NTNU
Norwegian University of
Science and Technology

Short- and Long-term Memory: A Complementary Dual-network Memory Model

William Peer Berg

Master of Science in Computer Science
Submission date: June 2016
Supervisor: Keith Downing, IDI

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

I would like to express my sincere thanks to my supervisor, Keith Downing, for his valuable support and guidance during the work on this thesis.

I would also like to express my gratitude towards Lester Solbakken, who recommended me the book ’On Intelligence’, which lured me into the field of artificial intelligence.

Abstract

In recent years, the possible applications of artificial intelligence (AI) and deep learning have increased drastically. However, the algorithms which constitute the learning mechanisms in deep learning are based largely on the same principles as when formalised about half a century ago. Namely using feed-forward back-propagation (FFBP) and gradient based techniques in order to train the artificial neural networks (ANNs). When training an FFBP ANN within a novel domain, it seems inevitable that this training will largely, and quite rapidly entirely disrupt the information which was formerly stored in the network. This phenomenon is called catastrophic interference, or forgetting, and remains a long-standing issue within the field.

An architecture addressing this issue is the dual-network memory architecture, which by addressing two fundamental aspects of memory acquisition in neural networks, namely short- and long-term memory, reduces or eliminates catastrophic forgetting, as well as suggests biological implications. However, former implementations reduce catastrophic forgetting by employing pseudorehearsal, by implicitly re-training on the former weight configuration. While this provides a means of interleaving the former information with the new, it remains a slightly unrealistic training scheme.

In order to address these crucial issues within the dual-network memory architecture, this thesis implements a more biologically plausible dual-network memory model, and a novel memory consolidation scheme. Building upon the work of Hattori (2014)[1], a more biologically realistic short-term memory model is attained, from which information may be consolidated to a long-term memory model. The model and its associated behaviour is analyzed, and a novel parametrization and resulting memory consolidation mechanism is demonstrated. This mechanism reduces catastrophic forgetting without employing pseudorehearsal, when exposing the dual-network memory model to five consecutive and distinct, but correlated sets of training patterns. This demonstrates a potential neural mechanism for reducing catastrophic forgetting, which may operate in synthesis with, or instead of pseudorehearsal. This novel memory consolidation scheme is regarded as fairly biologically realistic, as it emerges from several hippocampal aspects that are empirically observed and documented within the literature. Furthermore, the mechanism illuminates several interesting emergent qualities of pattern extraction by chaotic recall in the attained hippocampal model.

[1] Hattori, M. (2014), A biologically inspired dual-network memory model for reduction of catastrophic forgetting. *Neurocomputing 134*, 262-268.

Table of Contents

Abbreviations

AI	=	Artificial intelligence
NN	=	Neural network
ANN	=	Artificial neural network
LTM	=	Long-term memory
STM	=	Short-term memory
RNN	=	Recurrent neural network
LSTM	=	Long short-term memory
GRU	=	General recurrent unit
FFBP	=	Feed-forward back-propagation
BP	=	Back-propagation
BPTT	=	Back-propagation through time
HPC	=	Hippocampus
PFC	=	Prefrontal cortex
GD	=	Gardient-descent
SGD	=	Stochastic gradient-descent
SDM	=	Sparse distributed memory
k-WTA	=	k-winners-take-all
CTRNN	=	Continuous-time recurrent neural network
MTRNN	=	Multiple-timescales recurrent neural network
CPU	=	Central processing unit
GPU	=	Graphics processing unit
GPGPU	=	General processing graphics processing unit
I/O	=	Input/output

Chapter 1

Introduction

In recent years, the possible applications of artificial intelligence (AI) have increased drastically. From autonomous self-driving cars (Urmson et al., 2009), to facial recognition systems with super-human performance (Sun et al., 2014), to IBM's Watson performing medical diagnoses (Wagle, 2013), to Google's DeepMind playing Atari 2600 games (Mnih et al., 2015). Yet there is a vast amount of potential which has yet to be explored within the field; as IT is becoming increasingly ubiquitous, so does the potential applicability of AI. One of the reasons is that the presence of IT enables the harvesting and analyzing of data. Two of the common factors for recent advances within AI are increased computational power, and algorithmic improvements. Not only have algorithms performing facial recognition using deep learning become tractable on an average high-end desktop-computer, there has also been an explosion in data generation as well as availability within recent decades. In the future, we are likely to see a similar expansion in available data within the public domain, with a vast array of applications. Also quite noteworthy, this is occurring in synthesis with the Internet of things, i.e. connecting ordinary objects to the Internet. In other words, visual recognition will be far from the only tangible deep learning domain in the future. Note also that a synthesis of data from different domains may enable a cross-domain data synthesis, which although increasing the complexity of the data set may also increase the quality of the predictions performed by deep learning algorithms. Cross-domain syntheses have already shown promising results in tasks such as feature extraction, and image classification, and in recommendation systems (Huang and Wang, 2013; Shapira et al., 2013).

At the core of several of the aforementioned algorithms lies the sub-field of AI known as deep learning, as previously mentioned. Deep learning is a class of algorithms where multiple layers of processing are employed, enabling the extraction of intricate correlations and patterns in data sets without any prior domain-knowledge nor supervision during learning. Employing a priori knowledge through convolutional filters, has further enabled deep learning algorithms to attain excellent performance in domains such as speech recognition, image classification, and genomics (LeCun et al., 2015), displaying the wide applicability of deep learning, as well as the potential benefits it may bring to society. Furthermore, the convolutional filters, elements of pre-processing that essentially perform feature detection in data sets, may be constituted solely by deep neural networks, i.e. by employing the same structure and algorithm as in the rest of the network (LeCun et al., 2015). This may enable a network to learn convolutional operations, i.e. desired filters, as data sets is presented to the model. Resulting in a more dynamic and generally applicable unsupervised learning algorithm, although possibly requiring larger data sets for training. Summarising, the same technique for extracting intricate correlations in a data set may in fact be used to find the filters that reduce raw input data to feature vectors, which the remaining deep network may then process. This is the core algorithm employed in deep learning; the artificial neural network (ANN). Two of its variants, the traditional feed-forward back-propagation (FFBP) network, using gradient-descent, and a slightly more complex and biologically realistic network using k-winners-take-all and Hebbian learning, are employed in this thesis. I will get back to the research goal as well as clear research questions later in this introductory chapter.

The neural network is a biologically inspired class of algorithms which borrows its vocabulary from neuroscience. Going back to 1943, McCulloch and Pitts (1943) proposed to formalize neural functioning within logical propositions. However, they could only create a logical calculus based upon the abstract assumptions of the neurological and psychological basis of the time. Formalizing neural networks was nevertheless a seminal contribution, in which the important assumption that neurons could be considered as binary processing units was made, due to the observation that they seemed to fire in an ”all-or-none” fashion. This created a framework with which psychological phenomena could be regarded in a reductionist way; through the lens of two-valued logic. The way in which neural networks were defined, and thus changed during termination, however, remained obscure. McCulloch and Pitts (1943) noted that: ”With determination of the net, the unknowable object of knowledge, the ”thing in itself”, ceases to be unknowable” [1] Thus describing the long-standing symbol-grounding problem, which has yet to be elucidated.

In his 1895 manuscript for ’Project for a Scientific Psychology’, Sigmund Freud had in fact already suggested that synaptic transmission could promote post-synaptic neural excitation (Kiernan, 2011). Despite predicting a variant of Hebbian learning more than half a century before Hebb himself, his manuscript was not published until 1950. Therefore the phenomena of synaptic modification resulting from, and occurring relative to the correlation in neural activity among neurons, is accredited to Donald O. Hebb (Kiernan, 2011). As Hebb (1949) eloquently put it: ”One cannot logically be a determinist in physics and chemistry and biology, and a mystic in psychology” [2]. He further outlined the paradigm in which neural functioning is deterministic, and proposed the seminal concept of Hebbian learning, in his 1949 book ’The Organization of Behavior: A Neuropsychological Theory’. Hebbian learning is widely recognized as the fundamental mechanism underlying synaptic modification and learning in biological neural networks. Simply put, Hebbian learning may be summarised as; ”fire together; wire together”.

In less than a decade after the proposal of McCulloch and Pitts (1943), a formal cal-

[1] McCulloch, W. S., & Pitts, W. 1943. ’A Logical Calculus of the Idea Immanent in Nervous Activity’, *Bulletin of Mathematical Biophysics*, **5**: 131.

[2] Hebb, D. O. 1949. *The Organization of Behavior: A Neuropsychological Theory*. New York: JOHN WILEY & SONS, Inc. Pp. xiii

culus for neural network computation, as well as a neuropsychological theory containing principles for how neurons may perform synaptic modification was proposed. The latter suggesting a mechanism for neural network computation, and thus possible mechanisms for memory and learning. From this historical point in time and forward, there has been a continuous synthesis between psychology, neuroscience and computer science with regard to neural functioning: Computer science lending itself to construct models within computational neuroscience, which seek to explain certain neurological aspects of brain functioning. These, in turn, may explain or be related to aspects of cognition. Conversely, insights from neuroscience and psychology may lend themselves to, and inspire researchers within more computationally related disciplines in creating more powerful neural network algorithms, performing tasks such as pattern extraction, clustering, classification or segmentation. Naturally, this synthesis has spawned different scientific fields. Two of which are known as connectionism and computational neuroscience. While computational neuroscience attempts to tackle problems within neuroscience, as briefly touched upon above, connectionism may be described as aiming solely towards attaining neural network behaviour, potentially disregarding the biological plausibility. This does not necessarily mean that researchers seek to use the model only for computer scientific purposes, as one may argue that certain psychological or neurological aspects may be studied without all model aspects being biologically plausible. In fact, until we fully fathom brain functioning neurologically speaking, we cannot hope to create a model which will encompass all of its biological aspects. Note, however that this may not be a necessity in order to have certain phenomena emerge from a neural network model. In fact, it may be hypothesized that certain principles underlie a certain neural functioning, and thus if simulated algorithmically, the same behaviour may emerge in a model, despite possibly being implemented quite differently. This is what largely forms the basis for connectionism, where researchers aim to study cognitive phenomena through usually simplistic computational models. The main point being that the constraint of biological plausibility is relaxed. Quite often, a synthesis between the two aforementioned fields emerges in engineered models and solutions. This might enable researchers to study both how the brain may implement certain functionality, as well as to study possible algorithmic implications within the more purely applied disciplines of AI such as deep learning.

This is the primary motivation for the thesis topic.

Before introducing the model which will be used in this thesis, I would like to introduce the concept of catastrophic forgetting. Hebb (1949) described the general case of learning new information which may disrupt old, as the sensitivity-stability dilemma, in which one has to decide on how sensitive a network should be to new information through its parametrization. Making it more sensitive to recent information would of course most likely, depending on the network model, largely disrupt the old information contained within the network, thus making it very unstable, and conversely; maintaining the current configuration too heavily may result in failing to acquire new knowledge or to extract new information. This problem is also known as the stability-plasticity (Carpenter and Grossberg, 1987) problem, placing network models on a scale from being very plastic to very stable. Catastrophic forgetting (McCloskey and Cohen, 1989; Ratcliff, 1990) is a term which describes the phenomena of when an ANN model ”forgets” large parts, or everything, that it has previously learnt. I.e. the weight tuning which corresponds to correlation extraction in a data set is erased. This may occur to such an extent that the model performance is equal to that of randomly assigning its network weights. Catastrophic interference is a term capturing both catastrophic forgetting, as well as other types of interference, such as when a network model fails to attain new knowledge, i.e. the network being unable to capture further correlations. For instance, in a recurrent network using Hebbian learning, catastrophic interference occurs when increasing the number of training patterns beyond a certain extent. This results in the network not only failing to learn new patterns, but it also disrupts old, previously learnt patterns, possibly making the model useless. This is the case in Hopfield networks, which I will get back to in chapter 2, where stable states may be considered as basins of attractions in a three-dimensional space. If there are too many basins of attraction, the behaviour will be unstable, and the state of the network will oscillate in a chaotic manner. When it comes to traditional feed-forward back-propagation ANNs; training a network in a novel problem domain using gradient-descent will adjust the weights according to the new domain only, neglecting all knowledge that may have been previously attained (McCloskey and Cohen, 1989; French, 1999; French et al., 2001). This necessarily results in catastrophic forgetting if training solely on new patterns, which then disregards old. If all patterns are included, the old and new alike, the algorithm of FFBP using gradient-descent will seek to minimize the error signal over all patterns, thus maintaining old knowledge equally well as new, as long as it is represented in the current training data. Note, however, that rehearsing on all previously learnt information may be regarded as biologically unrealistic. This provides the basis for investigating complementary or optional, and potentially more biologically realistic mechanisms for reducing catastrophic forgetting. Note also that catastrophic interference may occur if a model is given a training set which exceeds the complexity that the network may capture. If the network complexity is insufficient to extract the desired distributions from the data set, gradient-descent may also fail to converge towards a solution, or only extract the most principal components. The fact that catastrophic *forgetting* occurs in an FFBP ANN model when it is trained in a novel problem domain, reflects that the network is only a local stochastic extraction of correlations from a probability distribution constituted by data sets from that particular domain. Interestingly, recurrence in networks may enable a network to capture more complex dependencies, such as temporal ones, but it may also introduce a larger state-space in doing so.

McClelland et al. (1995) propose that the brain solves the problem of catastrophic forgetting by a dual-network memory architecture, implemented by the hippocampus and neocortex. However, the body of research within AI on the architecture, is to the best of my knowledge fairly limited. Furthermore, proposed implementations suffer from issues related to simplification or obscurity (French, 1997; French et al., 2001; Hattori, 2010, 2014). Note also that it is only recently that a more biologically plausible hippocampal network in such a model has been studied. Hattori (2014) investigates how seeking to capture the chaotic macro-scale behaviour of the CA3 region of the hippocampus affects the model's behaviour, and concludes that the model is significantly improved when compared to his former work. His former work does in turn significantly outperform previous implementations of the dual-network memory architecture. A short abstract definition of the model is that it consists of two networks, where one represents working or short-term memory, and the other long-term memory. As patterns are learnt by the first network, they are consolidated to the second through the use of pseudopatterns. A pseudopattern remains roughly the same throughout different papers investigating the dual-network memory model, such as (French, 1997; Ans and Rousset, 2000; French et al., 2001; Hattori, 2010, 2014). Fundamentally, a pseudopattern is a pattern that reflects the configuration of the long-term memory network, which may be used to represent former training patterns such that they may be re-learnt along with new, minimizing the loss of old knowledge. Pseudopatterns are introduced formally in the following chapter; chapter 2.

Deep learning has led to a tremendous advance in the capabilities of AI within recent years. However, the plasticity needed to combine memories and patterns in a more general and abstract way may not be not yet be present in today's state-of-the-art deep learning algorithms. One observation supporting this statement is the fact that catastrophic forgetting occurs in FFBP ANNs. Because modern state-of-the-art algorithms still largely employ variants of gradient descent, it is likely that they are prone to the same type of interference. Possibly ameliorated by the recently drastically increased storage capacity of today's ANNs and deep learning algorithms. It is worth mentioning that in addition to seeking to reduce catastrophic forgetting through a more complex short-term memory model, slightly more intricate network structures may also enable more sophisticated abstraction, as exemplified by authors such as Tani (2014). This basis may be formed by the dual-network memory architecture, which enables abstraction of short- and long-term pattern-associations. In this thesis the main research topic is to study memory abstraction in the dual-network memory architecture, investigating possible implications for general deep learning algorithms, as well within computational neuroscience. Note that memory is an abstract term used to refer to functions, patterns, or data set correlations that a network extracts and maintains for a variable time span.

In this thesis, the primary research goal is: *"To study how the brain might implement short- and long-term memory using the dual-network memory architecture, and to implement a novel dual-network memory model"*.

I aspire to do this by further investigating the dual-network memory model of (Hattori, 2014), and related work. More specifically, building upon Hattori's (2014) model, my aim is to answer the following questions:

- What are the limitations of pattern extraction in the hippocampal model?
- How does asyncronous or synchronous CA3-neuronal simulation affect hippocampal model behaviour?
- How does neuronal turnover, and a synaptic weight coefficient for the outgoing synapses of the DG impact hippocampal model behaviour?
- What information seems to be inherent in the patterns extracted by chaotic recall in the hippocampal module?

- Can the original training patterns be consolidated to the neocortical network by solely using chaotically recalled patterns?
- Can chaotic interference be reduced in the novel dual-network memory model without pseudorehearsal?

I believe that a central aspect for advancing the frontier of deep learning is to investigate how high-level level cognitive behaviour and functionality may emerge in ANNs. Investigating mechanisms associated with reasoning over different memories may potentially provide insights for attaining greater plasticity and generalization in ANN models. This may be seen by considering that a crucial aspect of being able to combine different memories is simply remembering what has previously been learnt. Therefore the foremost goal in this thesis is to investigate the dual-network memory architecture in this context.

The structure of the thesis will be as follows: After this introductory chapter, a background chapter follows. This chapter contains two fairly short reviews of the fields of connectionism and computational neuroscience. Furthermore, it formally introduces catastrophic interference and forgetting, and lays out a theoretical foundation for neural networks and the dual-network memory model. In chapter 3, the methods and implementation of the model is outlined. This includes describing the elected programming environment, a formal definition of the dual-network memory model which is implemented, along with model decisions and system design. Preceding the chapter on the methods and implementation is the chapter containing the experiments and results, namely chapter 4. This is the main chapter of this thesis, and the initial as well as designed experiments are outlined, along with the attained results. Furthermore, the specific sections, containing the individual experiments, also contain a fairly detailed analysis of the experiment results, as these provide the basis for designing further experiments that are also implemented. Lastly, in the final chapter the main thesis findings are summarized, particularly from chapter 4. Furthermore, the research questions are addressed, after which the methods of the thesis are discussed. In the final section aspects and areas of research that I would like to focus on and address in future work are introduced, along with biological and slightly more philosophical parallels and hypotheses.

Chapter 2

Background

2.1 Connectionism and deep learning

The processing unit in an artificial neural network (ANN) is the artificial neuron. This unit may be represented as a single activation value, symbolising a neuron's internal state. In order to process information, vectors of activation values representing neuronal activation in a layer may be multiplied with matrices of weights, representing connections between neurons of different layers, i.e. synapses and synaptic connection strengths. This linear algebra operation propagates information throughout the network, and is commonly known as the feed-forward process in the classical domain of neural networks. In order to arrive at a weight configuration which lets a network perform a certain task, it may be trained using gradient descent in weight-space (Hinton, 1989). A common implementation is by back-propagating an error signal through the network whilst attempting to minimize it, adjusting the network weights accordingly. The error is usually a loss function such as the l2-norm (see appendix A) of the difference between an acquired output and a target output, which may be thought of as the Euclidean distance in space. This technique of training a neural network; back-propagation, was largely popularized by Rumelhart et al. (1986). Furthermore, Rumelhart et al. (1986) made the important choice of electing the logistic function, also known as the sigmoidal function, as their candidate transfer function for propagating activation values through synapses in their experiments and models. That is, the sum of a neuron's input is run through the sigmoidal function, which has two important characteristics: **(1)** it puts a lower and upper bound on the activation values of a neuron, (-1, 1), and **(2)** it is continuous and differentiable, resulting in numeric methods of differentiation being applicable for weight adjustment relative to the change in activation values. In other words, the transfer function may be thought of as a crude mathematical approximation to a neuron's internal dynamics. When ANNs are constructed in this manner; as simple activation values, weights between the values, and transfer functions between the different layers, the resulting models are often referred to as connectionist models. An example includes the aforementioned traditional feed-forward back-propagation (FFBP) neural networks of (Rumelhart et al., 1986).

When it comes to deep learning, this is primarily an engineering discipline in the sense of its models being more focused on creating applicable systems and solutions, rather than on explaining the biological systems from which they originate. I would like to emphasize that despite this; it is the synthesis of neuroscience, psychology, and computer science that has given rise to the field of artificial neural networks (McCulloch and Pitts, 1943) and its various sub-fields. Furthermore, they continue to be factors in advancing the field's applications. This is exemplified by the recent deep learning algorithm in which the biologically inspired long short-term memory (LSTM) unit (Hochreiter and Schmidhuber, 1997), and the even more recently proposed, and perhaps simpler, gated recurrent unit (GRU) (Mnih et al., 2015), enables deep networks to capture temporal dependencies in data sets - adding a fundamental and crucial richness to what correlations and structures that may be captured by this class of general learning algorithms; namely long-term temporal dependencies within data. While a unit such as the GRU does not necessarily demonstrate the workings of the biological brain, it does demonstrate that studying aspects which the biological brain captures, and translating them into algorithmic principles or requirements, may significantly improve engineered solutions, having both computer scientific value and impact. In other words; attaining further knowledge within the domain of computational neuroscience may lead to algorithmic advances within deep learning, and vice versa. Had the GRU been discovered first, this could have led to the hypothesizing of recurrence being crucial to capturing temporal dependencies within neural networks. Even though this is already a widely appreciated fact within neuroscience, the LSTM and GRU may still have an impact on the field of neuroscience, as we continue to discover why they enable algorithms to perform more sophisticated types of processing. While it is not the aim of this thesis to categorize algorithms as belonging to one branch or another within the associated fields of neural networks, it is my intent to outline connectionism and computational neuroscience in order to *clarify* the synthesis of fields related to neural networks and AI, as well as to establish the context for the model which is studied in this thesis. A further review of computational models is included later in this chapter, after a foundational neuroscientific background has been visited.

2.2 Catastrophic forgetting

Catastrophic forgetting (McCloskey and Cohen, 1989; Ratcliff, 1990) is as outlined in the introduction (chapter 1); the forgetting of model parameters for a domain in which the model has previously been trained. This may occur to such an extent that the network performance is equal to that of random weight initialization.

McCloskey and Cohen (1989) were some of the first to analyse catastrophic interference, noting that it seems inevitable during sequential learning in connectionist models. Furthermore, they found that the cause for interference is that the newest weight adjustments reflect the newest information more heavily. In other words, old information is by the nature of the sequential learning disrupted by the correlations currently being extracted from the current pattern. This remains a fundamental issue for algorithms employing sequential learning, including algorithms such as back-propagation, and other gradient-based algorithms. In fact, it remains a central issue for all network-based algorithms which perform a type of sequential learning.

Ratcliff (1990) studied forgetting in neural networks, and found similarly to McCloskey and Cohen that back-propagation networks are prone to catastrophic forgetting. Furthermore, he found that as more information is learnt by an FFBP ANN, its ability to discriminate between previously presented patterns and the ones currently being learnt is reduced. This contrasts empirical studies of biological neural networks (Ratcliff, 1990), and led Ratcliff (1990) to propose a mechanism for reducing catastrophic forgetting in ANNs. Namely by using neurons which would selectively *respond* to certain input, which he termed 'response nodes'. However, his proposed model only alleviated, and did not successfully eliminate the problem of catastrophic forgetting. Note that various attempts were made by both of the two aforementioned authors, (McCloskey and Cohen, 1989; Ratcliff, 1990), none of whom attained a model which eliminates catastrophic forgetting in FFBP ANNs. French (1992) examined FFBP ANNs in a very similar context, attempting to eliminate catastrophic forgetting by implementing 'node sharpening'. Arguing that catastrophic interference occurs due to overlap in input patterns, French (1992) suggested that the issue could be alleviated, or even resolved, by creating a non-overlapping memory. Therefore he reviewed previous research such as the ALCOVE algorithm (Kruschke, 1992), on using a sparse distributed memory, in which information would rarely overlap. He found this to alleviate catastrophic forgetting to a certain extent. However, it did so by implementing a very large address space. Once this address space would become digested, catastrophic forgetting would indeed still occur. It is worth noting that using a sparse distributed memory requires an algorithm to be able to construct hyperplanes in which the input data is successfully separated. This may result in less generalizability in the network model, if representations are distributed across distinct sub-networks due to the hyperplane mapping. Node sharpening as proposed by French (1992) locally constrains input patterns to enhance the most prominent input features, effectively resulting in sparse representations. This results in less "noise" being propagated throughout the network, the most principal input nodes being in focus, and a significant reduction in catastrophic forgetting. However, this is done at the cost of attaining a less generalisable model, as node sharpening only looks at the *n* most prominent nodes.

The aforementioned process of node sharpening is strikingly similar to using convolutional networks for feature extraction, which was one of the inventions that re-instated neural networks as the state-of-the-art within domains such as image classification (LeCun et al., 2015). It should be noted that today's solutions such as the seminal deep convolutional network of Krizhevsky et al. (2012) performs a much more rigorous analysis of the input vector before passing it to the subsequent layers of the network. Additionally, the method with which node sharpening is performed is a fairly biologically implausible algorithm, though feature extraction itself, e.g. extracting the most important features within a visual scene, is believed to occur biologically speaking. The mechanism with which this occurs is believed to be more similar to feature extraction using ConvNets, the convolution operation, however, being performed by the plastic neural network itself.

Today's state-of-the-art algorithms employ different transfer functions, network topologies, and neuron-models, when compared to French's (1992) node sharpening model. This may render the claim that a trade-off between catastrophic forgetting and generalisation is inevitable obsolete. Building on French's (1992) former work, French (1994) proposes a model which *dynamically* sharpens the most relevant input nodes. Once two different outputs have been presented to a standard FFBP network, a context bias is calculated in the hidden layer, and propagated back to the input layer. Shortly put, this emphasizes the differences of the distinct categories, focusing on segmenting and orthogonalizing on the most prominent properties in differentiation. This results in more orthogonal, well distributed patterns being learnt. Although this type of segmentation works well, it remains similar to the former approach of French (1992), and suffers from the same type of trade-off between remembering and generalisation. As in other FFBP networks performing gradient descent in weight space, the algorithm will only succeed to find the most principal components, scaling with how much information the network is able to store (mainly affected by its size). As a consequence, failing to attain a sufficient accuracy for a given task could be due to ignoring the detailed information present in the segmentation process. Furthermore, it may be the case that only finding the most principal correlations in a distribution does not reveal the distribution's true nature, failing to extract precise correlations and properties. This is at the very core of the sensitivity-stability dilemma (Hebb, 1949), i.e. the trade-off which occurs between the learning of new and disruption of old information. One way way of addressing this dilemma is by multi-network systems, which French (1994) suggests in his conclusion. This may produce refined solutions and abstractions and more sophisticated pattern-associations, although seemingly less computationally efficient. In this thesis, the dual-network memory architecture (McClelland et al., 1995) is the elected architecture for model construction and implementation, addressing catastrophic interference, and how short- and long-term memory may be implemented by the brain by implementing two network modules. The state-of-the-art of dual-network memory architectures is reviewed below in chapter 2.4, after further background material from computational neuroscience has been presented.

2.3 Computational neuroscience

At the other end of the scale of neural networks, used to study emergent behaviour, we have computational neuroscience. In this discipline, Hebbian learning is often seen as the elected learning mechanism, particularly when it comes to the modeling of short-term memory, as it is generally regarded to be fairly biologically realistic, and results in rapid convergence in the computational models (one-shot learning is possible, and sometimes occurs in the model which is outlined in chapter 2.4). Another aspect which may regarded as more biologically realistic is the algorithm k-winners-take-all (k-WTA), where the plausibility arises from regarding it as implementing lateral inhibition. In biological networks, lateral inhibition may occur due to inhibitory neurons depressing the activation of other neurons (Rolls and Treves, 1998a); thus the term long-term depression. As for k-WTA; the k-winners may be regarded as inhibiting the neighbouring neurons of the layer. Note that the algorithmic approach diverges from the biological in that lateral inhibition may be arbitrary, and for a fixed number of k neurons in k-WTA, whereas the brain is not likely to implement such a hard firing rate threshold (number k). Furthermore, depending on the implementation, activation values may be fixed, and possibly binary in the algorithmic approach to lateral inhibition; k-WTA.

While it is assumed that the reader is familiar with neural networks within AI, it is not assumed that the reader has any prior knowledge within neuroscience. Therefore I will

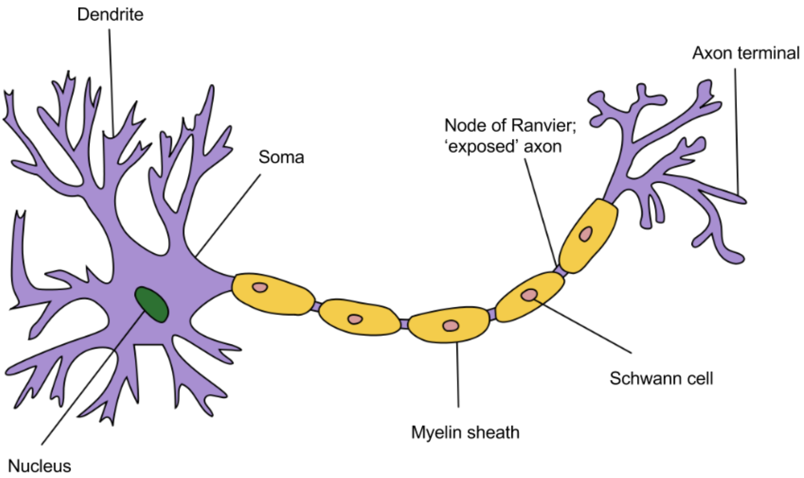

Figure 2.1: Illustrating a **neuron with its dendritic and axonal arborization**, i.e. branching. Please see figure 2.2 for an illustration of a synaptic connection between a dendrite and an axon. The figure is adapted from Wikipedia: By Quasar Jarosz at English Wikipedia, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=7616130

present an overview of the neuroscientific concepts used in this thesis, with the aim of sufficiently covering the neuroscientific background knowledge that is used.

As touched upon in the introduction, AI and neural networks borrows quite a bit of its vocabulary from neuroscience. Readers are very likely to encounter terms such as synapses, axons and dendrites when consulting the literature in computational neuroscience and other related disciplines. These terms refer to the connections between neurons, and the branches which physically enable them, respectively. A synapse actually consists of an axon, also referred to as a neuronal terminal, and a dendrite. Action potential may be propagated through an axon and its synaptic cleft to its connected dendrites, whose small branches are stretched out from other neurons' somata, or cell bodies - lying very closely to an axon terminal, and thus being referred to as connected (see figure 2.1). When a neuron is intracellularly, i.e. internally, excited above a certain level, this triggers a response which results in the release of neurotransmitters, specific molecules, through its axon terminal. These neurotransmitters may then bind to the receptors (see figure 2.2), other molecular structures that bind to specific neurotransmitters, which are on the surface of the dendrite, triggering an intracellular response in the receiving neuron (Campbell and Reece, 2015). Usually, neurons are multipolar, meaning that they consist of several dendrites, and one axon. While the dendrites branch continuously, the axonal arborization occurs at the axon terminal, i.e. the end of the axon, where it emerges from the myelin sheath; an isolating coating which enhances its conductance (see figure 2.1). Note that there are periodically reoccurring gaps without myelin sheathing in figure 2.1, called nodes of Ranvier (Byrne et al., 2014a). These allow the absorption of ions through

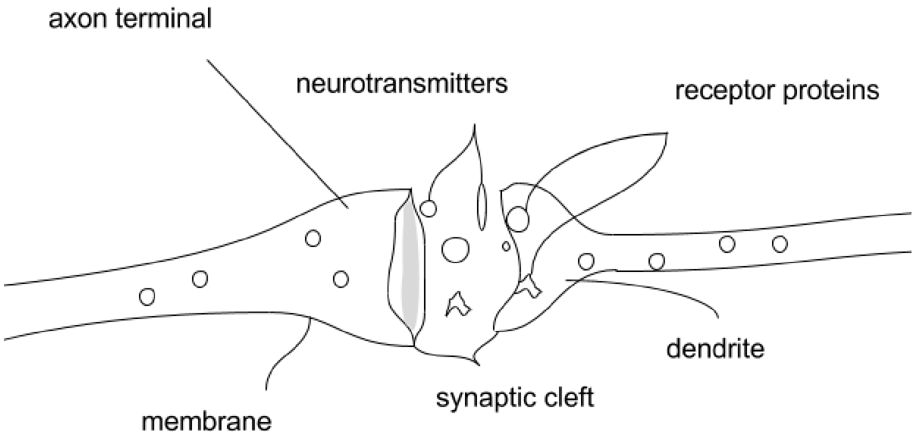

Figure 2.2: Illustrating **a synaptic cleft** in which two neurons are connected. Note that the axon of one of the neurons has released neurotransmitters, which may either excite or inhibit the receiving neuron after these neurotransmitters have bound to the receptors of its dendrite.

ion channels, which are thought to facilitate the action potential propagation coordination and timing (Byrne et al., 2014b). This may also, however, perform a type of information exchange in interacting with extracellular ions. The Schwann cell is a type of support cell, which synthesizes myelin in the central nervous system (Byrne et al., 2014a). From this short outline of synaptic communication, it is possible to see how biological neuronal functioning potentially provides for a much richer type of information processing, when compared to the crude mathematical approximations that are used in most neural network models. Specifically, in the biological brain, different information may be carried by different types of neurotransmitters, triggering different intracellular responses, one of which may be altering the genome of the cell. Furthermore, a synapse may carry information at different speeds, depending on its structure, and synapses may be of different lengths, possibly carrying information not only at different paces, but also performing different intra-synaptical processing of the received relay molecules. Thus potentially performing a rich amount of information processing.

I will regard emergent model aspects related to neuroscience and psychology through a connectionist lens; trying to keep what is biologically inspired biologically plausible where algorithmically feasible, and at the same time employing Ockham's razor (Russell and Norvig, 2009a). Both because the most simple hypothesis achieving the desired behaviour may be argued to be more likely, as it may potentially be more easily attained through evolution, but also because it is easier to implement, and provides for an axiomatic framework with basic building blocks, i.e. the researched algorithms. As argued by Hebb, the framework with which we study aspects of cognition has to be scientific. Therefore it has to be axiomatic, and deterministic in the sense of being algorithmically implementable. Furthermore, I argue that although approximations are made in artificial models, it may be possible to extract mechanisms, properties, and principles of biological neural network functioning. Upon extraction, these principles may be implemented and executed, studied, and perhaps even successful in capturing specific emergent aspects of cognition. I would like to emphasise that although a complete implementation of the biological knowledge we have of neural functioning is infeasible, it may be the case that algorithmic principles underlying the emergent neural functioning and neural network behaviour, may in fact be extracted. In this case, not only does it let us build a framework for studying neural networks - artificial and the like - it also lets do so using the scientific method, and Ockham's razor.

