NTNU
Innovation and Creativity

Reducing catastrophic forgetting in neural networks using slow learning

Mikael Eikrem Vik

Master of Science in Informatics
Submission date: June 2006
Supervisor: Keith Downing, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Abstract

This thesis describes a connectionist approach to learning and long-term memory consolidation, inspired by empirical studies on the roles of the hippocampus and neocortex in the brain. The existence of complementary learning systems is due to demands posed on our cognitive system due to our environment and the nature of our experiences. It has been shown that dual-network architectures utilizing information transfer successfully can avoid the phenomenon of catastrophic forgetting occuring in multiple sequence learning. The experiments involve a Reverberated Simple Recurrent Network which is trained on multiple sequences with memory reinforcement by means of self-generated pseudopatterns. My focus will be on the implications of how differentiated learning speeds affect the level of forgetting, without explicit training on the data used to form the existing memory.

Preface

Before I came to Trondheim and NTNU I had finished a bachelor in Computer Engineering from Bergen University College. Along with my interest in programming I also brought with me a long-existing fascination of science fiction and visions of the future, often dystopic, with artificial intelligence as a recurring theme. This has set me up with an urge to discover the how far we have progressed in the creation of artificial intelligence, and see where the current frontiers are.

I am also intrigued by the philosophical issues around the definition of intelligence and the perception of our selves. What is the essence of our personalities, is it just electrical signals propagating through the neurons of our brains? These last years have made me realize some of the possibilites and limitations of artificial intelligence. Knowledge from the field does give you a more realistic understanding of what can be achieved in the near future. In my opinion the neural basis of cognition constitutes the most promising endavour in our quest to develop real artificial intelligence in self-aware systems.

The master studies in artificial intelligence and learning have provided me with the opportunity to explore this field in-depth. I have not only read about breakthroughs and achievements of real AI systems, but also had the opportunity to get hands-on experience implementing accomplished systems. It has been rewarding to do my masters under someone with a genuine interest in my field of my choice. I would like to say thanks to my advisor, Professor Keith Downing, for our meetings and conversations during the last year. I have enjoyed the freedom to explore different directions of the research area, but always received good advice when the possibilites became overwhelming and my ambitions exceeded my capabilities. I am also grateful for my sessions with Diego Federici, who acted as a stand-in advisor during Keith's sabbatical.

Thanks for the inspiration and encouragement.

Trondheim, May 2006

Mikael Eikrem Vik

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The role and functionality of the hippocampus has attracted a large body of research and is an active topic in both neuroscience and information science. While brain scientists use connectionist models to test models of brain function, information scientists are interested in how to build more effective AI learning systems and push the limits of what they can achieve.

Brain research has looked to the implications of lesions and damage to the hippocampus and related areas when exploring its role in memory, as well as electrical activity recordings by EEG or electrodes in vivo (only in animals). There is good evidence that the hippocampus in rats encodes a spatial map which enables the rat to navigate. It has been proposed that the hippocampus plays an important role in the formation of declarative memories and their incorporation into existing brain structures.

Connectionist system are biologically inspired and represent simplified models of the mind. This neural basis of cognition constitutes in my opinion, the most promising approach to creating intelligent and self-aware systems. McClelland, McNaughton & O'Reilly (1994) investigates the complementary roles of the hippocampus and neocortex. Their model propose that knowledge is not organized hierarchically, but by incremental storage of concepts that depend on the responses the learning system must learn to produce. Slow interleaved learning is presented as a way of getting AI systems to generalize properly during learning. thereby achieving the task of extracting the underlying structure of a domain. During extraction the network often experiences interference between the patterns drawn from a domain. French, Ans, Rousset & Musca (2002) describes how dual-network architectures can utilize information transfer to overcome the problem of catastrophic forgetting in multiple sequence learning.

1.1 Motivation

Due to the diversity of background theories and possible directions available in the literature it is impossible to capture every subtlety in one single experimental setup. This made it important to focus the project and concentrate on a well-established theory and find a relevant test problem. My first encounter with a connectionist approach presented the need for complementary learning systems in the brain (McClelland et al. 1994). This article suggested that the hippocampus provides an initial storage of memories which later are transferred into long-term storage in the neocortex. The dual-network architecture of French et al. (2002) seemed to be a good starting point for further investigations. With its use of two networks the system is able to overcome catastrophic forgetting in multiple sequence learning.

The main goal of this thesis is to investigate how *learning speed* affects the memory of an artificial neural network. Dual-network models use a very elaborate process of memory transfer which result in stable and resistant back-up memory. I will therefore try to make the memory more volatile and examine what the consequences are. A network is presented with the task of learning of multiple sequences and interleave new learning with pseudopatterns. The purpose of these pseudopatterns is to enforce the current memory of the network. By the use of a reverberation technique they settle into attractor states that reflect the network function.

1.2 Organization

This thesis is organized as follows; first, a theoretic background from neuroscience with a focus on the interplay between long-term memory storage in the neocortex and the rapid memory acqusition abilities of the hippocampus. This is followed by an introduction to artificial neural networks as a model of the mind. Related work includes connectionist approaches to learning, models of hippocampal function in rats, and last, computational principles of neural networks. Second, I present how the hippocampus might act as a teacher to the neocortex, along with how dual-network architectures successfully has overcome the problem of catastrophic forgetting. Third, the design and implementation of the experimental framework and the methods, is described. Fourth, the performance of the networks in the simulations is analysed and discussed. Finally, the conclusion summarize the results and describe some of their implications, and how this might affect the direction of future research.

Chapter 2

Background

2.1 The quest for common sense

Creating real self-aware and conscious artificial intelligence is the utltimate goal of the AI community. Can actual intelligence ever emerge and thus result in artificial systems which achieve common sense? Artificial Neural Networks (ANNs) are a biological approach to creating a model of the mind and cognition, but current research is usually not aimed at creating the newest and most advanced artificial life forms. ANNs are useful when exploring and testing biological theories of brain function. Although it is presently impossible to capture the full complexity of the human brain, or even the rat brain, theories can be tested and cast aside, or preferably developed further.

2.2 Biological foundation

According to Burgess & O'Keefe (2003), the hippocampus is *"the primary region in the mammalian brain for the study of the synaptic basis of memory and learning"* (page 1). Much of the interest in the hippocampus and its role in learning is due to the existence of memory deficits produced by lesions or damage to the hippocampus and related structures. This includes anterograde amnesia affecting the acquisition of new memories, but also retrograde amnesia causing the forgetting of events within a temporal window prior to the hippocampal insults.

2.2.1 Architecture of the hippocampus

This brief introduction to the architecture of the hippocampus is based on Rolls & Treves (1998) and the view of hippocampus as a buffer store for memories before they are consolidated into neocortex as proposed by McClelland et al. (1994).

The hippocampus is located in the temporal lobe of the brain. It receives *"inputs from virtually all association areas in the neocortex"* (Rolls & Treves 1998, page 107). This information has already gone through extensive processing before it reaches the hippocampus through the perforant path of the parahippocampal gyrus and entorhinal cortex. It provides the opportunity for stimuli originating from sensory systems such as the visual, auditory and olfactory systems to be associated together rapidly. Information flow then continues to the different stages of the hippocampus, which are formed from separated sheets of cells.

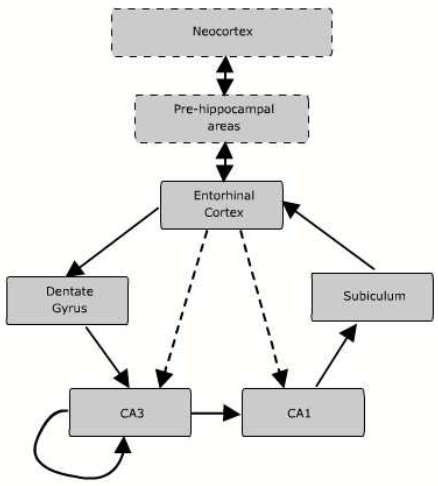


Figure 2.1: Architecture of the hippocampus

It is proposed that cells in the dentate gyrus produce the sparse, but yet efficient, representation which is required for the autoassociation in the CA3 stage to perform well. Competetive learning with Hebb-like qualities acts to remove redundancy. It has also been hypothesized that it is this competetive learning which allows overlapping inputs to the hippocampus to be separated even in the presence of non-linearity. The dentate granule cells project to the CA3 cells via mossy fibres. This provides a small number of strong inputs and may be efficient in forcing firing patterns onto the CA3 cells during learning.

If the hippocampus is provided with only a small cue it can achieve the retrieval of stored firing patterns. The CA3 stage of the hippocampus contains extensive recurrent

connections which enables it to act as an autoassociative memory. This final stage of convergence allows the conjunctive features of an episode to be associated together and stored as one event. Limited storage capacity and its consequences will be discussed more thoroughly in section 3.1.

The structures that follow CA3 should be optimized to preserve information content. Because the entorhinal cortex connects to most of the later stages within the hippocampus it can influence learning in the later stages. Associatively modifiable synapses between CA3 and CA1 provides an intermediate recoding stage, allowing sparse representations produced by CA3 and the previous stages to be redistributed over a larger number of CA1 cells. The integration of reduced CA3 representation with inputs from the EC may be useful both in consolidation and in immediate use of the retrieved memory.

The next stages recode the information and redistribute it over a larger set of neurons. The number of CA1 cells in monkeys expand relative to the number of CA3 cells in the rats, and even more so in humans (Rolls & Treves 1998, page 121). This results in the forward connections leading into the CA3 network showing great convergence, and similarly the backprojections to the neocortex showing great divergence. While there must be a limit to the size and storage capacity of CA3 for it to remain as one efficient auto-associative network, once the information has passed this bottleneck this is no longer a requirement. Information content continues through the subiculum and is maintained while the sensitivity to noise and information loss in the next stages is reduced.

2.2.2 Functional boundaries

The neocortex and the hippocampus have different roles in memory, but how is functionality shared between them? The neocortex is responsible for long-term storage and involved in higher brain functions; spatial reasoning, sensory perception and generation of motor commands. In the case of humans this also involves language and conscious thought. It is not suited for the quick formation of new snapshot quality memories, which would depend on a network of limited size.

There are many theories of what the hippocampus is actually doing. My emphasis will be on McClelland's theory of the hippocampus acting as a buffer store and teacher to the neocortex long-term storage. Figure 2.2 is adopted from McClelland et al. (1994)(page 38). It shows a simple model of the functional boundaries of the hippocampus and the neocortex. The initial strength of synaptic updates in the hippocampus is much higher than those in the neocortex; $S_h(0) \gg S_c(0)$. D_h and D_c is the rate of decay in the hippocampus and neocortex, respectively. C refers to the rate of consolidation. Some initial changes do occur in the neocortex, but these are not large enough for memory storage. The neocortex relies on the hippocampus for these types of memory formation

of which it is not itself capable. The necessity for rapid learning would actually cause catastrophic forgetting in the neocortex. By having the hippocampus act as a teacher for long-term incorporation of memories these problems can be overcome. The strength of the synaptic updates are also indicated by the breadth of the arrows.

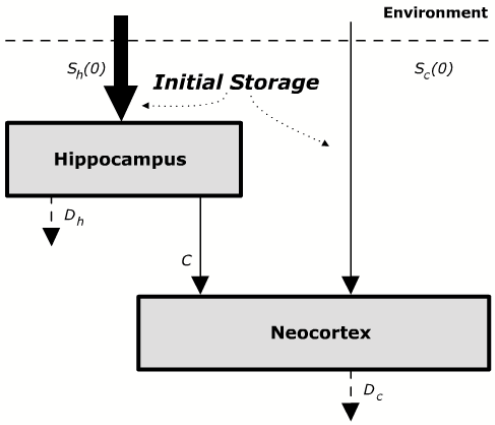


Figure 2.2: Two compartment model of memory storage and decay.

I've already mentioned the limited storage capacity of the hippocampus due to the functionality demands required by the CA3 stage. Hence, memories must be moved out of this area to make place for new learning. If the brain is not to simply forget these types of memory they must be transferred into long-term memory. The process of recall is one such method for memory reinstatement and transfer; backprojections from the hippocampus to the neocortex are one of the two major outputs of the hippocampus, indicated through the presence of dreams and reminiscense (Rolls & Treves 1998, page 99). Recall will be discussed in section 3.1.3. The second, more action-directed, major output involves information stored in the hippocampus in rapid visual to spatial response mapping. This is beside the main focus of this thesis, long-term consolidation.

Rolls & Treves (1998) see the hippocampal CA3 recurrent collateral system as most likely to be involved in memory processing forming new associations between arbitrary events. The hippocampus is necessary for the storage of information characterized as declarative (page 96), including episodic and semantic memory. In plain language this is information regarding *knowing that*, as opposed to procedural, or *knowing how*. There is no impairment to procedural learning when the hippocampus is damaged, but subjects might be unable to reflect and verbalize on what he or she has learnt. This will be presented in more detail in section 2.4.4.

In a discussion of a special declarative task of incidental conjunctive learning tasks O'Reilly & Rudy (2000) uses exploratory behaviour in rats as an example (page 393).

Control rats and rats with damage to the dorsal hippocampus were both tested in an environment containing a set of objects relative to a distinct visual cue. Rearranging the objects prompted exploratory behaviour only in the control rats, but both groups of rats reacted when new objects were introduced to the mix. This shows that the rats depended on the hippocampus to encode the conjunctions of objects necessary to represent the spatial arrangement of the objects, but not the mere presence of the same objects.

