performs substantially better when using plastic MTL to cortex connections.

3.5 Cortex scalability

A set of simulations was run on networks ranging in size from 100 to 400 neurons, the results are displayed in figure 3.9. As can be seen, there is no significant difference in the number of patterns learned between the different networks when using a static MTL to cortex projection. When using plastic connections the performance increases steadily with the number of nodes up to about 250 nodes. It seems that the cortex size is the limiting factor of the network up until 250 nodes, where other

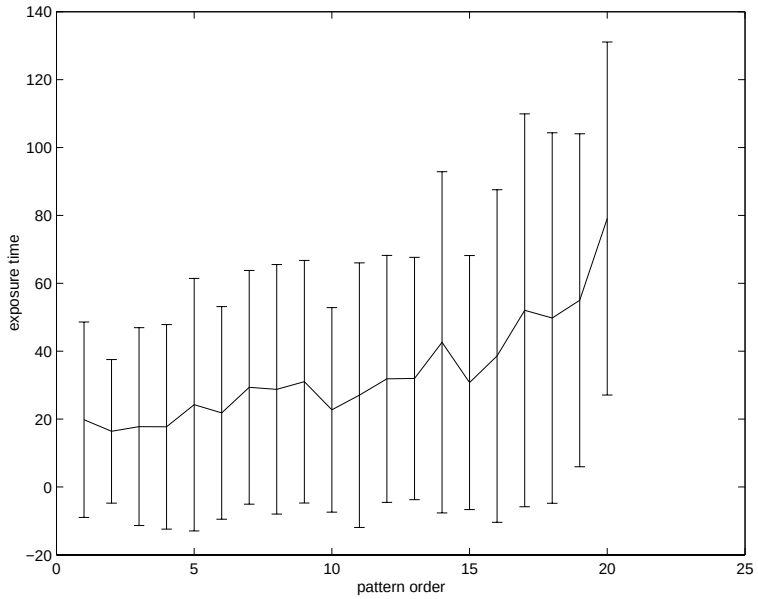


Figure 3.5. The exposure time (the time a pattern is active in the MTL and projected onto the cortex) of patterns plotted against time of day when the pattern was shown. As can be seen, there is a strong tendency towards showing only the last patterns of each day.

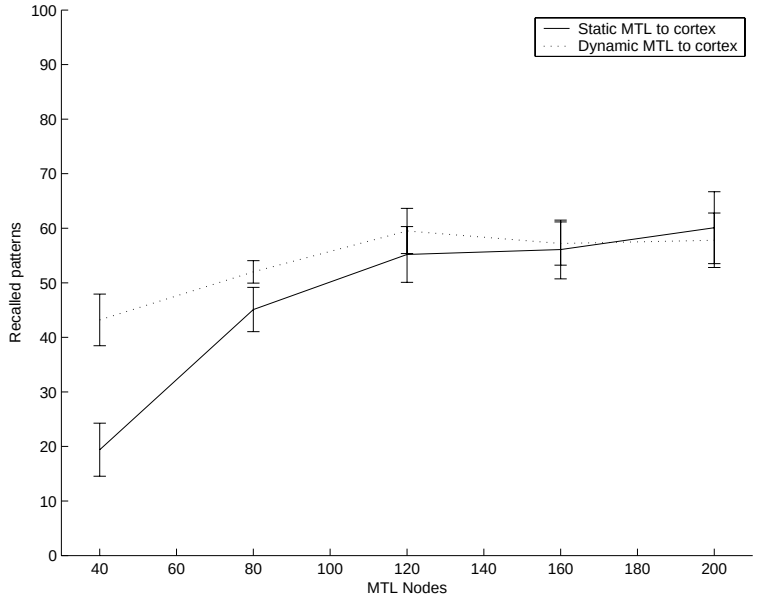


Figure 3.6. The number of patterns learned as a function of the number of MTL nodes.