Some examples of computational models which have been able to model emergent neural functioning are models of spatial navigation in place fields and place cells (O'Keefe, 1976; O'Keefe and Burgess, 1996), and the more recently discovered grid cells (Hafting et al., 2005). Whether a mechanistic neuron-level understanding of the brain can be attained, and whether it encompasses what is required to have cognition emerge, remains another discussion. For as long as proven otherwise, the scientific method has to be employed, in which a testable framework needs to be used. We cannot but assume that aspects of cognition may be captured by a mechanistic neural level understanding of the brain. For a further discussion on this topic, using examples from spatial navigation as those mentioned above to draw parallels from computational models to cognition, the interested reader may be referred to appendix E.

Propagation of action potentials may lead to synaptic modification in that dendritic branches grow closer to axon terminals, thus being more respondent to a connected neuron's release of neurotransmitters and activity. This process may be simplified and modeled as an activation value and function for the internal neuronal dynamics together with a weight symbolizing the synaptic connection strength. Various algorithms may be employed for synaptic weight modification in the computational model, as well as for adjusting neuronal activation values. These different approaches may be hybrids of primarily three neural network types as outlined by Rolls and Treves (1998a); **(1)** conditioned stimulus with associatively modifiable synapses, i.e. Hebbian learning, or standard feed-forward back-propagation networks, **(2)** purely recurrently connected associative networks, also known as Hopfield networks (Hopfield, 1982), and **(3)** k-winners-take-all architectures, in which lateral inhibition is algorithmically simulated through letting the k most active neurons fire, possibly with the remaining firing beneath a certain, low threshold. All of these approaches can be said to capture aspects of long-term potentiation in neural networks. However, the algorithm with which a network learns, i.e. how a network attains its weight-configuration, cannot be said to be biologically plausible in the case of minimizing an error signal through back-propagation. Hebbian learning, however, is more biologically realistic, as it updates its weights according to the firing activity between neurons, often representing firing rates. Interestingly, ANNs using Hebbian learning converge very quickly when compared to FFBP ANNs using gradient descent.

This thesis studies how memory may be implemented by the brain in an artificial and simplified neural network model. Below I review some of the material related to memory within the literature of computational neuroscience. More specifically, these theories are quite often based upon studies of the hippocampus. Nearly half a century ago, Marr (1971) argued that the hippocampus acts as a type of short-term or working memory, (Rolls and Treves, 1998b). This has influenced later work, such as (McClelland et al., 1995), in proposing models where the hippocampus functions as an intermediary storage. The reasons for why the hippocampus has been hypothesized to constitute a type of working memory are multiple. One is that it is generally recognized that the hippocampus receives input from nearly every part of the brain, either directly or indirectly (Rolls and Treves, 1998a), and so has the required input to integrate across different stimuli, or memories. Another is that the computational models of hippocampal operation, built upon the empirical data of its structure, i.e. topology and functioning, have been shown to have qualities that are well suited for performing essential operations that are related to memory and cognition.

Firstly, being able to quickly form new memories due to Hebbian learning is a crucial aspect for processing and remembering what happens during temporally constrained events. Therefore a mechanism for rapid learning needs to be present. As Hebbian learning performs weight adjustment according to the current firing pattern, this provides for a quickly converging learning mechanism in artificial neural networks. It is worth mentioning that one-shot learning is also made possible by Hebbian learning. When it comes to the potential storage capacity of memories in the hippocampus, there is a fairly large body of evidence suggesting that the hippocampus most likely employs a distributed type of encoding, resulting in that the capacity of *patterns* which it may store is exponential to the number of neurons in a layer (Rolls and Treves, 1998b). However, this does not imply that an exponential number of *pattern associations* may be stored, i.e. an exponential number of different stimulus-response patterns, as this has been found to increase only linearly with the number of neurons in empirical studies (Rolls and Treves, 1998b).

Secondly, integrating across memories by using an auto-associative network, which is thought to be constituted by the highly recurrent CA3-layer of the hippocampus, may in fact be part of what cognitively enables the brain to contextualize the self and the present.

Thirdly, very much related to the former point of being able to integrate across several sources due to the recurrent nature of CA3, a mechanism for implementing a form of episodic memory is essential in order to remember events, and to temporally associate different memories with one another. This is thought to be constituted by the CA3- and CA1-layers, where activity is both projected from the CA3 to CA1 through so-called Schaffer collaterals, and also recurrently to CA3. In other words, CA1 may then possibly temporally associate consecutive integrated memories that are relayed from CA3.

Fourthly, the hippocampus has been found to back-project its activity from CA1 to the EC (Rolls and Treves, 1998b), and to the neocortex. The output from CA1 to neocortical areas may be hypothesised to perform a type of both recall, and memory consolidation. Recall in that the patterns may simply invoke activity given its output, and memory consolidation in that this activity, after having been processed by the hippocampus, may be stored in the neocortex. Addressing the latter, the neocortex has been found to perform significant long-term potentiation in its circuitry (Rolls and Treves, 1998b). Such consolidation has also been hypothesized to occur primarily during sleep (French et al., 2001), in which a type of chaotic recall from a hippocampal network could produce pseudopatterns which may be learnt by the neocortex.

It may now begin to become clear to the reader how certain aspects of hippocampal function may be algorithmically described, and thus how certain aspects of cognition may emerge from implemented models. Furthermore, these aspects may then be analyzed both from a connectionist perspective with regard to cognition, and also with regards to the information processing capabilities that the emerging phenomena gives or may give rise to. A short outline has now been presented on how hippocampal functionality is connected to psychology and neuroscience. I will proceed to connect these aspects to computational modeling and algorithmics below.

One of the key components in the hippocampal neural processing as outlined by Hattori (2014) in his model, is using k-WTA in the layer-wise processing. He notes that k-WTA seems to enable a much more sophisticated type of pattern-separation than ordinary transfer functions. This matches the observations and findings as outlined by Rolls and Treves (1998c,b). Namely that k-winners-take-all will reduce overlap in that only the k most prominent units and neurons will dominate and inhibit the other neurons in a layer. Furthermore, they enable a more powerful separation of overlapping inputs. It should be emphasised that while k-WTA may resemble lateral inhibition, selecting an arbitrary value for the parameter k, quickly becomes biologically implausible. Therefore, computational models often rely on empirical studies of levels of activation within different brain regions in order to attain more realistic models. One example is the model which is outlined below, where Hattori (2014), building upon the work of Wakagi, Yuko; Hattori (2008), finds that selecting k such that the layer-wise firing rate in his hippocampal module corresponds to the biologically observed firing rates, interestingly significantly improves the model performance compared to other k-values. I would like to emphasise that selecting only k winners for a layer also leads to a quicker convergence and faster learning within the network, because weight updates are performed primarily by the k winners relative to the current pattern. In a sense, less ”noise” from the other neurons is present, and a more clear correlation and possibly pattern association is likely to be present for the network to extract. This may enable a network to reduce the redundance during feature or pattern correlation extraction, both functioning as an orthogonalizer quite similarly to principal components analysis (PCA), which extracts the most principal components of a data-set - only that k-WTA is more biologically realistic, and is able to perform a more separated type of extraction by activating different neurons. Lastly, it is worth mentioning that while k-WTA may reduce overlap and separate inputs, it will also tend to result in similar patterns activating the same set of neurons. Therefore, k-WTA may also be regarded as pre-processing which will categorize patterns into categories of similar types, (Rolls and Treves, 1998a). In other words, while k-WTA enables interesting information processing capabilities, the reader should bear in mind that this does not imply that the approach is not prone to some aspects of simplification due to orthogonalization - i.e. a type of diluted connectivity when using k-WTA due to reasons such as; too complex input data, sub-optimal convergence due to the initial random weight configuration, or simply basins of attraction being too close to one another, leading to divergence or spurious pattern extraction.

Another key aspect of computation previously outlined and present in the hippocampus, is auto-association and associative memory. Auto-association such as in a Hopfield

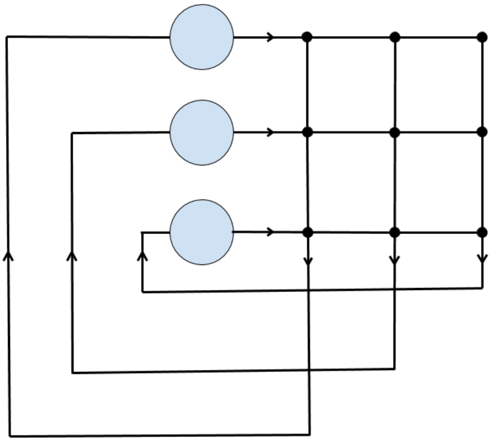

Figure 2.3: Illustrating a Hopfield net, which is **a simple auto-associative network**. Note that every neuron feeds into all other neurons of the network, which means that all information will be available to all neurons. Thus propagating it throughout the network until it reaches an equilibrium.

network (see figure 2.3), associates one pattern with another very much like in a traditional network, only that the next activations are fed back into the very same layer, resulting in a mechanism for pattern completion. I.e. for only partial information of a previously learnt pattern, the values from the pattern will be propagated through previously learnt weights, which in turn reflect the pattern association previously learnt. This will most likely result in that the consecutive patterns will more closely resemble the learnt target pattern in the pattern association. This abstract outline of the process hopefully provides the reader with a gist about how auto-association enables recall through association of similar events and memories, as well as storage of these memories, and also why only a certain amount of memories may be stored in a short-term memory operating on these principles. This has been demonstrated in cases where an auto-associative network is taught two patterns that are too correlated. In that case, a simple Hopfield network will fail to perform pattern-completion, as it will be stuck between the two basins of attractions that are formed by the two different patterns (Rolls and Treves, 1998d). This may be thought of as completing parts of for instance a pattern representing a letter, say 'a' - once part of the letter has been presented to the network; if another letter which is too similar, such as 'b' has been taught to the network, chances are that the Hopfield network then will push the network state towards both the 'a' and 'b' states at the same time. This may then lead to a new attractor state in which none of the letters is correctly recalled. Instead, it is likely that a combination of the two will be the result, or a cyclic pattern which cycles through hybrids of the two letters. There is a topographical influence from the closeness in the input or feature space. In the general case, the possible state-space of an auto-associative network may be visualised as a 3-dimensional space, where the current input represents a point which

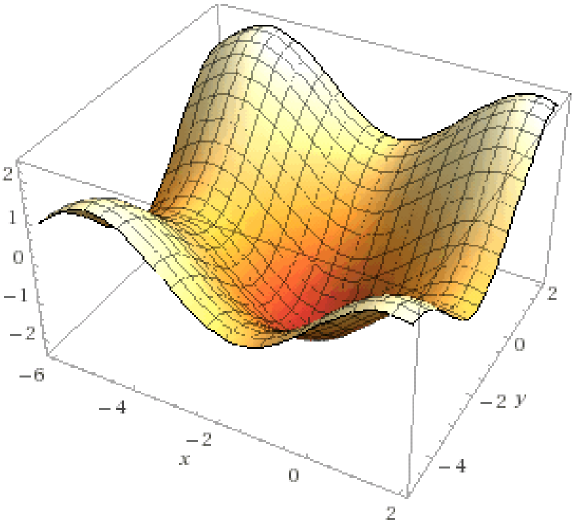

Figure 2.4: Illustrating a single basin of attraction, the **axes representing the weight-space** which is traversed during learning and recall. Note that the further down into the basin one gets, the more likely the solution is to stay within the boundaries of converging towards the given solution (the red area of the basin).

moves in the space relative to gravity which is exerted on it in the space. Curvature in the space is then formed by training the network, which will result in areas curving downward towards an attractor point or state. Analogously, when such a point is created, its surroundings is necessarily also affected a bit by this spatial curvature, and will therefore alter its curvature, although slightly less, downward, giving a direction during traversal. From this analogy it may be seen that if two such points are created in each others' neighbourhood, there will be a given distance between them where the curvature exerted by the creation of both may form a new such basin. Additionally, if too many such points are created, the landscape may become impossible to traverse, or even pointing to one or very few basins - which may very well also be hybrids of previously learnt patterns, i.e. spuriously created patterns. This brings us toward the importance of other qualities that are present in hippocampal models which largely removes these problems of redundancy and convergence in the CA3-layer.

While the classic XOR-problem is a matter of linear separability, solvable by the introduction of non-linear transfer functions in FFBP ANNs, issues are not quite as trivial in hippocampal models. When it comes to auto-associative memory, the forming of spurious basins of attraction (see figure 2.4) in auto-associative networks may be addressed by qualities which are introduced by the hippocampus, and more specifically largely by the dentate granule cells of the dentate gyrus (DG), which project to cornu ammonis region 3 (CA3) (which is believed to constitute an auto-associative network in the HPC), both receiving projections from the entorhinal cortex (EC).

The dentate granule cells of the DG have been empirically observed in the rat hippocampus to be nearly approximately a sevenfold relative to the number of pyramidal cells which project to it from the entorhinal cortex (Rolls and Treves, 1998b). Furthermore, while the firing rate of the EC has been observed to be about 10 %, the firing rate of the dentage granule cells have been observed to be approximately 1 % (Rolls and Treves, 1998b). Largely due to these facts, it is hypothesized that the role of these cells is to reduce overlap between and to separate overlapping inputs using inhibitory interneurons, i.e. k-WTA algorithmically speaking. Furthermore, the low contact between the dentate granule cells and the CA3-cells (i.e. low firing rate), referred to as expansion encoding (Rolls and Treves, 1998b), also necessarily leads to a type of sparsification, in addition to further orthogonalization resulting from k-WTA in the expanded layer. Furthermore, it is believed that the dentate granule cells are strongly connected to the CA3-cells, which receive projections from EC, DG-cells, and recurrently from itself. This would then enable the DG-cells to strongly influence synaptic modification of the CA3-cells during learning, further enforcing new pattern correlations to be considered and possibly learnt (Rolls and Treves, 1998b).

2.4 Existing models

In their seminal paper, McClelland et al. (1995) propose a dual-network memory architecture in which the hippocampus is responsible for the consolidation of memories to the neocortex, with the neocortex storing semantic and episodic memory. The synthesis of recall from the deeper layers of the neocortex and representations in the working memory itself enables contexts to be distinguished or connected in the proposed model. The learning and consolidation to the neocortical module is essentially performed in an interleaved fashion; slowly potentiating and instantiating the memories from the hippocampal to the neocortical network. An approach closely resembling a bottom-up and top-down synthesis, where recall is combined with novel patterns. This raises the question of how such an interconnectedness is constituted both topologically speaking, as well as in terms of local information-processing. Furthermore, the question is whether principles from the environment of the neocortex and hippocampus need to be extracted and implemented to successfully have this functionality emerge in computational models, i.e. whether a synthesis of brain functionality constituted by additional parts would be crucial in regard to memory consolidation in the artificial model. For instance for the successful integration across memories. The proposed model of (McClelland et al., 1995) suggests that the hippocampus and neocortex constitute the mechanisms for successful integration across memories, as well as keeping memories fairly intact. The specifics on how these mechanisms are constituted, however, remain obscure or undiscovered. This constitutes a core inspiration and foundation for the artificial neural network (ANN) model in this thesis. More specifically it lays out the foundation for investigating long-term memory and memory consolidation, by which I hope to attain more insight into mechanisms that enable generalizability and plasticity in ANN models.

The model of McClelland et al. (1995) largely ameliorates the problem of catastrophic forgetting in ANNs, outperforming other algorithms of the time by far. However, work on this model is fairly limited, and it is with the aim of further extending the architecture that I review implementations of it.

French (1997) proposes a series of experiments that address the sensitivity-stability dilemma (Hebb, 1949), demonstrating that a pseudo-recurrent network model performs significantly better than traditional feed-forwad back-propagate networks, mainly inspired by McClelland et al. (1995). Different experiments are used to illuminate several aspects of the pseudo-recurrent network model. The key finding is that pseudo-recurrent networks using pseudopatterns perform significantly better in terms of less catastrophic forgetting, suggesting that the brain may perform a type of pseudopattern compression and storage of information. Another point worth noting is that the networks that are simulated are of a fairly small scale, making them generalisable only to a certain extent due to network capacity. This seems to have been largely ignored by the authors. Therefore it would be interesting to look at the implications of both increasing the complexity as well as the scale of the network, such as in the models of (Hattori, 2010, 2014). Note that the semi-distributedness of this paper's model arises naturally from the pseudo-recurrent neural network, as opposed to in the former papers of French (1992, 1994). This may suggest that the mechanism, which acts as an auto-associative memory, may also act as a predictor. In addition to completing incomplete, partial or fuzzy memories and retrieving them, it might therefore also provide a mechanism to filling in a story, or even imagining a story, creating it on the go by using the pseudo-recurrent mechanisms. This could suggest that the interleaving of memories in a pseudo-recurrent manner is at the heart of creativity, prediction and not the least; cognition. In relation to the thesis, I wish to investigate various successful or partly successful approaches for addressing convergence in the dual-network memory architecture. A key finding is that of pseudo-recurrence performing a crucial mechanism in interleaving and pattern generation. This may form a foundation for a comparative analysis in my future work, as well as providing insights into some underlying principles for the emergence of such mechanisms.

French et al. (2001) address the issues of the dual-network memory models in (French, 1997; Ans and Rousset, 1997), illuminating key issues related to episodic memory, contextualisation, and pseudopattern generation and optimisation. In doing this, they conclude that the brain is likely to perform some kind of pseudopattern optimisation. Possibly in a stochastic way relative to how well it evaluates its performance and understanding of a currently perceived concept or state. When it comes to episodic memory, the work of Ans and Rousset (2000) is elaborated on by French et al. (2001), in which only dissimilar pseudo-inputs were used for consolidation to a neocortical network. Results demonstrate that the model is capable of generalising to and thus learning all patterns (20 patterns, where only 13 are explicitly taught to the neocortical network). This strengthens the view that a dual-network memory model is crucial for successful integration across memories. Another aspect that is addressed in (French et al., 2001) is contextualisation (in the dual-network memory architecture). Ans and Rousset (2000) demonstrate that their implementation of a dual-network memory model performs better using pseudopatterns with random initial input, rather than when retrieving more similar patterns from the neocortical module. This does pose an inconsistency both biologically and algorithmically speaking, because it is biologically implausible to retrieve an output representing the activity of the entire neocortical network, and algorithmically inefficient or possibly intractable with increasing network size, in order to interleave new memories with old. Furthermore, retrieving similar patterns mostly leads to a failure of convergence. This suggests that there may in fact be other, or more, principles that play a crucial role in the dual-network memory architecture. Summarising; the dual-network memory model offers significant benefits and advances in neural network models, but there seems to be an oversimplification related to pseudopattern formation and generation, affecting the integration across similar, yet separate memories. This is an aspect which I wish to further investigate in this thesis. It is also important to note that French et al. (2001) find that the pace at which memories are consolidated to a traditional ANN is of little relevance to the amount of information it may store - it is the mechanism with which the information is learnt that is of central importance in this regard. This makes the learning using pseudopatterns the central mechanism for increasing the storage capacity of the traditional ANNs.

Ans and Rousset (1997) address catastrophic interference in traditional backpropagation networks, demonstrating that it may be significantly reduced by using a pseudo-rehearsal mechanism (Robins, 1995, 1996). Such a mechanism is shown to be implementable by two reverberating neural networks, in which the first learns a certain pattern correlation, i.e. stimulus-response pair, after which the learnt pattern function is taught, or consolidated, to the second network by producing pseudopatterns - random input and corresponding output pairs; extracting the learnt pattern-mapping. Furthermore, in order to maintain previously learnt patterns in this traditional architecture, the way in which novel patterns are interleaved with old, is by consolidating the network configuration of the second network by pseudopatterns consisting of random input along with corresponding output, thought to represent the network configuration, to the first network. This results in that the first network considers both new information along with old. Ans and Rousset (1997) demonstrates that this does in fact significantly reduce catastrophic forgetting, and suggests that the brain may implement memory consolidation and interleaving in similar ways, although the mechanisms proposed by reverberation seems biologically unrealistic. Furthermore, they demonstrate that the same mechanism may be implemented by a slightly more realistic single-network model, in which the two networks are coupled to one single hidden layer, which then projects back to both input layers, thus enabling interleaving of new information with old, as well as re-injection of an extracted pattern. Note that this paper demonstrates a mechanism for transferring knowledge or information from one neural network to another, which one may argue is fairly realistic, given that biological neural networks tend to relay information to others. The idea of consolidating information is more closely studied in this thesis, and is also built upon by the architecture which is presented below.

Hattori (2010) proposes a model which fundamentally differs from the former implementations of French (1997); French et al. (2001), and Ans and Rousset (1997) in that the hippocampal module is constituted by a chaotic neural network. Keep in mind that the phrases of hippocampal and neocortical networks should only be considered as borrowed terms for symbolising the networks of the model. The model networks are only very loosely coupled to biological functioning, with the approach outlined being only inspired by it. Hattori (2014) presents a novel ANN model based on his former work on the dual-memory architecture, where the short-term memory network is now a more complex and biologically plausible network; namely a simplified hippocampus model. Hattori (2014) demonstrates in several experiments that catastrophic forgetting is reduced to a large extent in the new model, and furthermore that the model performance, in terms of recall rate and reducing catastrophic forgetting, is significantly improved relative to the dual-network memory models of Ans and Rousset (1997); French (1997); Hattori (2010). Hattori (2014) further demonstrates that the hippocampal network is capable of acquiring information rapidly, consolidating this to the neocortical network when it is successfully extracted in the hippocampal module. However, the model is still not used to solve complex tasks, as the model is rather heavy in terms of computational complexity due to introducing more complex neuronal dynamics. The hippocampal network consists of McCulloch & Pitts neurons, using the Oja rule for learning and updating connections between the different sub-modules, and less contrained Hebbian learning with forgetting in the recurrent CA3-connections. Please see below for further details on the model. As for his experiments and results, he finds that mean goodness and perfect recall to be worse in the hetero-associative case when compared to the auto-associative for the model. This may suggest that a significant amount of plasticity is missing in the model. This is further supported by the observation that a much higher turnover-rate than observed biologically speaking has to be employed for tuning the model. Looking to complex systems theory, some noise or chaos is required to arrive at a phase-transition between regular and chaotic dynamics, i.e. the critical phase, in which learning is made possible and efficient, Langton (1990); Newman (2003). By using a very high neuronal turnover in the model, this suggests that the high turnover rate could be what alleviates the lack of plasticity in the proposed model. Despite improving training performance and pattern extraction, using too high a turnover-rate may also introduce too much randomness, rendering the representations too coarse-grained. Therefore it would be interesting to see parameter adjustments for the hetero-associative case in the hippocampal module, as this may pose different constraints on the network. Particularly an analysis of the edge of chaos for the CA3-part is something which I wish to further investigate. Another important aspect would be attempting to temporally extend the model, in an attempt to capture how episodic memory may be constituted in a complementary memory model. Such a synthesis could potentially introduce novel aspects of high-level cognition.

2.4.1 State-of-the-art

Ans and Rousset (1997) demonstrated the important and interesting aspect of being able to increase a traditional FFBP ANN's storage capacity through embedding a learning algorithm using pseudopatterns in a dual-network architecture. More recently, Hattori (2010, 2014) demonstrates that a dual-network memory model which is more biologically plausible may in fact extract patterns in a more effective and biologically resemblant way, including one-shot learning, in an intermediary storage network. Thus having several qualities of short-term memory. Furthermore, his dual-network memory models show promising results in that they both outperform previous models of the architecture such as the models of (French, 1997; Ans and Rousset, 1997), in classic sequential learning experiments for retroactive interference (McCloskey and Cohen, 1989), where stimulus-response pairs of different training bases are learnt. Hattori's (2014) model is reviewed and outlined below, and the methods and implementation used in this thesis is contained within chapter 3.

The neocortical module

Hattori (2010) trains the neocortical module on the novel chaotically extracted patterns along with pseudopatterns. A set of pseudopatterns represents the current weight configuration of a network. In his paper, Hattori (2010) employs two types of pseudopatterns, which he refers to as type I and II, in order to interleave the former weight configuration with the novel patterns it learns. Type I is constructed in a very simple way: A random input is presented to the neocortical network, and the output is retrieved. This is then stored as an input-output pattern, called a pseudopattern of type I.

Pseudopatterns of type II are constructed in a slightly different manner, the approach being as follows:

1. Retrieve an extracted pattern from the hippocampal network.
2. For each element of the pattern, reverse it with a probability p_r.
3. Present the pattern to the neocortical network, and store the retrieved input-output pair (pseudopattern II) in a set.
4. Repeat step 2. and 3. until a certain number of pseudopatterns of type II are obtained.

Performing steps 1.-4. above results in a set of pseudopatterns of type II.

After pseudopatterns have been obtained, the neocortical network is simply trained on them along with the novel patterns extracted from the hippocampal network by chaotic recall, by using FFBP, i.e. standard gradient descent in weight space (as outlined in appendix A). Note that due to the nature of the pseudopatterns, old memories are actually interleaved with old. This may be seen by considering that pseudopattern I is in fact the output obtained by presenting a random input to the network, the output reflecting a compressed representation of the network weights at the time. When the network is trained on the pseudopattern along with a hippocampal pseudopattern, the algorithm of BP and gradient descent attempts to minimise the error between the old configuration of weights and the new hippocampal pseudopattern. Thus interleaving the old representation of memories with the new memory. In fact, Hattori (2014) uses the exact same type of mechanisms for memory consolidation to the neocortical network. Similarly, for a set of pseudopatterns II, as elements are reversed in the chaotically extracted pattern, the resulting pseudopatterns II reflect the old network configuration for distorted versions of the novel input patterns (for the current training set in the auto-associative scheme). This is a clever way of separating the old configurations from the new, as by creating slightly similar inputs, with outputs reflecting the former configuration the network will explicitly focus on these dissimilarities, and create separating hyperplanes for them. I.e. when training on the pseudopatterns II along with the new extracted patterns, it is ensured that the neocortical network has to minimise the loss between novel inputs and former *similar*, yet separate inputs. Note that the only risk here is the case when the permutation of extracted pattern outputs by chaotic recall from the hippocampal network are permuted in a similar manner, which may possibly bias the neocortical network during training. This issue may be alleviated, or avoided altogether by training on several such pseudopatterns, which is part of Hattori's (2014) algorithm. Lastly, note that this is likely to be another factor resulting in the reduced goodness of fit for the hetero-associative patterns, other than the fact that the recall rate of the hippocampal module is lower. Namely that the hippocampal module *output* is what

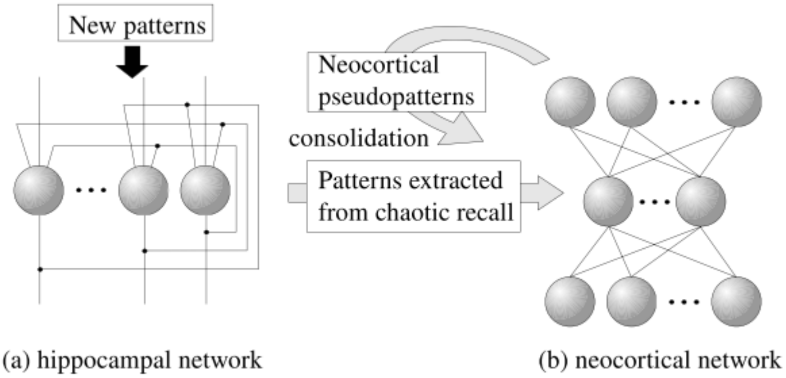

Figure 2.5: Illustrating **Hattori's (2010) dual-network memory model**. (a) represents the hippocampal module, whereas (b) represents the neocortical module. Note that the hippocampal module implements a Hopfield network, from which seemingly chaotic behaviour emerges when combined with the neuronal dynamics. Adapted from Hattori, M. (2014). 'A biologically inspired dual-network memory model for reduction of catastrophic forgetting', *Neurocomputing*, **134**: 262-268. Adapted with permission.

is permuted and given to the neocortical network as *input*. In the hetero-associative case, for it to have the same similarity to the extracted patterns as in the auto-associative case, it is the hippocampal pattern inputs which would have to be slightly permuted and given to the neocortical network in order to create pseudopatterns type II. This could be justified, and achieved, by means of bilateral neural network firing, i.e. firing from the output to the input, or simply relaying a distorted signal of the model input, biologically speaking.

Memory recall in the neocortical network may be performed by presenting input patterns to the neocortical network, and obtaining the resulting output from the network. Note that as outlined above, the neocortical network only learns using the pseudopatterns that are extracted by the hippocampal network. Therefore, some success criteria, such as perfect recall in the neocortical network, depends strongly on the perfect extraction rate of the hippocampal network (i.e. when all details of the pattern have been successfully learnt).

The hippocampal module

Hattori (2010)

As may be seen in figure 2.5, Hattori (2010) proposes a model in which the hippocampal (HPC) module is a single layer Hopfield network. However, the HPC module is not trained using gradient descent, but rather by Hebbian learning, which may be summarised as; fire together, wire together. Using Hebbian learning leads to faster convergence when compared to stochastic gradient descent (Hattori, 2010). Adopting Hattori's (2010) notation, the model may be formally outlined as follows, beginning with the equation for Hebbian learning;

$$\omega_{i,j}(t+1) = \gamma\omega_{i,j}(t) + x_i^{(k)}x_j^{(k)}, \tag{2.1}$$

where $\omega_{i,j}(t+1)$ is the weight between neurons i and j for time step $t+1$, γ is a constant forgetting factor, $\gamma \in (0,1)$, and $\mathbf{x}^{(k)} = (x_1^{(k)}, x_2^{(k)}, ..., x_N^{(k)})$ is the k-th pattern that we want the network to learn. Note that $\mathbf{x}^{(k)} \in \{-1,1\}^N$, which constrains $x_i^{(k)}x_j^{(k)} \in [-1,1] \implies \omega_{i,j} \in [-1-\gamma, 1+\gamma]$. N is the number of nodes in the input patterns.

Further, Hattori (2010) outlines the neuronal dynamics as follows,

$$x_j(t+1) = f\{\eta_j(t+1) + \zeta_j(t+1)\} \tag{2.2}$$

$$\eta_j(t+1) = k_m\eta_j(t) + \sum_{i=1}^{N}\omega_{i,j}x_i(t) \tag{2.3}$$

$$\zeta_j(t+1) = k_r\zeta_j(t) - \alpha x_j(t) + a_j \tag{2.4}$$

Adapted to the thesis notation, u_j is neuron j's activation value, where the value for the next time step is determined by two functions, namely $\eta(t+1)$ and $\zeta(t+1)$. Equation 2.3 takes into account its former input values through $\eta_j(t)$ for the current time step, in addition to summing over the inputs of its incoming synaptic connections. Equation 2.4 includes a relationship to the neurons' previous activation values. Note that an external input parameter a_j is also included in $\zeta_j(t+1)$, and that both equations 2.3 and 2.4 have damping factors of refractoriness k_m and k_r, respectively, discontinuing the impact of former function-values exponentially relative to the temporal distance. $f(u)$ is the sigmoid function as defined in equation 5.1, note however that a steepness parameter ϵ is also included, θ being divided by ϵ such that,

$$f(\theta) = \frac{1}{1+e^{\frac{-\theta}{\epsilon}}}$$

Introducing a Hopfield net as the hippocampal module lets the first network attain qualities of an auto-associative network, namely those similar to a short-term memory, or working memory. More specifically, it results in a content-addressable memory, along with the ability to perform perfect recall, unless the network's memory becomes digested as in the above general example (see the figures representing basins of attraction for an illustration of how this phenomenon may occur). To some extent, a Hopfield net can be said to have a graceful degradation; i.e. its patterns are gradually distorted when the memory is diluted. It should be noted that this degradation is somewhat rapid in the case of a single-layer Hopfield network. In either case, this yields an interesting synthesis between the two networks, as the algorithm for memory consolidation does not have to reverberate the network configuration between two traditional neural networks that are trained using back-propagation. Instead, the associative network may be designed such that its parametrization lets it forget previous 'memories' at a pace such that it may learn up to a certain number of patterns, and thus consolidating the patterns as it learns. This is where algorithmic design of memory consolidation, i.e. learning in the second network, becomes of central importance. Note that the first network needs to be able to extract the patterns or memories that we wish to consolidate - but assuming that it does, it is the mechanism with which these are stored in the long-term memory network that is of central importance. This does not contrast previous models when it comes to the importance of interleaving old memories with new - it does however pose a significantly more biologically realistic model, as well as to open new opportunities for the short-term memory and pseudopattern generation. Pseudopattern generation is discussed further in the next chapter on the methods and implementation employed (chapter 3).

Hattori (2014)

Hattori (2014) proposes a more biologically realistic dual-network memory model, based upon, and outperforming his former model. In the novel model, Hattori (2014) proposes a significant architectural change in the hippocampal network, now modeling a more complex hippocampal network model, while the neocortical module remains the same. The hippocampal module consists of five layers, the three middle layers being inspired by different parts of the hippocampus; namely the entorhinal cortex, dentate gyrus, CA3, and two enclosing input and output layers. See figure 2.6 for the topological structure of the novel model of Hattori (2014). Considering the neuroscientific background that is outlined above, the reasons for the significant performance increase can now be addressed by considering the different layers' algorithmic properties. Shortly put, the hippocampal module can be described as a competitive network, using k-WTA in all layers. Furthermore, sparsification results from partially connected layers, as well as from expansion encoding, i.e. a significant increase in the layer size relative to the preceding projecting layer, in the DG-layer. Note that the ratios between the DG-layer and the EC- and CA3-layer is the same as the ratios in the rat hippocampus, with the number of modeled neurons being about one thousandth. The relatively large DG-layer results in an expansion encoding, potentially recoding the neural activation values in the succeeding auto-associative layer of CA3, resulting in separated, yet initially similar patterns. It is worth mentioning that the DG model also implements neuronal turnover, which may further reduce overlap between inputs by re-instantiating some of the neurons' synaptic connections, contributing to the learning of separate new, yet similar patterns. Further model details are outlined below and in chapter 3.