2.2.3 Learning mechanisms

Below I present some of the important biological processes involved in the possible hebbian learning mechanisms. The section is based on a summary presented by Rolls & Treves (1998). Long-term potentiation was discovered by Terje Lømo in 1966 and is a potential synaptic mechanism underlying some types of memory formation. It is long-lasting and becomes evident rapidly, typically in less than 1 minute. Although it cannot be said to always be an excact model present in learning, many of its properties are often required.

Long-term potentiation (LTP) and long-term depression (LTD) are useful models of biological learning mechanisms that occur in the brain. Adjustment of connections between neurons, or synapses, are performed to reflect the result of new experiences. LTP indicates an increase, and LTD a decrease, in synaptic strength. They form a basis for synaptic modifications which appear to be synapse-specific and depend on local information. This synaptic modification is in accordance with a hebbian, or associative, form of learning.

Synapses connecting two neurons become stronger in the presence of conjunctive presynaptic and postsynaptic activity. NMDA receptors are crucial to both LTP and LTD because of their role in detecting the existence of such activity. NMDA receptor channels are voltage dependent, which introduces a treshold and thus non-linearity in the firing properties of neurons. The importance of non-linearity is also discussed for ANNs in section 2.3.1.

Joint synaptic activation must exceed a treshold to induce LTP, this becomes evident through some main phenomena. *Cooperativity* means that many small co-active inputs are enough produce sufficient depolarization to exceed the treshold of a neuron. A weak input alone is not enough, but sufficient if there is also a strong input for it to be *associated* with. The *temporal contiguity* of LTP requires pre- and postsynaptic activity to occur at the same time, within a time-window of 500 ms. LTP is also *synapse-specific*, in that only synapses with active inputs to a cell will show traces of potentiation subsequently. On the other hand, inactive synapses to a cell that exceeds the firing treshold will not show LTP even if other inputs are strongly activated.

The activation of NMDA receptors depends on the neurotransmitter glutamate, an amino acid released by the presynaptic terminals. After its initial establishment trough NMDA receptors, LTP is expressed through K-Q receptors. Some evidence of this is provided by infusing drugs, such as the antagonist AP5, to block glutamate. The drug blocks the establishment of LTP in the NMDA receptors as well as subsequent spatial learning mediated by the hippocampus, but the K-Q receptors are not affected. Thus, AP5 will not block subsequent expression of LTP trough the K-Q receptors, only its establishment.

Long-term depression is the process of weakening synapses. There are two types of associative LTD. *Heterosynaptic LTD* occurs in the case of strongly activated postsynaptic neurons and low presynaptic activity. *Homosynaptic LTD* occurs when the presynaptic neuron is strongly active, and the postsynaptic neuron has some, but low, activity. In the first type, the synapse that weakens is another one than the one through which the postsynaptic neuron is activated. For the second type it is the same synapse as the one that is active. These types of LTD are found both in neocortex and hippocampus (Rolls & Treves 1998, pages 7-11). They are required in order to minimize interference between memories held at any one time in the store, rather than in order to gradually delete older memories.

2.3 Artificial neural networks

Artificial neural networks are models of the mind inspired by electrical activity being propagated through biological neuronal networks. The many types of different networks exist because they tailored to fit a diversity of tasks. This presentation will only include those important to the simulations of this thesis.

2.3.1 Feed-forward networks

The general idea behind my simulations are based on feed-forward networks. Perceptrons were the first type of feed-forward networks (Callan 1998). They suffered under very limited capacities due to their inability to solve problems which were not linearly separable. Multi-layer perceptrons employing a non-linear, but continous, activation function was able to overcome these difficulties (figure 2.3).

Units in the hidden layers do not have target values, which made it difficult to train such networks. The problem of assigning blame in the hidden layers was finally solved with the introduction of the supervised backpropagation learning algorithm by Rumelhart, Hinton & Williams (1987). This spawned a new wave of research in neuronal networks.

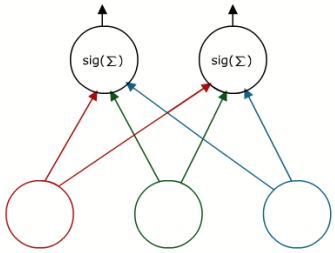


Figure 2.3: Weights connecting two layers of a feed-forward network. Output is calculated by applying the sigmoid activation to a perceptron's summed net input.

The algorithm is described in more detail in section 4.2.5.

Feed-forward networks can be both *auto-associative* and *hetero-associative*. The first type trains the network to reproduce its input on the output units. This is not only useful for pattern completion, but also enables pattern compression in the hidden layer. With the second type of network, training is an effort to map inputs to a different output. These networks are often used to generalize and classify patterns.

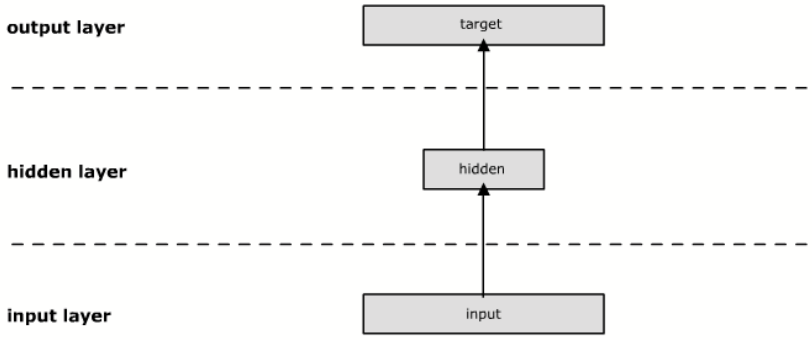


Figure 2.4: Feed-forward network

2.3.2 Simple Recurrent Networks

Simple Recurrent Networks (SRNs) were invented by Elman (1990). They develop an internal representation of time, whereby hidden units recode the input patterns in a way that enables the network to produce the correct output. This representation provides the network with a memory of the previous state of the network. Input to the network

consists of regular input units along with a copy of the previous hidden activation to the context units. Thus, the network must map both external input and internal state to some desired output. The internal representations are implicitly sensitive to temporal context, but still highly task- and stimulus-dependent. The demands of the task are intermixed with the demands imposed by carrying out a time-dependent task.

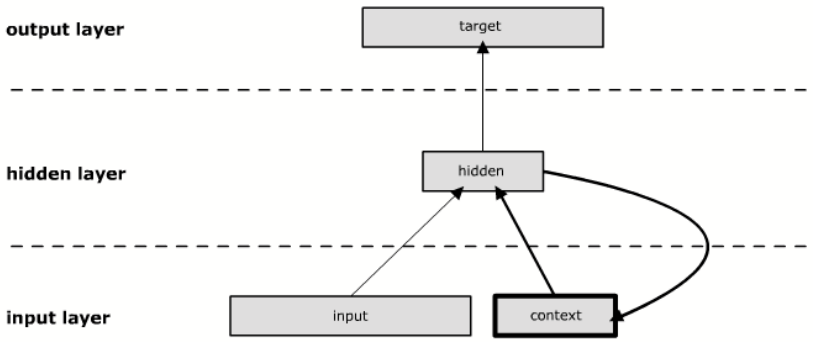


Figure 2.5: Simple Recurrent Network

An SRN is able to sort out ambiguous input in a way a standard feed-forward network is not. Ambiguity provided by identical input patterns can be mapped to different output patterns based on the internal representation of time or, in other words, context. This is an important aspect of the simulations which follow in Chapter 5.

2.4 Related work

I will now refer to some of the key articles and breakthroughs in subsymbolic artificial intelligence and the crossover area of neuroscience and information science. The first section presents a computational motivation for the existence of complementary learning systems in the brain, represented by the hippocampus and the neocortex. The overview presents how their contributions and cooperation result in extended capabilities. Much research has also been directed to the role of the hippocampus as a spatial map enabling navigation in rats. Some of the development and refinement of these models will thus be presented next. I then move on to some of the computational principles involved in neural network training, and aspects of learning speed compared to the resulting learning abilities. The chapter closes with a presentation of the advantages of subdivided networks and the the role of sleep in learning.

2.4.1 Complementary learning systems

In their corner stone article "Why there are complementary learning systems in the neocortex and hippocampus", McClelland et al. (1994) tries to give a clearer computational motivation on the role of the hippocampal learning system for long-term memory consolidation. Is consolidation just an arbitrary parameter, or an important aspect, of learning?

Hippocampal roles

Research on hippocampal roles are vast, but McClelland et al. (1994) build their theories on four main points:

- Some evidence has been provided of hippocampal damage leading to a deficit in new learning, anterograde amnesia, as well as some retrograde amnesia.
- The deficits are selective to some forms of learning, arbitrary associations or conjunctions. This is typically a problem for explicit memories, consisting of episodic and semantic memories, memories that can be verbally described.
- Learning of non-declarative or implicit memories are unaffected, these are gradually acquired skills which influence behaviour without being part of conscious thought.
- Although it is subject to controversy, it has been shown that insults to the hippocampal system, such as lesions or electro-convulsive shock treatment, appear to give rise to temporally graded retrograde amnesia.

Retrograde amnesia affects events preceeding the time of lesion or other insults. In its presence the performance on recent data is worse than performance on very old data. The retrograde gradient in humans can extend over periods of 15 years. It is less substantial in monkeys (4 weeks) and rats (10 days). This result has only been shown for learning which depends on the hippocampus.

Organization of memory in the brain

Based on the data summarized above, McClelland presents a theory of the organization of memory in the brain. The neocortical system consists of a distribution of interconnected neural populations. Usually these connections are bidirectional projections allowing activity in one region to give rise to activity in other regions. Representations

of experiences consist of widely distributed patterns of activity. Elicitation of patterns of activation are propagated over the population of neurons in different regions. For instance, reading a word produces a pattern in the visual cortex that might elicit activity corresponding to the sound of the word in some other region. This type of memory is content addressable, so that only a limited aspect of the content of the memory can serve to retrieve the entire memory.

The popular view of short-term memory loss is usually a manifestation of deficits in long-term memory encoding. The knowledge underlying all cognitive capacities must ultimately be represented in connections of the neocortex. Although the initial activations in the neocortex will produce slight changes, these are not sufficient for permanent storage. Small adjustments to these neocortical connections are made during initial information processing, but only show subtle effects incapable of storing an event. Learning is gradual and these adjustments are not sufficient for adequate performance.

The contents of a specific episode are initially stored in the hippocampus, which acts not only as a memory store, but also as a teacher. Associations between activations in neocortex and a corresponding pattern of activation in the hippocampal system is established through bi-directional implastic pathways. Long-term potentiation makes this pattern an attractor. Reinstatement of attractor patterns in the hippocampus may occur when needed or in offline situations such as sleep, providing a training trial for the neocortex. Over time small adjustments accumulate to acquired cognitive skills.

Results of this learning includes preservation of common aspects across different contexts, alongside inclusion of idiosyncrasies of the different episodes.

Discovery of shared structure through interleaved learning

An example of how shared structure can be discovered through interleaved learning was performed using a connectionist network. The goal was to learn to distinguish relations between concepts and their respective abilities.

Semantic networks have tried to organize knowledge hierarchically, but people's knowledge is not represented in this way (McClelland et al. 1994, page 12). The relations between concepts are not always predictable from their surface properties. Structure is not constructed on the basis of an omniscient perspective, but tend to rely on on the relations between concepts which are known at the time. Similar to this the connectionist models capture conceptual similarity. They depend on generalizations assigned by the internal representations in the hidden layers and the responses the system must learn to produce, and not hierarchic structures.

Weight connection adjustment is based on shared structure common to the environ-

ment from which associations are sampled. Conceptual similarities are captured in the connections from input to hidden units, a mapping from inputs to an internal representations. Connections from hidden to output units captures the mapping from internal representations of concepts to response patterns on the output layer. Similarity is thus obtained, not because of intrinsic similarities in input, but because of the similarity in the responses the network must learn to make. A previously trained concept network was presented with the task of of learning the internal representation of a new input based on only one of the relations. After training it was able to generalize and supply the correct answer for all the other relations as well.

Catastrophic interference

Interleaved learning systems are not appropriate for rapid acquisition of memory. Such tasks leads to catastrophic interference and a loss of ability which is far from gradual. Performance on old memories is completely demolished even before the acquisition of new memories is showing results. Reducing crosstalk between input patterns can reduce this problem (French 1991).

Another approach reduces the overlap between hidden representation (French 1991). Activation sharpening is used to develop semi-distributed representations in the hidden layer. A certain fraction of the hidden units with the highest activations are selected for a slight activation increase, the rest of the units are slightly decreased. Sharpened nodes have significant effect on the output, resulting in less interference with the weights of the network than for an unsharpened network. Learning is thus forced through a representational bottleneck which limits the amount of information in the hidden representations. Although this helps on the problem of catastrophic interference, it does reduce the network's ability to generalize.

Focused learning of new information can be acquired fairly quickly, but this is destructive to previous concepts (McClelland et al. 1994, page 21). Interleaving new knowledge with existing knowledge allows the network to carve out a place for it in the existing structured system. This is an important aspect of the role of the hippocampus in learning and memory. It may provide an initial storage of memory which allows a slow incorporation process which avoids the rapid synaptic changes that would lead to catastrophic interference with previously acquired knowledge.