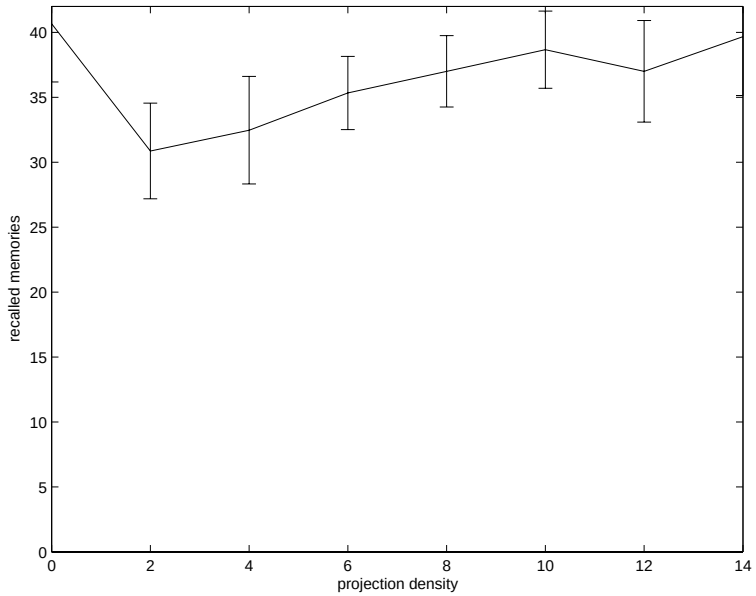


Figure 3.7. The number of patterns learned as a function of the projection density. Note that even with a projection density of 0, there is a 1 to 1 mapping of the MTL and a subset of the cortex.

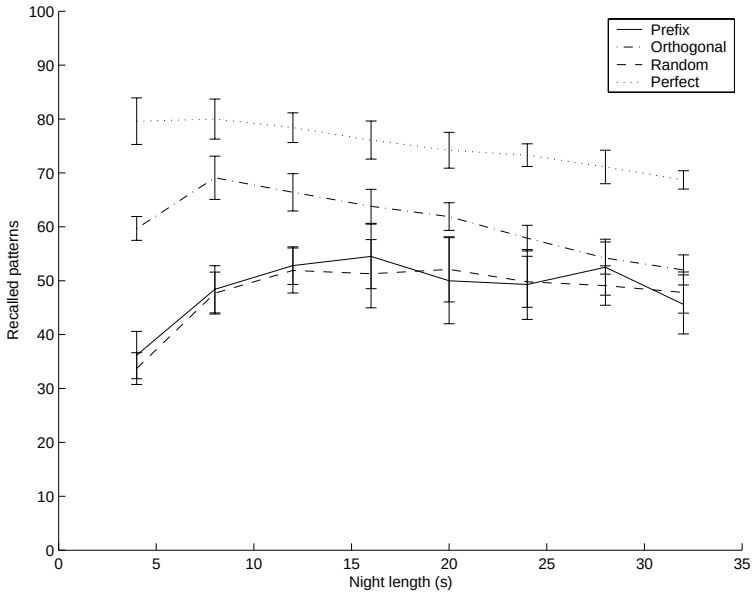


Figure 3.8. The figure shows network performance with different learning rates using various types of plastic MTL to cortex projections. These projections were described in section 2.8.

factors, possibly the adaptation projection, limit the pertformance of the network. Increasing the network size beyond 250 nodes does not increase performance.

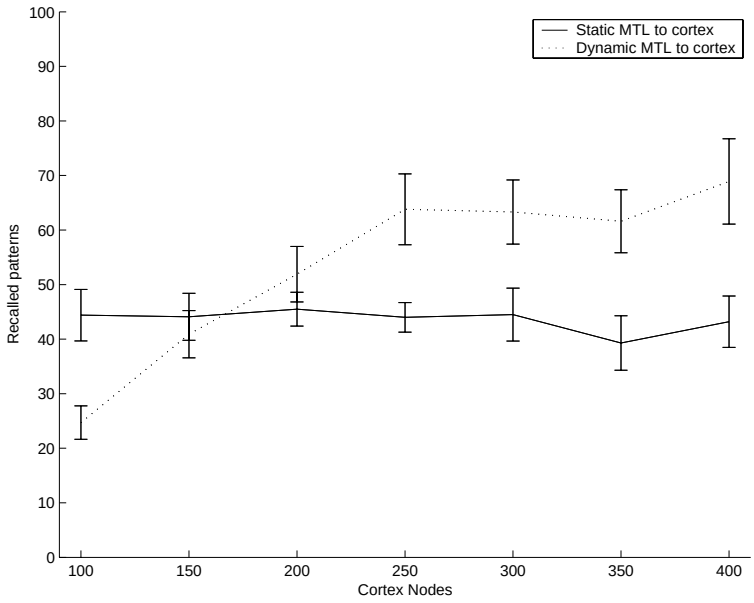


Figure 3.9. The number of patterns learned as a function of the number of cortex nodes.