Note that in figure 2.6, the CA3-layer is fully connected both recurrently as well as to the output layer. As the EC-layer is connected somewhat sparsely to the DG-layer, and the DG-layer is very sparsely connected to the CA3-layer, this may constitute a form of compression mechanism as seen in auto-encoders. It is also worth noting that this poses a time-delay from when a certain input has been directly presented to the CA3-layer from the EC-layer, until the possibly compressed input arrives from the DG-layer. This might further constitute mechanisms similar to those of operating at multiple timescales, as well as mechanisms for abstraction. It is worth mentioning that a slightly different transfer function is used by Hattori (2014). Namely,

$$f(\theta) = tanh(\tfrac{\theta}{\epsilon}),$$

where ϵ still is a steepness parameter.

Hebbian learning is still used as in equation 2.1 for the CA3-layer and the CA3 to output-layer, relative to its former output;

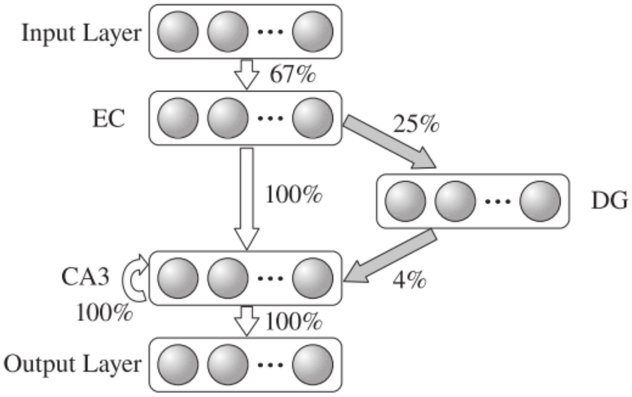

Figure 2.6: This figure illustrates **Hattori's (2014) proposed dual-network memory model**. Note that the EC is connected to both CA3 and DG, which in turn is also connected to CA3. The gray arrows are connections which are used solely during training. Hattori, M. (2014). 'A biologically inspired dual-network memory model for reduction of catastrophic forgetting', *Neurocomputing*, **134**: 262-268. Adapted with permission.

$$\omega_{i,j}(t+1) = \gamma\omega_{i,j}(t) + u_i u_j$$

However, between the EC and DG, EC and CA3, and DG and CA3 parts, Oja's rule is used (Hertz et al. (1991), cited in Hattori (2014)). Oja's learning rule is a modified type of Hebbian learning, restricting the weight space (to prevent divergence as a result of the chaotic behaviour). It may be formally outlined as follows,

$$\omega_{i,j} = \omega_{i,j}(t) + \lambda u_j(u_i - u_j\omega_{i,j}(t)), \tag{2.5}$$

where λ is the learning rate for the Oja neurons. Note that the input and output layer neurons are bipolar (±1), whereas the other neurons are binary. Every region is trained by a k-winners-take-all (k-WTA) approach, in which a fixed number of the k most active neurons' activation values are propagated throughout the neurons' synapses. Neuronal activity is determined by firing frequency. Interestingly, (Hattori, 2014) notes that the non-linear separation of k-WTA seems to be far more powerful than that of non-linear transfer functions. Furthermore, he notes that non-linear transfer functions may actually reduce the performance of k-WTA.

When it comes to the DG-layer of the model, it consists of 1600 artificial neurons, whereas the EC consists of 240, and the CA3 of 480. The input and output layers can be set to an arbitrary number of neurons, and are in most experiments set to 49 neurons each. Because of the ratio between the DG-layer and the EC-layer, the DG may perform expansion encoding. This may enable more intricate pattern correlations to be extracted by having a multiple number of neurons available for computation when compared to the layer which projects its output to the layer. Furthermore, it is believed that this constitutes a mechanism for decorrelating input patterns (Rolls and Treves, 1998e), enabling the storage of similar, yet separate memories. Along with k-WTA, which biologically resembles lateral inhibition, it has been shown that including these mechanisms in a hippocampus-model increases the number of patterns that the model may store (Wakagi, Yuko; Hattori, 2008; Hattori, 2014). It is worth noting that while Hattori (2014) employs the DG-layer in order to enable pattern decorrelation in the hippocampal module, this is only the case during learning, and the layer is not used during recall. The justification is that the DG is thought to function primarily as a pattern separator during learning. Biologically speaking, the DG is hypothesized to be able to strongly influence the synaptic modification in the CA3 during learning (Rolls and Treves, 1998b). One of the final keys to attaining a successful dual-network memory model is the introduction of neuronal turnover in the dentate gyrus. Neuronal turnover is the birth and extinction of a percentage $\beta\%$ of the neurons, here in the DG. Note, however, that while this is believed to occur to a very low degree biologically in the DG, a very high rate of $\beta = 50\%$ is employed by Hattori (2014), with turnover after every training set. Possible reasons for this and associated implications, related to plasticity and convergence, is discussed earlier in this section, in that a high level of randomness, or noise, needs to be introduced in the artificial model, as it is less plastic due to mathematical approximations and model simplification relative to the biological system by which it is inspired. Hattori (2014) demonstrates that input patterns become less similar when introducing neuronal turnover, which in turn drastically increases the number of patterns that may be stored in the HPC module. This demonstrates that neuronal turnover is a crucial mechanism for separation of similar patterns in the model, and possibly key to learning to distinguish between similar, yet separate memories and patterns.

Memory recall may be performed in the hippocampal network by chaotic recall after learning, i.e. presenting random input to the hippocampal model, waiting until it reaches some convergence criterion such as having a stable output for a number of time-steps, and considering the current input-output pattern as a 'chaotically' recalled memory, representing a learnt memory of the short-term memory model. Such a mechanism for pseudopattern extraction has been shown to create pseudopatterns that enable the neocortical network to extract the underlying function, i.e. mapping, of the pattern-association. It has further been hypothesized that a similar mechanism may be implemented by the brain (French et al., 2001; Ans and Rousset, 1997). Note that French et al. (2001) found consolidation speed to be of no importance when it comes to retroactive network interference. Thus making learning through pseudopatterns the mechanism which itself enables the interleaving of several memories, as well as storage of them. This in turn makes the quality of the generated pseudopatterns a key key component in memory consolidation. When it comes to the biologically implausible convergence criterion for when pseudopatterns should be generated, and memories consolidated, it may be hypothesized that when encountering a certain stability, the brain may release certain neurotransmitters that promote learning, such as glutamate, GABA, or dopamine, enabling memory consolidation. While remaining only a hypothesis, note that the algorithmic criterion for pseudopattern generation and memory consolidation is an aspect where the model diverges significantly from being the biological mechanisms from which the model is inspired. Interestingly, when recalling previous memories in the HPC-model without using the DG-layer of the model, memories may yet be successfully distinguished. It seems that using the expansion encoding in the DG-layer together with k-WTA enables the model to extract the different properties in the similar pattern-associations. An observation which demonstrates that this mechanism is not sufficient, however, is the fact that hetero-associative patterns are not as easily extracted. The question remains whether they may still be as easily separated, only that the more complex function-mapping is more computationally heavy to extract.

To summarise, the outlined model of Hattori (2014) introduces a novel hippocampal neural network model, inspired by hippocampal functioning. This results in a short-term memory network capable of acting as a working memory, having desired emergent qualities resembling those of the biological ones when it comes to; separation of overlapping, yet separate pattern-associations, integration across different memories (represented by different pattern-associations), quick convergence and the capability of one-shot learning, and the ability of extraction and perfect recall of low numbers of pattern-associations. Note that the model is significantly simplified topologically speaking when it comes to the input and output connections. As outlined previously related to computational models and empirical studies on the hippocampus, the activity of CA3 is believed to be projected to CA1, which again relays the activity to cortical regions, as well as back to the entorhinal cortex (Rolls and Treves, 1998b). This means that there is another layer which might be responsible for among others, two crucial aspects of hippocampal functioning. The first being episodic memory, in that the current memory is relayed back to the EC, and thus projected through the hippocampus and back to the CA3, possibly enabling relating consecutive memories and/or events to one another. The second being recall, and possibly memory consolidation, through back-projection to the neocortex after another synaptic information processing stage has been performed by projection to the CA1. Thus leaving out CA1 in the hippocampal module results in that the model is incapable of episodic memory, i.e. relating consecutive memories, events or episodes to one another when temporally close. This is one of the limitations to the model that is studied. However, it does not imply that the underlying mechanisms for integrating across different stimuli and single pattern-associations, are not generalizable. Furthermore, even though back-projecting from a CA1-layer is not implemented, output is consolidated from an output layer, which is synaptically modifiable, and may thus be regarded as an approximation to the neocortical back-projections through which learning may occur. In other words, memory consolidation through pseudopattern generation and hippocampal module learning *is* a very crude approximation to biological brain function and memory consolidation. Because of the interesting lines that may be drawn between neuroscience and psychology and the emergent model behaviour, I aspire to further study memory consolidation in the model of Hattori (2014), drawing parallels from the context which is established in this chapter.

Biologically, the experiment in (Hattori, 2014, 2010) is an adapted version of a classical learning experiment (Barnes and Underwood (1959); cited in McCloskey and Cohen (1989)) created to test interference in connectionist models. McCloskey and Cohen (1989) address sequential learning in ANNs, and explore retroactive interference and possible links to this type of interference in connectionist networks by visiting the classical experiment of (Barnes and Underwood, 1959), where lists, or sets of pattern-associations are learnt by subjects. Specifically, these experiments are meant to highlight retroactive interference occurring during sequential learning of different sets of pattern-associations. McCloskey and Cohen (1989) conclude that interference may be catastrophic in connectionist models when it is not in human subjects. However, as more recently addressed and outlined in chapter 2.2; more biologically realistic models have been created within the paradigm, implementing memory in a dual-network fashion. So far, the comparative analysis has not been done with these models and the findings as outlined in (McCloskey and Cohen, 1989), related to performance by human subjects, as models would have to implement more sophisticated autobiographical memory within these dual-network memory systems. I.e. the context is relevant to determining which pattern-association the current pattern should invoke (as is the case in the model of Hattori (2014)). This could be simplified by regarding the context as part of the input-pattern - thus making the results attained in single-pattern-associations generalisable to a certain extent within this domain. Considering context as encoded in single pattern-associations remains a biologically implausible simplification, however, and the mechanisms for more complex types of memory remain unexplored in the domain of dual-network memory systems. I wish to perform a comparative analysis of the model of Hattori (2014) and this thesis' implementation, drawing parallels to how the parametrization affects the quality of the attained pattern-associations, and the overall model performance. Furthermore, I wish to draw parallels between these relationships and the biological context addressed in among others (McCloskey and Cohen, 1989; Barnes and Underwood, 1959). This may give rise to hypotheses or suggestions for further experiments and discussions, as well as potential research spaces and future work.

Chapter 3

Methods and implementation

3.1 Programming environment

Theano is a Python library for building high-performance mathematical tools (Bergstra et al., 2010). It lets you write library-specific code which will be analyzed, optimized, and compiled to C or CUDA, enabling execution of efficient bytecode. Furthermore, Theano lets you symbolically define an algorithm in a high-level programming environment. For those familiar with Mathematica, symbolic definition in Theano is fairly similar. In this thesis Python and Theano is used for model implementation, the experiments being outlined in chapter 4.

Theano is tightly integrated with NumPy, another Python package for scientific computation. Moreover, Theano supports parsing of several NumPy objects into objects which Theano will be able to later use efficiently after its optimization process. Please consult LISA lab's webpage (LISA-lab, 2015b) for a complete documentation of the framework. Theano operates on symbolic constructs, termed tensors; general mathematical constructs. In order to define a function, one may write the actual mathematical expression, for instance:

```
import theano.tensor as T
import theano

A = T.fmatrix('A')
y = A ** 2
f = theano.function([A], y)
```

Calling the library function theano.function then analyses the symbolic expression, and constructs executable C or CUDA. See appendix B for a slightly more advanced example using the scan-operator in theano.

When defining symbolic expressions such as functions using Theano, Theano constructs a graph of the provided symbolic expressions, see figure 3.1. This allows for

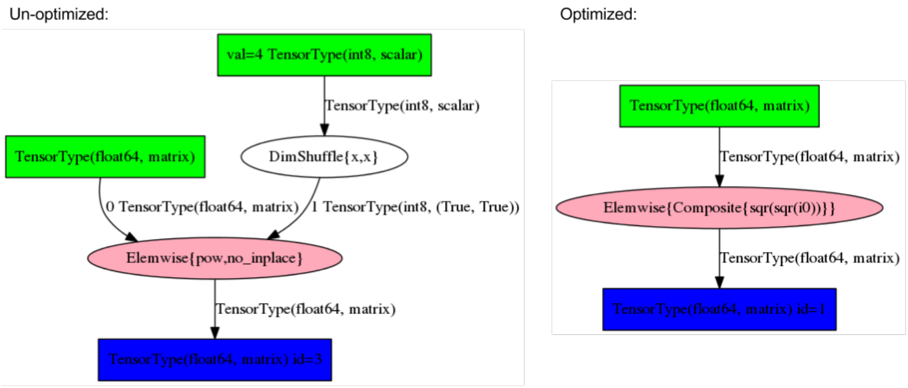
Un-optimized:
val=4 TensorType(int8, scalar)
TensorType(int8, scalar)
TensorType(float64, matrix)
DimShuffle{x,x}
0 TensorType(float64, matrix)
1 TensorType(int8, (True, True))
Elemwise{pow,no_inplace}
TensorType(float64, matrix)
TensorType(float64, matrix) id=3
Optimized:
TensorType(float64, matrix)
TensorType(float64, matrix)
Elemwise{Composite{sqr(sqr(i0))}}
TensorType(float64, matrix)
TensorType(float64, matrix) id=1

Figure 3.1: Graph manipulation of tensors in Theano during the optimization and compilation process of an expression such as $y = A^2$. Note that the left-most graph is the graph which has been constructed directly from symbolic differentiation, whereas the right-most hast been optimized before compilation.

differentiation and manipulation through a syntactic and semantic analysis of the resulting expression graph, optimizing the graph for expression evaluation and graph traversal before code generation. The compiler may then generate optimized code, compiling it according to the provided environmental parameters. This leads to highly efficient C or CUDA executable code. Note, however, that as the library provides a high-level abstraction for generation of efficient bytecode, this may render debugging somewhat harder. More specifically, the executable code will be further processed from that in pure Python. This means that the programmer may need to predict further how the code will be compiled, making it highly recommendable or necessary to have some previous knowledge within C and/or how the Python-library's compilation process works. A remedy for this is however that Theano supports profiling, providing the programmer with information about what data structures the variables are compiled to, as well information about as their usage. Furthermore, if compiling to a GPU, Theano may output warnings whenever an operation is performed on the CPU.

In order to use Theano, certain dependencies need to be setup. These are fairly straightforward when it comes to compiling to C-code for CPU-execution. In a preliminary study for this thesis, Theano was setup for use with a GPU on a system with an NVIDIA card, using Ubuntu 14.04 LTS, consulting a guide found on (LISA-lab, 2015a). In doing so I found that k-WTA requires the dual-network memory model to be synchronized, thus requiring transferring control to the CPU and interpreter, including memory transfer from the GPU to the CPU. Therefore, running the model on the GPU did not provide the basis for a substantial run-time performance gain for the hippocampal model. Therefore I decided to target the CPU, particularly when it comes to the hippocampal module, which may only be executed on the CPU in its current form. Running only the neocortical network on a data-set would, however, be far more efficient using the GPU. In fact, the compiler may target the GPU if the framework for it is setup, and the Theano flags allow a hybrid approach (i.e. no device parameters are set).

3.2 The dual-network memory model

3.2.1 Hippocampal model details

For the hippocampal network, shared theano variables are instantiated, storing the values of vectors for the activation values, and matrices for the weights between the different layers. Topologically, it remains the same as illustrated previously in figure 2.6. In the hippocampal network, the first step of propagating values throughout the model is to simply multiply the activation values of the first layer with the associated weight matrix, representing its connections, storing the result in the subsequent layer; the EC-layer. After this operation, k-WTA is performed. For all layers except for the CA3-layer, these operations are simply performed according to the following equations,

$$\mathbf{x}_j = tanh(\frac{\mathbf{x}_i \mathbf{W}_{i,j}}{\epsilon}), \tag{3.1}$$

where ϵ is a steepness parameter. Succeeding this straightforward propagation between the layers, using a common thresholding function, is setting the activation values in a binary fashion according to the k-WTA algorithm;

$$f(x_i, x_{threshold}) = \begin{cases} 1, & x_i >= x_{threshold} \\ 0, & otherwise \end{cases} \tag{3.2}$$

where $x_{threshold}$ is calculated simply as $x_{threshold} = \frac{x_k + x_{k-1}}{2}$, where x_k is the k-th largest activation value in the layer after action potential potentiation according to equation 3.1. For pseudocode on the k-WTA implementation, please see appendix C.

When it comes to the CA3-layer, information is summed from several layers, after which the equations for chaotic neurons are used to attain the next η-values, which are finally sent through the thresholding function $tanh$ before k-WTA is performed according to the firing rate of the layer. The equations, as outlined in chapter 2.4 in equations (2.2, 2.3, 2.4), are here given in vector form;

$$\mathbf{x}(t+1) = f\{\vec{\eta}(t+1) + \vec{\zeta}(t+1)\} \tag{3.3}$$

$$\vec{\eta}(t+1) = k_m \vec{\eta}(t) + \sum_i \mathbf{W}_{i.j} \mathbf{x}_i \tag{3.4}$$

$$\vec{\zeta}(t+1) = k_r \vec{\zeta}(t) - \alpha \mathbf{x}_j(t) + \mathbf{a} \tag{3.5}$$

where k_m and k_r are a damping factors of refractoriness, $\mathbf{x}_j$ is the input values (i.e. former activation values of the CA3-layer), and $\sum_i \mathbf{W}_{i.j} \mathbf{x}_i$ is the sum of all input values from the preceding layers for $i \in \{ec, dg, ca3\}$, and,

$$\mathbf{x}_i = \mathbf{x}_{ec} \mathbf{W}_{ec,ca3} + \mathbf{x}_{dg} \mathbf{W}_{dg,ca3} + \mathbf{x}_{ec} \mathbf{W}_{ca3,ca3}$$

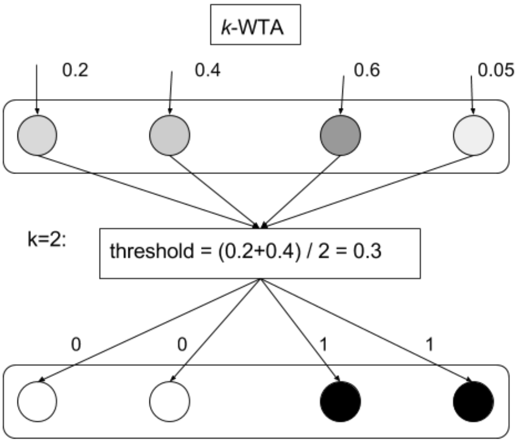

Figure 3.2: Illustrating k-WTA for an arbitrary network layer of size $n = 4$. Note that the figure depicts information flowing to the same layer after binary thresholding has been performed. Furthermore, the circles, illustrating the neurons, are shadowed relative to their excitation, i.e. activation values, before and after k-WTA. This results only in completely excited, or completely depressed (i.e. binary) activation values after k-WTA has been performed (as 0 is the lowest value for internal neurons).

Having shared variables store the current values for the eta- and zeta-vectors provides a sufficient basis for iterating through time-steps for the chaotic neurons, given that the surrounding activation value vectors and weight matrices also are instantiated. By using Theano, the above definitions of the equations translates more or less directly to symbolic definitions, thus implementing the model. Please see appendix C for a code excerpt of the hippocampal module, including other code examples, as well as an architectural code overview, and a link to a public git-repository containing the full code.

3.2.2 Neocortical model details

When it comes to the neocortical module, this is implemented essentially as a traditional back-propagation network; having an input and output-layer, and one hidden layer, along with two weight matrices for the connections between the layers. The L2-norm was used as error-function, and diracs delta function was used to chain the error-signal during back-propagation for an efficient execution, using the gradients starting at the output-layer. While the reader may consult Appendix A for the mathematical details and derivation of the equations associated with the traditional feed-forward back-propagation artificial neural network, the essential equations are also provided here, being the following,

$$\Delta\omega_{i,j} = -\alpha \frac{\partial \mathbf{E}}{\partial \omega_{i,j}}, \tag{3.6}$$

which minimizes the error loss-function $\mathbf{E}$ w.r.t. the partial derivative of the weight $\omega_{i,j}$

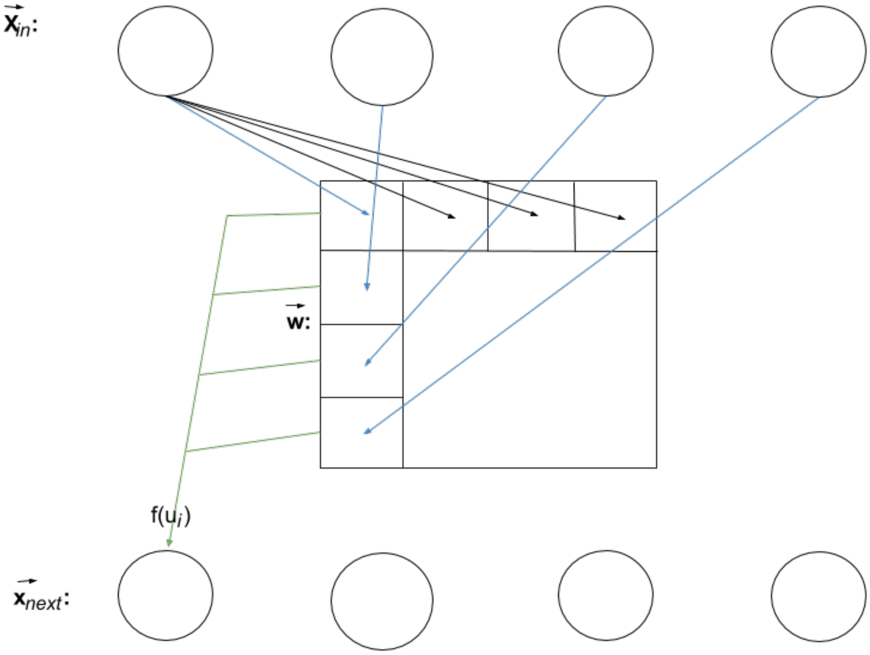

Figure 3.3: Illustrating **two network layers and their associated weight matrix**, representing the synapses and the synaptic connection strengths between the two layers. Note that the blue lines symbolise the input from the neurons that produce the next activation values for the first neuron of the second layer. These values are multiplied by the corresponding weights **w** (squares) and summed (green lines), before finally yielding the first neuron's next activation value by calculating the result of the input sum through the transfer function, f.

from neuron i to neuron j. In its full form, the term for the derivative of the error loss-function may be written using the chain rule as,

$$\frac{\partial \mathbf{E}}{\partial u_j}\frac{\partial u_j}{\partial \theta_j} = (\sum_{l \in L} \frac{\mathbf{E}}{\partial u_l}\frac{\partial u_l}{\partial \theta_l}) f(\theta_j)(1 - f(\theta_j), \tag{3.7}$$

where u_j is the activation value of node j, and θ_j is neuron j's total input. Starting at the output layer, one may simplify the equation to,

$$\tfrac{\partial \mathbf{E}}{\partial u_l}\tfrac{\partial u_l}{\partial \theta_l}\omega_{j,l} = u_l(u_l - d_l)\omega_{j,l},$$

which may then be used in the preceding layers after updating the subsequent values, as these will then embed the former terms of the chain into the updates. This allows for a simple implementation and a very efficient algorithm/model. Acquisition of a weight-configuration may also be performed on the GPU. However, this is regarded as unnecessary in the current paradigm, as the data set and number of iterations remain fairly small.

3.2.3 Chaotic patterns and memory consolidation

Ans and Rousset (1997) demonstrate that pseudorehearsal may be used to successfully reduce or eliminate catastrophic forgetting in FFBP ANNs. The mechanism which makes this possible is pseudorehearsal, as previously demonstrated by Robins (1995). Furthermore, Ans and Rousset (1997) demonstrate that pseudorehearsal may be performed solely in ANNs. I.e. the entire previous network configuration may be transferred to another network, which may then continuously send patterns (pseudopatterns reflecting the old configuration) to the first network, while the first network also learns novel patterns, thus interleaving the new patterns with old. As FFBP ANNs minimize an error-loss function, typically a distance measure from the attained output to a target output over all patterns in a training set, this results in a weight configuration which minimizes the error for both the old and the new patterns, thus attempting to maintain the new and the old information equally well, the weighting being determined by the number of patterns for each weight configuration when training using gradient-descent.

While it is interesting to note that pseudorehearsal may be used to successfully interleave previously learned patterns with new, it is not my aim to demonstrate this in this thesis. Storing patterns in a data structure that maintains the previous network configuration, and generating training patterns from this configuration in order to interleave the previous configuration with the new patterns, may also be regarded as storing the information simply in another neural network, such as demonstrated by French (1997). It remains, however, outside the scope of this thesis to demonstrate that catastrophic forgetting may be reduced to a large extent by pseudorehearsal, as it is fairly well-documented in the literature. What I am addressing, is the potential information transfer capabilities inherent in the patterns produced by chaotic recall itself, evaluating the potential emergent qualities of the hippocampal model. Therefore, four different schemes for information transfer and memory consolidation from the hippocampal model to the neocortical network are implemented. Namely:

1. Solely using the patterns extracted by chaotic recall as training patterns for the neocortical network.
2. By employing in addition to the chaotic patterns hippocampal pseudopatterns type I, created in the same manner as neocortical pseudopatterns type I, i.e. by a random input and the corresponding output. This does in other words correspond to adding further chaotically recalled patterns.
3. By in addition to the chaotic patterns employing hippocampal pseudopatterns type II, which consist of permuted chaotically recalled pattern outputs, and the corresponding hippocampal output after having recurrently fed them through the hippocampal network.
4. By using chaotically recalled patterns, and both hippocampal pseudopatterns I and II.

3.2.4 Model decisions

During the implementation of the model, I encountered some model aspects that were not clearly stated in the papers on which I base the model; (Hattori, 2014, 2010). To proceed with the implementation, certain decisions had to be made.

Asynchronicity may introduce more randomness in searches in algorithms such as in dynamic networks, cellular automata, and Boltzmann machines (Bar-yam, 1997). Therefore, having the CA3 neurons update their values asynchronously may enable the layer as an auto-associative network to traverse more of its search space - i.e. it is not as limited in what values that the neurons will take after a certain update/propagation from the preceding layers. This may possibly result in being able to recall more patterns within the hippocampal model, but may on the other hand introduce more noise during learning. Thus the question remains whether the benefit during the search and recall outweighs the possible downsides during learning. Because this extra 'jiggle' is in fact present during learning as well, chances are that having an asynchronous updating scheme is more effective. The justification of having a synchronized layer updating scheme in the model is that it is a far more efficient implementation - enabling for instance the exploitation of hardware parallelism for matrix operations. Whether asynchronous updating is more effective and biologically plausible is tested and discussed further within the next chapter (4).

As for calculating the next activation values of the chaotic neurons in the CA3-layer the paper of Hattori (2014) did not specify whether the vector and matrix products are simply summed, or whether coefficients for some of the weight matrices and synapses are included. Therefore I consulted other references such as (Wakagi, Yuko; Hattori, 2008), granting insights into, and underlining important topological decisions such as that the inclusion of the DG-layer during learning, which may enable pattern separation. Furthermore, the paper demonstrates and outlines that from the DG-synapses may be magnified (in their model by a factor of 25), such that the connections may impact the CA3-neuronal activation values more heavily.

Norman and O'Reilly (2003) implement a complementary learning system containing a hippocampal model, which uses expansion encoding in their DG-layer, arguing that the

DG physiologically speaking seems to have the ability to heavily influence the firing patterns of CA3. Based upon these papers, a variable weighting of the DG-CA3 pathway is implemented.

Regarding the implementation of eta- and zeta-functions in the CA3-layer, the thresholding is performed after the sum of the products of activation values and weight matrices is calculated. In other words, the hyperbolic tangent transfer function is applied to the sum of the eta- and zeta-functions:

$$\mathbf{x}_{ca3}(t+1) = tanh(\tfrac{\eta(t+1)+\zeta(t+1)}{\epsilon})$$

Lastly, it may be argued that a type of recall may include several iterations of auto-associative recall. Such details, however, are omitted, as average model trends after several iterations may be observed, and the activity of the EC- and DG-layers remain unchanged as long as the input does, given that no neuronal turnover is performed.

3.3 System layout

The dual-network memory model is implemented using Python and the library Theano, as previously discussed. In order to instantiate the model, test it, run experiments, and store results; a complete framework within Python is implemented. The core components of the system are the classes wrapping the artificial neural network models, namely; HPC and NeocorticalNetwork. These contain all methods associated with the hippocampal and neocortical networks that are required to perform the algorithmic operations of the dual-network system as outlined in (Hattori, 2014). Furthermore, these core classes make use of certain static functions such as displaying a visualization of the current network output during run-time, or writing to a log. These were defined in a Tools-package. In order to verify the functionality of the core modules, a small test suite using unit-testing is implemented, which may be used to automate debugging. Furthermore, there is a hippocampal module wrapper implementing an abstraction of learning a set of patterns using a hippocampal model object, performing chaotic recall for a given model parametrization and object instance, and the generation of hippocampal pseudopatterns. This wrapper is used in the experimental suite, which implements the experiments that are outlined in the following chapter, (chapter 4).

For each experiment that is run, the results are stored in a folder containing the saved data. Furthermore, each distinct experiment is stored in its own distinct folder, where the attained extracted patterns and optionally generated pseudopatterns are stored, both in the form of PNGs, and as binary data which may be imported and used for memory consolidation under different neocortical module schemes. All experiments write directly to a log that is located within the saved data folder. This log contains information such as the number of training iterations, the number of extracted patterns (perfectly recalled and spurious), and the goodness of fit for the neocortical module experiments. In association with the log and the employed logging formats for the hippocampal and neocorical module experiments, the system implements a parser. This parser is specifically designed to read the corresponding log-files and log-file data, and is connected to a plotting library, which creates figures and plots of the different experiment results. Examples include the average convergence and recall rates by the neuronal turnover rate.

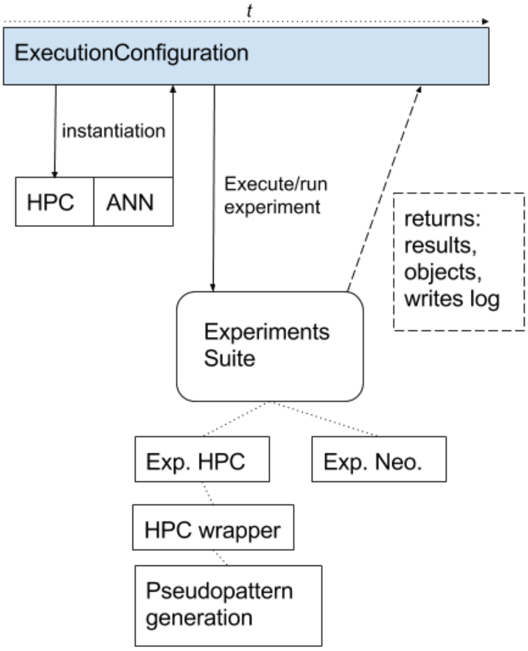

Figure 3.4: Illustrating the **execution and layout of the system** containing the dual-network memory model and the experimental environment. First; objects are instantiated containing the two networks of the model, then experiments are run from the experimental suite according to the provided run-configuration. This configuration specifies which experiment(s) to run, and with what experimental and model parameters. Note that the experiment makes calls to the HPC-wrapper, and implements calls to the FFBP ANN, i.e. the neocortical network object directly.

In order to generate images of the different layers' activities, simple formulas are used, which generate a rectangular view of a vector of activation values, along with the Python Image Library (PIL). Furthermore, in order to save previous results to disk, simple Python file handling is used to write logs, and cPickle is used to store the binary data of objects, enabling later retrieval and analyses of specific hippocampal models.

For the code of the entire implementation, please see appendix C, which also contains a link to a public git-repository which the reader may clone and run in order to demonstrate the model and experiments him-/herself. Note that a setup-script and short guide is only given for the operating system Ubuntu LTS 14.04.

Chapter 4

Experiments and results

This chapter structures experiments into sub-sections, which also contain a discussion of the specific experiment results. While remaining central in this chapter, the research questions are addressed, along with a more general abstract discussion, in the next chapter, 5. Beginning with a fairly thorough analysis of the hippocampal module, including a subsection on hetero-associative training patterns, this chapter is continued by an analysis of the neocortical module.

In the hippocampal model experiments, several variables are chosen as free variables, for which traditional experiments testing the variables (schemes and configurations) are run. Novel aspects are proposed and implemented according to the attained results, and even further experiments are also implemented. Roughly speaking, the free variables which are tested in the hippocampal model experiments are initially asynchronous and synchronous updating of the CA3-layer within the hippocampal model, followed by introducing a synaptic connection weighting between the dentate gyrus (DG) and the CA3-layers which is held as a free variable in the experiment. Furthermore, two different neuronal turnover modes, and the neuronal turnover rating itself are instantiated as free variables, resulting in data on different variable values along the aforementioned axes. Lastly, the convergence criterion during learning and recall is reformulated into a slightly less stringent, and more consistent criterion, being training or recalling for a static number of iterations, which results in a basis for further analysis.

Following the analysis on the hippocampal module, is an analysis on memory consolidation to the neocortical network for several of the hippocampal model schemes. I.e. learning by the chaotically recalled and extracted patterns, and by using hippocampal pseudopatterns. In this section I begin with demonstrating some of the network's properties, including catastrophic interference when training sequentially on the training sets using subsets, such as 2x5, or 5x5. Thereafter, experiments on memory consolidation based upon the previously attained chaotic patterns and pseudopatterns, are presented. These form the basis for further experiments and results, in which I propose and implement a novel memory consolidation scheme. Shortly put; using the patterns extracted by chaotic recall to span over the neocortical network, in order to reduce catastrophic interference.

Table 4.1: Lists the number of neurons in each layer within the hippocampal module.