Theory comparison

The phenomenon of temporally graded retrograde amnesia calls for a theory where the hippocampal system has a time-limited role, not only assigning distinct cortical repre-

sentations to novel conjunctions and telling the neocortex what to learn. This does not explain the necessity for slow learning and the occurence of retrograde amnesia. Three different kinds of roles have been suggested for the hippocampus; selecting representation at the time of storage, a rapid acquisition which is not possible in the neocortex without interference, and a time-limited role in the formation of neocortical representations (McClelland et al. 1994, page 46-8).

McClelland's theory places itself within the last category and is also the first to provide a computational account of why the involvement must be temporally extended. Hippocampus facilitates snapshot quality episodic memory. Together with related structures it exists to hold episodic memories while avoiding interference with structured knowledge in the neocortex. Reactivation and pattern completion of hippocampal memories allows for reinstatement in the neocortex, not only in task-relevant circumstances, but also in task-irrelevant contexts and offline activity. Slow updates in the long-term storage area allow the extraction of general environment tendencies and similarities between situations, thus discovering statistical structure.

The model fits with empirical findings and life in general. Learning rates that change throughout life can explain differences in human learning related to age. Babies are learning machines which acquire new knowledge with impressing speed, but the result also includes a memory which doesn't contain much from the first years of living. As the learning rate decreases with age, one experience a decreased ability to adapt to new situations and learn new things fast.

2.4.2 Spatial models of the hippocampus

The most serious result of hippocampal damage in rats appears to be a deficit in spatial navigation. Burgess & O'Keefe (2003) provide an overview of how the hippocampus in rats acts with spatial models of neuronal activity which might be used in navigation.

By recording the firing from single units one can see that this firing is restricted to small portions of the rat's environment. Such units, for example in CA3 and CA1, are thus given the name place cells (PCs). Their activity can be manipulated by rotation of environmental cues or variation of the environment size. PC firing rates appear to depend on the rat's direction of travel as well as its location. Recording the θ rythm of the EEG signal reveals that PC firing coincides with movement. Firing in a late phase places the position ahead of the rat, and firing in an early phase behind. This inspired the introduction of cognitive maps to explain place learning in rats.

A model of PC firing can be built on how the spatial firing might develop as the rat explores the environment. One type of cells fire at a given distance from a particular cue, another type fires when the cue is both at a given distance and within the range of

a certain angle from the rat's head. If navigation depends on direction, then the firing in a simulated rat will also reflect this and correlate with direction.

A more elaborate model was proposed by O'Keefe (1991). The centroid model characterize an environment by a centroid and the slope of the positions of environmental cues. These can be used as the origin and the direction of polar coordinates. PC firing represent mini-centroids which can be used to average an environmental centroid. Position could be represented by single cells using the firing amplitude for proximity and the phase relative to the θ rythm as angle. Summing PC activity should thus provide a vector pointing to the centroid of the environment. Learning the centroid and slope of a goal occurs when the rat encounters it. A translation vector between current position and the goal is then computed whenever the rat felt like moving towards the goal. This model provides the advantageous possibility for taking short-cuts, but is also sensitive to a unique reference direction to movement and occlusion of cues.

The simplest models are not capable of latent learning. At their best they need to encounter goals many times to slowly build a surface definiton which allows gradient following. The model presented in the next section (2.4.3) has tried to face all the disadvantages of the models mentioned above.

2.4.3 A model of hippocampal function

The population vector model of Burgess, Recce & O'Keefe (1994) is based on the action of hippocampal cells resembling a radial basis function (RBF) (Mitchell 1997, page 238). It builds a spatial firing rate map of place cells (PCs) from tuning curve responses to the distance of cues around the rat. Although limited, the model is a close approximation to biological data on hippocampal activity, including synaptic modification and local inhibition.

An adaptive form of competitive learning is used where the connections are either on or off. They are activated in the presence of maximum pre- and postsynaptic firing. The computational mechanisms include a temporal aspect related to the θ rythm exhibited by hippocampal EEG. Place fields positioned ahead of the rat fires late in the cycle, those behind fires early. Direction is not represented by a single cell, but many (in contrast to the centroid model). The output of the model is a set of goal cells which provide a population vector for the instanteneous direction from interesting locations in an environment. The vector sum is then weighted by firing rates to give the direction and proximity of interesting objects. The opportunity to avoid obstacles is achieved by subtraction to the population vector.

This model does not rely on biologically implausible learning rules and it shows latent learning of directional output. The one-shot modification of connections to goal cells

when encountering a goal only once is enough to enable subsequent navigation to the goal, including the ability to take short-cuts. The cells are organized in layers, comparable to entorhinal cells (ECs), CA1 cells (PCs) and subicular cells (SCs). Sensory input arrives at the ECs and propagates to the PCs and the SCs before it is passed on as directional output.

Feedback inhibition is modelled by arranging the cells in each layer in groups. Only the cells with the largest input is allowed to fire, the rest remain silent. ECs usually set up a large spatial firing field to cover the entire area. PCs and SCs suffer from more severe inhibition, and thus produce spatial firing fields that span a lesser area, where PCs represent the smallest of them.

Advantages of this model includes stable firing fields which are built quickly, a fast learning process resulting in good, but not optimal, trajectories (short-cuts), and separate environmental representation and goal locations (Burgess et al. 1994, 1079).

2.4.4 Computational principles

O'Reilly & Rudy (2000) presents a computational approach to the understanding of contributions from the neocortex and hippocampus in the aspects of learning in artificial neural networks, principles which account for empirical findings. This complements the findings from the other articles on related work.

The level of interference is intimately connected to learning rate and representational overlap. This has implications for the need to keep memories separate while extracting generalities across episodes. Learning is reflected in weight updates, regardless of the learning mechanism. Overlapping patterns will share more weights and thus interfere more with each other. Similarly, as rapid learning depends of faster learning rates it leads to more weight change and a network that mainly reflects the most recent memories. For the weights to reflect the underlying statistical structure the learning rate must be low. Weights are reused, and integrated across many experiences.

Capturing episodic memory and extracting generalities are in opposition. Avoiding interference is incompatible with the extraction of generalities by integration over experiences. Still, these processes have clear functional advantages and should by all means be present in a learning model. The hippocampus facilitates episodic memory through the rapid acquisition of separated patterns. Its activity is able to shift between pattern separation for new memories and pattern completion for existing ones. Extracting general statistical structure requires slow learning and is provided by connections in the neocortex. By the use of slow weight updates similarites between overlapping representations can be integrated.

Hebbian learning is well-suited to bind together conjunctive information of co-occurring memory features. It is constantly operating by reinforcing active representations. Procedural learning of task demands is accomodated by the mapping of sensory input to action output, also known as *error driven learning*. Hebbian and error-driven learning are present both in the hippocampus and the neocortex, but to different degrees.

The presence of retrograde amnesia in the case of hippocampal damage is a debated issue. O'Reilly and Rudy's principles suggest that hippocampal reactivation of cortical activity patterns can aid consolidation. Still, they view the cortex as highly capable learning system on its own. None the less, all of the above-mentioned principles apply to the simulation results of this thesis.

2.4.5 Subdivision and sleep

The organization of the brain into different processing units resembles those of subdivided networks as presented by (Bar-Yam 1997, pages 328-370). Feed-forward networks perform a longitudinal subdivision into layers, a parallel computation paradigm which partly is a consequence of the limitations of single layer networks. In a lateral subdivision *"the connections within each subdivision are of greater number or of greater strength than between the subdivisions"* (Bar-Yam 1997, page 329). Subdivision enables the network to separate independent information and develop areas capable of performing special tasks. This is one reason for why the brain relies on subdivision. Usually information isn't completely independent, as can be experienced in vision by the separation of color, shape and motion. In the presence of such inter-dependence, some connectivity between the areas of a network allows for useful communication.

A subdivided network has reduced storage capacity, but subdivision introduces the ability to recall composite states which offer significant advantages. The level of subdivision must strike a balance between the number of imprinted patterns and the number of composite patterns the network can store. Simulations shows that optimal performance is obtained using seven subdivisions, which also is consistent with the 7 ± 2 rule for short-term memory (Bar-Yam 1997, page 348).

Since sleep represents a time of reduced awareness it must offer an advantage over what is provided by simple rest (Bar-Yam 1997, page 376-389). The complexity of sleep increase with the evolutionary complexity of the organism. Sleep is a "temporary dissociation of the brain into its components" (page 389). A subdivided network must be offline to perform subdivision training under periods of dissociation. This can be seen as a difference in patterns of activity between the waking and asleep brain. Dissociation is achieved by diminishing the synaptic weights that connect between subdivision. The extent of dissociation correlates with how deep the sleep is. The activity is a filtering process that reinforces some memories at the expense of others to prevent overload and

thereby allow for additional learning.

The consequences of sleep deprivation are catastrophic. Dissociation between brain subdivision allows a juxtaposition of composite states which wouldn't normally occur when awake. In humans the *"waking experiences are reflected in sleep"* (Bar-Yam 1997, page 393). Along with a decoupling of self-awareness this can account for the lack of surprise by the often bizarre content of dreams. This also explains why it is difficult to recall dreams, they are just an arbitrary side effect of other learning processes.

Chapter 3

The problem of catastrophic forgetting

3.1 Hippocampus as a teacher

This section is an effort to present a biological approach to hippocampal theories based on the work of Rolls & Treves (1998). I will discuss relevant empirical evidence from lesion studies and present the possible hippocampal functionality that is tested in this thesis.

3.1.1 Limited storage capacity

Experimental simulations and results are based on the role of the hippocampus as a teacher to the neocortex. This approach is adopted from Rolls & Treves (1998) and McClelland et al. (1994). Initially the imprints in the hippocampus is of great help in restoring neocortical activations, but for these memories to last they must become independent of the hippocampus. Hippocampal imprints disappear gradually, but because of long-term consolidation some of them become part of neocortical activation patterns. As explained in section 2.2.1, CA3 acts as a representational bottleneck due to the need to maintain it as an efficient autoassociative network. The maximum number of memories in the CA3 cells depends on its level of recurrent connectivity and on the sparseness of the representations. This limited storage capacity has implications both on the number of concurrent memories that can be stored and the time gradient of consolidation into neocortex long-term storage. Memories in the hippocampus must gradually be overwritten to prevent the storage capacity from being exceeded, resulting in memory retrieval

becoming impossible.

3.1.2 Amnesia

Anterograde amnesia is a deficit in the ability to transfer new events into long-term storage to form new memories (Rolls & Treves 1998, pages 95-99). The existence of anterograde amnesia in the presence of hippocampal damage contributes to the theory of the hippocampus playing a role in long-term memory consolidation. Hippocampal damage affects different types of memory to different degrees. While a person might be unable to form new semantic memories, the ability to learn procedural skills could still be intact. But still, the memory of how such new skills were acquired would not be available.

Anterograde amnesia often occurs alongside *retrograde amnesia*, an inability to remember events prior to hippocampal damage. Retrograde amnesia is occasionally temporally graded and restricted in time, having the newest memories show the worst impairment. It is theorized that this is an effect of short-term memories that have still not been consolidated in long-term storage. The hippocampus acts as a tempory buffer store, due to representational issues discussed in section 3.1.5

3.1.3 Recall

Rolls & Treves (1998) describes operation during information recall from the hippocampus to the neocortex (pages 124-126). The hippocampus must be able to recall the whole of a previously stored episode based only on fragments of the inital memory. The presence of NMDA receptors in superficial layers of the cerebral cortex implies the existence of the Hebb-like learning needed for such a content-addressable memory (see section 2.2.3).

Pattern completion is made possible by the recurrent connections in CA3, which acts as an autoassociative network (see section 2.2.1). The feedforward connections from association areas of the cerebral neocortex show major convergence as information is passed to CA3, the processing stage with the smallest number of neurons. Here a compressed version of neocortical activations is stored by modification of CA3 synapses.

During the formation of a new memory there is strong feed-forward activation progressing towards the hippocampus. Backprojecting synapses from the hippocampus must be set up with appropriate weights to allow for pattern reinstatement in the neocortex. The modifiable connections between CA3 and CA1 neurons are set up as the new memory is established. This allows the compressed information of CA3 to be restored

in CA1 before it is passed on to the neocortex via divergent connections through the subiculum and the deep layers of the entorhinal cortex. The backprojecting synapses undergoes a process of associative modification allowing the hippocampus to reproduce the activation in neocortical cells caused by forward inputs, based on the forward input to the entorhinal cortex propagating through the hippocampus. This is formally known as pattern association.