3.6 Learning rate scalability

The cortex must be many orders of magnitude slower than the MTL if it is to retain memories for several decades, whereas the MTL only needs to retain memories for about one day in rats and about one week in humans. The difference in time for memory extinction is of the order 1 to 10000. For performance reasons, it is not realistic to run all tests with such a large difference in time constants. During most simulations the difference in time constants between the cortex and the MTL has been a factor 20, but it is necessary to verify that the system behaves similarly with larger differences. Therefore a set of simulations were run using varying time constants and sleep time, see figure 3.10. The upscaling of time constants has a slightly positive effect on the performance of the network, so long as the sleep period is chosen correctly.

3.7 Sleep time and learning

The probability of the cortex learning a pattern is very much related to the exposure time of the pattern during sleep. Figure 3.11 shows the pattern exposure time and whether each patterns could be correctly recalled during a typical simulation. It can

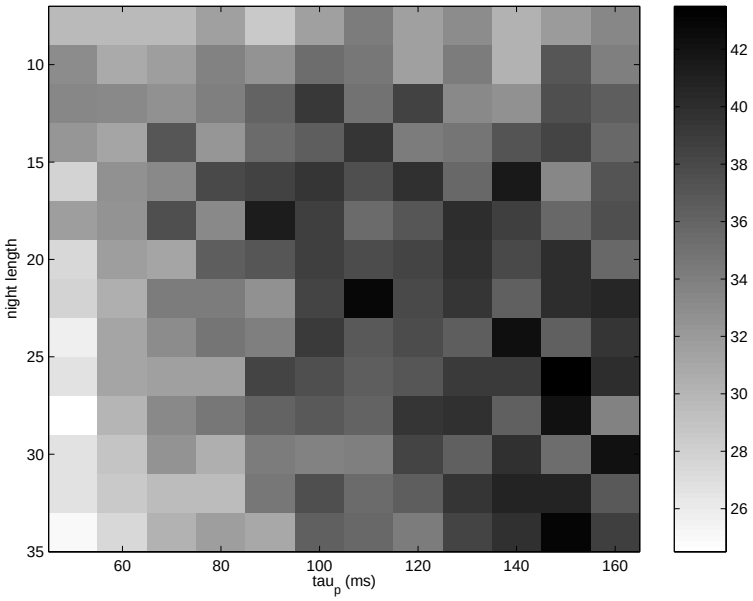


Figure 3.10. The number of recalled patterns as a function of the cortex time constant and the sleeping time.

be seen that above a certain exposure time, the probability of successfully learning a given pattern increases sharply. This threshold time is very close to the time constant of the cortex (100 time steps), a result that should not be surprising.

3.8 Retrograde amnesia

As can be seen in figure 3.12, temporally graded retrograde amnesia is indeed present in the simulation if the MTL circuitry is deactivated. It can be seen quite clearly that memory performance for the last day is significantly impaired with the simulated lesion. This result should be compared to the biological equivalent, such as those in figure 1.3. The difference in timescale between this simulation and the biological result depends on the choice of network parameters. Figure 1.3 displays results from a simulation using the default of 20 patterns per day.

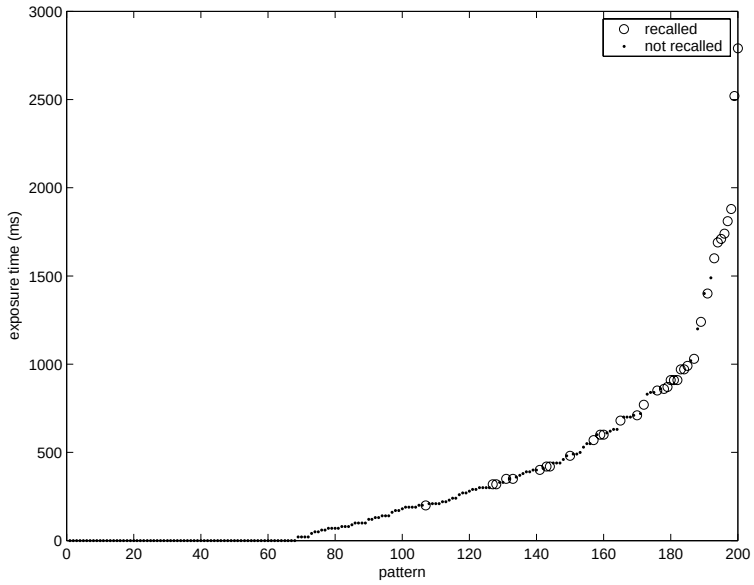


Figure 3.11. Exposure time and learned patterns. It can be clearly seen that the probability of learning a pattern is strongly dependent on the exposure time.