	Input	EC	DG	CA3	Output
Neurons	49	240	1600	480	49

Table 4.2: Displaying the firing rates for the different layers, and the resulting values k for k-Winners-Takes-All.

	EC	DG	CA3
Firing rate	0.10	0.01	0.04
Resulting k	24	16	19

These experiments also form the basis for a discussion of the entire dual-network memory model and architecture.

4.1 Setup and data set

When it comes to the initial model setup, the initial parameters are set in accordance with the findings of the literature study, and mainly based on the parametrization used in (Hattori, 2010, 2014; Wakagi, Yuko; Hattori, 2008). See tables 4.1, 4.2, and 4.3.

Furthermore, the DG-weighting is set to either 1 or 25, and the neuronal turnover rate to either 0.50 or 0.04. The initial setups are chosen in order to test the configurations of both models upon which the implemented model is based. Furthermore, some values are tested due to their closeness to the values observed within the biological brain (Rolls and Treves, 1998b).

As previously mentioned, the input and output layers consist of 49 neurons each. When it comes to the data that is used, classical experiments of sequentially learning pattern associations, meant to highlight retroactive interference, are adapted. These correspond roughly to the experiments of (McCloskey and Cohen, 1989), as adapted by (French, 1997; Hattori, 2010, 2014).

In this thesis, two sets of 25 distinct patterns are used, namely the first 25 letters of the alphabet, in uppercase as I/O for the auto-associative training set, and in uppeercase as input and lowercase as output for the hetero-associative training set. Each letter is compressed to a 7x7 image of binary pixel values (black or white), corresponding to bipolar (± 1) input or output values. I.e. bipolar vectors of length 49, where -1 corresponds to

Table 4.3: Displaying the parameters for μ and σ^2, with which the weights are normally distributed. Note that the parameters found in the table correspond to the tuple (μ, σ^2). 'n/a' denotes 'not applicable'.

	DG	CA3	Out
EC	(0.5, 0.25)	(0.5, 0.25)	n/a
DG	n/a	(0.9, 0.01)	n/a
CA3	n/a	(0.5, 0.25)	(0.0, 0.5)

Table 4.4: Initial model variable constant settings. Neuronal turnover was calibrated through initial experiments along with the CA3 neuronal updating scheme, as well as the DG-weighting.

Parameter:	Gamma	Epsilon	Nu	k_m	k_r	a_i	alpha
Value:	0.70	100.0	0.10	0.10	0.95	0.80	2.00

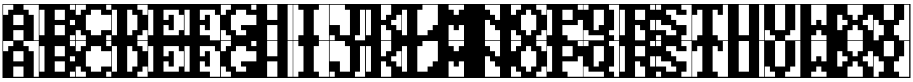

Figure 4.1: Illustrating **the auto-associative training set** as 7x7 images, the first row being the input and the second the corresponding output of the patterns.

black, and 1 to white. Note that the input and output space is reduced by constraining the vectors to bipolar values. However, catastrophic interference is observed in the neocortical module when training on the subsets of the original training patterns, and the memory is digested in the hippocampal module if training on the entire training set at once. For for most hippocampal model schemes, memory is digested even in when training on the larger subsets. Thus, the particular dimensionality and data set size is regarded as of the sufficient complexity needed to study the desired model properties. That is, to study and attempt to reduce or eliminate catastrophic interference, as well as to improve the memory span of the models. Unless stated otherwise, 20 trials are used per set size and model configuration in order to calculate the average values. In other words, 20 trials are used per distinct experimental scheme.

Lastly, I would like to clarify the following: While the reader may interpret 'perfect recall rate', and 'extraction rate' as two different measures, these are in fact considered to be the same. Adapting the same conventions and notation as in the literature study, in chapter 2, the perfect recall rate, or extraction rate, denotes the number of extracted output patterns relative to the total number of output patterns in the training set, where extracted means a one to one correlation of the output patterns. Thus, perfectly recalled becomes equivalent to extracted, as a correlation of ¡1.0 (for bipolar output values) is defined as a spuriously recalled pattern. As we will see, patterns tend to, in fact, either be fairly spurious, or perfectly recalled, due to the hippocampal model's pattern completion qualities. Furthermore, the spurious recall rate is always included either in the same graph, or in the next figure within the sections, such that the reader may easily compare the perfect extraction rate with the spurious recall rate, as well as the convergence rate where applicable. This will hopefully make it easy to infer the quality of the extraction, while yet exposing and making the original measures readily available. For instance, the perfect recall rate relative to the sum of all recalled patterns may be 1.0, although the extraction rate is only 50%, such as for some of the attained model schemes. Thus, by omitting this relative rate, the intention is that the results provide a clearer picture of model performance.

4.2 Hippocampal module experiments

Before I introduce the hippocampal model experiments, I would like to briefly introduce the model's functionality by demonstrating model activity in a single experiment at a more detailed level. The intention of this is to create a more complete and sound picture of the model elements, as well as to provide the reader with a more thorough understanding of the representations that are used as input and output data. This is followed by experiments designed to test specific aspects of the hippocampal model, presenting aggregate results such as graphs of mean convergence ratios along with corresponding analyses.

4.2.1 Low-level demonstration

Following is a demonstration of hippocampal model learning, using two distinct model schemes for two different examples, with figures of model output after each learning iteration for chaotic recall and normal recall, i.e. pattern association. The first two subsets of auto-associative patterns, that is $\{$A→A, B→B$\}$, and $\{$C→C, D→D$\}$, are used as training sets in this example. Furthermore, the hippocampal module is instantiated with the parametrization described above in tables 4.4 and 4.2, and with a neuronal turnover rate $\tau = 0.04$ and a DG-weighting of 1 in the first example, and $\tau = 0.50$ and DG-weighting$= 25$ in the second example. Neuronal turnover is performed between every learnt training set - that is, only once after learning the $\{$A→A, B→B$\}$ associations, before commencing learning of the next input-output pair. Here the convergence criterion is set to a static number of training iterations, equal to 15. In further experiments, results are generated and analysed for both a dynamic convergence criterion for learning and stability during recall, as well as for two configurations using a number of $i = 15$, and $i = 50$ training iterations as the learning and stable recall criterion. Furthermore, the weight matrices are instantiated according to their respective firing rates, with weights being randomly assigned according to Gaussian distributions using the parameters presented in table 4.3, for the number of neurons corresponding to the layers' firing rates, respectively (as outlined in table 4.2).

Instantiating the hippocampal model using the parametrization as outlined above, images are generated of the network output for pattern recall (i.e. association), and chaotic recall for every training iteration. These are generated for both examples, i.e. synchronous and asynchronous updating of the CA3-layer in the model.

When studying the two figures (4.2, 4.3) that display the recalled patterns during and after learning pattern-associations for patterns A→A, and B→B, note that pattern separation seems fairly successful, as the associated output for the aforementioned letters is fairly stable. However, some spurious patterns appear, even in the case of having the actual, non-distorted input of the pattern as the model's input. This may indicate weak model convergence, which may be due to incorrect model calibration, such as a too heavy DG-weighting. Because the connections from the DG-layer are both instantiated as normally distributed with rather high weight values ($\mu = 0.9, \sigma = 0.01$), and weighted 25 times stronger than the connections from the EC- and CA3-layer, this may possibly disrupt the strength of the attained basins of attraction, or prolong the training period needed in order for the model to converge for its (EC-CA3, CA3-CA3, and thus CA3-output) connections,

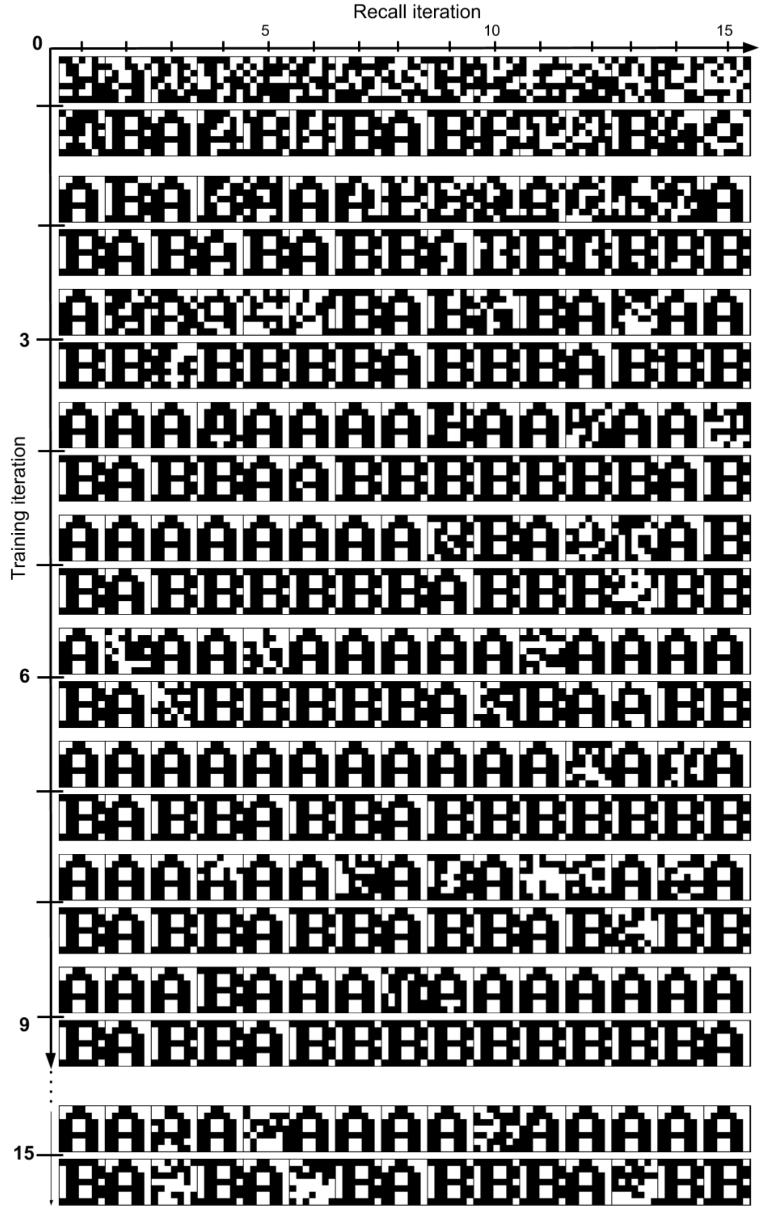

Figure 4.2: Displaying the **neural network recall for the training inputs after each training iteration**. Note that the origin is now in the upper left corner, time flowing downward along the y-axis for each training iteration, with the first axis denoting the model recall iteration. Note that for each training iteration, there are two rows of recalled output, displayed for two corresponding inputs, namely the two inputs of the current training subset. In this figure, these are the patterns 'A', and 'B', for the upper and lower rows, respectively. The model configuration is using **synchronous** CA3-updating, turnover only after the set is learnt, and a DG-weighting of 25.

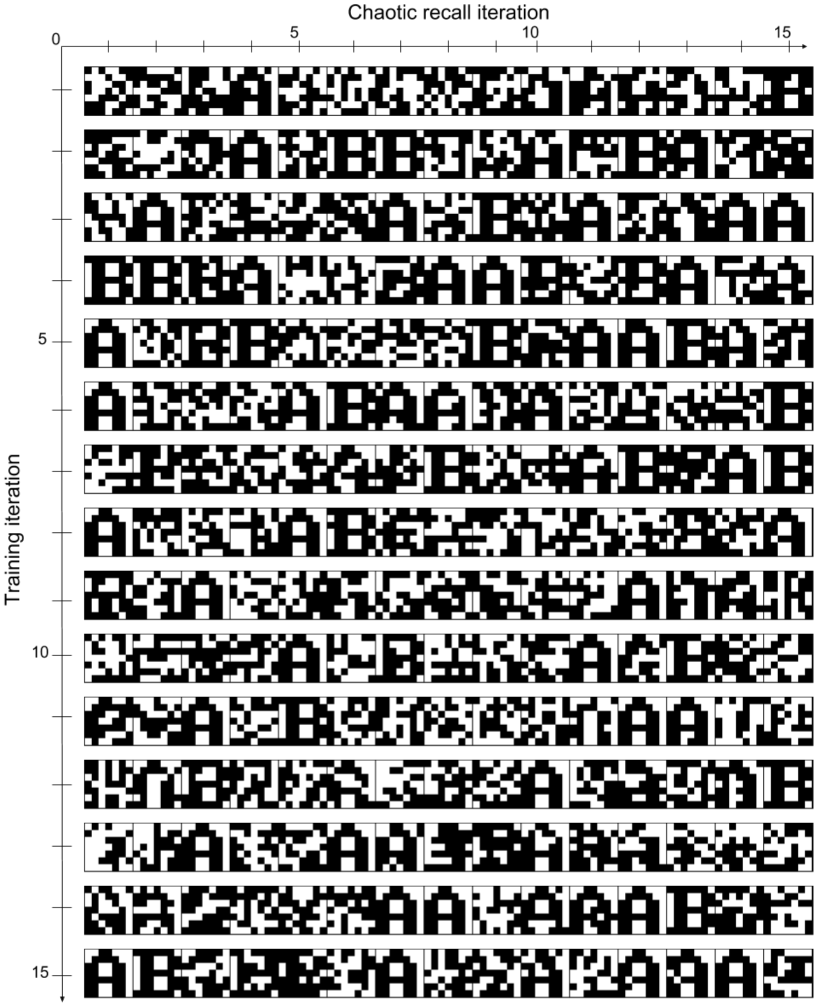

Figure 4.3: Displaying the **patterns recalled by chaotic recall** - that is, for each training iteration over the subset A, B, a random input is set as the model's input, for which the attained output after each recall iteration is displayed, for 15 iterations. The model configuration is the same as in figure 4.2, using **synchronous** CA3-layer updating.

and thus for the CA3-layer to extract non-overlapping pattern-associations. The latter resulting in unstable behaviour within the auto-associative CA3-layer.

Note that one run is not sufficient to determine whether this is the case. Therefore, these are parameters that will be tested in further experiments, where 20, or at least 10 trials per training set size, per configuration, are used in order to statistically determine patterns of the emergent model behaviour. I.e. 40-80 trials per experiments and model configuration are performed.

Lastly, it is worth noting that although model convergence and recall is fairly good in the case of having a non-distorted training pattern input as the network's input, a great number of spurious patterns are recalled in the chaotic recall scheme. Thus, the number of spurious patterns extracted by chaotic recall is one of the measures that will be used in further analyses. It will be used particularly with regards to successful pattern separation and the establishing of basins of attraction. I.e. the perfect recall rate may be erroneously observed as close to 100 % of the number of training patterns - however, when the number of spurious patterns extracted by chaotic recall exceeds this number (possibly by a great deal), the quality of the model performance may still be considered poor. Note, however, that it might be interesting to also consider memory consolidation in such a scheme to the neocortical module. It might be that a functional mapping is contained within the recalled spurious patterns, as random inputs and outputs may still reflect parts of the functional mapping. Note that in the case of attaining a spurious pattern output for an input from the *training set*, the functional mapping is most likely not present, and will only disrupt the convergence in the network which attempts to learn the pattern-associations. When it comes to the measures of model performance, another measure which will be used in terms of pattern consolidation is the goodness of fit measure, defined in section 4.3.1.

In order to demonstrate another hippocampal model configuration, low level results are presented for the case of asynchronous updating of the CA3-layer in figures 4.4 and 4.5. Neuronal turnover is still only performed between the two learning sets, which may be argued to be biologically unrealistic (some recoding of weights, i.e. topological synaptic modification) and turnover may be argued to be continuously present at a low rate. The latter is further addressed in later experiments. Nonetheless, this constructed low level example introduces randomness during training and recall by asynchronously updating the neurons the CA3-layer for each training and recall iteration. This may be considered to be somewhat more biologically realistic, although still implausible in terms of only one neuron being updated at a time algorithmically in the implemented model. In a more biologically realistic model, all neurons should be updated simultaneously, where hardware level parallelism could potentially determine a pseudo-random order of neuronal updates, making sub-sets of neurons fire and wire simultaneously. However, as aggregate activity on the population-level is what is studied in this thesis, I consider the attained results to yield valid results and associated suggestions at the neural network level.

Note that while the quality of chaotic recall appears to be better for asynchronous CA3-layer when comparing figures 4.3 and 4.5, the asynchronous scheme is slightly less accurate for the output for a learnt pattern association's input after 15 recall iterations. This is expected, as more randomness is introduced in the random sequence of updating the neurons of the CA3-layer, which will tend to have the model find the other basin of attraction more easily. Because these results were attained from only one run, the gener-

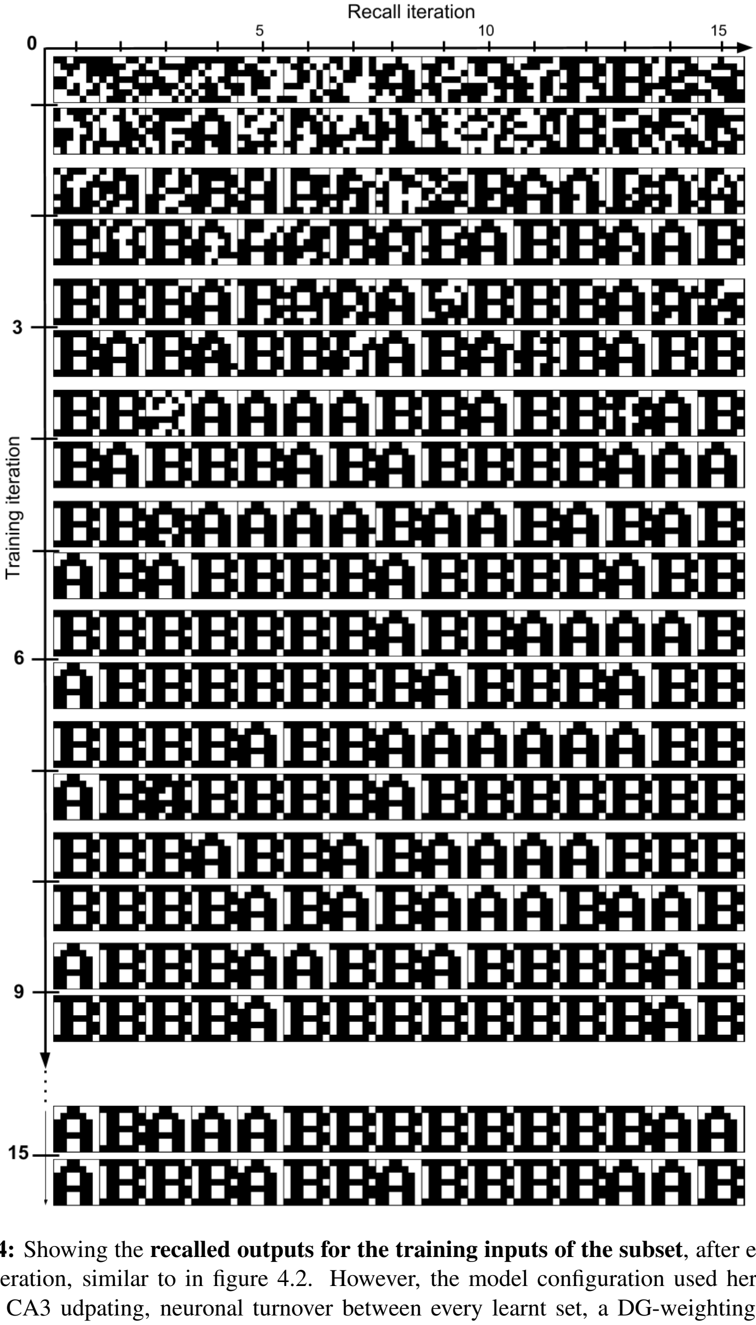

Figure 4.4: Showing the **recalled outputs for the training inputs of the subset**, after each subset training iteration, similar to in figure 4.2. However, the model configuration used here is **asynchronous** CA3 udpating, neuronal turnover between every learnt set, a DG-weighting of 1, and $\tau = 0.50$.

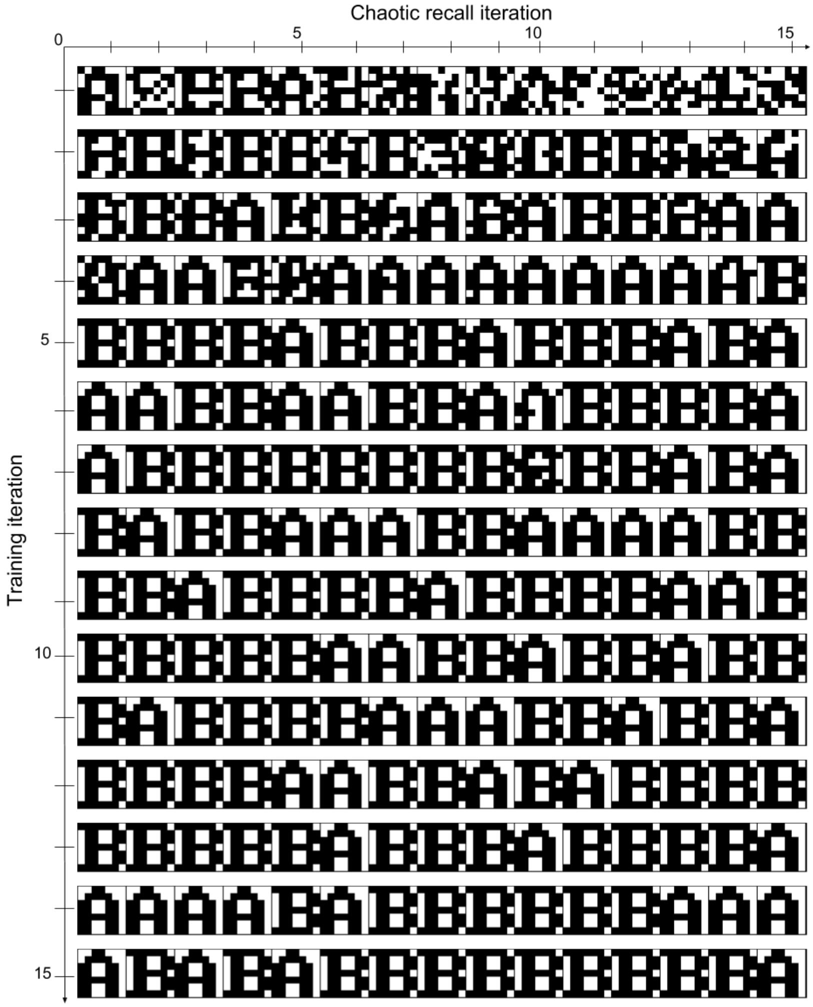

Figure 4.5: Displaying the **chaotically recalled patterns** for the same model configuration and results as displayed in figure 4.4 (**asynchronous** CA3-updating, a DG-weighting of 1, $\tau = 0.50$, turnover between learnt sets). Note that spurious patterns diminishes very quickly, and only learnt patterns are chaotically recalled after very few iterations. This configuration seems to be capable of one-shot learning.

alizability is very constrained. Nevertheless, the results do suggest a trend towards a more specific extraction capability at the cost of less accuracy. It is worth keeping this in mind when considering further experiments and results. Furthermore, it is worth noting that the synchronous updating scheme seems less stable than the asynchronous in these low-level demonstrations. This will be further discussed in experiments where average results, as well as convergence rates, are considered. If the reader wishes to see the figures presenting the attained output for the next training set, i.e. subset of the training patterns, please refer to appendix D, in which they are contained for the asynchronous model updating scheme.

4.2.2 Experiment: CA3-updating and neuronal turnover schemes

In order to evaluate the impact of synchronous action potential propagation and synaptic weight modification, four different model schemes are used. These consist of combining two types of CA3-layer neuronal activation and weight modification with two types of neuronal turnover. More specifically; updating the CA3 neuronal activation values synchronously or asynchronously, and performing neuronal turnover between every learned training subset, or for every training iteration. Synchronous CA3-layer updating effectively results in reducing the propagation of values from one layer of neurons to another to a set of vector and matrix operations, whereas the asynchronous scheme requires updating each neuron independently. In the simplest case of synchronous updating, i.e. for all non-chaotic layers, the synchronous propagation scheme is simply reduced to a vector of activation values multiplied by a weight matrix, which is adjusted through the activation function. For each of these schemes; synchronous or asynchronous updating, two turnover modes are tested. Firstly, turnover is performed only once before every new training subset, i.e. between learnt training sets. Secondly, turnover is performed between every training set iteration - that is, for every iteration over the current subset.

In this experiment, 20 trials are performed for every full auto-associative training set (that is 2x5, 3x5, 4x5, and 5x5), for every configuration. In other words, for 4 training set sizes $20 * 4 = 80$ trials/experiments are run for each model configuration. In these experiments the model attempts to learn to associate the n first capital letters auto-associatively, where n corresponds to e.g. 2x5 $= 10$, 3x5$= 15$, 4x5$= 20$, or 5x5$= 25$, the x denoting that the training set consists of 5 subsets that are used to train the model, sequentially.

Furthermore, the convergence criterion is defined as the following: For each training pattern in the current training set, if the model output is the correct pattern output for the undistorted pattern-input for three recall iterations, the pattern is considered to be successfully learnt. If this is the case for every pattern in the current training subset, convergence is considered to be attained, and chaotic recall is performed. Furthermore, when generating figures, the model is considered to have successfully learned the current training set if converging in less than 50 training iterations. Chaotic recall is performed similarly to the procedure in (Hattori, 2010, 2014). That is, during chaotic recall, when the output remains unchanged for three recall iterations, the pattern is considered to be learnt, or successfully extracted in the case of chaotic recall. Chaotic recall is always performed for 300 time-steps, with the input to the network being a new non-changing random input for every extracted pattern.

Note that spurious is defined as any distinct non-perfectly recalled pattern in figure 4.6. Non-perfect is used as a term rather than imperfect to signify that the pattern may be either

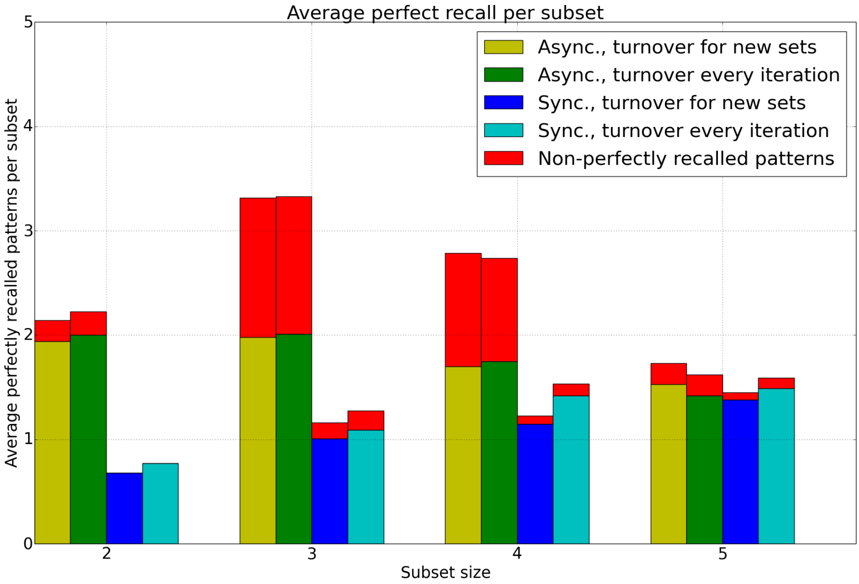

Figure 4.6: Displaying the average number of perfectly recalled patterns for each model configuration, along with the average number of spuriously recalled patterns for each configuration. Note that the dentate gyrus weighting is set to 25, and the turnover rate to 0.50 in all of the configurations, which might impact particularly the turnover mode in which turnover is performed between every training iteration.

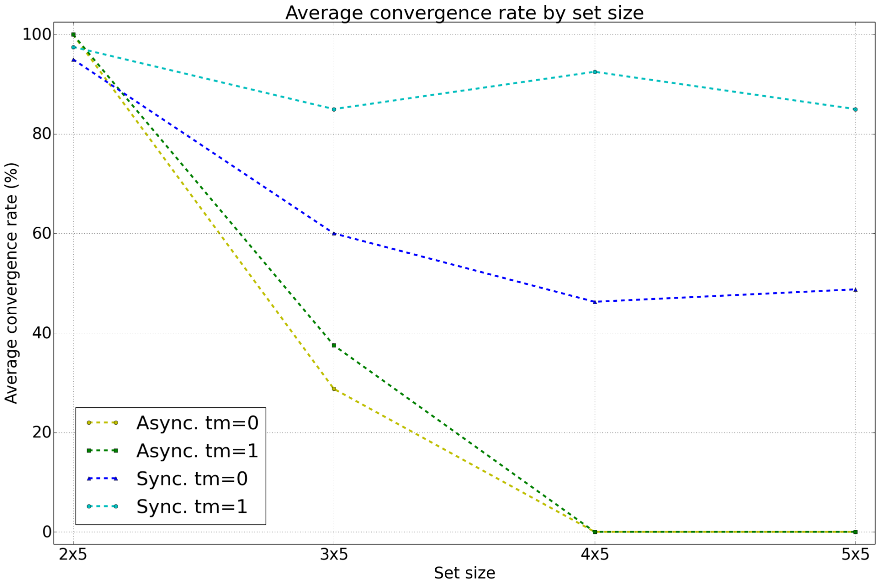

Figure 4.7: Presenting the average convergence rate in the four experiment-schemes, using the strict convergence criterion. Learning and convergence is considered successful if attained within 50 training iterations. Note that the synchronous updating schemes seem to converge significantly better than the asynchronous. Furthermore, when employing neuronal turnover for every training iteration, the convergence rate does not seem to fall as the set size increases.

nearly perfect, or in fact completely random, i.e. spurious. Note that in contrast to the introductory examples where the convergence criterion was defined as training or recalling for a fixed number of iterations i, ($i = 15$ in the introductory low-level examples), having a stricter convergence criterion naturally resulted in no spurious patterns being recalled in the synchronous updating schemes, which may be expected when considering figure 4.3. However, it remains unclear whether model convergence is attained successfully in terms of pattern separation. In order to possibly elucidate this, the convergence rate is also considered, and presented in figure 4.7 for asynchronous CA3 updating.

Although extraction of all patterns using chaotic recall is unsuccessful in nearly all of the current experiment schemes, one of the synchronous schemes, using turnover for every training iteration, converges well for all set sizes. However, it remains the scheme which performs the worst in terms of the perfect recall rate. Furthermore, this scheme recalls nearly no other than 1-2 of the patterns from each training subset, which it recalls perfectly. This may suggest that while the scheme is likely to successfully be able to separate patterns during training, it might not be able to separate them well enough to be able to chaotically recall them. Or even more likely; it might be that the chaotic recall criterion, being the same as the convergence criterion during training, is too stringent. I.e. a stability criterion of having the same output for three training iterations may be sensible when the correct pattern input is given to the network. However, when a random input is provided, it is expected that the network will oscillate more, and thus be less stable in its basins of attraction. Therefore, a relaxed convergence criterion might be more sensible during recall.

Because every neuron of the CA3-layer is updated synchronously, it may also be that the model lacks a certain 'jiggle' during recall, making it prone to being stuck in a subset of its basins of attraction. This view is further strengthened by considering that the DG-layer is not used during recall, as it performs expansion encoding, which along with its sparsity may recode and separate similar, but distinct, inputs into separate patterns for the CA3-layer to then be able to auto-associatively learn. Because convergence is attained during training, but not recall, this suggests that the DG-layer may in fact give rise to these emergent qualities, but that the CA3-layer possibly favours a subset of the learnt patterns too strongly, resulting in pattern completion for only this subset of patterns.

Note that due to the quick learning capability of the hippocampal module, it is unlikely that the EC to CA3 connections, as well as the CA3-CA3 connections have not been able to converge towards the solutions and extracted pattern correlations. However, if the recoding does not settle into a steady pattern, changing the input to the CA3-layer significantly for every training iteration, it may of course be the case that the weights do not converge. This theoretical scenario, however, is very unlikely to occur, as k-WTA will strengthen the connection weights to the first k winners, thus tending to favor the initial winners for the same input, only changing the output which it projects to the CA3-layer very little, if at all. This is exactly the reason for why neuronal turnover is performed in the first case - to recode the weight configuration such that the resulting k winners will vary slightly more, thus increasing its recoding, potentially improving pattern separation and model performance. Note that it is the expansion encoding and sparsity in the DG-layer that recodes and separate similar, yet distinct patterns. Together with the input of the EC-layer, the current values of the CA3-layer result in updating of weights for the recurrent connections to the layer itself, as well as backwards through the DG- and EC-layer, which again are adjusted according to the 'observed' input for the current pattern. In other words, a desired model configuration is when the model dynamically separates patterns due to its recoding qualities, and yet is able to recall them without the use of the pathways from the DG. It may be argued that completely omitting the DG-layer during recall is somewhat unrealistic. However, according to physiological findings, the DG-CA3 pathway is used very little during recall (Wakagi, Yuko; Hattori, 2008), justifying the omission during recall. Furthermore, due to the auto-associative nature of the CA3-layer, if this layer has successfully converged for patterns with little overlap, it is likely to fall into a basin of attraction, performing pattern completion for partial pattern input. Now, one may think that the EC-layer will not necessarily project a partial, recoded pattern to the CA3-layer, as this recoding is performed in synergy with the DG-layer. However, as the observant reader might have noticed, it is in fact in *synergy* with the DG-layer, and as recoding has already been performed, this recoding is propagated to all layers of the network model by the Hebbian learning. In other words, the synaptic weight modification between the EC- and CA3-layer will reflect the recoded pattern. Furthermore, if only partially reflecting it, the CA3-layer may perform pattern completion.

Lastly, the nature of the CA3-neurons, i.e. chaotic neurons, may also impact the chaotic recall capabilities of the model. While it appears evident that the zeta- and eta-equations do not limit successful training of patterns, it may be that they impact the next activation values of the CA3-layer too heavily during recall. This may be argued by considering that in the current implementation, the next eta- and thus zeta-values are based on the former eta- and zeta-values along with the sum of the raw input sum, i.e. the sum of the dot products between the anteceding layers' activation value vectors and their corresponding weight matrices, without computing the terms' transfer function values before summing them. However, as damping factors are used, the former values will be disregarded at an exponential pace. This leaves only the possibility that the sum of the former values along with the new input sum enhances the gravity of the current basin of attraction. This is unlikely, as the model attains successful convergence when an additional term in the input equation during training is present, namely the dot product from the DG-layer, which may also be magnified by a weight-coefficient.