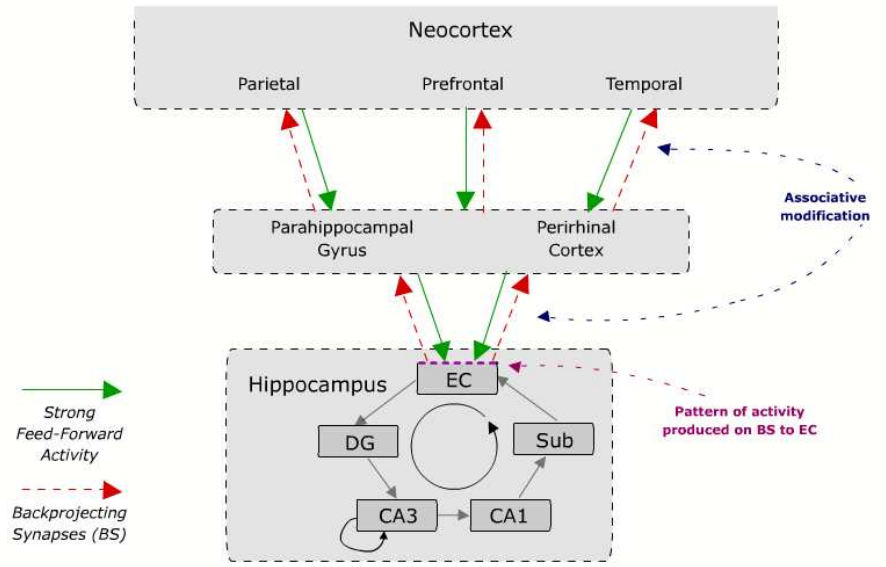


Figure 3.1: Backprojecting synapses are set up with appropriate strengths for recall, figure adopted from page 97 of Rolls & Treves (1998)

This hippocampal-dependent reinstatement of neocortical activation shows great divergence. It allows the cerebral cortex to initiate action or to incorporate the recalled information into long-term storage. Neocortical activations are restored in just those cortical pyramidal cells that were active when the memory was originally stored, and the synaptic modifications last only as long as the memory remains in the hippocampal buffer store.

3.1.4 Sleep

According to McClelland et al. (1994) there are two types of memory reinstatement mediated by the hippocampus, task-relevant and task-irrelevant. While task-relevant reinstatement depends on the re-occurence of situations, most animal experiments on consolidation have investigated *"periods when the animals have no exposure to the task or even the locations in the environment in which the memory was originally formed"* (page 30). Such task-irrelevant reinstatement may occur *"when the hippocampus is not actively*

engaged in processing external inputs" (page 30). In humans this can be experienced in offline situations such as sleep, or as reminiscense and daydreaming (where behavioural responses are controlled).

Electrical activity recordings have shown the presence of optimal conditions (sharp waves) for synaptic plasticity in the hippocampus under periods of quiet wakefulness and slow-wave sleep. Sharp waves arise in CA3 and are propagated to both CA1 and the output layers of EC. This can be signs of pattern completion which provides an opportunity for reinstatement in the neocortex. While common aspects of events might be incorporated at the first encounter of a new association, idiosyncratic content depends heavily on the decay of hippocampal traces. Section 2.4.5 also mentioned the process of dissociation during sleep to train subdivided networks.

3.1.5 Representation

"What can be performed by neuronal networks is closely related to how the information is represented" (Rolls & Treves 1998, page 99). Catastrophic forgetting is attributed to interference between patterns. Patterns with much overlap are subject to much interference, but the use of separated representations keeps the level of overlap low and helps preventing it. Another factor is due to learning rates; the faster the learning rate, the more interference.

There are several characterizations of pattern distributions (Rolls & Treves 1998, page 12). From a biological perspective a look-up table would necessitate a single neuron to represent every feature conjunction that exist in your memories. This is an example of a *local distribution*, which offers no possibility for generalization or statistical extraction of structure. As an example, you would need one neuron to represent your grandmother taking a walk in the park, and another one if she was sitting in the sofa knitting. In a *fully distributed representation* all the information of a particular episode or event is provided by the full set of neurons. At its most distributed a binary version of such a representation would need half the neurons to be active for any stimulus or event. A *sparse distributed representation* is a distributed representation where a small proportion of the neurons is active at any one time. The sparseness of the distribution is a measure of the proportion of active neurons.

With a distributed encoding the patterns of activity represent different stimuli. Viewing the set of activities on input axons as vectors, the similarity between different stimuli is reflected by the correlation of their respective vectors. Correlation will be high for similar representations, and lower and lower as more axons differ. This enables generalization to similar stimuli, or to incomplete versions of a stimulus, to occur.

Whereas the number of stimuli that can be encoded with a local representation grows

only linearly with the number of components, the number of possible stimuli for distributed encodings grows exponentially with the number of components in the representation. A neuron with a limited number of input can still receive a great deal of information about which stimulus was present. This is probably one of the factors that makes computation by the brain possible. There is now good evidence that many brain systems, such as the hippocampus, use distributed encoding. This includes the properties of representing similarity and an encoding capacity which is increasing exponentially with the number of neurons found in the representation.

In an artificial neuronal network it is also important to choose a correct size for the hidden representation. Too many units in the hidden layer can result in a look-up table where each input is represented by one hidden unit. It will lack the abstraction ability over the data and lead to poor generalization and no chance of discovering any statistical structure.

In McClelland's view, reinstatement of neocortical patterns in the hippocampus need not be an exact copy, but can be stored as compressed representations (McClelland et al. 1994, page 48). These hippocampal patterns must simply encode enough information about the original pattern for the neocortex to reconstruct it. This does not mean that essential information has to be lost, additional knowledge can be found in the connections within the cortical system and in the connections leading to and from the hippocampus and neocortex. Compression is then carried out by connections leading into the hippocampal system, and reinstatement of the neocortical pattern will be done by connections leading from the entorhinal cortex to the neocortex.

The question regarding the acquisition of additional information which is not stored by the hippocampal system is still left. Some synaptic modification will occur in the neocortex at the initial exposure to a new event or stimuli, but these small changes in connection weights would not be enough to store this new information. McClelland et al. suggest that the remaining information is part of already discovered knowledge and structure from previous learning. Patterns with much difference to already acquired information embodies different constraints and will be difficult to learn, at least for adult memory systems. Research has shown that the way the human memory *"conforms to familiar structures is far better when the material to be learnt"* (McClelland et al. 1994, page 49).

3.2 Dual-network architectures

3.2.1 Preventing catastrophic interference

French et al. (2002) presents a dual-network architecture which is able to overcome the problem of sudden and catastrophic interference in multiple-sequence learning. As the problem occurs in neural networks with the learning task of new static input-output patterns, it is amplified when presented with the task of learning multiple sequences. Newly learned information suddenly and completely erases previously learnt information. The use of internally-generated pseudopatterns has been one remedy for this phenomenon of catastrophic forgetting.

3.2.2 Pseudopatterns

Pseudopatterns are produced by presenting random activation on the input layer and recording its related output. They reflect, but are not similar to, the input output pairs used to train the network originally. By storing them externally they can be put to use later by including them in the training set along with the new patterns. Pseudopatterns can also be used to transfer memory between similar networks in a dual network architecture. This approach consists of an *awake state* where the network interacts with the environment to learn new sequences, and a *sleep state* where memory is transferred to long-term storage (see section 3.2.5).

In the case of sequence learning it is not necessary to construct temporal pseudo-sequences of patterns. Pseudopatterns are non-temporal compressed representations of the function learnt by the network, not the individual patterns used in training. Even if they constitute static input-output patterns they do represent a dynamic state of the network. It is possible for pseudopatterns to be similar to the original patterns, but this is not a necessity. In an earlier article, French et al. (2001) showed that the original patterns are transferred between networks even when the pseudopatterns are forced to be dissimilar. The same research also showed that using pseudopatterns whose input resembles that of the new patterns often will prevent the network from to converging. In other words, rehearsal with pseudopatterns prevents catastrophic forgetting of previous learning.

3.2.3 RSRN - SRN with autoassociator

A standard Simple Recurrent Network (SRN), as presented by Elman (1990), is capable of learning sequences of patterns (see figure 3.2). It is more powerful than a standard

backpropagation network, which would depend on distinct input patterns. The standard input $S(t)$ is associated together with context input from the previous hidden activation $H(t-1)$ This enables the network to sort out ambiguity represented by identical input patterns at different sequence positions, and let the resulting output activation be influenced by the temporal context.

Adding autoassociative units on the output layer of a standard SRN provides the additional possibility of reverberating the input when producing pseudopatterns. These networks are called RSRNs, Reverberated Simple Recurrent Networks, and provide a hybrid of the two types of network types discussed in section (2.3.1). In figure 3.2 the additional autoassociative units have been emphasized. For these units the error is calculated according to the difference between output and input. The error calculation for hetero-associative units is done by error backpropagation and the weight updates are performed as usual. More about the reverberation technique can be found in section 3.2.4.

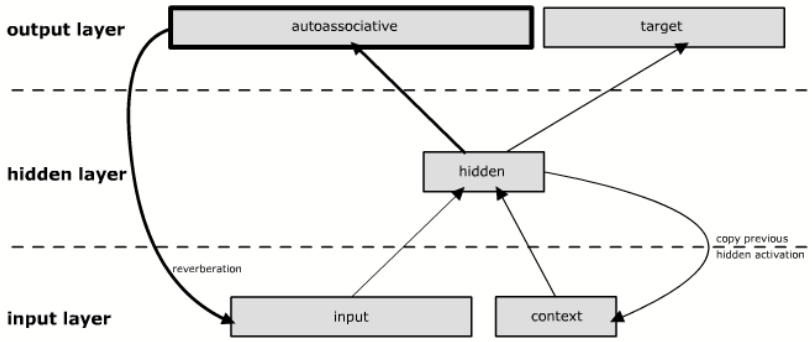


Figure 3.2: Reverberated Simple Recurrent Network

3.2.4 Attractor pseudopatterns

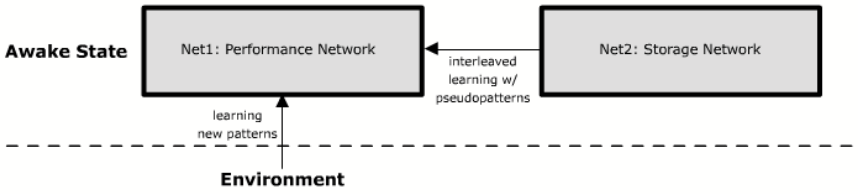
It has been proved that pseudopatterns offer a powerful way of avoiding catastrophic forgetting (section 3.2.2). Reverberation includes a cycle of re-presenting the information used to produce pseudopatterns. Random activation i_ψ is presented to the units of the input layer and fed through to the output layer. This produces activations on the autoassociative units, i'_ψ, which constitute a new input which can be re-presented to the input layer. Repeating this results in i''_ψ. The final reverberated output i^R_ψ is used as the last input to the network to produce the final activation o_ψ, resulting in the pseudopattern $\Psi : i^R_\psi \rightarrow o_\psi$.

> An "attractor pseudopattern" provide a much better reflection of the old patterns than pseudopatterns produced from simple random noise on input. It is this reverberation technique that is largely responsible for the power of this technique (French et al. 2002, page 3).

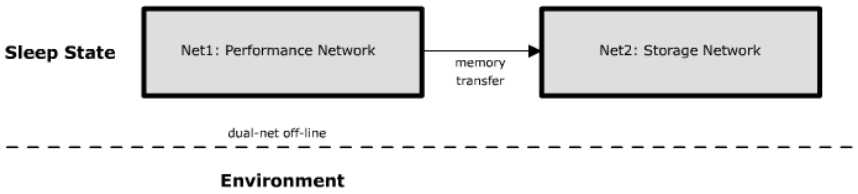
3.2.5 Information transfer

External storage of pseudopatterns is one possible way to prevent catastrophic forgetting. The network first learns a set of patterns P_i. Before the network is presented with the next set of patterns, Q_i, noise is fed through the network to produce a set of pseudopatterns Ψ_i. These pseudopatterns are added to the new patterns and the network trains on this larger set of patterns until the new patterns Q_i are learnt to criterion.

But where should these pseudopatterns be stored? Single RSRN networks can be used when expanding the concept to a dual network architecture. This architecture consists of two similar RSRNs coupled together. Net1 is a *performance network* that interacts with the environment and learns new sequences. Net2 is a *storage network* used to store previously learnt patterns. In the *awake state* training on new patterns in Net1 is interleaved with pseudopatterns from Net2 (Fig. 3.3a). The *sleep state* is when learning from Net1 is transferred to Net2 using pseudopatterns (Fig. 3.3b).



(a) Awake State



(b) Sleep State

Figure 3.3: Learning in the dual-network architecture

3.2.6 Experimental results

The measure of network performance is based on the number of incorrect units over all items of the sequence. 22 distinct random binary vectors of length 100 are generated. One half of the patterns is selected for the first sequence, the rest for the second sequence. To introduce a degree of ambiguity, two of the patterns in both sequences are identical. This ambiguity had to be sorted out by the temporal context.

Simulation 1 shows how the training of a single SRN on multiple sequences lead to catastrophic forgetting. After first learning sequence A to criterion the network is trained on the new sequence B. The performance on the old memories drops very fast. The network's memory of sequence A is forgotten very early on in the process of learning sequence B. Even before the network shows any learning on the new training data from sequence B, the performance has dropped to 50%. When sequence B is fully learnt it has caused sequence A to be completely forgotten.

For *Simulation 2*, a dual-network architecture of two coupled RSRNs is used to overcome catastrophic forgetting (see section 3.2.5). Network 1 is the performance network which act as an interface to new learning from the environment. Net1 first learns sequence A, afterwards 10^4 pseudopatterns are used to transfer this learning to Net2. Information transfer between the networks is done by means of pseudopatterns. Each learning epoch of sequence B is interleaved with pseudopatterns. Net1 receives 10 pseudopatterns from Net2 and performs one feedforward-backpropagation pass for each of them. The result shows that forgetting is no longer a problem with a dual-network architecture.

3.3 The importance of speed

Learning speed influences several qualites of the learning results. Section 2.4 showed how McClelland et al. (1994) argues for the importance of slow interleaved learning in extracting general tendencies and statistical structures. At the same time the limited storage capacity in CA3 requires that memories are pushed out of the hippocampus to allow for new learning, and puts a time-limit on consolidation (Sec. 3.1).