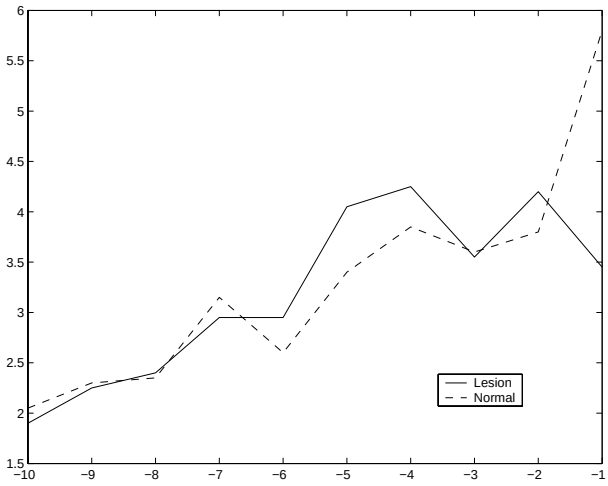


Figure 3.12. Simulated retrograde amnesia. The deactivation of the MTL circuitry in the simulation has a clear impact on recall performance.

Chapter 4

Discussion and conclusions

'The time has come,' the Walrus said,
'To talk of many things:
Of shoes — and ships — and sealing wax —
Of cabbages — and kings —
And why the sea is boiling hot —
And whether pigs have wings.'

— Lewis Carroll

If the memory storage capacity is proportional to the number of connections (synapses) in a network, then the human brain should be able to store about $1.5 \cdot 10^{10}$ times as many memories as the network in this thesis. Given a maximum capacity for the simulated network of about 100 patterns, and a life expectancy on the order of 100 years, we should be able to store $4 \cdot 10^7$ memories per day, or 475 patterns per second. This number obviously exceeds any number which could be expected in real life. Even if only a very small part of the brains synapses are used for memory storage, and if the synapses memory storage functionality is overloaded on top of other functions, it should be fairly obvious that our capability of recalling old memories is not limited by the maximum memory storage capacity of the brain. One common answer is that the brain is very selective about what to store.

A static projection from the MTL to cortex did not give acceptable performance. The best performance when using static projections was obtained by using a simple 1 to 1 connection between the MTL and a subset of the cortex, and no other MTL to cortex connections. It seems that when using static MTL to cortex projections, the network does not benefit from having a cortex which is larger than the MTL. This can also be seen from the very large performance decrease with a smaller MTL showed in figure 3.6.

The performance of the network is much better when using plastic MTL to cortex projections. Depending on the type of pattern used in the MTL, between 60 and 80% of the networks maximum capacity can be reached. The network is robust with regard to its parameters.

Increasing the cortex time constants seems to present no problem for the network, so long as the sleep phase is increased accordingly. If this is the case, the recall rate increases slightly with increasing cortex time constants. This suggests that learning at two different timescales does not present a problem.

When using plastic connections, increasing the size of the cortex also increases the storage capacity of the network. This means that the plastic connections do not limit the performance of the network. Instead, the limiting factors seem to be the cortex size and the adaptation performance.

One possible way of improving adaptation performance is using more than one adaptation projection in the MTL. Multiple projections with different time constants could conceivably improve the performance of the system. Unfortunately, this does not seem to be the case. A large set of simulations with two adaptation projections where tried, and no improvement was evident.

The network exhibits temporally graded retrograde amnesia. Because of the adaptation behavior during sleep, the same neurons which are active during learning are active during sleep periods shortly after learning, similar to place cell firing in rats. The pattern is more drawn out in higher animals such as monkeys and humans, where the retrograde amnesia can draw out for several days or even weeks. This does not seem to be the case in the simulations, however. There are at least two explanations for this. Firstly, when storing new patterns, older ones are simultaneously weakened. This means that some degree of repetition is necesary to keep a constant level of recall. Secondly, the patterns that are repeated by the MTL one or more nights after initially learning them are usually the patterns that where most strongly learned by the MTL, and therefore these are the patterns that the cortex most probably learned already on the first night. In conclusion, the network behavior is similar to that of primitive test subjects.