4.2.3 Experiment: Dentate Gyrus Weighting

In Wakagi, Yuko; Hattori's (2008) hippocampal model, upon which Hattori's (2010; 2014) models are based, the DG-layer, performing separation of similar, yet distinct patterns, has the ability to influence the activity of the CA3-layer strongly during learning. The idea that the DG is able to strongly influence the activity of CA3 during learning, is also confirmed by physiological findings (Rolls and Treves, 1998b), and employed in work such as (Norman and O'Reilly, 2003).

When it comes to this thesis' model; as synaptic connections from DG to CA3 are used solely during learning, this may in fact be what is needed in order to encompass and attain the desired emergence of successful pattern separation. Note also that the DG-CA3 weight matrix is initialized with rather high weight values ($\mu = 0.9, \sigma = 0.01$), and a low deviation from those values for the neurons whose synaptic connections become instantiated. This may result in the layer being able to highly influence preceding neurons through its connections. However, this does not necessarily hold, as weights from EC-CA3 may grow towards 1 as well. In either way, I decided to model the potential impact of adjusting the DG-weighting by implementing a DG-CA3 weight matrix *coefficient*, also referred to as the DG-weighting. For each DG-weighting variable value from 0 to 29, 40 experiments are performed - i.e. 10 experiments per set size. Furthermore, these experiments are performed for four different hippocampal model configurations, namely:

1. Asynchronous updating of the CA3-layer values and weights, turnover for every training iteration, using a turnover rate $\tau = 0.04$.

2. Asynchronous updating of the CA3-layer, with neuronal turnover between learnt training subsets, $\tau = 0.50$.

3. Synchronous updating of the CA3-layer and its associated values and weights, turnover for every new training subset, $\tau = 0.50$.

4. Synchronous updating of the CA3-layer and its associated values and weights, turnover for every training iteration, $\tau = 0.04$.

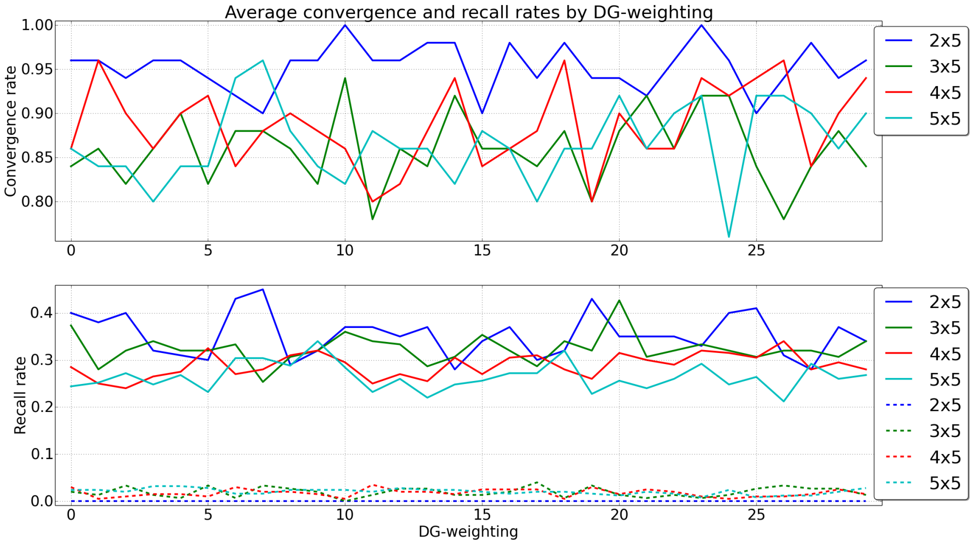

Figure 4.8: Displaying the **convergence rate (upper subplot) and recall rates (lower subplot) by DG-weighting** for the scheme of synchronous CA3-layer neuronal updates, using a turnover rate $\tau = 0.50$, and neuronal turnover between learnt subsets. Note that the dashed lines in the lower subplot denote the spurious recall rates, whereas the continuous lines indicate the perfect recall rates of the corresponding set sizes. It seems that using the DG-layer altogether does not impact the extraction rate of the model with the current parametrization. Furthermore, the extraction rate relative to the pattern size of perfectly recalled patterns remains fairly uniform across all training set sizes.

Note that the model configuration now employs a turnover rate, denoted by τ, of $\tau = 0.04$ in the neuronal turnover schemes where turnover is performed for every training iteration. This is due to the fact that performing turnover for 50% of the neurons between learnt training subsets, does in fact result in less neurons being re-initialized than when performing turnover for every training iteration, when the number of iterations i are above 12 before reaching convergence. Furthermore, $\tau = 0.04$ is, albeit performing turnover implausibly frequently, a more biologically realistic rate in itself. Nevertheless, employing the different rates does algorithmically test slightly different model parametrization and computation aspects. Even though turnover for every training set iteration may be unrealistic, the may provide a basis for further analysis on the topic of randomness in the DG-layer, as well as whether the model may be linked to some hippocampal aspects that operate at a larger time-scale.

Note in figure 4.8 that the perfect recall rate is fairly equal for all training set sizes. This may indicate that either few basins of attractions are formed, or that only few basins are reached during chaotic recall. As the model converges in nearly 100 % of the cases, the latter is likely to be the case. Figures for synchronous CA3-layer updating, using turnover for every set iteration, with the neuronal turnover rate $\tau = 0.04$, are included in appendix

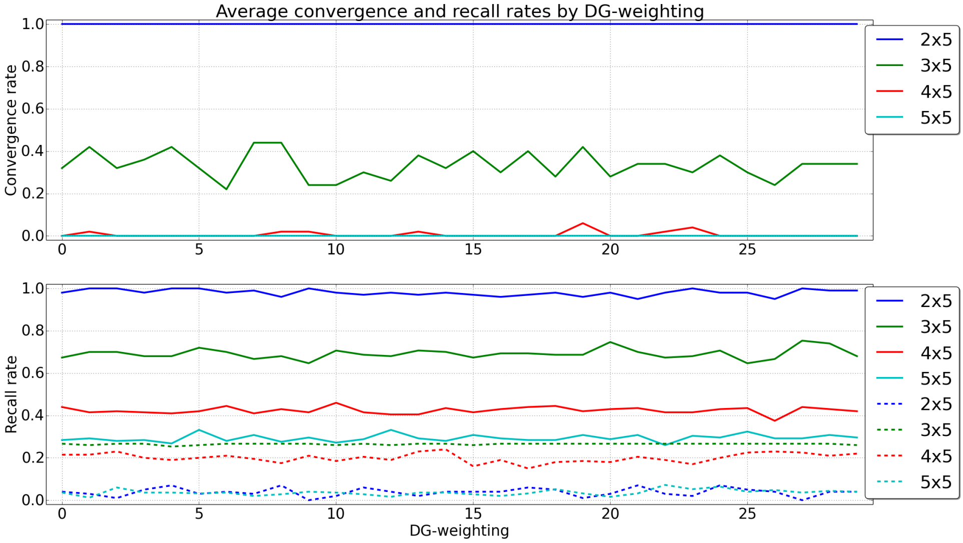

Figure 4.9: Displaying the **recall rates by DG-weighting** as in 4.8, however here for the configuration of asynchronous CA3 updating, and with $\tau = 0.50$, turnover being performed between learnt sets. Interestingly, note that when convergence is near perfect *and* very poor, no spurious patterns are extracted.

D. These figures are very similar to the ones attained for the scheme using turnover for every new set only, with $\tau = 0.50$.

As for the DG-weighting itself, note that there seems to be no significant correlation nor performance gain with the DG-weighting. Therefore, pattern separation from the DG-CA3-connections seem to be largely unsuccessful during recall. Note also that discarding the activity of the layer altogether during learning does not result in significantly worse model performance, which strengthens the hypothesis that the activity of the DG-layer is unsuccessful in heavily influencing the activity of the CA3-layer. This may be the case due to the synchronicity in the CA3-updates. However, when looking at the figures for the asynchronous CA3 updating schemes, both schemes generate very similar figures. Note that the turnover mode with turnover between learnt subsets only is also the only included figure for the synchronous model scheme in this chapter, the latter being contained within appendix D. Furthermore, the poor convergence results in approximately no spuriously recalled patterns. This could suggest that the two other set sizes partly successfully separate patterns. However, when considering that the convergence rate drops rapidly when increasing the training set size, this suggests that pattern separation is unsuccessful, as introducing more patterns (and more overlap) results in very poor model performance. Furthermore, the recall rates for both perfect recall and spurious recall are highly correlated with the convergence rate, which strongly suggests that introducing asynchronicity may increase the perfect recall rate, but does so highly due to the increase in randomness, including spuriously recalled patterns. Furthermore, the model performance for the largest set size is so poor that it may only learn to recall one to two patterns for set sizes 4x5 and 5x5. These basins of attraction are the only ones that the model may converge towards, also strengthening the claim that the model capacity is reduced to one or two patterns due to unsuccessful pattern separation.

4.2.4 Experiment: Neuronal turnover rates

As the neuronal turnover rate may directly impact the model's separation capabilities, and is shown to be correlated with model performance by (Hattori, 2014), I here investigate model convergence, perfect recall rate and spurious pattern recall rate, here defined as non-perfect pattern recall, relative to the turnover rate for several model schemes, see table 4.5. These experiments may elucidate why pattern separation is unsuccessful for all of the DG-weightings in the former experiments.

Interestingly, the asynchronous updating mode and dentate granule neurons' weighting seemed to to have no impact on model performance, nor did varying the turnover rate, τ. Note that asynchrounous CA3-updating, along with a DG-weighting of 25 and turnover mode 1 is not tested in this experiment. This is justified by considering that the DG-weighting had no impact on model behaviour in the previous experiments under this configuration (figure 5.4).

Figure 4.10 raises the question of whether the model contains any issues related to the DG-layer, as the layers' parameters do not seem to affect model performance. Because highly similar results were attained for all three neuronal turnover configurations when using asynchronous CA3-layer updating, figures for the two remaining asynchronous model schemes are contained in appendix D. It may be that the asynchronous CA3-layer updating scheme introduces a certain robustness to the model, seeing that permuting and

Table 4.5: Showing the **setup schemes used for investigating the impact of the neuronal turnover rate** on model performance. Note that neuronal turnover modes 0, and 1, correspond to turnover between every learnt set, and turnover for every training iteration, respectively.

Setup		
CA3 updating mode	DG-weighting	Turnover mode
Async	1	0
Async	25	0
Async	1	1
Sync	1	0
Sync	25	0
Sync	25	1

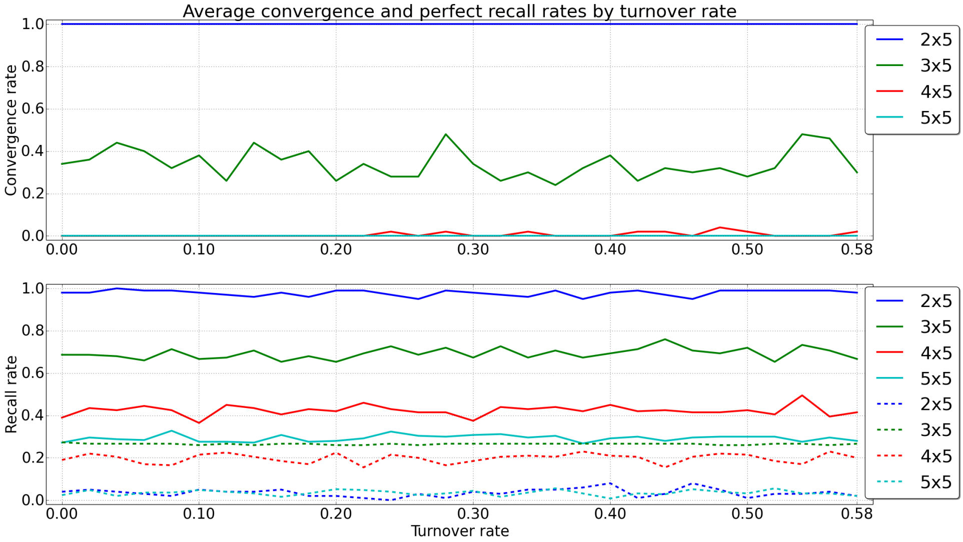

Figure 4.10: Displaying the average **convergence rate (upper graph) and perfect and spurious recall rates (lower graph) by the neuronal turnover rate**. In this figure the model employs asynchronous CA3 neuron-updates, with neuronal turnover being performed between every learnt training subset, and a DG-weighting of 1. Note that the perfect recall rate seems unaffected by a changing turnover rate. Average convergence by neuronal turnover rate, for asynchronous CA3 neuronal updating, and turnover between every learnt training (sub-)set. Note that convergence seems unaffected by a changing turnover rate.

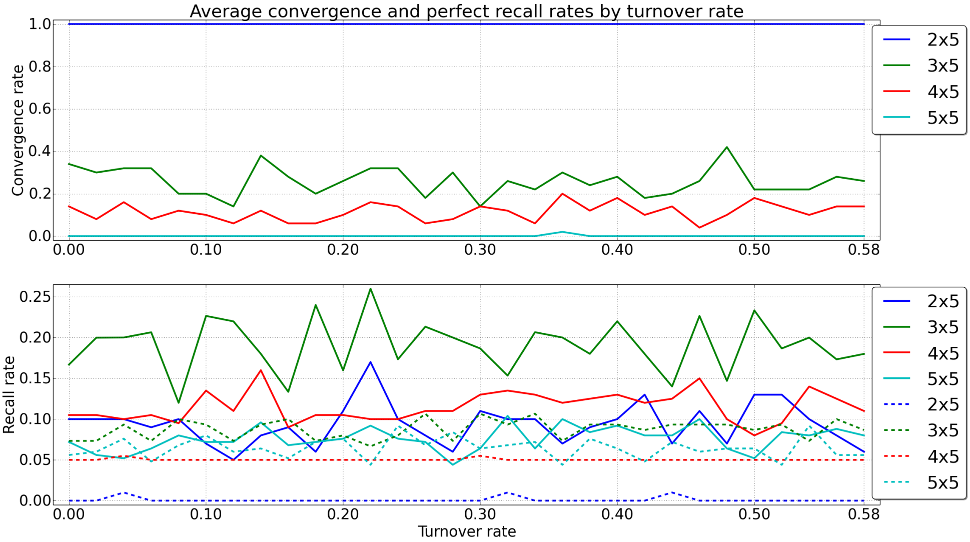

Figure 4.11: Illustrating the **average model convergence and recall rates by neuronal turnover rate** for synchronous CA3-layer updating, using a DG-weight coefficient of 1, and neuronal turnover between every learnt training subset. Note that the model seems to largely fail to converge for any set size other than 2x5, which may indicate that pattern separation is unsuccessful, which would also explain why changing the neuronal turnover rate does not affect the model performance in figure 4.10.

re-instantiating a large number of the EC-DG and DG-CA3 synapses for every training set iteration does not reduce the model performance. However, the convergence rates indicate poor convergence in all scenarios. Furthermore, when investigating the figures (4.12, 4.13) for synchronous CA3-neuron updating, neuronal turnover rate does in fact positively impact the perfect recall rates of the model when the DG-weighting is set to 25.

Model convergence is not attained in the synchronous CA3 neuronal updating scheme when a DG-weighting of 1 is employed, with the perfect recall rate remaining poor. This suggests that the model is both unable to separate the patterns for the correct input patterns during learning and recall when the DG-weighting is 1. Note that when increasing the DG-weighting to 25, the model converges for 80-100 % of the training patterns, irrespective of set size. Further, it results in perfect recall rates about twice as high as employing a DG-weighting of 1, suggesting that increasing the connection weighting of the synapses from the DG-layer to the CA3-layer may in fact enable pattern separation during learning and recall. Note that the neuronal turnover rate seems uncorrelated with model performance when performing neuronal turnover between learnt subsets. Interestingly, when performing neuronal turnover for every training iteration (DG-weighting = 25, synchronous updating), results in the best model performance attained so far. Namely in nearly 80 % of the training sets from the 3x5 auto-associative training set being perfectly

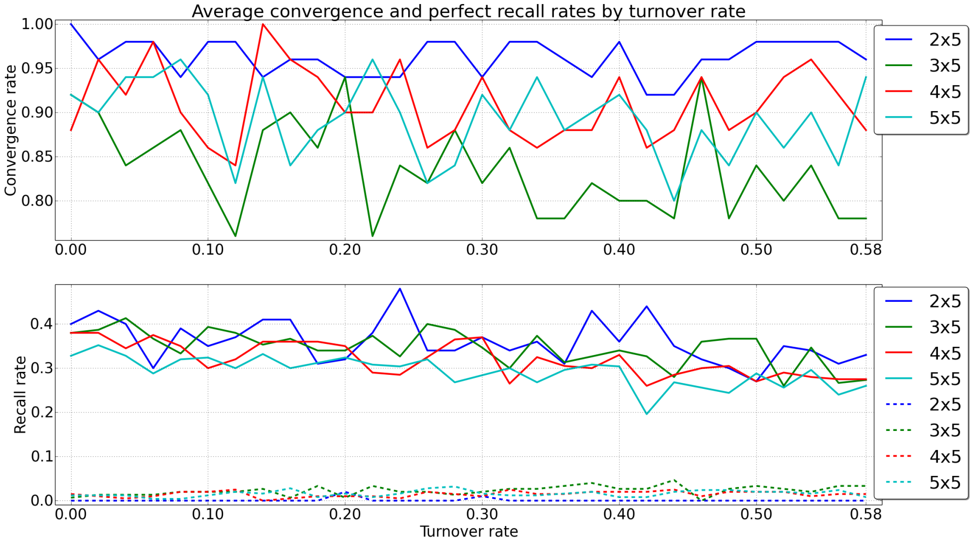

Figure 4.12: Presenting the **average perfect recall rate by neuronal turnover rate** for the scheme of synchronous updating, now using a DG-weighting of 25, turnover being performed for every new training subset. Note that neuronal turnover does not seem to affect model performance significantly. However, there is a slight tendency towards a worse perfect recall rate as the turnover rate grows towards 0.50. Convergence is attained in about 80 to 100 % of the cases, but is not correlated with the training set size.

recalled for $\tau \in \approx [0.40, 0.58]$. Furthermore, there is a slight increase in the recall capability during learning of the other set sizes too. It is important to emphasise that while perfect recall increases significantly for higher turnover rates, with the highest rate in the aforementioned interval; so does spurious pattern recall. While such low spurious recall rates may be acceptable, strict convergence is only attained for sufficiently low turnover rates. Interestingly, the figure suggests that rather high turnover rates may be employed, and yet having low spurious recall rates (such as $\tau = 0.30$). Furthermore, note that some patterns are spuriously recalled for very low turnover rates. This may suggest that when the turnover is too low, pattern separation is unsuccessful, resulting in spurious pattern extraction. Shortly put, figures 4.12 and 4.13 elucidate pattern separation in the outlined model, demonstrating that a certain level of continuous turnover is preferable for successful pattern separation, and further suggesting that employing strongly connected synapses in the DG-CA3 pathway enables pattern separation altogether.

As asynchronous neuronal updates use the newest updated values in the CA3-layer through its recurrent connections, this may in fact introduce too much randomness in the model. One solution to this may be to introduce hardware-mediated asynchronicity through algorithmic parallelization, as this is more likely to mainly use the current neuronal values, thus largely reducing the randomness and the combinatorial explosion of the

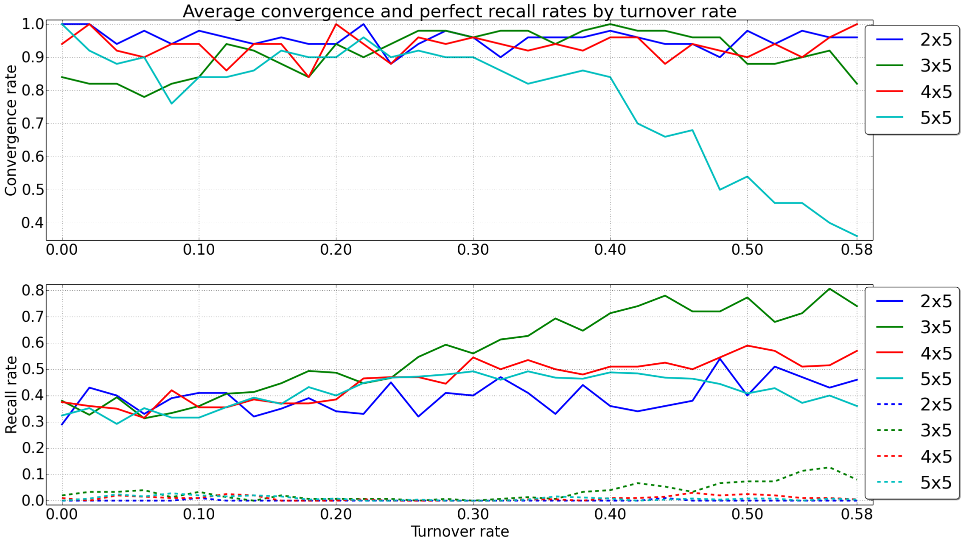

Figure 4.13: Showing the **average perfect recall rate by turnover rate** for the scheme of synchronous updating with a DG-weighting of 25, turnover being performed for every training iteration. Interestingly, model performance is now highly correlated with the turnover rate, when turnover is being performed very frequently. Convergence is attained in about 90 to 100 % of the cases, and seems to be best for a neuronal turnover rate τ in the interval of approximately $[0, 30]$. Note that the convergence drops rapidly for the largest training set size when τ goes above this value (for $\tau \in [0.30, 0.58]$).

outcome space during neuronal updating and wiring. This reduction is somewhat relaxed, enabling a trade-off that maintains a certain randomness in the model during training and recall. Furthermore, this scheme is may be more biologically realistic, as action potential propagation in the biological brain is performed continuously and only slightly synchronously (and at different time-scales). However, both implementing and elaborating more on such a scheme remains outside the scope of this thesis.

Note the successful increase in perfectly recalled patterns for set size 3x5, but not 2x5 when increasing the neuronal turnover rate in the synchronous CA3-updating scheme using turnover for every training iteration, figure 4.13. Because both patterns are successfully recalled when the correct corresponding input is present in the 2x5 training scheme, but not during recall, this suggests that one of the patterns in the 2x5 scheme covers most of the weight space, thus being the only pattern which is reached during learning. Furthermore, increasing the number of patterns also necessarily increases the need for pattern separation in order for the model to converge. Therefore, each pattern is more likely to occupy more of the weight space, thus increasing the area of its basin of attraction (i.e. the area inside which the network will converge towards the output pattern). Additionally, figure 4.12 demonstrates that when turnover is only performed between training sets, increasing the turnover rate only slightly decreases the perfect recall rate. This suggests that while increasing the frequency of performing turnover to every training iteration may provide the model with more randomness, expanding the model's search through learnt patterns, it only slightly improves the recall capabilities. Furthermore, expanding the search space also introduces some more spuriousness to the recall process. This provides the basis for the next experiments, where the convergence criterion is designed to be less stringent, as well as to potentially assign patterns more evenly throughout the model's weight space.

4.2.5 Experiment 4: Relaxing the convergence criterion

Because the results presented above in experiments 1-3 indicate issues related to chaotic recall, these experiments investigate how reformulating and relaxing the convergence criterion may impact model behaviour. Convergence is now considered to be attained once a static number of training or recall iterations have been performed. In this experiment the number of iterations is set to 15 as the criterion during recall, and 15 for most of the experiments during training, based on empirical data from previous experiments.

Although the model is shown to be capable of one-shot learning in the former experiments, convergence is only attained in less than 15 training iterations for set size 2x5 in the introductory experiments using the stringent convergence criterion (of stable output for three recall iterations). On average, the synchronous CA3 updating mode converged in 2-3 training iterations, which demonstrates a clear one-shot learning capability, while the in the asynchronous mode it converged in on average about 7 iterations for training set size 2x5. Nevertheless, as the set size grows larger, i.e. 3-5 per subset, the number of required training iterations grows slightly larger than 15 for the synchronous CA3 updating mode (approximately 20 for the remaining set sizes), and linearly towards 50, i.e. no convergence at all for the asynchronous CA3 updating mode, under the stringent convergence criterion. While convergence thus may not be attained for 15 training iterations according to the previous learning criterion, the model will definitely have had time to be completely exposed to the new training subset, adapting its weights accordingly (to the subset). Furthermore, 15 iterations is also more than sufficient in order to have former short-term memory diminish, as may be seen in the low-level example figures contained in appendix D. However, 50 iterations during training is also used in some of the experiments in order to investigate the potential effects this has on model performance and the quality of extracted patterns.

What I wish to investigate in this experiment are the trends that may arise under a more constant training scheme. Even though convergence is not strictly attained, the number of iterations do allow for learning pattern associations successfully. Furthermore, exposing the model to a constant and uniformly distributed continuous flow of patterns, is more likely to generate trends that are representative of model behaviour, both during recall and learning. As such, observing trends for a more static scheme may ameliorate the unsuccessful chaotic recall that is observed under the more stringent chaotic recall scheme and convergence criterion. Thus, observed trends may in fact provide a better picture of the model behaviour, and the observed trends may potentially help further illuminate the research questions, as well as the pattern separation aspects related to synchronicity and neuronal turnover that remain slightly obscure from previous experiments and results.

Note that perfect recall by chaotic pattern extraction is significantly better for synchronous rather than asynchronous CA3 updating in the local, as well as global training set exposure scheme. Furthermore, the fact that performance is lowered very little in the synchronous updating scheme suggests that the model configuration is capable of fairly robust pattern separation.

It is worth noting that the synchronous CA3 updating scheme results in a relatively large number of spuriously recalled patterns for small set sizes, at least as long as the neuronal turnover rate is fairly low. This may in indicate that the learned patterns' basins of attraction have merged, corresponding to spurious pattern(s) which may then be learned by the model. As previous empirical data, as well as the low-level demonstration has shown that the model on average successfully recalls patterns for the correct pattern input, this may point towards the issue residing within the synchronous updating schemes' recall procedure. Note however that once the model fails to separate two similar patterns and forms a novel spurious pattern and basin of attraction, this basin may increase the likelihood of further such spurious basins being created. This may be seen by considering that such a basin is in fact an overlap between the other basins, which is then likely to contain parts of other patterns and letters, too. In this case, the overlapping inputs will necessarily disrupt the pattern-completion of the CA3-layer, as it cannot settle into patterns that overlap too much. This phenomena has been demonstrated in work on auto-associative networks, as discussed in the background chapter, as well as in (Hattori, 2014), which employs a more complex hippocampal model specifically to alleviate the issues of pattern separation and memory congestion in a simpler more Hopfield-like network, such as used in (Hattori, 2010). Looking to the low level demonstration of the synchronous training and recall scheme, a fair stability is attained for learned pattern inputs. However, chaotic recall seems to be very unstable, only visiting learned pattern outputs for one time-step. In other words, the *correct* basins of attraction have not been sufficiently consolidated for the given letters, as the output oscillates. Whether consolidation is unsuccessful due to the creation of basins of attraction for spurious pattern correlations, or only because pattern separation is unsuccessful - which necessarily renders successful learning and convergence unsuc-

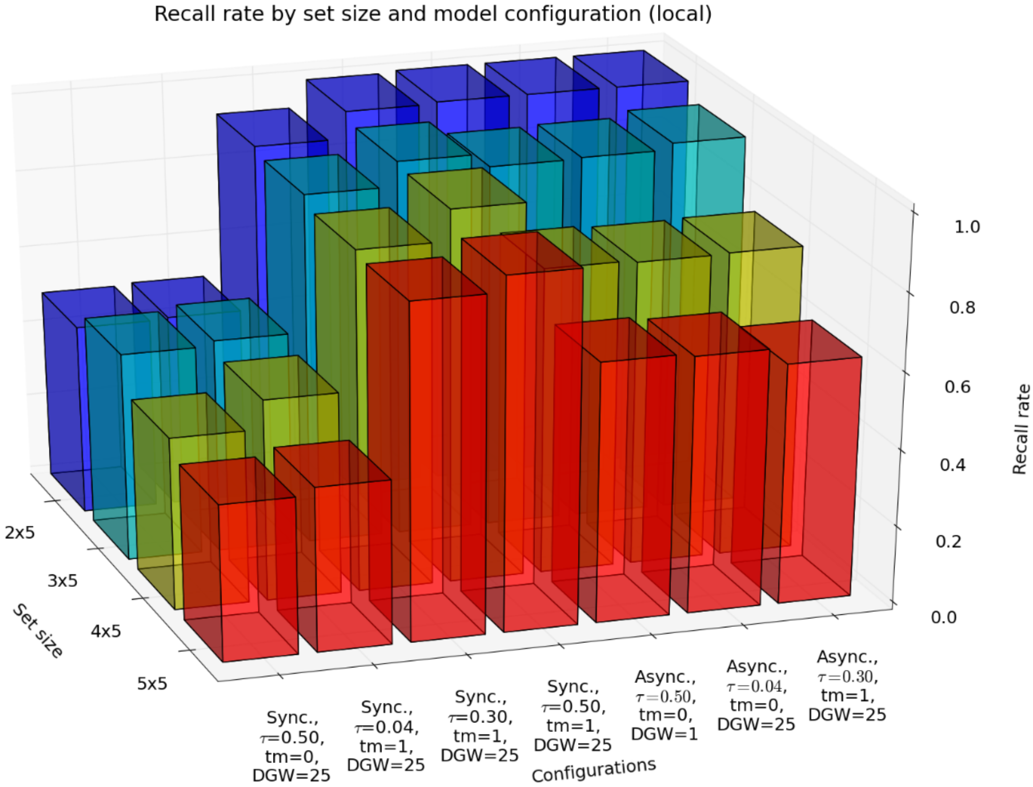

Figure 4.14: Perfect recall rates in several hippocampal model schemes for both synchronous and asynchronous CA3-layer updating, the *'local'* keyword in the title denoting that the models are trained on each subset of the corresponding set sizes, sequentially. Note that performing turnover for every training iteration seems to enhance the perfect recall rate when using synchronous CA3-updating significantly. However, this does not seem to have an effect for low turnover rates, such as $\tau = 0.04$, nor under asynchronous updating schemes. Note also that "tm=0", and "tm=1", denotes that neuronal turnover is performed between every learnt training subset, or for every training iteration, respectively.

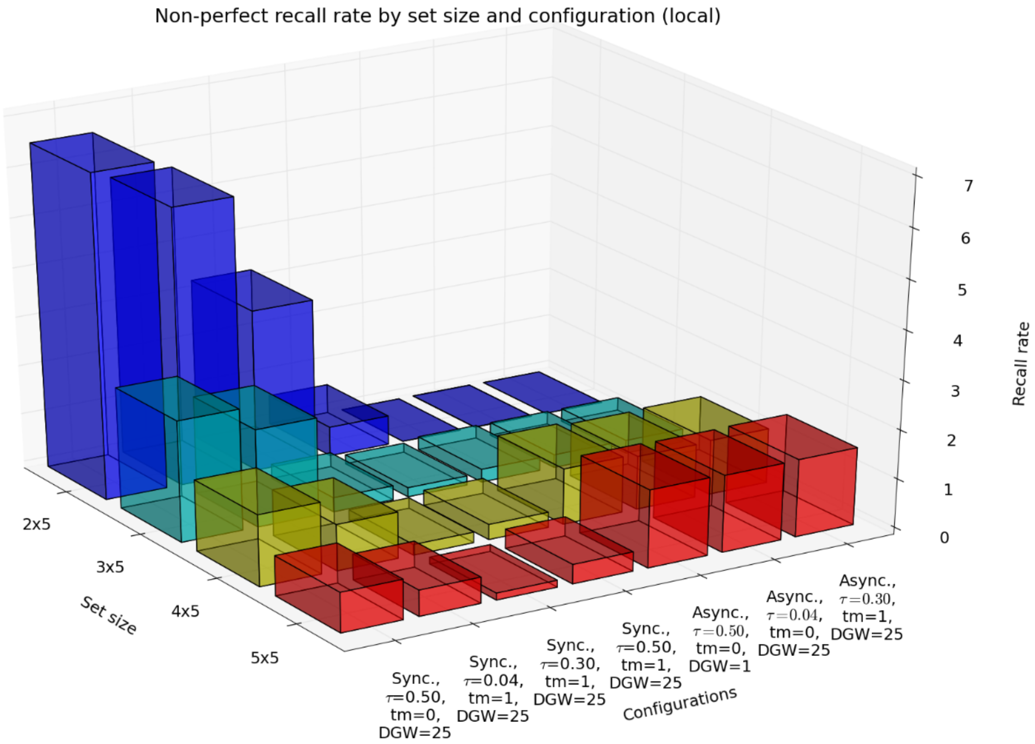

Figure 4.15: Displaying the spurious extraction rates corresponding to the perfect recall rates and model schemes of figure 4.14, for local recall, i.e. training on subsets sequentially.

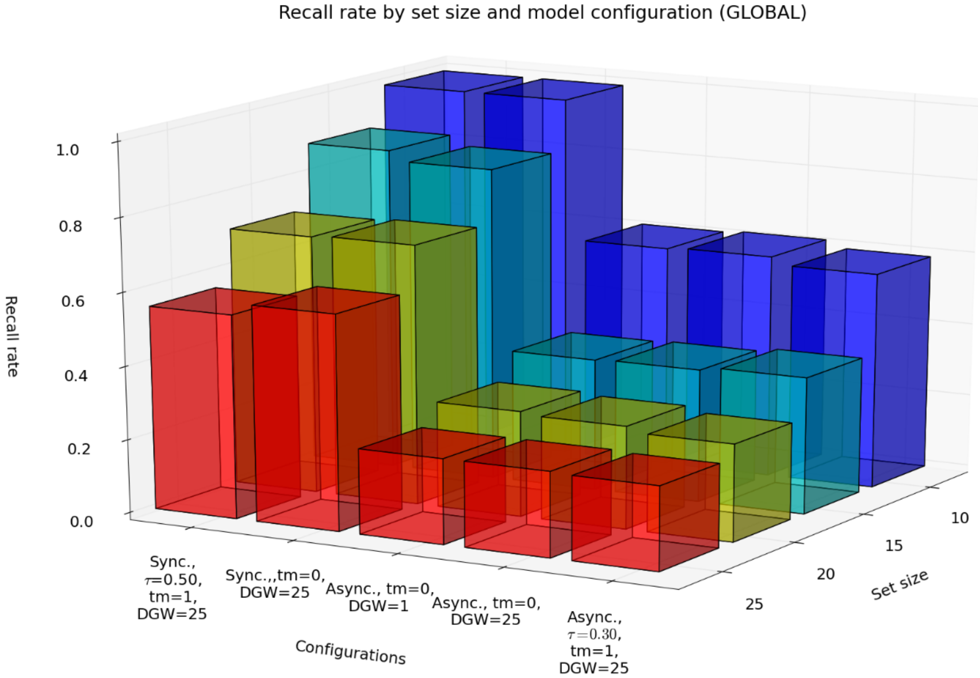

Figure 4.16: Displaying the **perfect recall rates under global training set exposure** attained for five different model schemes when the hippocampal model is exposed to all of the training patterns.