The dual-network architectures are an efficient way of preventing catastrophic interference, but what role does speed play in this aspect? If the learning rate is too fast the network will end up emphasizing the last experiences when updating the memory. The ability to incorporate idiosyncrasies is also weakened. Slow learning also facilitates the conservation of pseudopatterns that are capable of reflecting previous learning. Pseudopatterns are sensitive to changes in the network connections the same way the sequence patterns are. The inital learning of new patterns can disturb the weights so rapidly that

other memories vanish. Chapter 5 will explore the implications of varying the learning rate, and hence the consolidation speed, to the degree of forgetting.

Chapter 4

Design and methods

4.1 Research goals

This section starts out describing why I decided to restrict the architecture used by French et al. (2002). Then follows a biological justification of the new experimental framework, alongside a description of the methods used to reduce memory loss.

4.1.1 Restricted architecture

Dual-network architectures do prevent catastrophic forgetting, but the method results in a very elaborate process for remembering. With dual-network memory a representation of the network function exists in the storage network without being subject to interference from new learning. Speed vulnerability to pseudopatterns is simply not present because the network weights are kept in separate networks.

After reimplementing the dual-network from the French article I made various changes to accomodate the needs of my experimental framework. I cut down to a single network and generate pseudopatterns which are used to re-learn, or self enforce, the current network function. In this more restricted architecture the many stages of passing information back and forth between the two network areas are removed.

4.1.2 Biological justification

Unlike the dual-network architecture of French et al. (2002), the neocortex and the hippocampus are not represented by separate storage networks, but by the different parts of an RSRN (figure 4.1). Long-term consolidation is the only memory process which is taking place. The Simple Recurrent Network (SRN) part represents neocortex. This resembles a performance network, and is the part which is learning the sequences. The hippocampus part only contains the pattern completion functionality of CA3, the reverberated autoassociative connections. Both the new patterns and the self-produced pseudopatterns are part of hippocampus. Besides CA3 the hippocampus is not deconstructed into its various parts, but kept as a black box. Sequence B snapshot memory episodes learnt by hippocampus is represented by the system, and not a network.

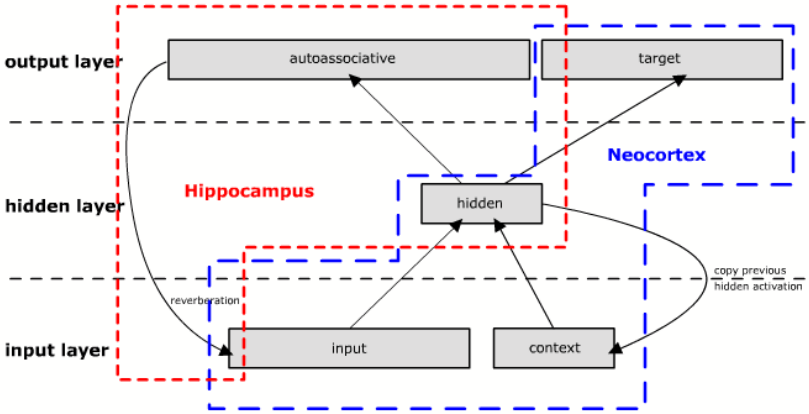


Figure 4.1: Biological justification

4.1.3 Reducing memory loss

For every epoch all the input patterns are presented to the network in a randomized order, with immediate weight updates performed after each presentation. The learning of a new sequence is interleaved with training on self-generated pseudopatterns, but they are not interleaved between epochs, as in French et al. (2002), but between each pattern within an epoch. Because these pseudopatterns are self-generated they do not obstruct new learning, they always reflect the present network function.

Self Enforce involves one backpropagation pass of a self-generated pseudopattern after each pattern presentation. With this present implementation the number of pseudopatterns depends on the length of the sequences. A different approach to learning speed variation could be based on the ratio between the length of the new sequences and the

number of interleaved pseudopatterns. *Circulated Storage* keeps a pseudopatterns list of variable, but predefined, size. These pseudopatterns are replaced randomly by storage of new pseudopatterns. This makes the memory more resistant, or robust, compared to simple self enforcing, but also includes an explicit external storage.

4.1.4 Learning speed

A learning rate of 0.01 causes modest weight updates, but also restricts interference between weight updates. I needed to push those limits and settled with a learning rate of 0.04, which is substantially higher. If it had been increased more, the fluctuations produced by interfering patterns would make the network unable to learn anything. The learning speed is not defined only with respect to learning rate. Weight updates caused by the pseudopatterns are intended to enforce memory already contained in the network. They are working to stabilize the interference caused the weight updates intended to learn a new sequence. The network must learn gradually to sort out how it can achieve adequate performance on both sequences.

The terms fast and slow learning are viewed as the relative strengths of the learning rates between weight updates intended to learn a new sequence and those intended to prevent memory loss. With *fast learning* both types of weight updates are done with the same learning rate. *Slow learning* gives the learning rate of new sequences in a fraction relative to the learning rate of the self enforce method.

4.2 Implementation

I now move on to how the dual-network architecture is modified to meet the research goals. This implementation has to meet the demands of both the experiments performed by French et al. (2002) and the new experiments. The implementation related issues are based on examples and equations.

4.2.1 Programming tools

Python® is a cross-platform dynamic object-oriented programming language (Pyt 2006), and it is also my programming language of preference. Although Python code runs quite fast, modules with critical performance demands can be implemented in C, or another language, and easily imported into your Python modules. Python also comes with substantial standard libraries and has available third-party packages which rendered this unnecessary in my project.

Writing code in Python is fast, simple and fun. It is your own skills and creativity which limit your solutions, not Python. The language gives you the freedom to solve problems fast and in the way you want to, but this freedom also demands great responsibility from the programmer. A project with many participants would have been susceptible to problems much because of the excact same reasons that makes it so attractive for doing an individual project.

I decided to implement a graphical user interface (GUI) to make the results of my project more accessible, and also free my eyes from the stress of viewing ascii print-outs all day long (see section 4.2.3). The possibility to save and load networks allowed me to change between the GUI and the more powerful *Python Shell Window*, which gives access to the *Python interactive mode*. Accessing the objects in the interactive mode allows direct control and manipulation for all instance variables of any object.

4.2.2 Network architecture

The RSRN concept was presented in section 3.2.3. The Python implementation does not utilize a further subdivision of the layers into subclasses, but represent the layers as weight matrices. Figure 4.2 shows the class structure for the network classes and their respective GUIs. The C-implemented library *Numerical Python* is used to improve execution speed in the computation of feed-forward activations and error calculation. Its structure *array* supports fast matrix operations and provided a substantial performance improvement compared to a previously self-implemented Python module.

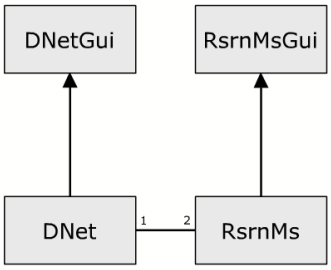


Figure 4.2: Class structure for network classes and GUI

Pseudopatterns

Pseudopatterns are generated by presenting noise on the input layer. The resulting activation on the autoassociation units of the output layer is then repeatedly presented at the input layer in the same manner. These pseudopatterns represent attractor states

of the network which reflect the network function better than simply using input noise and its respective output activation. Five repetitions provided efficient pseudopatterns when measuring the resulting memory transfer rate of learning.

$$\Psi : i_{\psi}^{R=5} \rightarrow o_{\psi} \tag{4.1}$$

The values corresponding to the standard input units of the autoassociative output units are tresholded with θ of 0.5. The context unit activations are kept unchanged.

$$f(o_i) = \begin{cases} 1 & if \ o_i \geq \theta \\ 0 & if \ o_i < \theta \end{cases} \tag{4.2}$$

4.2.3 Graphical user interface

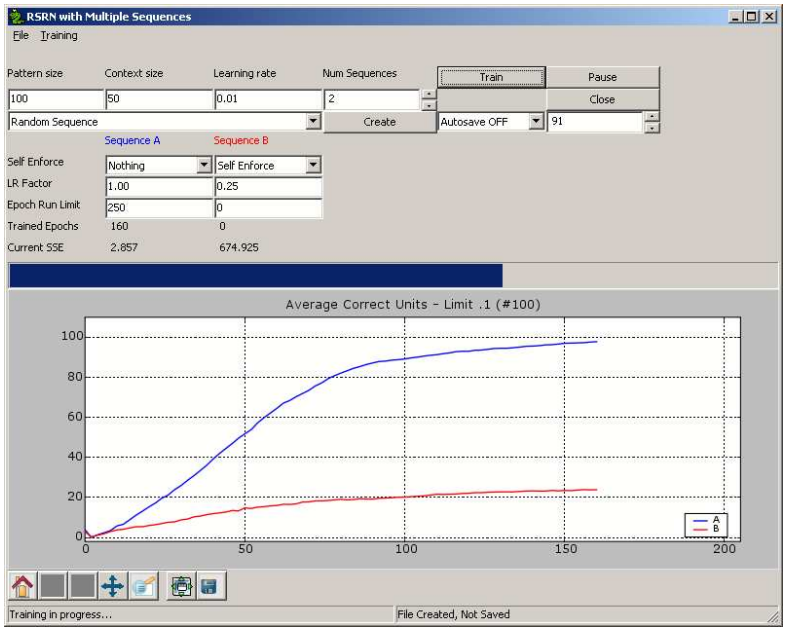
The graphical user interface (GUI) allows you to set and change all the necessary paramaters both before and during training, as well as providing a continued presentation of simulation results and progress. Based on previous experiences I decided to use *wxPython* which provides Python bindings to the wxWindows cross-platform toolkit. *Matplotlib*, a matlab style graphical package, was selected for the graphical presentation of training results. This package was also used to produce the graphs which show the average number of correct units for both sequences. Figure 4.3 shows two screenshots of the GUI during training.

4.2.4 Encoding

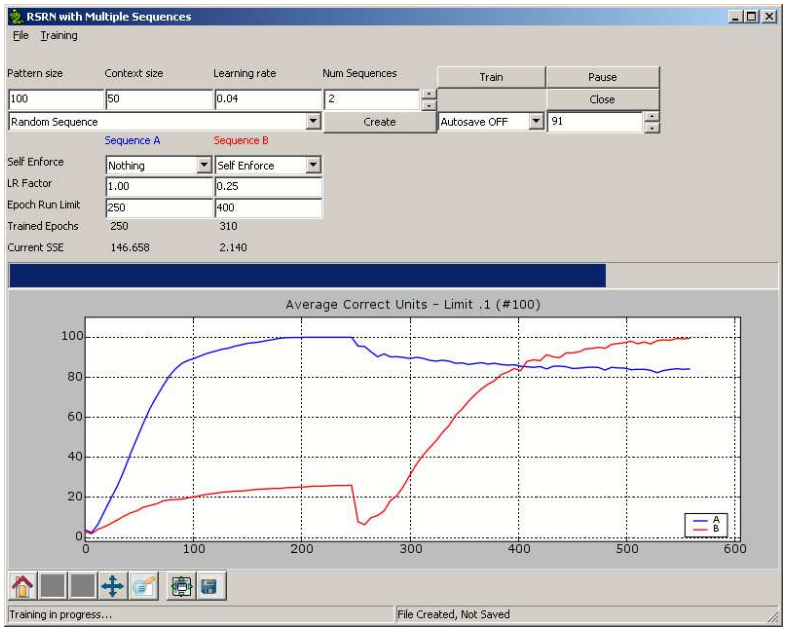
Simulations use input patterns of length 100, identical to that of French et al. (2002), to make it easier to compare our results. The high number of input neurons allows pattern combinations far larger than the number of disctinct memories used in the simulations, $2^{150} \approx 1.42 * 10^{45}$ distinct states of the input layer. As catastrophic forgetting is experienced even with dissimilar patterns the size limit is not put under pressure.

SRNs are able to sort out ambiguity due to the application of context. Ambiguity is added to the sequences by making two of the input patterns identical. Based on context the discrimination between these identical input patterns can be mapped to different output patterns.

The input sequences consist of fully distributed patterns of uniformly generated random binary vectors. Pattern similarity was controlled both within and between sequences.



(a) Training on Sequence A



(b) Training on Sequence B; changed learning rate and extended drawing area.

Figure 4.3: Graphical User Interface in wxPython

The result is a distribution without any underlying statistical similarites or idiosyncrasies. As with local encoding, this does not enable association between similar memories. Pattern overlap is thus kept low and the well established issue of how learning rate affects the extraction of statistical structure will not be an issue.

4.2.5 Backpropagation and biological plausibility

I apply a modified version of the backpropagation learning algorithm using gradient descent, originally invented by Rumelhart et al. (1987). This non-local learning rule exposes the networks to pattern propositions from the environment. Weight updates are performed with interleaved pattern presentations. After each presentation the discrepancy between the desired and the obtained output is calculated and the weights are adjusted proportional to the learning rate.