The optimal night length for a given network varies depending on what type of MTL encoding is used. When using orthogonal patterns, the optimal night length is shorter and the interval sharper than with other MTL encodings. The explanation for this might lie in the more even distribution of display times between patterns with an orthogonal MTL. When the orthogonal MTL has shown all patterns a sufficient time for the cortex to learn them, the uneven display times using prefix patterns in the MTL might mean that some patterns still have not been shown. Thus, memory performance might continue to improve using prefix patterns because new patterns are being shown for the first time, even though other patterns are being overlearned.

4.1 Predictions

The networks used in this thesis have an optimal consolidation time period. If the consolidation time is too short in comparison to the amount of data to be stored, storage will be strongly degraded. If the consolidation time is too long, storage will also be slightly degraded. This should be verifiable in test animals by controling how long they sleep each night. Making sure that a test animal sleeps longer than

normally may pose some difficlty, becuse any drugs given to the animal to control its sleep-cycle may also affect it's memory consolidation in other ways.

4.2 Future work

There are several future research possibilities in the same field.

- The adaptation performance is unsatisfactory. Increasing the size of the MTL has a positive influence, but a more economical solution would be desirable. One can imagine several possible ways to improve on the adaptation performance, including multiple adaptation projections as previously discussed. There are also several other ways to iterate over the MTL data, like resetting the MTL to a random state after a given interval, or performing a more detailed simulation of synaptic depression.
- Implement a network that is able to store and retrieve sequences of patterns, not just patterns.
- Investigate the properties of larger networks and a larger number of patterns.
- Investigate if the relevance of pattern can be modulated in the MTL. Increase the probability of the cortex learning the most important memories.

References

[1] Aristotle. *On Memory and Reminiscence.* Green Lion Press, 350.

[2] Aristotle. *On the parts of animals.* Oxford University Press, 350.

[3] Aristotle. *On the Soul (de Anima).* Peripatetic Press, 350.

[4] A.D. Baddeley. Memory. In R. Wilson and F. Keil, editors, *Encyclopedia of the Cognitive Sciences.* MIT Press, Cambridge, MA, 1999.

[5] C. Bell, V. Han, Y. Sugawara, and K. Grant. Synaptic plasticity in a cerebellum-like structure depends on temporal order. *Nature*, 387:278–281, 1997.

[6] A. Bibbig. *Hippocampal two-stage Learning and Memory Consolidation*, 1996.

[7] A. Bibbig, T. Wennekers, and G. Palm. A neural network model of the cortico-hippocampal interplay and the representation of contexts. *Behav Brain Res*, 66(1-2):169–75, Jan 23 1995.

[8] T.V.P. Bliss and A.R. Gardner-Medwin. Long-lasting potentation of synaptic transmission in the dendate area of unanaesthetized rabbit following stimulation of the perforant path. *J. Physiol.*, 232:357–374, 1973.

[9] T.V.P. Bliss and T. Lømo. Long-lasting potentation of synaptic transmission in the dendate area of anaesthetized rabbit following stimulation of the perforant path. *J. Physiol*, 232:331–356, 1973.

[10] J. Brown. Some tests of the decay theory of immediate memory. *Quarterly Journal of Experimental Psychology*, 10:12–21, 1958.

[11] B. Cartling. Neuromodulatory control of interacting medial temporal lobe and neocortex in memory consolidation and working memory. *Behavioural Brain Research*, 126:65–80, 2001.

[12] N.J. Cohen and L.R. Squire. Preserved learning and retention of pattern-analyzing skill in amnesia: dissociation of knowing how and knowing that. *Science*, 210:207–210, 1980.

[13] H. Eichenbaum and N.J. Cohen. *From Conditioning to Conscious Recollection: Memory Systems of the Brain.* Oxford University Press, 2001.

[14] J.M. Fuster. *Memory in the Cerebral Cortex.* MIT Press, Cambridge, Massachusetts, 1995.

[15] D.O. Hebb. *The Organization of Behavior.* John Wiley Inc., New York, 1949.

[16] J. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the Theory of Neural Computation*, volume 1 of *Lecture Notes.* Addison-Wesley, Santa Fe Institute for studies in the sciences of complexity, 1991.

[17] K.L. Hoffmanm and B.L. McNaughton. Coordinated reactivation of distributed memory traces in primate neocortex. *Science*, 297:2070–2073, 2002.

[18] J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc Natl Acad Sci U S A*, 79(8):2554–8, April 1982.