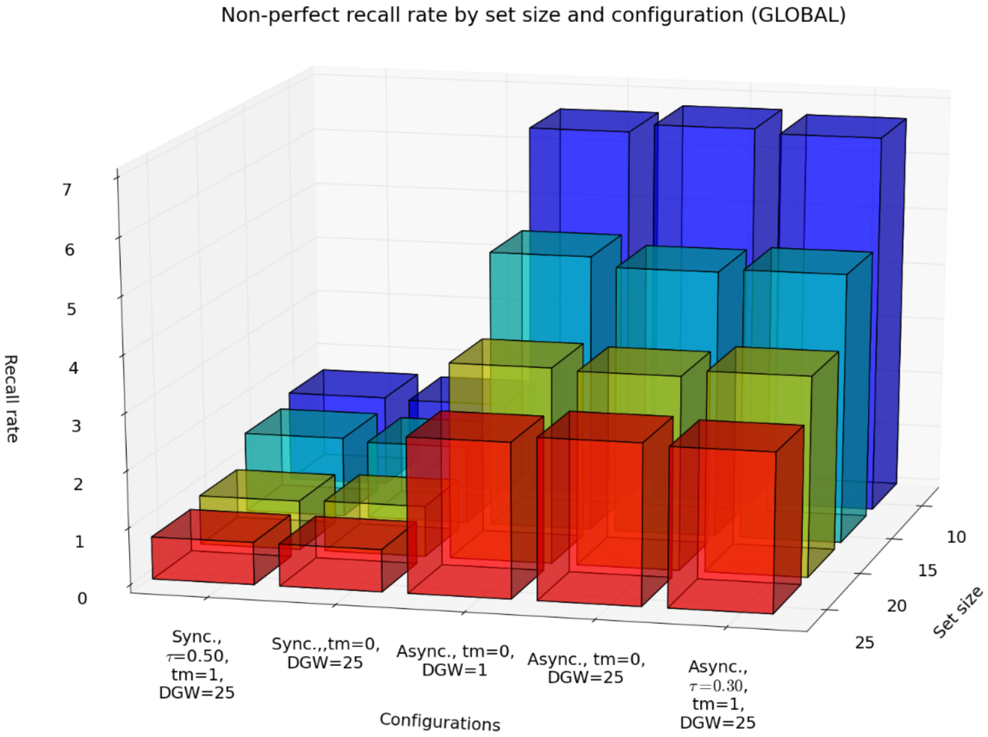

Figure 4.17: Displaying the **chaotically extracted spurious patterns for global training set exposure** under several model schemes.

cessful, remains slightly obscure. That being said, the chaotically recalled output seems to recur only for the actual training pattern output, with spurious chaotically recalled patterns only occurring once per spurious pattern. This suggests that chaotic recall is unsuccessful largely due to unsuccessful reduction in overlap of the training patterns. One thing that could ameliorate this issue is introducing a higher neuronal turnover for every training set iteration, although less biologically realistic. When considering the case where $\tau = 0.50$, neuronal turnover being performed for every training set iteration, the results suggest that this is in fact the case: Due to the small set size, as the turnover is performed relative to the number of training iterations, and thus implicitly set size, pattern separation is not successful. This is worth noting, as would be a logical conclusion that the number of recoding would have to increase relative to the set size. However, when regarding the spurious pattern recall for the different synchronous CA3-updating schemes, it seems that only a certain amount of turnover is required in order to successfully separate *all* training patterns. This may imply that the model, as well as the biological brain, needs a certain randomness to its learning procedure in order to enhance its pattern separation, recoding, and correlation extraction. This could potentially be more biologically realistically implemented by introducing random ”noise” in between training patterns, which would introduce randomness in the eta- and zeta-equations, and thus to the chaotic neurons of the CA3-layer. On the other hand, it may also imply that the model may be too simplified topologically speaking with regards to attaining the desired model behaviour (which presumably is the case with regards to certain aspects). More specifically, relaying the model output from CA3 to plastic connections to a CA1-layer in an extended and more complex model, could relay the activity back to the EC-layer. This may potentially provide the recurrence needed to obtain stability in the output layer during chaotic recall. This will be further discussed in chapter 5.

When analysing the spurious recall rate of the asynchronous schemes during chaotic recall, the asynchronous setups are far worse off than the synchronous. Making it apparent that the potentially increased perfect recall rate in the asynchronous schemes in figures 4.14 and 4.15 comes at the cost of drastically increasing the number of spuriously recalled patterns. Conversely and interestingly, spurious recall is still constrained in the synchronous updating mode. This may be due to the fact that asynchronicity simply may introduce more combinations of the previous basins of attraction, whereas the synchronicity greatly constrains the outcome space given the weight space (and training patterns during training). Thus, asynchronous CA3-updating may lead to the output after chaotic recall being a previously unseen, distinct spuriously recalled pattern for every spurious chaotic recall iteration. However, the fact that the synchronous mode is not prone to recalling that many chaotic patterns in the global training exposure mode may indicate that the model does in fact converge well during training, also in terms of pattern separation. The perfect recall rate remains very high, while the number of spurious patterns extracted grows very little when increasing the training set size by a factor of 5. This suggests that a more complete pattern-completion mechanism may be a central aspect which needs improvement in order to enhance the model performance, and possibly emergent pattern separation ability.

Addressing the success observed by using neuronal turnover for every training set iteration in the synchronous CA3 updating scheme: This will necessarily increase the pattern

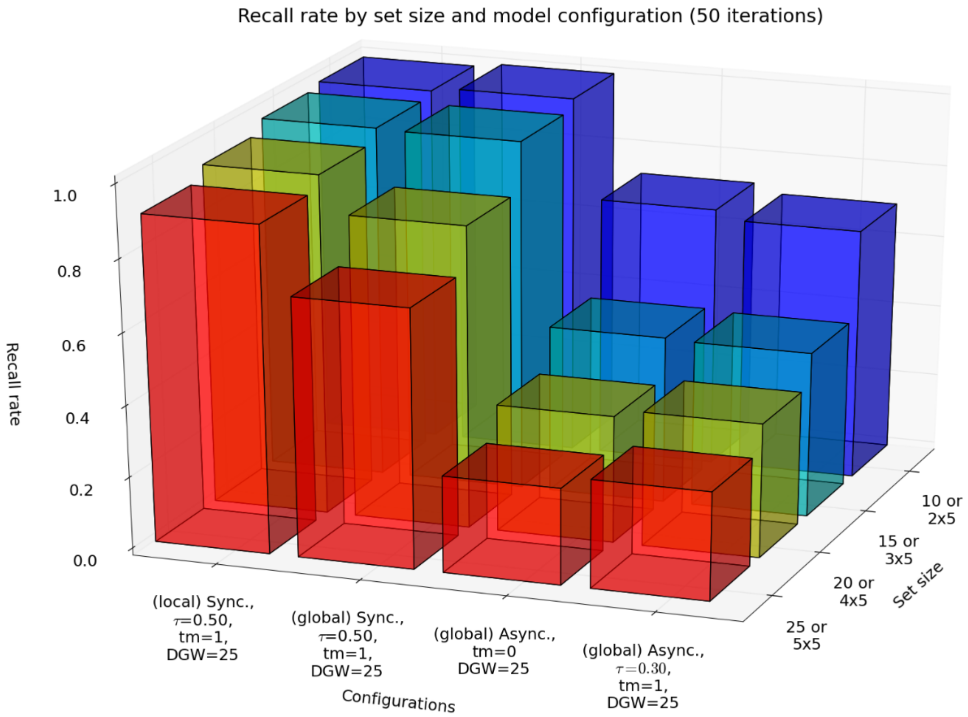

Figure 4.18: Displaying further recall results for four different model schemes, three of which are global, and one which is local. What distinguishes this plot from the previous are the **50 training iterations** that are used to train the hippocampal model, for each parametrization and configuration.

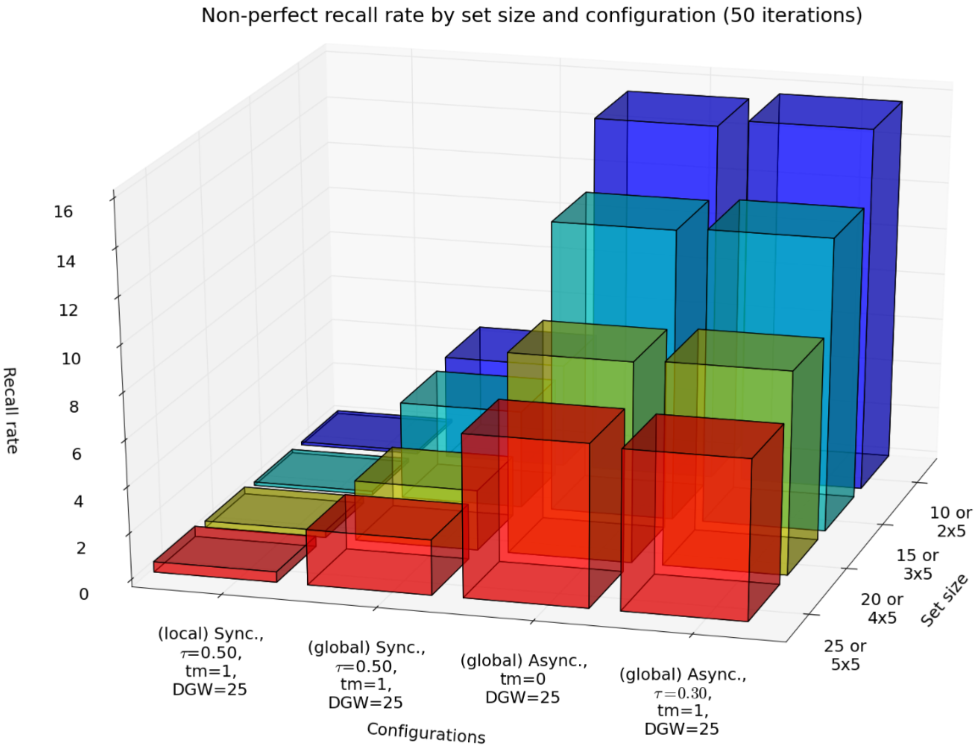

Figure 4.19: A 3-dimensional graph of the **spurious recall rate by set size and model scheme, each model trained for 50 iterations**, corresponding to the perfect recall rates attained and displayed in figure 4.18.

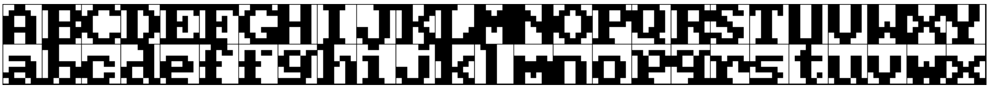

Figure 4.20: Displaying **the hetero-associative training set**, the first row being the pattern inputs, and the second being the associated outputs.

separation capabilities of the model dynamically, as the model by rewiring parts of its connections thus will re-code the presented k-winners and activation pattern to the preceding layer. However, as this is also performed for each exposure to the same pattern, it remains somewhat unclear how it still enhances the separation ability. My take on this is that more randomness in re-instantiating the synapses will lead to varying the k-WTA pattern, which again will be consolidated for the successfully separated patterns dynamically, as Hebbian learning wires the neurons that are simultaneously firing, enhancing the connections that are more persistent, and thus highly correlating, more than others. Thus, re-instantiation of synaptic connections, or neuronal turnover, becomes a type of dynamic trial-and-error in pattern separation, which may be successful due to the fact that lateral inhibition is simulated by the k-WTA algorithm, which favours a certain layer-wise stability.

4.2.6 Hetero-associative training patterns

In order to verify that the hippocampal model generalises to training sets of hetero-associative training patterns, which after all may be said to generally be the nature of more realistic training data, hippocampal model performance is evaluated on the hetero-associative training patterns, illustrated in figure 4.20.

Note that the results in figure 4.21 demonstrate a performance of approximately equal quality as in the auto-associative case. Furthermore, when using a turnover rate τ of 0.30, the same interesting phenomenon of a very large number of spuriously extracted patterns relatively is observed for set size 2x5. This may indicate that there is some mechanism at play which reflects internal model behaviour, rather than training set specific results for this training set size. One hypothesis on why this occurs includes that pattern separation might not have had time to occur succinctly before training is regarded as complete (the criterion of 15 iterations being reached). This hypothesis is further strengthened when considering that the increase in τ significantly decreases the number of spuriously extracted patterns. Furthermore, when the set size grows larger, a turnover of $\tau = 0.50$ shows an increase in the number of spurious pattern recall. Note that this also strengthens the view of the trade-off between successful pattern separation and spurious recall. Interestingly, the spuriously recalled patterns also seem to encode the principal correlations of the patterns in the data set, resulting in equal, or in fact enhanced neocortical model performance in terms of the goodness of fit in the previous section 4.2. Neocortical memory consolidation is not performed here for the hetero-associative training patterns, as this would require expanding the hippocampal model significantly, or designing a permuting input with the extracted chaotic outputs scheme, which would remain fairly biologically unrealistic. Therefore, memory consolidation in the case of hetero-association is considered to be outside the scope of this thesis.

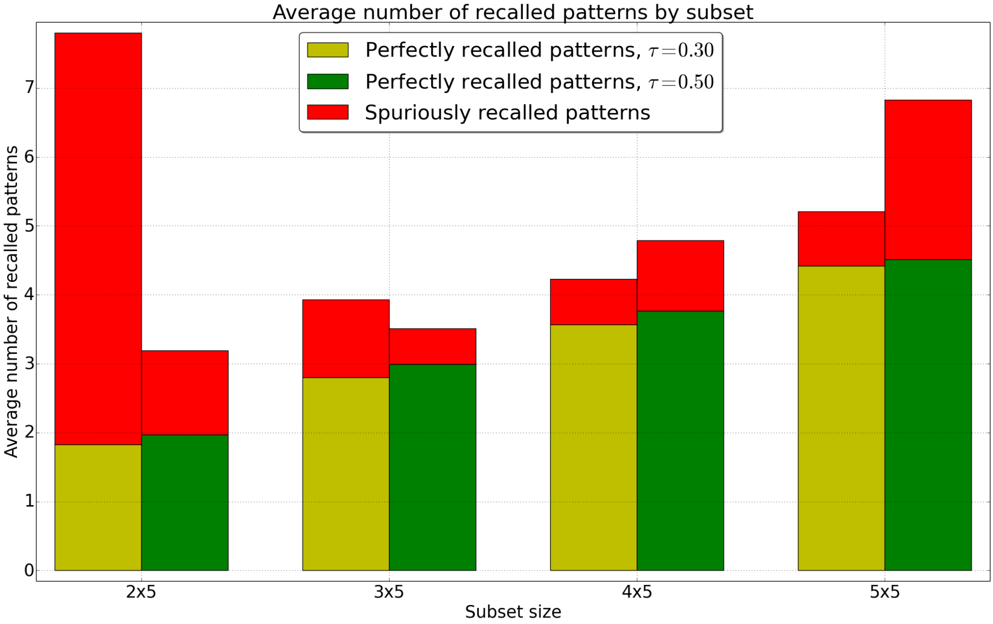

Figure 4.21: Displaying the number of recalled patterns when training on the hetero-associative training set, using the two best model configuration schemes attained in this thesis.

Table 4.6: Showing the approximate average perfect recall rate values using **auto-associative** training patterns for the conventional state-of-the-art model of (Hattori, 2014), and the novel model implemented in this thesis. See figures 4.14, 4.16, and 4.18 for figures illustrating the attained recall rates.

Perfect recall rates				
Pattern set size	2x5	3x5	4x5	5x5
Conventional model	1.00	0.98	0.91	0.81
Novel model	1.00	0.98	**0.96**	**0.90**

Table 4.7: Showing the approximate average perfect recall rate values using **hetero-associative** training patterns for the best model scheme as displayed in figure 4.21. Comparing the conventional model from (Hattori, 2014) with the novel implemented model.

Perfect recall rates				
Pattern set size	2x5	3x5	4x5	5x5
Conventional model	1.00	1.00	**0.97**	0.88
Novel model	0.99	1.00	0.94	**0.90**

4.2.7 Model comparison

When comparing the attained model results in tables 4.6 and 4.7, the novel model parametrization and CA3-updating scheme appears to be better at pattern separation for auto-associative training patterns, possibly due to its significantly increased neuronal turnover rate. In training on hetero-associative training patterns, the model performance appears to be approximately equal, perhaps slightly better in the conventional model for set size 4x5, having an extraction rate which is 3% higher than the novel model. However, as particularly evident for auto-associative patterns, the novel model still seems to be better at tackling the largest set size; 5x5, where the extraction rate is 0.02 more than for the conventional model.

4.2.8 Hippocampal module results summary

Under the less stringent convergence criterion of training or recalling for 15 iterations, it became apparent that the trends for one model configuration demonstrates significantly better performance. Namely the configuration of synchronous updating of the CA3-layer, using turnover for every training iteration, a DG-weighting of 25, and a turnover rate of $\tau = 0.30$. This configuration demonstrates a state-of-the-art performance for dual-network memory models in short-term memory pattern extraction and perfect recall rates. Perfect recall rates are significantly better than the models of (Hattori, 2014, 2010), with the comparative results being provided in table 4.6.

To summarise the main findings of the experiments specifically targeting the hippocampal module; the DG-weighting shows no correlation or impact in the asynchronous CA3-updating schemes. However, it seems to be crucial to the performance to have the $DGW = 25$ in the synchronous updating modes, which yields about twice the perfect recall rates.

As for the neuronal turnover rate, τ, it seemed to have no effect on neither the perfect recall rate nor the convergence rate using asynchronous CA3-neuronal updating. Furthermore, the average results reveal that there are no clearly visible impact under the synchronous updating scheme when turnover is only performed between the learnt subsets. However, when turnover is performed for every training iteration, the results demonstrate an increase in hippocampal model performance correlated with an increasing turnover rate - i.e. both the convergence rates and perfect recall rates increased as the turnover rate did, up to a certain limit of about 0.3, after which the perfect recall rate increased significantly for set size 3x5, but convergence and perfect recall decreased for set size 5x5, as well as spurious recall rates for all set sizes.

In the last experiment on auto-associative patterns, the relaxed criterion scheme demonstrates that the criterion of having a stable output for three recall iterations when a random input is provided to the network is far too stringent. Furthermore, in the local training set exposure scheme, synchronous CA3-updating, performing neuronal turnover for every training set iteration, as indicated in the previous experiments on the neuronal turnover rate, is significantly better than the other configurations. It is worth noting also that it displays a rather interesting property for the set size 2x5, i.e. a relatively large number of spuriously recalled patterns. As for the global training set exposure schemes, synchronous CA3-updating continued to significantly outperform the asynchronous. However, *not* performing neuronal turnover yielded slightly better model performance in terms of perfect recall for global set exposure. When increasing the number of training iterations to 50, the performance became more even. However, the model behaviour demonstrates a significant advantage in using local subset exposure to the hippocampal and short-term memory model, as this resulted in a perfect recall rate above 90%, along with few spuriously recalled patterns - even fewer when using 50 training iterations. For the best model scheme 50 training iterations deviated very little from when using 15, the main difference being the extraction of less spurious patterns.

Lastly, when training the most successful hippocampal model schemes on the hetero-associative training set, model performance remained fairly equal. This demonstrates a capability of successfully extracting hetero-associative patterns, and furthermore that the emergent model qualities such as pattern separation generalises to heterogeneous data sets. Note that the observed robustness in model performance is in slight contrast to the findings of (Hattori, 2014), as his model performance declined slightly when training on auto-associative training patterns, compared to hetero-associative training patterns. This may be due to an increased overlap in pattern-correlations in weight space.

4.3 Neocortical module experiments

This section contains the experiments testing the memory consolidation to the neocortical module within the dual-network memory model. Beginning with a fairly short demonstration of convergence and interference in the outlined network with the auto-associative training set, the section is followed by a short section on pseudorehearsal, preceeded by experiments on memory consolidation from the hippocampal module. Note that 'memory' is used throughout this thesis as describing any abstract representation such as a pattern association or functional mapping inherent in the network weight configuration, and thus embodied by the network itself, resulting in specific emergent network activity.

Based on the assumption that only successful extraction of patterns in the hippocampal module may provide the basis for successful information transfer to the neocortical module, the best synchronous CA3-updating mode and hippocampal configuration is selected as the model scheme for which neocortical memory consolidation is performed. Furthermore, as I wish to investigate the potential information inherent in spurious patterns, the asynchronous updating scheme is also included. Consolidation using chaotically recalled patterns and hippocampal pseudopatterns is also examined in this regard, and to draw potential biological parallels.

4.3.1 Goodness of fit

In evaluating the performance of the neocortical module, the goodness of fit measure, g, is adapted from (Hattori, 2010, 2014). It is defined as the following,

$$g = \frac{1}{N} \sum_{i=1}^{N} o_i t_i, \tag{4.1}$$

where o_i is the target output vector for pattern i, N is the number of patterns in the current training set, and t_i is the target output vector. Note that as the outputs are bipolar, i.e. 1 or -1, in the case of matching only 50 $\%$ of the output, the goodness of fit will be 0.

Note that the model in being a FFBP ANN is not necessarily able to extract perfect pattern correlations, and the goodness of fit measure is well suited for measuring a closeness in perfect correlation - where 1 is perfectly extracted, 0 is 50 $\%$ - i.e. random performance, and -1 is perfectly negatively correlated. The goodness of fit will therefore be the main measure in evaluating the consolidation quality, as well as the network performance.

4.3.2 Model demonstration and potential catastrophic forgetting

Before delving into the experiments of the neocortical module, I would like to outline and demonstrate how learning of all pattern associations may be successfully attained in the feed-forward back-propagation (FFBP) network studied in this section. This may be achieved by introducing the training set as a global training set, i.e. training on every single of the 25 patterns (in the 5x5-case) sequentially for a large number of iterations. This will lead to the successful convergence and a goodness of fit measure g of above 0.99 in most cases. However, when constructing sequentially detached training sets, i.e. 5 subsets of the 25 patterns, allowing the model to train only on one subset at a time, this results in catastrophic interference in the model. This brings us to the very core of the dual-network memory architecture; as everything cannot be learnt sequentially as a global training set, at least not biologically speaking, an architecture where subsets may be learnt rapidly by a short-term memory, may allow for the slow consolidation to a long-term memory such that its former memories are not disrupted.

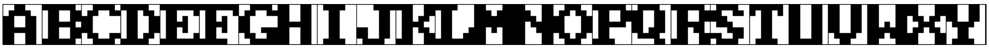

Figure 4.22: Displaying the bipolar, **recalled output for the neocortical network after training** on all of the associative training patterns sequentially (as one training set) for 15 iterations. The goodness of fit of the network is $g \approx 0.99$.

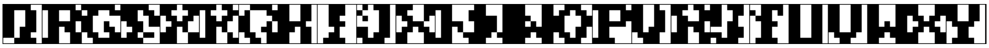

Figure 4.23: Illustrating **catastrophic forgetting in the neocortical network**, after it has been trained on each subsets of the associative training patterns sequentially (5x5), each for 15 iterations. Note that the recalled output is the model output after being presented with the corresponding input pattern, presented as bipolar values. The goodness of fit of the network is $g \approx 0.79$.

4.3.3 Experiment: Memory consolidation by chaotically recalled patterns

In this experiment, the random inputs and the corresponding chaotically recalled outputs (after recalling for 15 iterations), referred to as chaotic, are used in order to attempt to consolidate the functional pattern mapping to the neocortical network. Furthermore, hippocampal pseudopatterns are also used in order to attempt to consolidate the extracted patterns to the neocortical network - the hippocampal pseudopatterns being defined as outlined in chapter 3.2.3, now with the relaxed, constant convergence criterion embedded in their generation processes, defined as,

I. A random pattern is generated and input to the hippocampal network, which, with the corresponding output after 15 recall iterations, is a pseudopattern type I.

II. Each element of a chaotically recalled output changes its sign with probability P, here set to $P = 0.1$, and the pattern is input to the hippocampal network, which, with the corresponding output after 15 recall iterations, is a pseudopattern type II.

Note that pseudopatterns type I as described above essentially are chaotically recalled patterns. Thus the analysis of the inclusion of them does in a way only extend the number of chaotically recalled patterns used in the training set, which may potentially alter the performance.

Following are the hippocampal model configurations used for memory consolidation to the neocortical network model, ”DGW” denoting the dentate gyrus weight coefficient:

- Synchronous CA3-updating, DGW = 25, turnover every training iteration, $\tau = 0.30$, using 15 training iterations in the hippocampal model.
- Synchronous CA3-updating, DGW = 25, turnover every training iteration, $\tau = 0.30$, using 50 training iterations in the hippocampal model.
- Asynchronous CA3-updating, DGW = 1, turnover between every learnt subset, $\tau = 0.50$.
- Asynchronous CA3-updating, DGW = 25, turnover for every training iteration, $\tau = 0.30$.

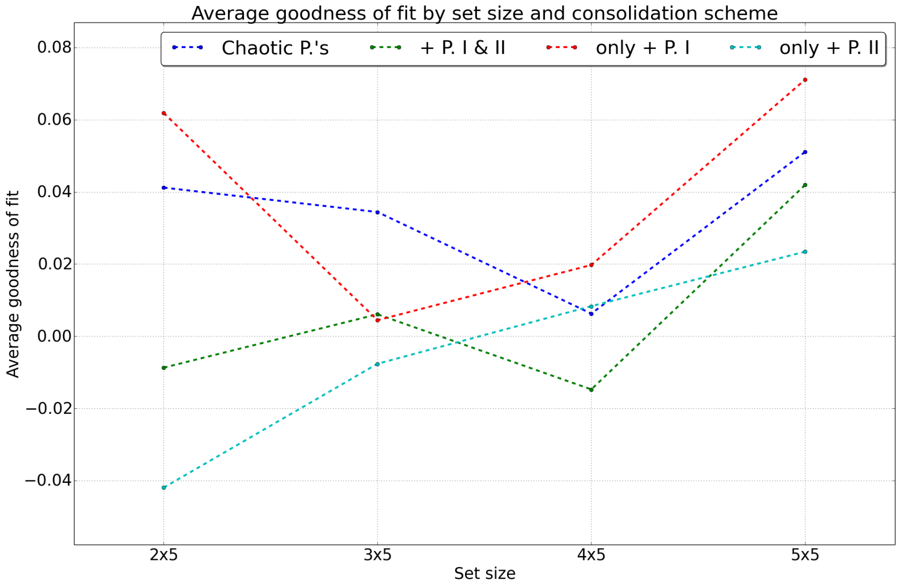

Figure 4.24: Displaying the average goodness of fit for four different consolidation schemes when training for 15 iterations per subset, namely; **(1)** by only using the chaotically recalled patterns as the training patterns, **(2)** by using the chaotically recalled as well as pseudopatterns I and II, **(3)** by using the chaotically recalled patterns and pseudopatterns type I, and **(4)** by using chaotically recalled patterns and pseudopatterns II. Here the first hippocampal model configuration as listed above is the one which is used, namely synchronous CA3-updating, DGW = 25, turnover for every training iteration, $\tau = 0.30$, and training for 15 iterations per subset. Note that the average goodness of fit seems to be approximately 0 in all consolidation schemes.

The results displayed in figure 4.24 suggest that no information is transferred by using any combination of chaotically recalled hippocampal patterns for the hippocampal recall scheme, as the goodness of fit remains approximately 0 for including pseudopatterns I, II, or both I and II. Furthermore, when halving the pseudopattern set size, or if increasing the number of training iterations per subset to 200, the performance remains the same; $g \approx 0$. Lastly, the consolidation performance, i.e. average goodness of fit, remains approximately 0 in all of the hippocampal model schemes listed above, figures being included in appendix D. This raises the question of whether the chaotically recalled patterns contain any qualities which may reduce catastrophic interference, or any information which may suggest biological parallels, which is addressed in the next experiment.

4.3.4 Experiment: A novel memory consolidation scheme

Due to the fact that the chaotically recalled patterns for a random input, and an attained fairly stable output, consolidated no information to the neocortical network, I here investigate the consolidation performance when using the chaotically extracted output as both input and output to the neocortical network. When only extracting the target outputs for the training set, this would correspond to training on each subset of the original training patterns sequentially for the auto-associative training patterns. This means that a baseline measure is that of the average goodness of fit when training the network on these original subsets, which results in an average goodness of fit $g \approx 0.79$ (see figure 4.23 for the results from one of the runs). Furthermore, the baseline averages, i.e. average goodness of fit when training the neocortical network on the original training subsets, are included per subset in the composite model scheme figures. That is, when investigating the performance of using the chaotically recalled output as both input and output for the auto-associative training sets, i.e. essentially relaying the activity of the input and output layers of the hippocampal module. I would like to point out that biologically speaking, the activity of CA3 is relayed to CA1 in the hippocampus, which is also relayed back to the EC. Although having a take at how this occurs is outside the scope of this thesis, the existence of such pathways may provide a mechanism for relaying heterogeneous patterns to parts of the cortex, such that the input and output of the neural network may correspond roughly to the pattern by relaying the activity of the EC and CA1.

In these experiments, the same four local schemes are used as in the previous experiment, being outlined in section 4.3.3 above. Each average value is the average over 20 trials for the corresponding model configuration and set size.

It is interesting to note that when distorting the original training patterns, the neocortical network is very sensitive to the number of training iterations used per subset. However, when using chaotically recalled patterns, it seems that the model is far more robust. This may suggest that the principal correlations are inherent in most chaotically recalled patterns. Furthermore, as the hippocampal model performs pattern separation, it may output patterns which are more separated in weight space, and thus more well suited for sequential consolidation to a neocortical network. This may lower the need for pseudorehearsal in order to maintain previous patterns, and is therefore a more biologically realistic mechanism for pattern consolidation.

Note that there seems to be a possible correlation between spuriously recalled patterns and the goodness of fit, when investigating figure 4.26. Furthermore, when investigating

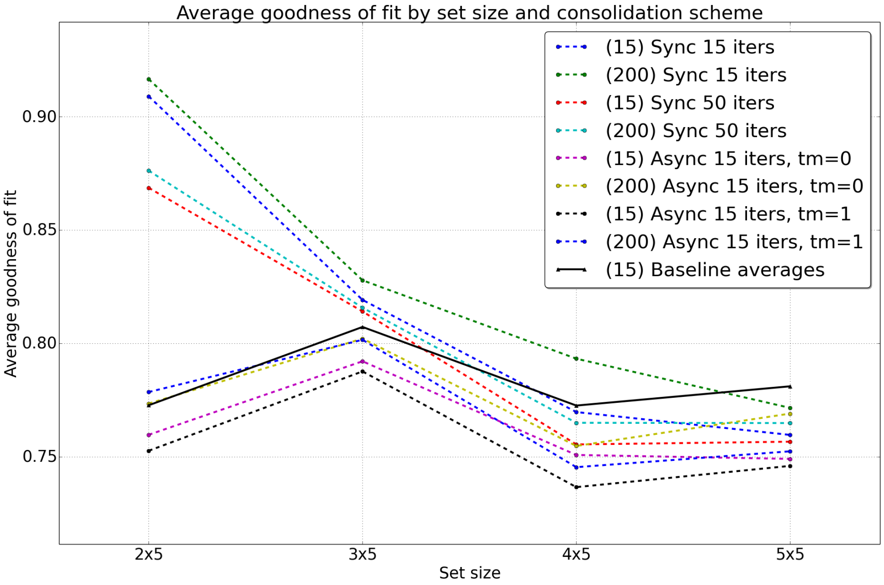

Figure 4.25: Illustrating the **average goodnesses of fit for four different hippocampal model schemes**, all **using the relaxed training and convergence criterion** ($i = 15$) in the hippocampal module. Note that the **bracketed numbers denote the number of training iterations used in the neocortical network** during pattern consolidation, being either 15 or 200, and the thick, **black line denotes the baseline averages for sequential training on the subsets of the original training set**. It seems that the averages do not deviate significantly from one another, with the exception being for set size 2x5 in the synchronous CA3 updating scheme, where the goodness of fit is significantly improved by using the chaotically recalled patterns as the spanning input and output of the neocortical network.