From a biological standpoint there does exist reciprocal connectivity and return connections between regions in the brain, but the error correcting weight adjustment required by backpropagation relies on a *"biologically implausible backward transmission across forward-going synapses"* (McClelland et al. 1994, page 28). Even though the biological plausibility of backpropagation has been a much debated issue, the important fact is that learning new patterns has a strong potential of interfering with previous learning. This is due to the general nature of the problem, regardless of the learning algorithm. The excact procedures of learning is not the most interesting point here, but the argument that slow, interleaved learning is necessary to discover shared structure in ensembles of patterns. Both biological and artificial neural networks store patterns in a distributed manner. The simulations in this thesis are a continuance of the work on dual network architecture by French et al. (2002), and hopefully they will be useful in furthering the understanding of the possible roles of the hippocampus in the interplay between short-term and long-term memory.

4.2.6 Cross entropy and pattern normalization

Problems with Sum of Squared Error

A standard backpropagation algorithm tries to minimize the network's cost function. The sum of squared error (SSE) is often used along with the sigmoid activation function. Joost & Schiffmann (1997) presented several drawbacks with this approach. First, with multi-layer networks the weight updates are based on a generalized error of other units, starting with the output layers (equation 4.3) and recursively computing the δ terms of the hidden units (equation 4.4), where S denotes the set of all successors of the hidden

unit y.

$$\delta_z = \frac{\partial f_z}{\partial net}(t_z - o_z) \tag{4.3}$$

$$\delta_y = \frac{\partial f_y}{\partial net} \sum_{z \in S} \delta_z w_{yz} \tag{4.4}$$

The cost function is minimized by changing the connection weights according to the negative gradient of the function's energy landscape. The derivation of the sigmoid activation function (equation 4.5) with respect to net_i is given by equation 4.6 This term yields a maximum value of 0.25.

$$f(net_i) = o_i = \frac{1}{1 + e^{-net_i}} \tag{4.5}$$

$$\frac{\partial f}{\partial net_i} = o_i(1 - o_i) \tag{4.6}$$

As this term can be recognized in the last part of equation 4.7, it attenuates the back-propagated error signal. Even output values far from the desired target values can result in a low residual error, a problem known as the *sigmoid prime factor*.

$$\delta_{SSE} = \sum_{n=1}^{N} \{(t_n - o_n)o_n(1 - o_n\} \tag{4.7}$$

The statistical approach

A different approach to network training is based on statistics where the neural network is seen as a model of the structure underlying the true distribution of the training data. Each output value is considered an estimate of the conditional posteriori probabilites of its input pattern belonging to a class.

In order to estimate the *joint probability* $p(\mathbf{x}, \mathbf{t})$ of the input vector $\mathbf{x}$ and the target vector $\mathbf{t}$, we define a *likelihood function* $\mathcal{L}$ with respect to a training set $\mathcal{T}$. This gives us a *model specific* likelihood function by assuming a parametric model with respect to

the weight vector $\mathbf{w}$.

$$\mathcal{L}_w(\mathcal{T}) = \prod_{n=1}^{N} (\mathbf{t}_n|\mathbf{x}_n) \tag{4.8}$$

Similar to the *maximum likelihood principle* used to estimate the distribution, an error function is defined by taking the negative logarithm of the likelihood function $\mathcal{L}_w$. The reason behind this is the need to *minimize* the error of the network.

$$E = -\ln \mathcal{L}_w(\mathcal{T}) = -\sum_{n=1}^{N} \ln p_w(\mathbf{t}_n|\mathbf{x}_n) \tag{4.9}$$

Two class case

For a two class problem with a single output unit the target values correspond to class conditional probabilities, in a closed form given by

$$p(t|\mathbf{x}) = o^t(1-o)^{1-t} \tag{4.10}$$

By taking the negative logarithm of the combination of this equation with the model specific likelihood function (equation 4.8) we get a *Cross Entropy* (CE) function between two distributions.

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \{t_n \ln o_n(\mathbf{w}) + (1-t_n)\ln(1-o_n(\mathbf{w}))\} \tag{4.11}$$

> The CE is minimized if the distribution of the model output $p(o_n|\mathbf{x})$ and the distribution of the target values $p(t_n|\mathbf{x})$ are equal. Because the outputs o_n depend on the vector $\mathbf{w}$ its distributions can be adapted to the distributions $p(t_n|\mathbf{x})$ of the target values (Joost & Schiffmann 1997, page 5). The following section is closely based on their work.

Multiclass case

When expanding this model to a multiclass case one faces the problem of probabilities not summing to unity. This can be overcome by viewing the problem as c separate

two class problems and use a c-1-of-2 coding scheme, which results in a modified CE function.

$$E_{CE}(\mathbf{w}) = -\sum_{z=1}^{c}\sum_{n=1}^{N_c}\{t_{zn}\ln o_{zn}(\mathbf{w}) + (1-t_{zn})\ln(1-o_{zn}(\mathbf{w}))\} \tag{4.12}$$

The error function is then derivated with respect to the weights

$$\frac{\partial E_{CE}}{\partial w_{yz}} = -\sum_{n=1}^{N}\{\frac{t_{zn}}{o_{zn}} - \frac{1-t_{zn}}{1-o_{zn}}\}\frac{\partial o_{zn}}{\partial net_{zn}}\frac{\partial net_{zn}}{\partial w_{yz}} \tag{4.13}$$

Using the sigmoid activation function from equation 4.5 and the weighted sum as the net input function, we get

$$\frac{\partial E_{CE}}{\partial w_{yz}} = -o_i\sum_{n=1}^{N}\{t_{zn}(1-o_{zn}) - o_{zn}(1-t_{zn})\} \tag{4.14}$$

Cross Entropy as an error function results in the first delta term found below. The network will still be trained in the same manner, but by comparing equation 4.15 to equation 4.16 (repeated here to ease comparison) it is shown that the attenuation caused by the sigmoid prime factor will no longer be present in the direct connections between the hidden units and the output units.

$$\delta_{CE} = \sum_{n=1}^{N}\{t_n - o_n\} \tag{4.15}$$

$$\delta_{SSE} = \sum_{n=1}^{N}\{(t_n - o_n)o_n(1-o_n\} \tag{4.16}$$

Pattern normalization

The second drawback relies on weight updates being computed on the basis of both delta terms and input activations. Because I am using binary input vectors approximately half the activiations will be zero and not render any weight updates, this is known as the *low input prime factor*. The need for auto-associative units in the input layer prohibits a mapping to negative and positive values. Instead the values are normalized and mapped to the range of $[0.1, 0.9]$ to help improve learning speed.

Chapter 5

Experimental results

This chapter present and analyze the results obtained from the experiments described in chapter 4. I will investigate a qualitative approach and describe the results compared to the theoretical background from chapters 2 and 3. The limited and controlled environment makes little room for variations, and I will thus not explore a quantitative analysis.

The first step was to reimplement the artificial neural networks used in French et al. (2002). This included both a standard Simple Recurrent Network (SRN), a Reverberated SRN (RSRN) and a dual-network architecture of two coupled RSRNS (DNet). These network implementations were used to perform the same experiments as they describe, and compare my simulation results to their results, catastrophic forgetting in a single SRN and overcoming catastrophic forgetting with DNets.

My simulations test the use of differentiated learning rates between the learning processes, or differences in learning speed. The first experiment performs a test on this variable with both fast and slow learning in DNets, a test which had not previously been performed. The next experiment looks into how pseudopatterns are distributed compared to the original patterns of sequence A used to train the network. Moving on, the third experiment shows the consequences of catastrophic forgetting when subsequent sequence training is performed without any remembering aid. The next experiment self enforces the current memory with self-generated pseudopatterns for both fast and slow learning. The last experiment is similar to self enforcing, but uses a circulated storage of pseudopatterns.

5.1 Common network parameters

The sequences are constructed from binary vectors of length 100 encoded as explained in section 4.2.4. Table 5.1 shows the parameters which are used for all experiments (when not stated differently).

Table 5.1: Common network parameters

Pattern size	100
Context size	50
Learning rate	0.01
Momentum	0.50
Error function	Cross entropy
Pattern normalization	[0.1,0.9]

The network setup is seen in figure 5.1. This is the type of RSRN which is used in all simulations. For the SRN the autoassociative units are dropped. The input layer consist of 100 standard input units plus 50 context units. The hidden layer is of the same size as the context, 50 units. The output layer has 150 autoassociative units that correspond to the units in the input layer, as well as 100 target units.

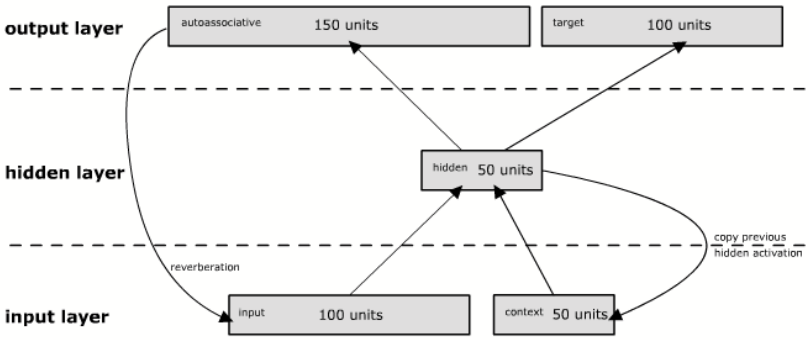


Figure 5.1: Neural network setup

5.2 Dual-network reimplementation

5.2.1 Simulation setup

The first step was to reimplement the dual-network architecture with two coupled RSRNs from the main article by French et al. (2002).

Table 5.2: Dual-network training parameters

Sequence A epochs	250
Information transfer patterns	10^4
Sequence B epochs	350

The article does not mention how many epochs sequence A is trained, but for my simulations a learning rate of 0.01 the choice of 250 epochs gives the network a wee bit time to settle in before it reaches a level of overtraining. Memory is then transferred from Net1 to Net2 with the use of 10^4 pseudopatterns.

5.2.2 Simulation results

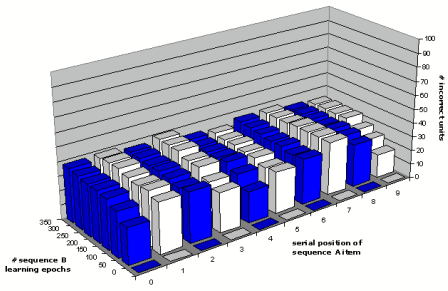
Catastrophic forgetting

> By 250 epochs, the network's performance on sequence A is essentially no better than chance and, by 450 epochs, sequence A is completely forgotten. In short, learning sequence B causes severe catastrophic forgetting of sequence A (French et al. 2002, page 5).

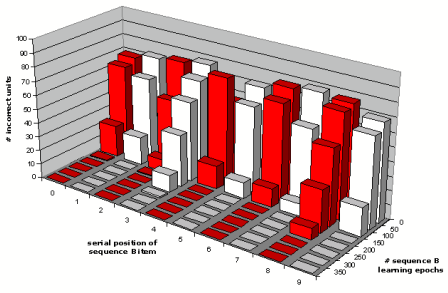
Figure 5.2 shows how my network learns the patterns of the new sequence faster than in the original implementation, after 300 epochs all sequences are learnt to criterion. The rate of catastrophic forgetting in a single SRN is not as severe either. It rises to 30% quite fast and is stabilized right below 40% by 200 epochs.

Catastrophic forgetting is overcome with pseudopatterns

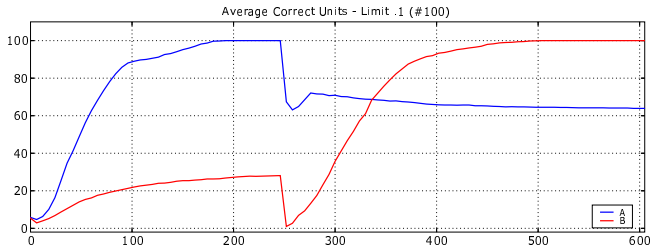
> ... there is virtually no forgetting of sequence A as the network learns sequence B. In short, catastrophic forgetting is completely overcome by the



(a) Catastrophic forgetting of sequence A



(b) Learning sequence B

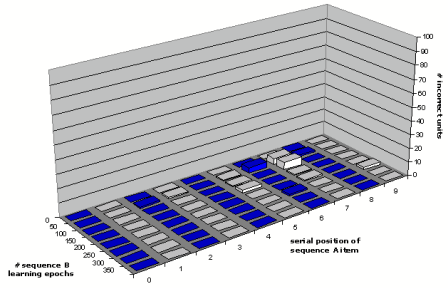


(c) Average number of correct units

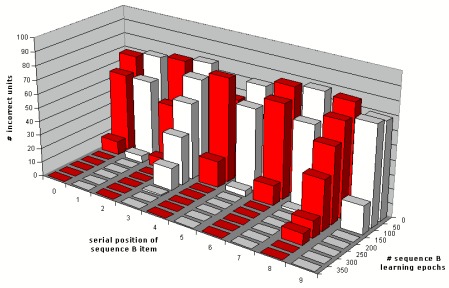
Figure 5.2: Catastrophic forgetting in SRN reimplementation. The bar graphs in 5.2a and 5.2b show the number of incorrect units for sequence A and B, respectively. The graph in 5.2c shows the average number of correct units for both sequences during training.

coupled system of RSRNs using pseudopattern information transfer (French et al. 2002, page 6).