[19] J.J. Hopfield and D.W. Tank. Neural computation of decisions optimization problem. *Biological Cybernetics*, 52:141–152, 1985.

[20] D.H. Hubel and T.N. Wiesel. The functional architecture of the macaque visual cortex. The Ferrier lecture. *Proc. Royal. Soc. B*, 198:1–59, 1977.

[21] W. James. *The Principles of Psychology.* Holt, Rinehart and Winston, New York, 1890.

[22] P. Lavenex and D.G. Amaral. Hippocampal-neocortical interaction: A hierarchy of associtivity. *Hippocampus*, 10:420–430, 2000.

[23] W. Levy and O. Steward. Temporal contiguity requirements for long-term associative potentiation/depression in the hippocampus. *Neuroscience*, 8:791–797, 1983.

[24] D. Marr. Simple memory: a theory for the archicortex. *Philosophical Transactions of the Royal Society of London, B*, 262::23–81, 1971.

[25] D. McCormick, B. Connors, J. Lighthall, and D. O'Prince. Comparative electrophysiology of pyramidal and sparesely spiny stellate neurons of the neocortex. *Journal of Neurophysiology*, 54:782–805, 1985.

[26] R.G.M. Morris. Learning, memory and synaptic plasticity: cellular mechanisms, network architecture and the recording of attended experience. In David Magnusson, editor, *The lifespan development of individuals: behavioral, neurobiological, and psychosocial perspectives*, chapter 7, pages 139–161. Cambridge University press, 1996.

[27] J.M. Murre. Tracelink: A model of amnesia and consolidation of memory. *Hippocampus*, 6:675–684, 1996.

[28] L. Nadel and M. Moscovitch. Hippocampal contributions to cortical plasticity. *Neuropharmacology*, 37:431–439, 1998.

[29] B. Pakkenberg and H.J.G. Gundersen. Neocortical neuron number in humans: Effect of sex and age. *Journal of comparative neurology*, 384:312–320, 1999.

[30] L.R. Peterson and M.J. Peterson. Short-term retention of individual verbal items. *Journal of Experimental Psychology*, 58:193–198, 1959.

[31] A. Sandberg. *Baysian Attractor Neural Network Models of Memory*. PhD thesis, NADA, Royal Institute of Technology, Stockholm, Sweden, May 2003.

[32] D.L. Schachter and E. Tulving. What are the memory systems of 1994? In D.L. Schachter and E. Tulving, editors, *Memory Systems*, pages 1–38. MIT Press, Cambridge, MA, 1994.

[33] W.B. Scoville and B. Milner. Loss of recent memories after bilateral hippocampal lesions. *J. Neurol. Neurosurg. Psychiatry*, 20:11–21, 1957.

[34] T. Shallice and E.K. Warrington. Independent functioning of verbal memory stores: a neuropsychological study. *Quarterly Journal of Experimental Psychology*, 22:261–273, 1970.

[35] Y. Shin and J. Ghosh. The pi-sigma network: An efficient higher-order neural network for pattern classification and function approximation. In *Proc. IJCNN*, Seattle, July 1991.

[36] W.E. Skaggs and B.L. McNaughton. Replay of neuronal firing sequences in rat hippocampus during sleep following spatial experience. *Science*, 271:1870–1873, 1996.

[37] R.E. Smith. *Psychology*. West Publishing Company, St. Paul, 1993.

[38] L. R. Squire, B. Knowlton, and G. Musen. The structure and organization of memory. *Annu Rev Psychol*, 44:453–95, 1993.

[39] L.R. Squire. Memory and the hippocampus: A synthesis from findings with rats, monkeys, and humans. *Psychological Review*, 99:195–231, 1992.

[40] L.R. Squire and P. Alvarez. Retogerade amnesia and memory consolidation: a neurobiologival perspective. *Current Opinion in Neurobiology*, 5:169–177, 1995.

[41] L.R. Squire and S. Zola-Morgan. The medial temporal lobe memory system. *Science*, 253:1380–1386, September 1991.

[42] E. Tulving. Episodic and semantic memory. In E. Tulving and W. Donaldson, editors, *Organization of memory*, pages 382–403. Academic Press, New York, 1972.

[43] R.W. Williams and K. Herrup. The control of neuron number. *Ann. Review Neuroscience*, 11:423–453, 1988.

[44] S. Zola-Morgan and L.R. Squire. The primate hippocampal formation: Evidence for a time-limited role in memory storage. *Science*, 250:288–290, 1990.