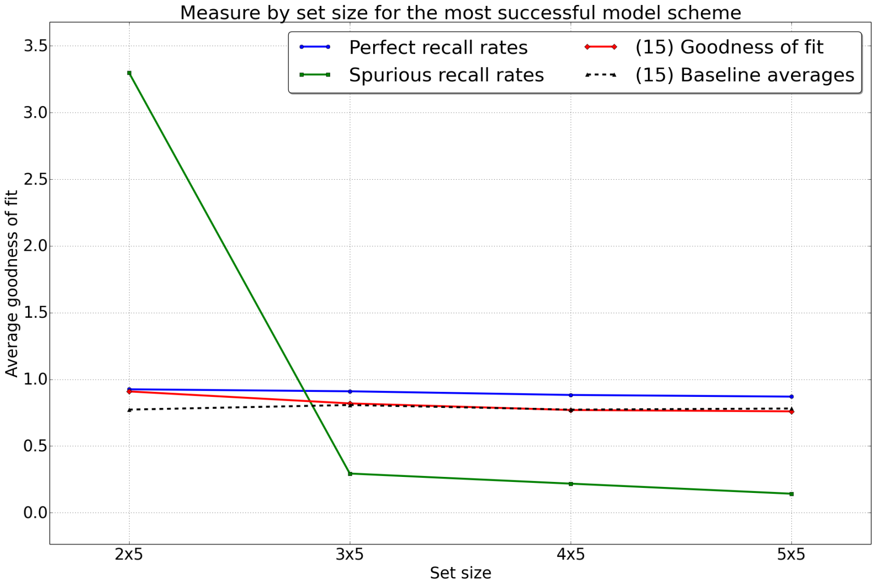

Figure 4.26: A plot of the **main measures used in this thesis to evaluate the dual-network memory model**, using one of the most successfull dual-network memory model schemes attained in this thesis; **synchronous CA3-updating,** $\tau = 0.30$**, tm=** 1**, DGW=** 25**, and training on all extracted patterns, using the output as input and output to the neocortical network** (i.e. symbolising a potential pathway relaying of the pattern signals). Note that the goodness of fit is approximately the same as the perfect recall rate for set size 2x5, when the relative spurious recall rate is more than 3 times the set size. However, when the set size increases, and the spurious recall rate drops below 0.5, the goodness of fit drops by 0.10. It continues to drop slightly for the preceding set sizes, while starting to flatten out, too. Note that the exact same description fits for the spurious recall rate.

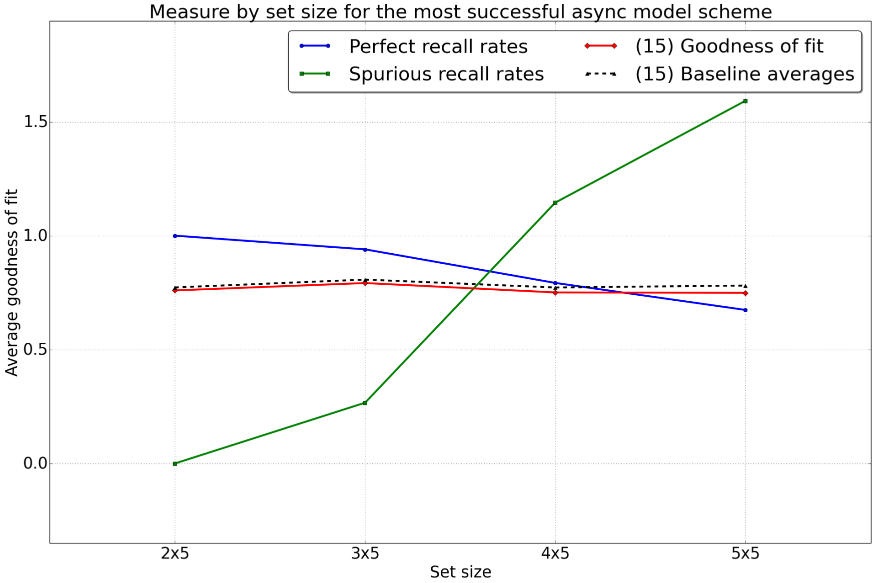

Figure 4.27: Displaying the **main thesis measures**, here under the same scheme as in figure 4.26, however, by employing **asynchronous CA3-updating**. Note that the spurious recall rate seems correlated with a decrease in the perfect recall rate here, as opposed to in the figure for the synchronous scheme.

figure 4.27, it seems that no correlation is present between the two variables - only a negative correlation between the spurious recall rate and the perfect recall rate. Looking back to previous figures in the more stringent convergence scheme, this may reflect the fact that the model when using asynchronous CA3-updating introduces more randomness, and so introduces more truly spurious patterns when failing to convergence. Quite differently, the spurious patterns in the synchronous CA3-updating scheme are present in the mode which is most likely to converge. This might possibly be because the model is allowed to extract patterns by chaotic recall for a larger number of iterations relative to the learnt patterns. It is nevertheless a noteworthy observation, that these seem to enable the neocortical network to attain, on average, a high goodness of fit. In fact, this goodness of fit remains approximately the same when varying the training iterations per subset of chaotically recalled patterns in the neocortical network from 15 to 200. This is yet another interesting quality of the chaotically recalled patterns. When training on the subsets of the original training patterns, the average goodness of fit is very insensitive to the number of neocortical training iterations (see among others figures 4.26, 4.27, 4.25). However, the goodness of fit is approximately the same for all set sizes, being approximately $g \approx 0.8$. Furthermore, it does not necessarily follow that this robustness to the number of neocortical training iterations will arise for the goodness of fit, strengthening the view that there may be some biologically realistic mechanisms at play in the creation of spurious chaotically recalled patterns in the best model scheme, particularly for set size 2x5. The same holds when increasing the number of training patterns in the original training set. To see this, it may be demonstrated that the average goodness of fit, if creating a number of very lightly permuted patterns of the original training patterns and training the neocortical model solely on these, falls rapidly. Furthermore, the neocortical model is highly sensitive to the number of training iterations. This strengthens the hypothesis that the spurious patterns reflect the principal correlations of the original training set, and further possibly contain information about the separating hyperplanes in weight space generated by the recoding capabilities of the CA3-layer (keep in mind that the hippocampal model is not a FFBP ANN, so it does not necessarily perform principal component extraction). In order to illuminate the discussion above, a figure of the aggregate neocortical output for the original input patterns is shown in figure 4.29.

Figures 4.29 and 4.28 suggest that the spurious patterns are an aggregate representation of the original training set. By comparison, the neocortical network output when training with catastrophic forgetting on the undistorted original training patterns does not output letters resembling the output patterns for any other than the last two training sets (figure 4.30).

In order to test the hypothesis about the qualities of patterns extracted by chaotic recall and spurious patterns containing crucial information, the scheme of synchronous CA3-updating, with DGW=25, tm=1, and $\tau = 0.50$ is also used to generate results which are compared with those of when using $\tau = 0.30$, as these model schemes result in slightly different spuriously recalled patterns (see figure 4.14). Furthermore, perfect recall and spurious pattern generation is also shown for a novel scheme which is introduced, namely; using $\tau = 0.30$, tm=1, DGW=25 in the synchronous updating mode, but having the number of training iterations vary with the set size by a factor of 1.5.

These results indicate that the hypothesis formed above about the number of spurious

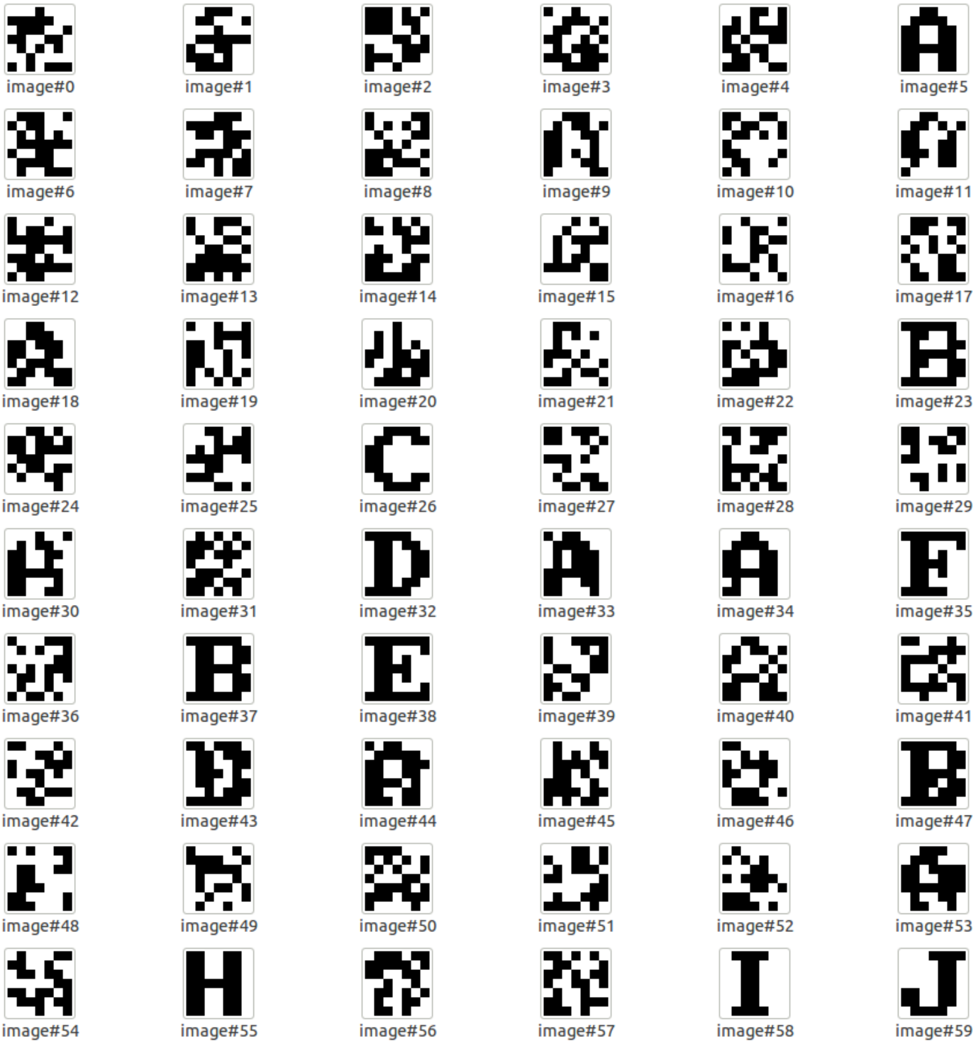

Figure 4.28: Displaying the **patterns extracted by chaotic recall for one single experiment where the dual-network memory model is trained on the 2x5 auto-associative training set**. Note that most of the patterns are non-perfectly recalled patterns, or spurious patterns.

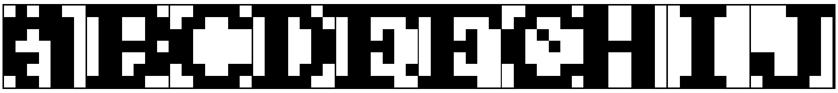

Figure 4.29: Illustrating the bipolar **output of the neocortical network, using the original pattern inputs, after training solely on the chaotically recalled patterns of a single experiment**. Note that the chaotically extracted patterns are displayed in figure 4.28. Note that the original pattern outputs are fairly maintained, and not only the principal elements, resulting in high readability, as opposed to in figure 4.30.

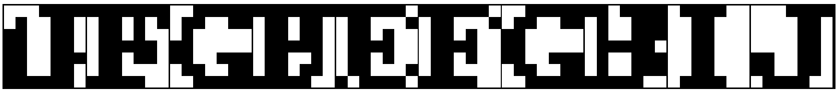

Figure 4.30: Displaying the **recalled output of the neocortical network after training on the original associative training pattern subsets**, 15 iterations per subset. Note that the first 3 out of 5 subset outputs are completely distorted, and do not resemble the desired output at all.

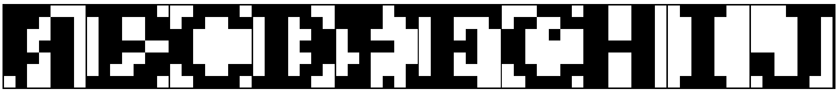

Figure 4.31: Displaying the **output of the neocortical network for the inputs of the 2x5 auto-associative training patterns** after training for 15 iterations on the chaotically recalled patterns in the case of the same model scheme as in figure 4.28 - however, here the **hippocampal model convergence criterion was set to** 50 **iterations during training**. It is worth emphasizing that the number of spurious patterns extracted by chaotic recall is 9, as opposed to in the former case, where it is 51 (chaotic patterns not displayed).

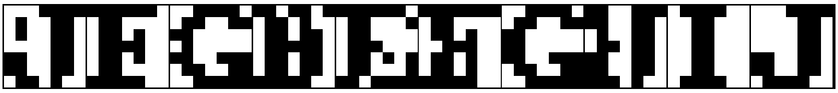

Figure 4.32: Displaying the **output of the neocortical network for the 2x5 auto-associative input patterns after training on the chaotically recalled patterns by the hippocampal model when *asynchronous* CA3-updating is used**. Note that 0 spurious patterns were recalled in this scheme, and that the pattern output of the neocortical recall seems to be erroneous for the two first subsets.

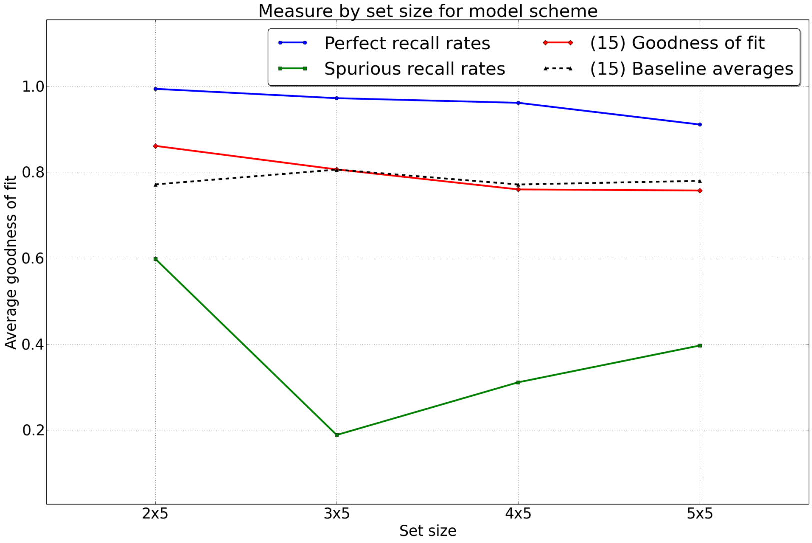

Figure 4.33: Displaying the **goodness of fit, perfect recall, and spurious recall rates** for the dual-network memory model when using synchronous CA3-layer updating, $\tau = 0.50$, DGW$= 25$, turnover for every training iteration, and the extracted output by chaotic recall as the training set (both input and outout) for the neocortical network.

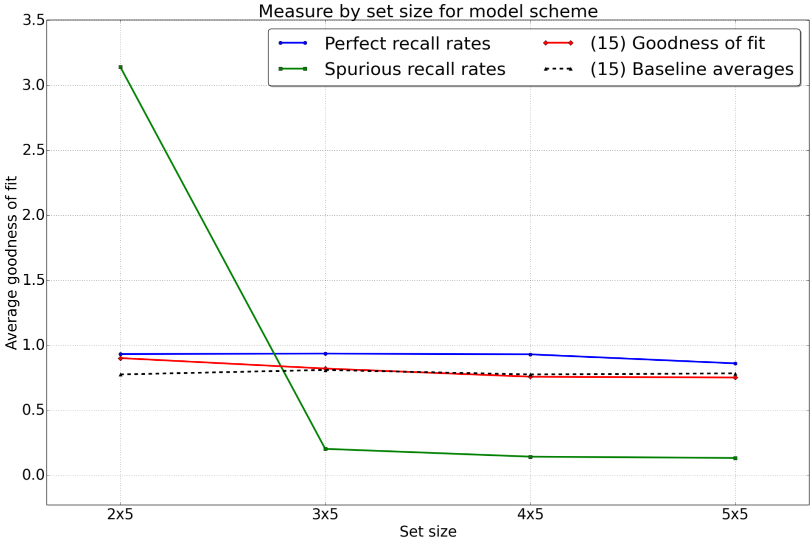

Figure 4.34: A figure of the **main measures** as plotted in figure 4.33, only that the number of **training iterations in the hippocampal module were set relative to the total set size**, by a factor of 1.5.

patterns being crucial to the successful reduction in catastrophic forgetting does not hold. However, the hippocampal model still separates patterns in weight space, resulting in less disruption of old patterns when learning new - i.e. successful reduction of catastrophic forgetting.

Notice that despite using no pseudorehearsal, model performance is significantly better than the model of (Hattori, 2010). This may be due to its lower hippocampal extraction rate, which then necessarily restricts the consolidation performance. Anyhow, it is noteworthy that the goodness of fit is only slightly lower than the conventional, which is achieved when employing pseudorehearsal. In fact, when globalising the training set constituted by all chaotically extracted pattern outputs, i.e. training over all of the extracted

Table 4.8: Showing the average goodness of fit for the best hippocampal model scheme, using only the chaotically recalled pattern outputs for memory consolidation in the novel scheme, along with the approximate goodness of fit from both (Hattori, 2014), and (Hattori, 2010), both employing pseudorehearsal.

	Perfect recall rates			
Pattern set size	2x5	3x5	4x5	5x5
Conventional model, (Hattori, 2010)	0.82	0.79	0.69	0.68
Conventional model, (Hattori, 2014)	0.97	0.95	0.91	0.89
Novel model, no pseudorehearsal	0.92	0.83	0.79	0.78

outputs, an average goodness of fit in the interval $(0.97, 1.0]$ is observed. This suggests that the hippocampal patterns contain some qualities which add a significant robustness to the learning process during memory consolidation, using gradient descent in the neocortical network (or FFBP ANN).

It was empirically observed that when creating multiple copies of the original training patterns, evenly copying them and permuting them only a little, the resulting goodness of fit when training the neocortical network using these patterns drops fairly quickly as the training set size increases. Furthermore, the goodness of fit also drops as the number of training iterations increases. This sensitivity to the number of training iterations is displayed when creating only two copies of each training pattern, for which 10% of the patterns are permuted and distorted. Note that this is in contrast to when training on the chaotically extracted patterns. Possible explanations for this observed emergent model behaviour under the chaotic output as input and output scheme includes those outlined above for extracting and encoding the principal components, as well as encoding the pattern recoding and separation in weight space into the pattern outputs. This may then be what is reflected in the successful significant reduction in catastrophic forgetting, displaying equal to or significantly better than baseline model performance. Where baseline is defined as the average goodness of fit attained when training the neocortical network on the subsets of the original training patterns only.

Note however that an increased sensitivity to the number of training patterns when copying and permuting the original training patterns is expected, because increasing the number of training iterations will 'overfit' the network fairly quickly to the actual permutation, as the model will regard these as undistorted, target patterns. In other words, using only few training iterations i, such as to $i = 15$, will tend to neglect some of the erroneous details of the permuted training patterns, possibly (and empirically on average) resulting in a better goodness of fit than when increasing the number of iterations to a number such as $i = 200$. Nevertheless, it is noteworthy that a greater number of patterns, acquired only by chaotic recall per subset, and trained only on subsets, successfully leads to attaining an equal or improved good goodness of fit, as well as a significant observed reduction in catastrophic interference.

4.3.5 Neocortical module results summary

A summary of the neocortical module experiments is presented in the list below:

- Solely using chaotically recalled patterns with the random input consolidates little or no information.

- Using the chaotically recalled patterns (output) as I/O to the neocortical network reduces catastrophic interference. This is likely due to the fact that the inclusion of spurious patterns in the extracted patterns by chaotic recall introduces reduction in overlap in the weight space for the consecutive training patterns and subsets, as neuronal turnover and pattern recoding constantly occurs. This is confirmed by the results from the global training schemes, as the performance remains approximately equal.

- Using the chaotic output as I/O in the global scheme globalizes the training set. This enables the FFBP ANN to extract the principal components, i.e. minimize the error-loss function in the weight-space spanning across all of the chaotic pattern outputs. This results in a performance (goodness of fit) near the case of simply iterating over the entire original training set.

- Using the chaotically extracted outputs from the most successful hippocampal model scheme as I/O-patterns to the neocortical network introduces a robustness to the learning procedure. This robustness is regarded as the insensitivity to the number of training iterations used, the fairly rapid convergence, as well as by allowing the long-term memory to learn only local subsets of the entire training set sequentially, yet disrupting former subsets fairly little. This is in contrast to when using the chaotically extracted patterns when using asynchronous CA3-updating in the hippocampal module, when training on the subsets of the original training set, or when training on the entire training set when only slightly permuting copies of the original patterns are regarded as the new training set.

Shortly put, the mechanism of using the chaotically extracted output patterns as both input and output to the neocortical network ameliorates catastrophic interference in the local training scheme. Using this novel memory consolidation scheme also introduces interesting and desired qualities to the dual-network memory model, such as eliminating the need for pseudorehearsal. Therefore, the novel memory consolidation scheme may be regarded as more biologically realistic. Furthermore, it introduces a robustness to the learning process in terms of insensitivity to the number of training iterations.

Chapter 5

Discussion

Summary

Building upon several experiments, a novel dual-network memory model is implemented. It is based upon Hattori's (2014) model, and attained through a novel model parametrization and memory consolidation scheme. Furthermore, the model employs synchronous neuronal activation value propagation and synaptic modification in all layers. The memory consolidation mechanism is implemented by using the chaotically extracted outputs of the hippocampal model as the training input and output to the neocortical neural network. The hippocampal model is comparatively speaking significantly better in performance relative to the state-of-the-art in terms of perfect recall, particularly for auto-associative patterns. I believe that by using the novel memory consolidation scheme, a more biologically realistic dual-network memory model is attained. Furthermore, the demonstration of reduction in catastrophic forgetting by employing the short-term memory constituted by the hippocampal model, *without* using pseudorehearsal in the neocortical module, is a novel emergent model quality. Based upon this quality, it is possible to contribute to discussions about aspects of memory consolidation, pattern recoding, and pattern separation in biological neural networks and corresponding models. While pseudopatterns transferring information as described in the literature review may describe a somewhat biologically plausible scenario for underlying mechanisms of memory consolidation in neural networks, it is not the aim in this thesis to replicate pseudorehearsal. Contrary, the implemented model contributes to the field by demonstrating information transfer from a biologically realistic network, namely the hippocampal model, to a more traditional network; the neocortical network. Furthermore, this is performed and demonstrated using a novel memory consolidation mechanism, without using pseudorehearsal. This novel memory consolidation scheme may be regarded as more biologically realistic than conventional consolidation mechanisms.

Research questions

- What are the limitations of pattern extraction in the hippocampal model?

The experiments and results of the previous chapter demonstrate that the hippocampal model may learn both auto-associative and hetero-associative training patterns equally well. This suggests that pattern separation is successful even for highly correlated patterns. However, only patterns with an input and output dimensionality of 49 were used to test the model, and as in all neural network models, a given topology and model will necessarily have a maximum capacity before its 'memory' is digested - a set of weights can only store a certain amount of information. Furthermore, an I/O-dimensionality of 49 greatly constrains the state space, as does having bipolar output values. That being said, the former chapter demonstrates that the model's memory capacity does not decline for the given training sets as the subset size increases, which is in contrast to when exposing different model parametrizations to the same training sets. However, when training on the complete sets of (global) patterns at once, such as 25 patterns simultaneously, the model's storage capacity is exhausted. That being said, it is still capable of having desired biologically resemblant qualities emerge, both for homogeneous and heterogeneous training sets.

- How does asynchronous or synchronous CA3-neuronal simulation affect hippocampal model behaviour?

  Asynchronicity introduces randomness to the model in its search, both during learning, and recall. This randomness might increase the model's separation abilities, as suggested by the results and figures of the previous chapter. However, this increased separation ability is shown to most likely come at the cost of an exponentially expanding state space. This results in successful convergence and a 100 % extraction rate for only the smallest of the set sizes, i.e. 2x5. A more desirable trade-off, which works more locally by recoding the synaptic connections of the incoming and outgoing connections for a subset of the neurons in the DG-layer, is demonstrated as being part of the most successful model scheme. It becomes evident that solely using the hippocampal network topology as outlined in this thesis does not suffice to successfully separate and recode patterns. The artificially drastically high neuronal turnover rate may reflect that a truly synchronous CA3-updating scheme is biologically implausible. Suggesting that a scheme in which subsets of neurons, constituting layers or local networks and operating according to rhythms, may be what is required in order to attain a more realistic and possibly improved simulation.

- How does neuronal turnover, and a synaptic weight coefficient for the outgoing synapses of the DG impact hippocampal model behaviour?

  Addressing the first; neuronal turnover turned out to be crucial in attaining all of the desired model qualities. This includes successful pattern separation, extraction of all patterns during perfect recall, as well as the successful extraction of patterns that may be used to reduce catastrophic forgetting during memory consolidation. The discovered mechanism with which patterns may be consolidated to the long-term memory without the use of pseudorehearsal, and yet significantly reduce catastrophic interference, is one of the main findings of this thesis. Furthermore, the DG-weighting is demonstrated to give rise to a performance nearly twice as high as when being set to 25, as opposed to being a neutral value of 1 in the synchronous CA3-neuronal updating scheme. Furthermore, it is demonstrated to be crucial with a stronger DG-weighting in order to have the neuronal turnover-associated behaviour and improved performance emerge.

- What information seems to be inherent in the patterns extracted by chaotic recall in the hippocampal module?

  Related to the former research question, the results attained in the previous chapter suggest that the hippocampal model, when being presented with random input, tends to converge towards outputs reflecting the functional mapping of the training patterns. This may possibly suggest a mechanism for compressing auto-associative patterns into a single layer's activity pattern, which may then be relayed to other parts of the model. Quite possibly also suggesting a mechanism which may be at play in biological neural networks, or which may be exploited in ANNs. Furthermore, catastrophic interference is significantly reduced by employing these extracted outputs. In fact, all of the original training patterns remain recognizable, unrelated to the number of neocortical training iterations (given that the number of iterations is above a certain low value such as 15), when only training the neocortical network on the chaotically extracted pattern outputs without using pseudorehearsal. This may further suggest that the extracted patterns are separated in weight space, due to the hippocampal model's recoding and separation qualities, which appears to be anchored in the neuronal turnover and topology of the DG-layer.

- Can the original training patterns be consolidated to the neocortical network by solely using chaotically recalled patterns?

  When considering the findings of the experiments and results, they demonstrate a clear and significant capability of memory consolidation and information transfer by solely employing chaotically extracted model outputs for the case of auto-associative training patterns. Thus, the scope remains that of auto-associative training patterns. It remains slightly obscure whether the entire memory model would easily generalise to hetero-associative patterns. Considering that the hippocampal model's extraction rate and performance did not decrease for hetero-associative training patterns, it may be hypothesised that it would generalise to this type of training patterns as well. The mechanism with which the pattern input could be obtained might be by bi-lateral hippocampal model action potential propagation. Another more realistic option is to extend the hippocampal model, such as by the inclusion of a CA1 layer and a relaying pathway from CA1 back to the EC. These may constitute pathways as well as a mechanism for relaying both auto-associative and hetero-associative patterns to the neocortical network.

- Can chaotic interference be reduced in the novel dual-network memory model without pseudorehearsal?

  All outputs of the neocortical model are recognized for the corresponding training inputs, and the average goodness of fit is above or equal to the baseline average, when training the neocortical model solely on the chaotically extracted patterns. Catastrophic forgetting is thus regarded as significantly reduced by employing the hippocampal model as a short-term memory. Pattern consolidation is performed by relaying the outputs extracted by chaotic recall in short-term memory as both

> input and output to the neocortical model. I am not aware of any previous findings demonstrating a reduction of catastrophic interference solely through the use of emergent neural network model mechanisms and behaviour such as that attained in the hippocampal model. It is without the use of pseudorehearsal in the long-term memory module that catastrophic interference and forgetting is reduced. Furthermore, the chaotic recall mechanism employed in the short-term memory resembles those believed to part-take in the biological brain during sleep. Namely random firing activity, which is thought to process the recently formed memories of among other parts; the hippocampus.

Limited training pattern dimensionality, a highly calibrated hippocampal model, and the fairly artificial memory consolidation mechanism, raise questions about the model generalizability. Further, they raise the question of whether it may provide a basis for drawing biological parallels. Presumably, the model is both incomplete and not biologically sound. This should be taken into account when considering the model results, as well as if using the research for further analyses. However, in algorithmically encoding some principles of biological neural network behaviour, the emergent behaviour may be regarded as somewhat generalisable.

Along the same lines comes the parametrization of the hippocampal model. In implementing and testing the model, some variables were set to constant values. These include among others the firing rates of the respective layers of the hippocampal model. Such values may usually be held as free variables, or tested for certain combinations of values, often designed with specific emergent network qualities in mind. However, this model configuration and these firing rates correlate with and are inspired by the literature: Both by similar computational models, and the biologically observed firing rates - which in fact have shown to be equal. While not necessarily justifying the lack of testing other firing rates, this justifies the elected firing rates, and enhances the model's biological plausibility.

When it comes to the parametrization of the neocortical model, the same configuration as implemented by Hattori (2010, 2014) is elected. This is in order to have a more accurate basis for comparing the results attained for memory consolidation.

Because the model is a connectionistic model leaning towards the branch of computational neuroscience, rather than industry-oriented deep learning, it becomes natural to address the question of biological plausibility. One of the main aspects which may be criticised in this regard, is the biological plausibility of using spurious patterns for memory consolidation. In particular, simply relaying the output of the hippocampal model as both input and output to the neocortical module remains a fairly artificial and simplistic approach. However, this may be regarded as an approximation to using actual neural network processes to relay the information. Furthermore, the attained results display a clear capability for reducing catastrophic forgetting *in the model*. It is worth mentioning in this regard that they are also statistically reliable, as they are average results of several trials displaying model trends.

Another aspect addressing the model's biological plausibility is employing k-WTA as a crude approximation to lateral inhibition. This is demonstrated within the literature to give rise to feature discovery and self-organization, as well as redundance reduction, which in turn are aspects thought to emerge partly from lateral inhibition (see chapter 2). Furthermore, while algorithms such as standard FFBP using gradient descent is generally not regarded as biologically plausible, it may perform similar types of orthogonalization. Thus, in this regard, gradient-descent may be regarded as approximating the aspect of orthogonalization, also performed by k-WTA. However, gradient-descent orthogonalizes data in a different manner; by minimizing an error-signal, separating patterns solely by weight adjustment according to error-minimization. k-WTA, however is in one sense less sensitive to the explicit output target distance, and may also be regarded as more dynamic in that it considers k values far more than the others. Note that as demonstrated in this thesis, combining k-WTA with weight re-instantiation may also introduce emergent behaviour of recoding which would not necessarily make sense using standard FFBP and gradient-descent. This is simply because k-WTA ensures a certain stability, while FFBP's signal usually operates on floating point numbers. Synaptic re-instantiation may however allow FFBP ANNs to traverse a local minima, although not necessarily resulting in an improved weight-configuration.

It is worth mentioning in relation to more traditional architectures that FFBP ANNs have recently been shown to be able to approximate biologically plausible neural network aspects under given constraints. An example being spike-timing-dependent plasticity, i.e. performing synaptic weight modification according to the relative timing of firings between neurons (Bengio et al., 2015).

Another example addressing the issue of the generalizability of the model behaviour revolves around the order of pattern extraction by chaotic recall. More specifically, note that the order of pattern extraction is not considered in the model and experiments. Nor is the subset which extracted patterns stem from. This poses a potential discrepancy in the extraction scheme, raising the question of whether these details may impact model behaviour, or create room for systematic errors. I allowed the order of extraction through chaotic recall to be neglected, because the output is likely to only reflect the current training subset, particularly in the i iterations training scheme. Furthermore, if reflecting former subsets, chances are that the current subset output will not have been extracted properly. This would in turn likely result in average model trends displaying catastrophic interference.

Future work

One of the issues related to the model of this thesis is the limited dimensionality of the input and output patterns which are used to test the hippocampal model. Regarding this limitation, there is in fact a fairly large body of evidence suggesting that the hippocampus most likely employs a distributed type of encoding. Such an encoding may result in that the capacity of patterns which it may store is exponential to the number of neurons in a layer. However, this does not imply that an exponential number of pattern associations may be stored, i.e. an exponential number of different stimulus-response patterns. In fact, this has been found to increase only linearly with the number of neurons in empirical studies (Rolls and Treves, 1998b). These are aspects that may be tested in future work on the model.

Another aspect which is more closely related to computational neuroscience that I would like to address further, is the transfer and learning mechanism which chaotic recall may constitute. While artificial neural networks do not usually process information using biologically plausible neural network models, the implemented hippocampal model does.

More specifically, the short-term memory employs Hebbian learning, and thus model behaviour such as pattern extraction may be regarded as fairly realistic at the network level. I would like to emphasize that the dual-network model constitutes in itself a mechanism for internal information transfer *from* a fairly biologically realistic network, to a more traditional one. I.e. the exposure of data to the LTM is through the pre-processing of the hippocampal model and STM, which additionally does not require pseudorehearsal in the neocortical model and LTM - at least not for data sets of constrained size, as demonstrated in this thesis. Therefore, I would like to further investigate extending the training sets which are used to analyse both pattern extraction in the STM, as well as memory consolidation to the LTM. This may then confirm or disconfirm a potential further extent of generalizability of the entire model.

Orthogonalization is a general machine learning mechanism which may lead to improved classification and/or categorization. By separating patterns from one another such that they are orthogonal in the input and output space, it is easier for auto-associative networks such as Hopfield networks, and similarly the CA3-layer of the hippocampal model, to separate these patterns from one another, and thus perform pattern completion which converges towards the actual undistorted target. Initial experiments show that a DG-weighting within the hippocampal model of this thesis approximately similar to that in (Wakagi, Yuko; Hattori, 2008), may improve model performance. More specifically, a weighting resulting in 25 times stronger connections between the DG- and CA3-layer, may yield better perfect recall rates and quicker convergence in the attained model. This suggests that pattern separation may be performed by the DG-layer, and confirms the hypothesis that it may be more strongly interconnected with the CA3-layer. This pattern separation mechanism is likely due to an orthogonalization, which occurs due to the pattern recoding, likely in turn caused largely by neuronal turnover. It would be interesting to further investigate the relationship between k-WTA, the expansion encoding of the DG-layer, and the neuronal turnover which it may perform, relative to the recoding and separation abilities of the layer. These may be considered relative to activity within the layer itself. Visualizations and/or layer-wise activation activity analyses are aspects that could be further addressed in this regard.