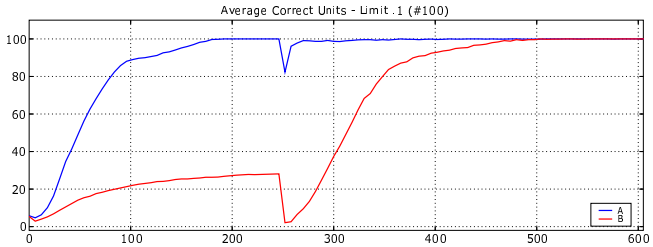
The results of the attempt to replicate how catastrophic forgetting is overcome is shown in figure 5.3. Also in this simulation the patterns of sequence B is learnt to criterion slightly faster in the reimplementation than in the original article, by 300 epochs all units produce the correct output for all sequence B items. Thus, catastrophic forgetting is here completely overcome with dual-networks and interleaved pseudopatterns.



(a) No catastrophic forgetting of sequence A



(b) Learning sequence B



(c) Average number of correct units

Figure 5.3: Dual-network reimplementation. The bar graphs in 5.3a and 5.3b show the number of incorrect units for sequence A and B, respectively. The graph in 5.3c shows the average number of correct units for both sequences during training.

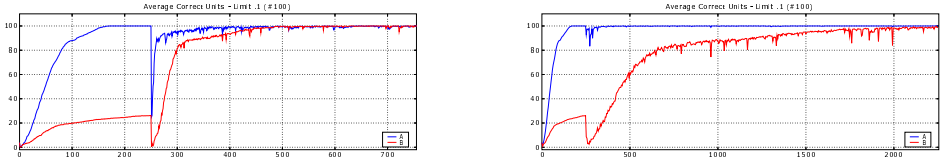
Explanation of differences

I was able to replicate the results from French et al. (2002) to a certain degree, but some discrepancies were found. The explanation of the different types of network implementations are clear, both with respect to the layout and the training methods. Both our simulations also depend on sequences with distinct patterns, but because the excact

sequence generating procedure was not explained we most likely use different procedures for pattern generation. I also tried varying the number of sequence A epochs without any success in producing more similarities. The important fact is, none the less, that catastrophic forgetting is overcome with the use of the dual-network architecture.

5.2.3 New simulations with dual-networks

The simulations in section 5.3 perform weight updates with differentiable learning speeds for RSRNs. In the simulations of figure 5.4 this approach has also been test on dual-networks. The learning rate for the pseudopatterns from the storage network have a learning rate of 0.04. With fast learning the new memories are incorporated with the same learning rate, for slow learning it is reduced to 25%, 0.01.



(a) Fast learning (b) Slow learning

Figure 5.4: New simulations with dual-networks

In dual-network the learning process has to continue longer before performance is adequate than in the previous simulations. With fast learning, weight adjustments from the pseudopatterns and sequence B disturb and interfere with each other when the learning rate is set so high. In the case of slow learning, sequence B has still not reached top notch performance by 2000 epochs. New learning is opposed by the updates caused by pseudopatterns with the relatively higher learning rate. As opposed to the simulations with a single RSRN, the pseudopatterns are never updated to reflect that new memories have been partially acquired. Occasionally putting the dual-networks offline would allow such a process in the sleep state (see section 3.2.5).

5.3 Single RSRN experiments

5.3.1 Learning sequence A

Although the subsequent simulations have differences, all networks are set up and trained on sequence A in the same manner. All weights are initialized to random values between -0.5 and 0.5. Pattern presentation order is randomized for every epoch, and the context

for the next input pattern is replaced by the last hidden activation and stored after each presentation; $C(t+1) = H(t)$.

Table 5.3: Training on sequence A

Epochs	250
Learning rate	0.01
Momentum	0.50

Learning stagnates when the network has to sort out the temporal context by means of hidden representation. When learning sequence A the curve flattens out for a while, before it ascends again. This can be observed as a plateau in the graphs around 120 epochs (figure 5.5). This is when the temporal context of the ambiguous patterns must be sorted out.

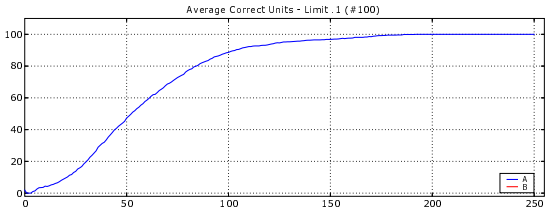


Figure 5.5: Training only on sequence A

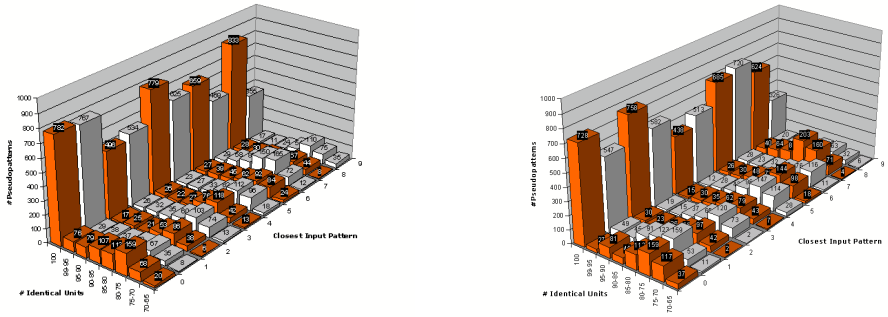
5.3.2 Pseudopattern distribution

How does the used pseudopatterns distribute when compared to the original patterns of sequence A after training to criterion? A large set of pseudopatterns is generated and the euclidean distance (equation 5.1) is calculated between all pseudopatterns and sequence A input patterns. The pseudopatterns are then assigned to the input pattern which it resembles the most (equation 5.2), i.e. the pattern for which it shows the lowest euclidean distance. The number of identical standard units between the patterns is then stored. Context similarity is only used to sort out assignment when a pseudopattern is equally similar to several input patterns.

$$d(x, y) = ||x - y|| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{5.1}$$

$$assign(A, \Psi) = argmin_{s \in A, \psi \in \Psi}\{d(s, \psi)\} \tag{5.2}$$

10^4 pseudopatterns are generated for a learning rate of both 0.01 and 0.04. The bar graphs of 5.6 shows that the results are very similar. Most pseudopatterns are identical to one of the stored patterns. Another peak is found in the range of 80-75 identical units. The concavity between these peaks, for 99-80 identical units, is due to what happens when a pattern becomes similar to one of the stored patterns. The autoassociative network has performed a pattern completion and attracted the pseudopattern to its respective input pattern. Input patterns have a forced dissimilarity of 40% which also seems to work as a downward similarity limit of 60 units. Those pseudopatterns which don't resemble a particular sequence pattern are those that best reflect the network function.



(a) Low Learning Rate (b) High Learning Rate

Figure 5.6: Pseudopattern distribution

5.3.3 Learning sequence B

All simulations train on sequence B for 500 epochs. This is how long it takes for the network to learn sequence B to criterion with the slowest learning speed. Although learning is faster with the other methods, it makes the graphs of correct units easier to compare. For the bar graphs I show only the interesting range of epochs.

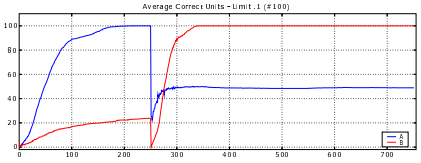
The same learning curve stagnation as observed for sequence A can also be observed when training on sequence B, but not as much with fast learning as with slow learning.

Table 5.4: Training on sequence B

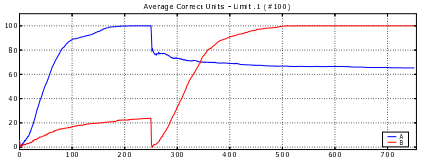
Epochs	500
Fast learning rate	0.04
Slow learning rate	0.01
Slow vs fast ratio	0.25
Momentum	0.50

5.3.4 No reimprinting

The first run is done without any form of reimprinting for the network memory. The simulations are similar to the one replicating the results of French et al. (2002), but the learning rates are different. As you can see from figure 5.7 the level of forgetting is approximately 50% with fast learning, and somewhat lower with only 40% for slow learning. The first dramatic drop is caused by the great weight changes demanded by the new patterns. When the conflicts of the new patterns have been sorted out and the red graph ascends, the blue graph for sequence A also shows an improvement. This is due to a certain representational overlap and can be attributed to chance. 50% correct units can thus be seen as a bench-mark of bad performance for my simulations.



(a) Fast learning, average number of correct units



(b) Slow learning, average number of correct units

Figure 5.7: Learning with no reimprinting

5.3.5 Only pseudopatterns

The previous section showed the limits of catastrophic forgetting for fast learning, but also that some of the memory was lost with the use of slow learning. It is also necessary to show the implications of continued training on only pseudopatterns. The simulation in figure 5.8 shows how performance drops to 93% without any interference from learning a new sequence. This is due to one of the ambiguous input patterns overwriting the memory of the other one, resulting in a drop to 55% on that sequence item. This is the

upper limit for how well a network trained on both sequence A and B will be able to perform.

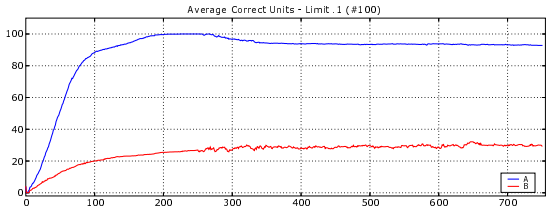


Figure 5.8: Training using only pseudopatterns

5.3.6 Self enforce

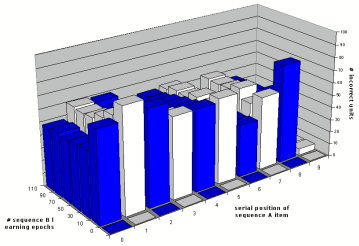
Fast learning

With the *self enforce method* the training on sequence B is interleaved with pseudopatterns generated on-the-fly (see section 4.1.3). Fast learning results in catastrophic forgetting of 40%, which is very close to the downward limit of 50%. The immediate dramatic performance drop which was observed without reimprinting is also observed here. Figure 5.9 shows the dramatic effect of forgetting; the pseudopatterns are not able to aid in remembering. The memory that could have been preserved is lost. The minor improvement in performance is only due to sequence B conflicts being sorted.

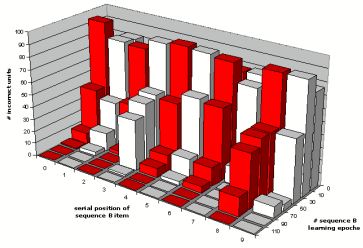
Ambiguity was added by making the input patterns of sequence items 3 and 8 identical. Figure 5.9b shows how their performance reaches criterion later than the other sequence patterns.

Slow learning

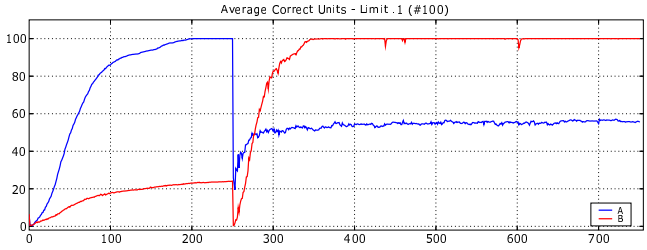
Figure 5.10 shows what can be achieved using slow learning. The plateau of learning stagnation from sequence A can also be observed when learning sequence B. Although it takes much longer time to learn the new sequence to criterion, the rate of forgetting is gradual. When performance is satisfactory, close to 90% of sequence A is still remembered. Once again it is one of the ambiguous sequence items that suffers the worst blow. Compared to training only on pseudopatterns, as in section 5.3.5, using slow learning does not represent a dramatic loss of memory. Although forgetting is not fully abolished, it is no longer catastrophic, and most of the memory is kept intact.



(a) Forgetting of sequence A

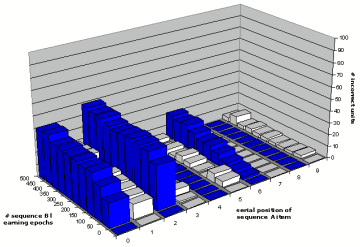


(b) Incorrect units of sequence B

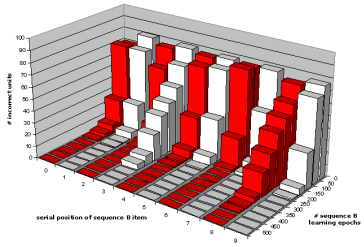


(c) Average number of correct units

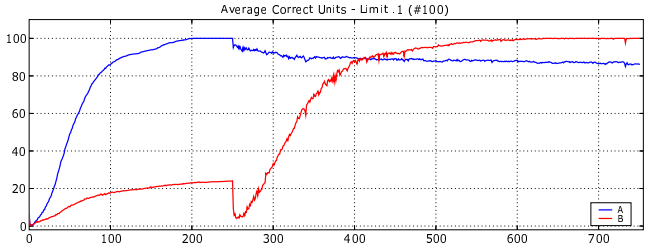
Figure 5.9: Fast Learning with self enforce



(a) Forgetting of sequence A



(b) Incorrect units of sequence B



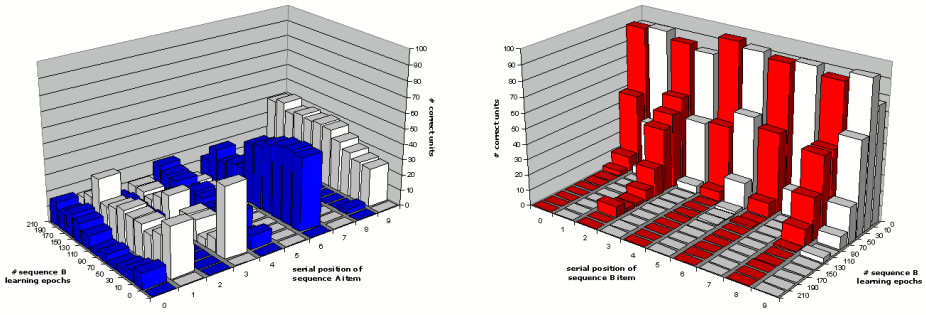
(c) Average number of correct units

Figure 5.10: Slow Learning with self enforce

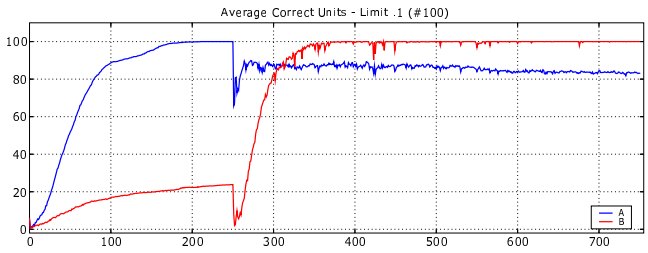
5.3.7 Circulated storage

Fast learning

The *circulated storage method* presents middle course between dual-networks and self enforcing. It keeps a list of patterns which are circulated with new ones created on-the-fly. See section 4.1.3 for a more thorough explanation. This method shows results that are very similar to those of self enforce in section 5.3.6, but the deflections are less extreme. This becomes very evident in the case of fast learning. Pseudopattens contained in the storage list are not susceptible to interference from new learning and are able to reduce memory loss from more than 40% to less than 20%.



(a) Forgetting of sequence A (b) Incorrect units of sequence B



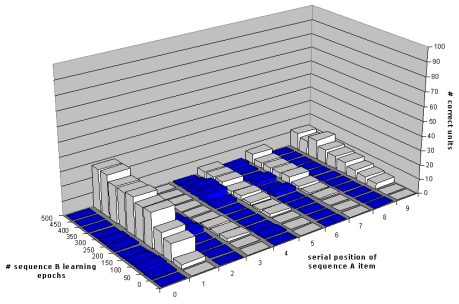
(c) Average number of correct units

Figure 5.11: Fast learning with circulated storage

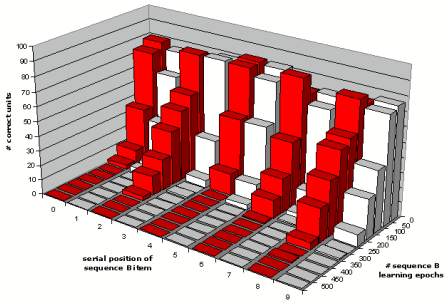
Slow learning

Storing a few patterns externally is able to reduce the level of catastrophic forgetting for slow learning compared to fast learning by 10%. One can see that with a low learning rate (figure 5.12 the graphical results are directly comparable to those of self enforcing. Although learning sequence B to criterion takes longer time than for self enforce, the

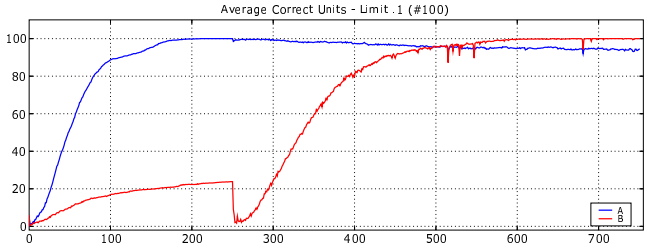
rate of forgetting is very gradual. As for self enforce, at satisfactory performance for sequence B as much as 90% of sequence A is still remembered.



(a) Forgetting of sequence A



(b) Incorrect units of sequence B



(c) Average number of correct units

Figure 5.12: Slow learning with circulated storage

Chapter 6

Conclusion

6.1 Summary

This thesis described an experimental framework using slow learning in artificial neural networks to reduce catastrophic forgetting in multiple sequence learning. Because of the elaborate process of information transfer in the dual-network architecture, its abilities was restricted by relying on a single RSRN (section 4.1.1). After first having learnt one sequence, the training of a new sequence is interleaved with self-generated pseudopatterns. These pseudopatterns are used to prevent memory loss by enforcing the current memory state of the network (section 4.1.3). Learning speed is varied by differentiating the learning rates for new and old memories. The use of fast learning results in catastrophic forgetting, rendering performance on old memories virtually non-existant even before the learning of the new sequence is showing results. Slow learning only shows a limited display of gradual forgetting where most of the previous memory is conserved. This result is achieved both with the self enforce method (section 5.3.6) and with the circulated storage method (section 5.3.7).

6.2 Discussion

As already stated in the introduction, the main goal of this thesis is to investigate how learning speed affects the memory of an artificial neural network presented with the task of learning multiple sequences, and only allowed to enforce its current memory by interleaving new learning with self-generated pseudopatterns.

The ability to extract generalities and idiosyncracies from the training data is one of the

most important reasons for slow interleaved learning (section 2.4.1). The sequences are created with a forced difference in their distribution, a choice of pattern encoding which effect is maximized interference between the patterns (section 4.2.4). Crosstalk between patterns should not occur according to this forced difference French (1991). Thus, the possibility to investigate the effects of pseudopatterns on concept learning and extracting similarities is not present in the setup

Ambiguous patterns are the first to be affected by interference from new learning. It takes much effort to make both of these patterns part of the network function, this is probably why one of them is dominated and disappears, while the other one remains unaffected. When they are imprinted equally strong which one remains in memory depends on what the first pseudopatterns used in training reflected.

The development of the hidden representations is not controlled, and 50 units are substantially more than the 20 input patterns the network has to learn how to distinguish. Thus, there is a possibility for the network to build a look-up table in hidden memory for all the distinct input patterns (section 3.1.5).

Pseudopatterns play an important role in these simulations, this makes it interesting to inspect how they distribute compared to the actual sequence patterns used to establish the memory of the trained network. Half the pseudopatterns undergoes a process of pattern completion and becomes identical to one of the original input patterns. The rest are distributed shows a normal distribtion with a peak similarity between 75 and 80 identical units. These pseudopatterns are static input-output pairs reflecting a dynamic state (section 3.2.2). In the dual-network architecture information can be transferred between networks by the use of pseudopatterns. It is the collective contribution from a large set of pseudopattern that is able to re-represent the network's function. Their distributed effect is able to mirror the training effects from the original patterns used in learning.

One simulation even shows how the weights updates produced only by means of self-generated pseudopatterns actually lead to interference with previous learning and certain degree of forgetting (section 5.3.5). Pseudopatterns are extremely sensitive to sudden and large weight changes. By keeping the learning rate low the pseudopatterns do a better job in reinforcing the present memory (section 5.3.6). Fast learning leads to very rapid and catastrophic forgetting. Performance shows an immediate drop due to the interference in the pseudopatterns, a memory which is never recovered. Although slow learning uses much longer time to reach optimal performance than fast learning, memory only suffers from gradual forgetting troughout this extended period. Using a low learning rate for the new sequence helps because the pseudopatterns are sensitive to weight changes, and training on the pseudopatterns stabilize the effect of the interference. Forgetting is less dramatic, but learning takes a long time leading to a small drop in memory throughout this extended period. Human memory is not perfect either, but

some loss of accuracy is the price to be paid for strength in other areas.

6.3 Future work

6.3.1 Extensions to the experiment

These experiments could benefit from the use of a more biologically plausible learning mechanism, but decisions made early in the project made it difficult to accomodate its inclusion. But as discussed in section 4.2.5, is biological plausibility really a necessity? Developing better neural networks and learning methods is a goal in itself.

With the current experimental settings the memory capacity of the network is not challenged due to its large number of possible input patterns. The memory capacity could be tested both by reducing the size of the input patterns and the length of the sequences.

Research has also showed that the information transfer between networks also occurs with a forced dissimilarity between the pseudopatterns and the original training patterns (section 3.2.2). Simulations in this thesis showed that it is those patterns that provide ambiguity to the sequences that suffers the worst blow of catastrophic forgetting. By applying a similarity constraint it could be possible to face some of the extent of forgetting which is experienced for the ambiguous patterns.

The methods in this thesis construct learning speeds by differentiating the learning rates of the different learning activities. Another way to impose this difference in learning speed could be achieved by varying the number of sequence patters compared to the number of interleaved pseudopatterns in every epoch. This would allow a more distributed weight update contribution from the pseudopatterns, something which is more similar to how pseudopatterns act in a dual-network architecture.

6.3.2 Other approaches

The learning task presented in this thesis does not embody any underlying structure to be extracted by the network. By adding autoassociative units to a concept network of the type presented by McClelland et al. (1994) it would be possible to produce self-generated pseudopatterns and investigate how the self enforce method affects the extraction of generalities. Pseudopatterns can be interleaved with the learning of new concepts and the level of interference can be investigated. This includes both the level of complete forgetting by interference to other types of concepts, and how much is remembered of the idiosyncrasies of similar concepts.

In general, it would be very interesting to conduct experiments with a more challenging learning task, not just sets of random vectors. The first step would be to construct training patterns which embody an inherent meaning, for example structural knowledge. The same learning process could be used to explore how initial memory from training on a set of knowledge A may experience forgetting due to interference from knowledge B. Similarity is here to interpreted as by McClelland et al. (1994), as *"overlapping sets of propositions"*. By performing an experiment where A and B involves dissimilar knowledge areas it is possible to examine the level of catastrophic forgetting. It would also be possible to construct training sets with mostly similar knowledge in A and B, which also include subtle differences. As for the concept network, this would lead to an experimental framework that could examine both the network's ability to generalize and how knowledge is integrated into familiar structure. Will the network be able to keep track of subtleties for the different concepts?

Appendices

The source code for the implementations of the reverberated simple recurrent networks and the dual-network architecture are both made available in the digital version of this thesis. Their respective graphical user interfaces are also included and can be run directly.

Python files:

DNet.py	Dual-network architecture
DNetGui.py	GUI for DNet
RsrnMs	RSRN with multiple sequences
RsrnMsGui	GUI for RsrnMs
Functions.py	Collection of various help functions

It is not enough to install only Python to run the code. Below follows a list of the other libraries that are needed and where they can be found.

Necessary downloads for running the Python files:

Python	$http://www.python.org/download/$
wxPython	$http://www.wxpython.org/download.php$
matplotlib	$http://sourceforge.net/projects/matplotlib$
Numerical Python	$http://numeric.scipy.org/$

Bibliography

Bar-Yam, Y. (1997), *Dynamics of complex systems*, Perseus Books, Cambridge, MA, USA.

Burgess, N. & O'Keefe, J. (2003), Hippocampus: Spatial models, *in* M. Arbib, ed., 'The Handbook of Brain Theory and Neural Networks', The MIT Press, Cambridge, MA, pp. 539–543.

Burgess, N., Recce, M. & O'Keefe, J. (1994), 'A model of hippocampal function', *Neural Networks* **7**, 1065–1083.

Callan, R. (1998), *Essence of Neural Networks*, Prentice Hall PTR, Upper Saddle River, NJ, USA.

Downing, K. L. (2005), The predictive basis of situated and embodied artificial intelligence, *in* 'GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation', ACM Press, New York, NY, USA, pp. 43–50.

Elman, J. L. (1990), 'Finding structure in time', *Cognitive Science* **14**(2), 179–211.

French, R. M. (1991), Using semi-distributed representations to overcome catastrophic forgetting in connectionist networks, Technical Report CRCC-TR-51-1991, University of Liège, 510 North Fess, Bloomington, Indiana.

French, R. M., Ans, B. & Rousset, S. (2001), Pseudopatterns and dual-network memory models: Advantages and shortcomings, *in* R. French & J. Sougné, eds, 'Connectionist Models of Learning, Development and Evolution', Springer, London, pp. 13–22.

French, R. M., Ans, B., Rousset, S. & Musca, S. (2002), Preventing catastrophic interference in multiple-sequence learning using coupled reverberating elman networks, *in* 'Proceedings of the 24^{th} Annual Conference of the Cognitive Science Society', NJ:LEA.

Joost, M. & Schiffmann, W. (1997), 'Speeding up backpropagation algorithms by using cross–entropy combined with pattern normalization'.

McClelland, J. L., McNaughton, B. L. & O'Reilly, R. C. (1994), Why there are complementary learning systems in the hippocampus and neocortex: Insights from the

successes and failures of connectionist models of learning and memory, Technical Report PDP.CNS.94.1, Carnegie Mellon University and The University of Arizona.

Mitchell, T. M. (1997), *Machine Learning*, McGraw-Hill Higher Education.

O'Keefe, J. (1991), The hippocampal cognitive map and navigational strategies, *in* J. Pailard, ed., 'Brain and Space', Oxford University Press, pp. 273–295.

O'Reilly, R. C. & Rudy, J. W. (2000), 'Computational principles of learning in the neocortex and hippocampus'.

Pyt (2006), *Python Programming Language - Official Website*, http://www.python.org/.

Rolls, E. T. & Treves, A. (1998), *Neural Networks and Brain Function*, Oxford University Press.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1987), Learning internal representations by error propagation, *in* D. E. Rumelhart, J. L. McClelland et al., eds, 'Parallel Distributed Processing: Volume 1: Foundations', MIT Press, Cambridge, pp. 318–362.