If the mechanism which enables a larger memory space in the hippocampal model is in fact pattern separation, this may be regarded as somewhat distributing the corresponding functional mapping in a diversifying manner. This may suggest that information transfer using chaotically extracted outputs in fact enable training the neocortical model on the consecutive knowledge separately in a way which interferes *less* with previous memories due to diversification in weight space. It may further be hypothesised that this mechanism participates in internal knowledge transfer and memory consolidation within brain-like structures. However, it remains unclear if this would suffice in explaining the capability of storing more information without disrupting old. Empirical observations of catastrophic interference occurring to some extent in the biological brain, may suggest that a trade-off is inevitable. However, this does not imply that it *must* occur, only that a type of it may occur in certain situations. Relating this to the artificial model of this thesis; as a network can only hold a certain amount of information, designing experiments exceeding this capacity will necessarily exhaust its memory, potentially resulting in catastrophic interference and forgetting. How and when such interference occurs in comparison with experiments performed on human subjects, should be addressed in future research. Using association training sets as outlined and used in this thesis provides the basis for such comparative analyses, where human subjects are taught pattern-associations. However, the generalizability of such comparisons remains somewhat limited, as these tasks may be highly context-dependent, and additionally harder to measure in human subjects. Nevertheless, whether a more explicit type of interleaving occurs in order to preserve former memories, remains slightly obscure, and also an aspect which I would like to address in my future work.

A major part of the work contained within this thesis is related to the hippocampus. One of the cognitive aspects constituted largely by this part of the brain is thought to be episodic memory. I would like to address some hypotheses that have emerged while working with material on the hippocampus, and relate these to the implemented model. Firstly, the attained model demonstrates mechanisms that may suggest biological schemes for memory consolidation, which provides the basis for drawing parallels to episodic memory. Episodic memory consists of several consecutive memories that are linked, and presumably recalled, through sequential evocation of the following memory. I.e. these patterns include temporal information in that they contain information which results in the recall of consecutive patterns, and thus hypothetically events. Cognition may be said to operate on a stream of input and output, relating a great deal of information temporally. In a simplified manner, the patterns of the model in this thesis may be regarded as integrating a temporal aspect through a single pattern-completion mechanism. This includes very little temporal information, but does provide the foundation for linking patterns in time. If extending the model such as proposed with respect to relaying hetero-associative patterns to the neocortical network model, this may in fact also provide the basis for a type of episodic memory. More specifically, relaying patterns back through the entire model would potentially provide the entire model with another layer of pattern-association, only on the level between patterns. Note that this may be performed in synthesis with the auto-association of the CA3-layer, and that input which is relayed back to the EC may then possibly change once the output has been recalled for the current specific and fairly precisely recalled output and former pattern. This may potentially result in a consecutive series of temporally linked patterns being recalled, or episodic memory and recall.

A further extension of this model could be to include spiking networks, as noted by Hattori (2014), in addition to the more sophisticated CA1-EC mapping, both which may be required in order to encompass episodic memory. It would indeed be interesting to analyse to what extent functional mappings including temporal correlations in data could be captured such an extended model. More specifically within the type of information transfer where the activity of the EC and CA1-layers are relayed to a neocortical network model. One clear issue related to this is the fact that an FFBP ANN using gradient descent simply learns single pattern associations without relating them temporally. It could be hypothesized that temporally related patterns are yet another type of compressed single pattern correlation. However, due to the temporal dimension, it appears more likely that both the STM and LTM models may require another level of complexity in order to capture such mappings. One suggestion for achieving this is simply to include GRUs in the neocortical network. Another is to modify the neocortical network such that it implements a type of topological recurrence, possibly enabling capturing temporal relations.

As neural activity is relayed from large parts of the brain to the hippocampus, this may enable the hippocampus to form memories which include information from all of these systems, i.e. to integrate across memories. Such memory integration is believed to be one of the fundamental functions performed by the hippocampus (Rolls and Treves, 1998a). In fact, I hypothesise that the hippocampus might solely and essentially perform mapping. Whether spatial, temporal, or any combination of those. For external and more sensory related data, for abstract patterns, as well as for internal representations and abstract patterns. Under the assumption that the hippocampus solely performs mapping, pattern association may be a central mechanism. Furthermore, being able to compress a pattern association into an activation pattern of excited neurons in a local network or layer, such as for the extracted chaotic outputs of the implemented model, may be thought of as a representation of this mapping. Extending this model and the model's mechanisms for pattern compression and association, and more explicitly investigating its relation to mapping, may therefore provide valuable insights into how high-level cognitive behaviour may be constituted. Furthermore, it may potentially provide the basis for a framework in which more sophisticated artificial intelligence may emerge. This is one of the main goals which I wish to pursue in my future work.

Before summarising the section on future work, I would like to describe a few biological parallels that I consider worth mentioning. The first is related to memory consolidation and dreaming in the biological brain. More specifically, the process of chaotic recall performed in the hippocampal model may be compared to the process thought to occur whilst dreaming. Note that these parallels may be questionable, as the hippocampal model's memory is quickly exhausted, requiring continuous consolidation or storage of extracted patterns. This raises questions both about how the model translates into these biological parallels, and how memory acquisition and storage may be constituted in such neural structures. It may be argued that due to the continuous learning process constantly occurring in the biological brain, it might not be entirely biologically implausible to extract chaotically recalled patterns continuously during learning. Furthermore, by re-formulating the learning and convergence criterion in the implemented model, a certain stability is maintained, and model trends are observable. This introduces the extraction of several spurious patterns, which is to be expected as a type of interpolation and prediction mechanism. Thus, while a certain stability is maintained, there is a more constant flow of chaotically recalled patterns which is generated and consolidated to the long-term memory model, which may be considered more biologically realistic than the former model.

In drawing parallels between dreaming and memory consolidation by chaotic recall, it should be noted that certain skills, or the enhancement of skills such as playing an instrument, may be observed after sleep. Concluding, chaotic recall might be a process consolidating memory to a more constant, long-term memory network, possibly enhancing the skills by associating the corresponding patterns. This may occur through episodic memory, or as a type of pattern-association compression which may be learnt by other neural structures. Interestingly, the attained pattern outputs extracted by chaotic recall in the hippocampal model may be regarded as a fingerprint in a hypothesized cognitive map, i.e. although seemingly spurious in fact containing a compressed form of pattern information. This may be a process which enhances, or supports the neural plasticity, although more precisely how remaining slightly obscure.

A central aspect in this thesis is switching between learning and recall in the hippocampal model. Interestingly, it was recently shown that the neurotransmitter acetylcholine may mediate learning in the hippocampus. Furthermore, these neurotransmitters are also demonstrated to cause inhibition of *dentate granule* cells, due to a GABAergic response (Pabst et al., 2016). In other words, the biological brain and hippocampus implements a type of depression of DG-cells, and may also use a type of switching between learning and recall through such neurotransmitters.

Another interesting aspect of note within biological parallels is the mutual inhibition simulated by k-WTA. This is mediated by inhibitory interneurons in the biological brain. It would be interesting to compare the neural network behaviour of model employing k-WTA, and in particular in the hippocampal model, with empirical data on network activity and mediation through neurotransmitters such as GABA, and acetylcholine. Furthermore, the stability criterion which initiates extraction of the model output, and thus implicitly memory consolidation, may be more explicitly compared with the stability of biological neural activity in vertebrates. Potentially, this may inspire adjusting the convergence criterion, or designing a novel implementation in which biological aspects are modeled more extensively.

To summarise, further research is needed on several of the dual-network memory model aspects. Designing and running further experiments is needed in order to test the current model more extensively, such as with respect to training pattern complexity and model behaviour. Furthermore, the model should be extended, including the synthesis of different network topologies and learning mechanisms in the hippocampal module. This includes implementing another layer of network recurrence in the hippocampal model, such as previously outlined by implementing a CA1-layer which may relay its activation values back to the EC. One aim of investigating this model extension is to further illuminate the information transfer mechanism constituted by the chaotically extracted patterns of the hippocampal model, along with the qualities of the patterns. These may potentially be used to develop more sophisticated or intertwined neural network algorithms and models, such as by implementing the novel memory consolidation scheme for hetero-associative patterns, and to draw further neuroscientific parallels.

References

Ans, B. and S. Rousset (1997). Avoiding catastrophic forgetting by coupling two reverberating neural networks. *Comptes Rendus de l'Académie des Sciences-Series III-Sciences de la Vie 320*(12), 989–997.

Ans, B. and S. Rousset (2000). Neural networks with a self-refreshing memory: Knowledge transfer in sequential learning tasks without catastrophic forgetting.

Bar-yam, Y. (1997). Dynamics of Complex Systems. In *Methods* (First Edit ed.)., Chapter 0 Overview, pp. 1–15. Westview Press.

Barnes, J. M. and B. J. Underwood (1959). Fate of first-list associations in transfer theory. *Journal of experimental psychology 58*(2), 97–105.

Bengio, Y., D.-h. Lee, J. Bornschein, and Z. Lin (2015). Towards Biologically Plausible Deep Learning. *arXiv*.

Bergstra, J., O. Breuleux, F. F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio (2010). Theano: a CPU and GPU math compiler in Python. *Proceedings of the Python for Scientific Computing Conference (SciPy)* (Scipy), 1–7.

Byrne, J., J. H. Byrne, and J. L. Roberts (2014a). *From Molecules to Networks: An Introduction to Cellular and Molecular Neuroscience* (Third edit ed.)., Chapter 1. Cellular Components of Nervous Tissue, pp. 11. London: Academic Press, Elsevier Inc.

Byrne, J., J. H. Byrne, and J. L. Roberts (2014b). *From Molecules to Networks: An Introduction to Cellular and Molecular Neuroscience* (Third edit ed.)., Chapter 1. Cellular Components of Nervous Tissue, pp. 26. London: Academic Press, Elsevier Inc.

Campbell, N. A. and J. Reece (2015). *Biology* (Tenth edit ed.)., Chapter 9: Cellular Signaling, pp. 216–237. Edinburgh Gate, Harlow, Essex CM20 2JE, England: Pearson Education Limited.

Carpenter, G. A. and S. Grossberg (1987). ART 2: self-organization of stable category recognition codes for analog input patterns. *Applied optics 26*(23), 4919–4930.

French, R. (1994). Dynamically constraining connectionist networks to produce distributed, orthogonal representations to reduce catastrophic interference. *Network 1111*, 00001.

French, R. M. (1992). Semi-distributed Representations and Catastrophic Forgetting in Connectionist Networks. *Connection Science 4*(3-4), 365–377.

French, R. M. (1997). Pseudo-recurrent Connectionist Networks: An Approach to the 'Sensitivity-Stability' Dilemma. *Connection Science 9*(4), 353–380.

French, R. M. (1999). Catastrophic forgetting in connectionist networks: Causes, consequences and solutions. *Trends in Cognitive Sciences 3*(4), 128–135.

French, R. M., B. Ans, and S. Rousset (2001). Pseudopatterns and dual-network memory models : Advantages and shortcomings. *Connectionist Models of Learning, Development and Evolution*, 13–22.

Hafting, T., M. Fyhn, S. Molden, M. Moser, and E. I. Moser (2005). Microstructure of a spatial map in the entorhinal cortex. *Nature 436*(7052), 801–806.

Hattori, M. (2010). Dual-network memory model using a chaotic neural network. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–5.

Hattori, M. (2014). A biologically inspired dual-network memory model for reduction of catastrophic forgetting. *Neurocomputing 134*, 262–268.

Hebb, D. O. (1949). *The organization of behavior: a neuropsychological theory*. New York: JOHN WILEY & SONS, Inc.

Hertz, J., A. Krogh, and R. G. Palmer (1991). *Introduction to the Theory of Neural Computation*, Volume 1. Westview Press.

Hinton, G. E. (1989). Deterministic Boltzmann Learning Performs Steepest Descent in Weight-Space.

Hochreiter, S. and J. Schmidhuber (1997). Long Short-Term Memory. *Neural Computation 9*(8), 1735–1780.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America 79*(8), 2554–2558.

Huang, D. A. and Y. C. F. Wang (2013). Coupled dictionary and feature space learning with applications to cross-domain image synthesis and recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2496–2503.

Kiernan, M. C. (2011). Freud, neurology and the emergence of dynamic neural networks. *Journal of neurology, neurosurgery, and psychiatry 82*(2), 120.

Kreyszig, E. (2011). *Advanced Engineering Mathematics* (10th ed.)., Chapter 20.4, pp. 864–871. JOHN WILEY & SONS, Inc.

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, and C. J. C. Burges, and L. Bottou and K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. Curran Associates, Inc.

Kruschke, J. K. (1992). ALCOVE: an exemplar-based connectionist model of category learning. *Psychological review 99*(1), 22–44.

Langton, C. G. (1990). Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D: Nonlinear Phenomena 42*(1-3), 12–37.

LeCun, Y., Y. Bengio, and G. Hinton (2015). Deep learning. *Nature 521*(7553), 436–444.

LISA-lab (2015a). Installing theano. Internet www page at URL: `http://deeplearning.net/software/theano/install.html#install` (accessed 15/12/2015).

LISA-lab (2015b). Theano 0.7 documentation. Internet www page at URL: `http://deeplearning.net/software/theano/index.html#documentation` (accessed 15/12/2015).

Marr, D. (1971). Simple memory: a theory for archicortex. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences 262*(841), 23–81.

McClelland, J. L., B. L. McNaughton, and R. C. O'Reilly (1995). Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review 102*(3), 419–457.

McCloskey, M. and N. J. Cohen (1989). Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. *Psychology of Learning and Motivation - Advances in Research and Theory 24*(C), 109–165.

McCulloch, W. S. and W. Pitts (1943). A Logical Calculus of the Idea Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics 5*, 115–133.

Mnih, V., K. Kavukcuoglu, D. Silver, A. a. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis (2015). Human-level control through deep reinforcement learning. *Nature 518*(7540), 529–533.

Newman, M. E. J. (2003). The structure and function of complex networks. *arXiv:Cond-Mat/0303516 45*(2), 167–256.

Norman, K. A. and R. C. O'Reilly (2003). Modeling hippocampal and neocortical contributions to recognition memory: a complementary-learning-systems approach. *Psychol Rev 110*(4), 611–646.

O'Keefe, J. (1976). Place units in the hippocampus of the freely moving rat. *Experimental Neurology 51*(1), 78–109.

O'Keefe, J. and N. Burgess (1996). Geometric determinants of the place fields of hippocampal neurons. *Nature 381*, 425–428.

Pabst, M., O. Braganza, H. Dannenberg, W. Hu, L. Pothmann, J. Rosen, I. Mody, K. vanLoo, K. Deisseroth, A. Becker, S. Schoch, and H. Beck (2016). Astrocyte Intermediaries of Septal Cholinergic Modulation in the Hippocampus. *Neuron 90*, 853–865.

Ratcliff, R. (1990). Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review 97*(2), 285–308.

Robins, A. (1995). Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science 7*, 123–146.

Robins, A. (1996). Consolidation in Neural Networks and in the Sleeping Brain. *Connection Science 8*(2), 259–276.

Rolls, E. T. and A. Treves (1998a). *Neural Networks and Brain Function*, Chapter 1: Introduction, pp. 1–22. Oxford, UK: Oxford University Press.

Rolls, E. T. and A. Treves (1998b). *Neural Networks and Brain Function*, Chapter 6: The hippocampus and memory, pp. 95–135. Oxford, UK: Oxford University Press.

Rolls, E. T. and A. Treves (1998c). *Neural Networks and Brain Function*, Chapter 4: Competitive networks, including self-organizing maps, pp. 54–74. Oxford, UK: Oxford University Press.

Rolls, E. T. and A. Treves (1998d). *Neural Networks and Brain Function*, Chapter 3: Autoassociation memory, pp. 42–53. Oxford, UK: Oxford University Press.

Rolls, E. T. and A. Treves (1998e). *Neural Networks and Brain Function*, Chapter 2: Pattern association memory, pp. 23–41. Oxford, UK: Oxford University Press.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Volume 1, pp. 318–362. US, Cambridge MA 02142-1209: The MIT Press.

Russell, S. and P. Norvig (2009a). *Artificial Intelligence: A Modern Approach* (3rd editon ed.)., Chapter 18: Learning From Observations, pp. 1152. Essex CM20 2JE: Pearson Education Limited.

Russell, S. and P. Norvig (2009b). *Artificial Intelligence: A Modern Approach* (3rd editon ed.). Essex CM20 2JE: Pearson Education Limited.

Shapira, B., L. Rokach, and S. Freilikhman (2013). Facebook single and cross domain data for recommendation systems. *User Modeling and User-Adapted Interaction 23*(2-3), 211–247.

Sun, Y., X. Wang, and X. Tang (2014). Deep Learning Face Representation by Joint Identification-Verification. *Nips*, 1–9.

Tani, J. (2014). Self-Organization and Compositionality in Cognitive Brains : A Neurorobotics Study. *Proceedings of the IEEE 102*(4), 586–605.

Urmson, C., J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. N. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. M. Howard, S. Kolski, A. Kelly, M. Likhachev, M. McNaughton, N. Miller, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, B. Salesky, Y. W. Seo, S. Singh, J. Snider, A. Stentz, W. Whittaker, Z. Wolkowicki, J. Ziglar, H. Bae, T. Brown, D. Demitrish, B. Litkouhi, J. Nickolaou, V. Sadekar, W. Zhang, J. Struble, M. Taylor, M. Darms, and D. Ferguson (2009). Autonomous driving in Urban environments: Boss and the Urban Challenge. In *Springer Tracts in Advanced Robotics*, Volume 56, pp. 1–59.

Wagle, K. (2013). IBM Watson: Revolutionizing healthcare? *Young Scientists Journal 6*(13), 17–19.

Wakagi, Yuko; Hattori, M. (2008). A Model of Hippocampal Learning with Neuronal Turnover in Dentate Gyrus. *INTERNATIONAL JOURNAL OF MATHEMATICS AND COMPUTERS IN SIMULATION 2*(2), 215–222.

Appendices

Appendix A

Artificial Neural Network Concepts

Formal Outline

Mathematically speaking, ANNs may be outlined as follows;
a network consists of a set S of N nodes. Furthermore, for every node $i \in S$: i may be connected to a node $j \in S$.
For every such connection, there exists a weight, $\omega_{i,j} \in \Re$, the sub-script denoting a connection weight from node i to node j.
A transfer function f is a function $f(\theta)$ of a neuron's total input, θ. A commonly used transfer function is the sigmoid transfer function, simply being

$$f(\theta) = \frac{1}{1 + e^{-\theta}} \tag{5.1}$$

Neuronal activation u_j of a node j may be written as,

$$u_j = f(\theta_j) \tag{5.2}$$

where

$$\theta_j = \sum_{i \in M} u_i \omega_{i,j} \tag{5.3}$$

and M is the set of all nodes that have incoming connections to neuron j, $M \in S$, and u_i is the activation value of node i. This is the principle which is used during the feed-forward phase in an FFBP ANN. In other words; the presented input is propagated throughout the network by calculating activation values for all nodes in the input layer, which then flows through the rest of the nodes in the network in the same manner until finally arriving at the output nodes.

Back-propagation

One way of updating the weights in an ANN is as previously mentioned by using back-propagation (BP) of an error signal. This may described as a fairly straight-forward algorithm for finding sub-optimal or optimal weights in an ANN. This is achieved by minimizing an error-signal which is back-propagated from the output node(s) of the network. Because the generation of an error signal requires an input pattern to be fed forward throughout a network, these two steps are commonly referred to together as feed-forward back-propagation (FFBP). Note, however, that the algorithm does not guarantee convergence towards a global optimum, as it is a gradient-based method, which traverses the weight space constituted by minimizing an error signal for a neural network. See figure 5.1 for an illustration of this. An analogy to the problem of convergence towards a local optimum is simulated annealing, which runs the risk of being stuck in a local optimum of the temperature cools down too quickly. However, with just enough temperature and movement, or ”jiggle”, the algorithm may be able to continue its traversal, possibly finding a more optimal solution.

Mathematically, the back-propagate algorithm requires us to be able to calculate a gradient $\Delta\omega$ for each weight $\omega \in \Omega$, where Ω is the weight space for the network. Arriving at a given output for a given FFBP ANN through feed-forward propagation using the equations (5.1), (5.3), (5.2), the squared error may be expressed as,

$$\mathbf{E} = \frac{(\mathbf{d} - \mathbf{o})^2}{2}, \tag{5.4}$$

where **d** is the desired output vector for all output nodes. Dividing by two to account for using two data points in finding the squared error.

This may then be used to calculate a gradient that may be used in updating every weight between the output layer and the preceding layer in the ANN,

$$\omega_{t+1}^{i,j} = \omega_t^{i,j} + \Delta\omega_t^{i,j}, \tag{5.5}$$

In order to perform a weight change in the direction of minimizing the error loss function **E**, the partial derivative of **E** w.r.t. the weight $\omega_{i,j}$ is used,

$$\Delta\omega_{i,j} = -\alpha \frac{\partial \mathbf{E}}{\partial \omega_{i,j}}, \tag{5.6}$$

where α is a learning rate parameter. Note that the sub-script denoting time is dropped for convenience. The negative is used in order to adjust for the error. Despite the fact that BP does not guarantee convergence towards the global optimum (here minimum), it can be shown that for a sufficiently fine-grained step-parameter (i.e. learning rate), convergence towards a local optimum can be guaranteed. This is due to the nature of the search space, which is continuous and differentiable, but may contain ridges and local minima in terms of the squared error, **E**. However, the smaller the learning rate α, the slower the convergence. Furthermore, for too low an α, the gradient's ”reach” will also decrease, making it more prone to small stationary points in the weight space. In other words, a learning rate parameter which will converge at a ”fair” rate towards the optimum is preferable. This rate is domain specific. Illustratively, if α is too large, the algorithm is chaotic, resulting in divergence when using gradient descent in weight space. If it is too small, the algorithm is very stable and unable to traverse large parts of the weight space. Lastly, if α is of a preferable size, the algorithm is at the edge of chaos - able to traverse larger parts of the weight space, yet eventually converging towards a solution. See figure 5.1 for an illustration of this.

Using the chain rule, one may obtain the weight change update in a given layer as (see below in chapter 5 for a derivation),

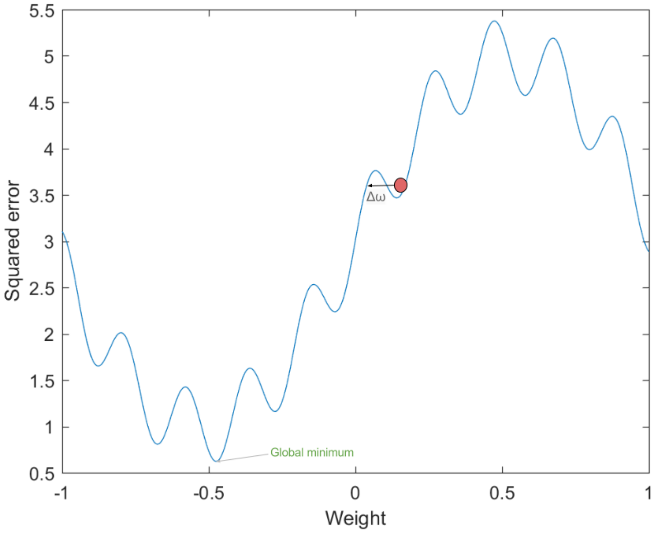

Figure 5.1: Illustrating the error-landscape formed by finding the weights in an ANN that minimize the sum of squared errors, shown for one particular weight and its associated squared error. When traversing the landscape formed by **E**, local minima may be encountered (in between ridges). An analogy which is often used in illustrating this is a ball rolling down the hills because of kinetic energy imposed on it by gravity. In this analogy α would in a sense be friction, constraining the work of gravity and resulting in a smaller or larger velocity, i.e. step sizes $\Delta\omega$ for each iteration of the algorithm. Too small an acceleration may result in the ball stopping when encountering a ridge. In the event of this being a local minimum, we would of course prefer for the ball to have enough kinetic energy to traverse the ridge and fall into a lower valley. Therefore, just enough acceleration to traverse the local minima is preferable. However, using too high an acceleration, the ball might not settle into any attractor at all. Possibly the weights will even diverge, with the ball being launched into outer space, sealing its fate to never again return to the hillside.

$$\frac{\partial \mathbf{E}}{\partial u_j}\frac{\partial u_j}{\partial \theta_j} = (\sum_{l \in L} \frac{\mathbf{E}}{\partial u_l}\frac{\partial u_l}{\partial \theta_l}) f(\theta_j)(1 - f(\theta_j), \tag{5.7}$$

which accounts for the weight change updates of the preceding layers too, the first layer being,

$$\tfrac{\partial \mathbf{E}}{\partial u_l}\tfrac{\partial u_l}{\partial \theta_l}\omega_{j,l} = u_l(u_l - d_l)\omega_{j,l},$$

Making it possible to obtain the partial derivatives recursively by starting at the output layer and back-propagating the values into the partial derivatives for **E** for every weight $\omega_{i,j}$.

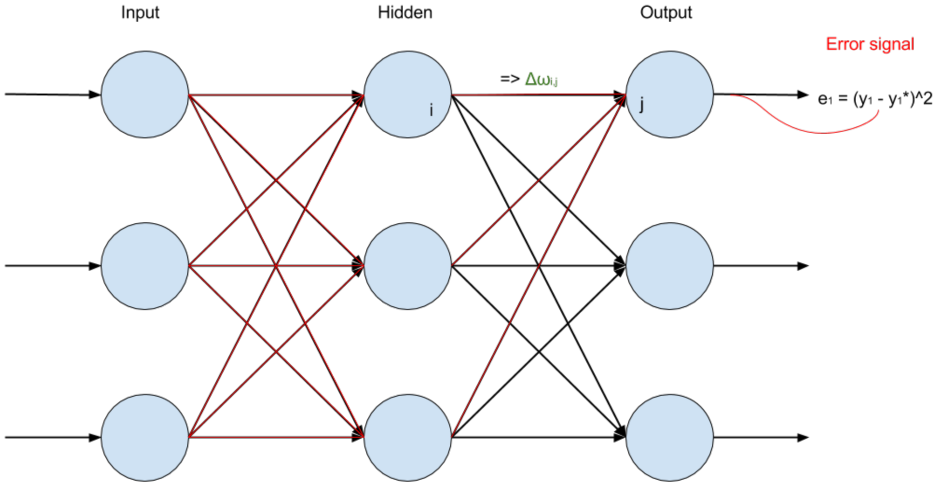

Figure 5.2: Illustrating the back-propagation of an error-signal, with the error signal from every output node having an impact on the resulting weight gradient that is calculated for every weight in the network. Note that the red lines are the lines symbolize the propagation of the error signal from output node j to all of its predecessors. y_1 is the obtained output, whereas y_1^* is the desired output, and e_1 is the squared error for the output node.

Back-propagation Through Time

The algorithm of section 5 may be extended by taking into account the k last time-steps of training in a network. Implementations may vary, the key here being a time-dependence. By introducing a measure which averages over the former weights, it turns out that convergence may be faster than when only using BP (as opposed to here BPTT). An alternative approach is to include k, or all previous weights, discounting the impact each former weight has exponentially as a function of time.

$$\omega_{t+1} = \textstyle\sum_{i=0}^{k} \gamma^i \omega_{t-i}, \gamma \in (0,1)$$

Derivation of the Back-propagation Rule

The following derivation is based primarily on the derivation of Rumelhart et al. (1986), as well as on that of Russell and Norvig (2009b). Using the chain rule, one may formally derive $\Delta\omega$ the following way,

$$\frac{\partial \mathbf{E}}{\partial \omega_{i,j}} = \frac{\partial \mathbf{E}}{\partial u_j} \frac{u_j}{\theta_j} \frac{\theta_j}{\omega_{i,j}}, \tag{5.8}$$

where the partial derivative w.r.t. the weight between nodes i and j will be cancelled out for all nodes other than j. Formally,

$$\tfrac{\partial \theta_j}{\partial \omega_{i,j}} = \tfrac{\partial}{\partial \omega_{i,j}} (\textstyle\sum_{k \in M} \omega_{k,j} o_k),$$

$$\tfrac{\partial \omega_{k,j} o_k}{\partial \omega_{i,j}} = 0 : k \neq i, \implies$$

$$\frac{\partial \theta_j}{\partial \omega_{i,j}} = u_i. \tag{5.9}$$

$$\frac{\partial u_j}{\partial \theta_j} = \frac{\partial}{\partial \theta_j} f(\theta_j) = f(\theta_j)(1 - f(\theta_j)) \tag{5.10}$$

$$\frac{\partial \mathbf{E}}{\partial u_j} = \sum_{l \in L} (\frac{\partial \mathbf{E}}{\partial \theta_l} \frac{\partial \theta_l}{\partial u_j}) = \sum_{l \in L} (\frac{\partial \mathbf{E}}{\partial u_l} \frac{\partial u_l}{\partial \theta_l} \omega_{j,l}), \tag{5.11}$$

where L is all nodes to which node j is connected - i.e. the set of nodes with *outgoing* links from j. From this it can be seen that when l is an output node,

$$\tfrac{\partial \mathbf{E}}{\partial u_l} \tfrac{\partial u_l}{\partial \theta_l} \omega_{j,l} = u_l(u_l - d_l)\omega_{j,l},$$

Making it possible to obtain the partial derivatives recursively by starting at the output layer and, surprisingly, back-propagating the values into the partial derivatives for $\mathbf{E}$ for every weight $\omega_{i,j}$.

In other words, the weight change update in a given layer accounts for the weight change updates of the preceding layers too. Formally,

$$\frac{\partial \mathbf{E}}{\partial u_j} \frac{\partial u_j}{\partial \theta_j} = (\sum_{l \in L} \frac{\mathbf{E}}{\partial u_l} \frac{\partial u_l}{\partial \theta_l}) f(\theta_j)(1 - f(\theta_j). \tag{5.12}$$

The Vector L2-norm

For a vector $\mathbf{v} = \begin{bmatrix} x_1 & x_2 & ... & x_n \end{bmatrix}^T$, the l^2-norm of the vector is defined as,

$$|\mathbf{v}| = \sqrt{\sum_{i=1}^{n} x_i^2}, \tag{5.13}$$

for real-valued vectors (Kreyszig, 2011).

Appendix B

Theano

To further demonstrate the flexibility of using symbolic expressions, an example of calculating numbers of the Fibonacci sequence is included below (imports as above being presumed):

```
def fib_acc(old, older):
    return old + older

fib_expr, updates = theano.scan(
    fn=fib_acc,
    sequences=None,
    outputs_info=[dict(initial=np.int32([0, 1]),
    taps=[-1, -2])], n_steps=n)

f_fib_scan = theano.function([n], fib_expr)
```

Note that the recursive definition is captured by theano.function. When theano.function is called, the first parameter it takes is the input parameters for the function that is to be calculated. Following the input parameters are the output parameters which are the expressions to be calculated. It is further possible to specify among other expressions the *updates* for new shared variables. By providing 'updates' = [variables] to the function, the variables will be kept in the shared variable *updates*. Furthermore, *givens* may be supplied for substitutions in the computational graph (i.e. if a function should be substituted with a given value such as a constant). In order to loop over a function, the *scan*-function of the Theano-library may be used. Scan performs an optimized looping over a symbolic graph, letting the user define parameters for optimizing the iteration. For instance it lets the user tell Theano that some variables do not need to be stored, thus the compiler re-uses a register for the variable as it is updated, potentially speeding up the loop. Scan may further be used to dramatically speed up matrix-vector multiplication. Note that Theano may take a normal Python function-object as an input function to the scan-operator. Standard Python code of the f_fib_scan loop would be to call fib_acc(...) in a for-loop. Furthermore, scan lets the programmer explicitly define recursive relationships for output expressions that are to be computed using the taps-variable, which here states that the two former variables are to be kept in memory. Note that this requires one to define the initial values of the parameters. Theano supports both exclusive and inclusive scan (appending one element at a time for input to the binary function as opposed to supplying all combinations of the input).

Appendix C

Architectural Overview

The entire code base of the implementation is contained in the public git-repository `https://github.com/williampeer/DeepBytes`. Furthermore, the repository contains a short 'readme' for how to setup the system such that the model and experiments may be run in Ubuntu 14.04 LTS.

Hippocampal code example

```
import theano
import theano.tensor as T
import numpy as np
import Tools

class HPC:
    # the parameters being omitted for simplicity
    def __init__(self, ..parameters):
        # variables are bound to the object instance (...)

        # ============== SETUP ==================
        input_values = np.zeros((1, dims[0]), dtype=np.float32)
        self.input_values = theano.shared(name='input_values',
            value=input_values.astype(theano.config.floatX),
            borrow=True)

        # the remaining activation values are bound (...), and
        #   weight matrices are setup similarly:
        input_ec_weights = Tools.binomial_f(dims[0], dims[1],
            self.connection_rate_input_ec)
        self.in_ec_weights = theano.shared(name='in_ec_weights',
            value=input_ec_weights.astype(theano.config.floatX),
            borrow=True)

        # (...)

        # Theano functions are symbolically defined and bound to
        #   the object instance, such as:
        local_in_vals = T.fmatrix()
        local_in_ec_Ws = T.fmatrix()
        next_activation_values_ec = T.tanh(local_in_vals.dot(\
            local_in_ec_Ws) / self._epsilon)
        self.propagate_input_to_ec = theano.function(\
            [local_in_vals, local_in_ec_Ws], outputs=None,
            updates=[(self.ec_values, next_activation_values_ec)])

        # wire after kWTA for this layer
        u_prev_reshaped_transposed = T.fmatrix(\
            ’u_prev_reshaped_transposed’)
        u_next_reshaped = T.fmatrix(’u_next_reshaped’)
        Ws_prev_next = T.fmatrix(’Ws_prev_next’)
        # Element-wise operations. w_13_next = w_13 +
            nu u_3(u_1-u_3 w_13).
        next_Ws = Ws_prev_next + self._nu * u_next_reshaped * \
            (u_prev_reshaped_transposed.T - u_next_reshaped * \
            Ws_prev_next)
        self.wire_ec_dg = theano.function([u_prev_reshaped_transposed,
            u_next_reshaped, Ws_prev_next],
            updates=[(self.ec_dg_weights, next_Ws)])

        # wrapper method for learning:
        def learn(self):
            self.propagate_input_to_ec(self.in_values.get_value(\
                return_internal_type=True), self.in_ec_weights.
                    get_value(return_internal_type=True))

            self.set_ec_values(kWTA(self.ec_values.get_value(...)))

            # -> repeat for all layers, using a different function as
            #       outlined for the CA3-layer in Chapter 3.
```

Note that for retrieval of shared variable values in the above code example, I return the internal value - this is to improve efficiency solely, and should not be prone to aliasing with the current implementation.

k-WTA

k-Winners-Takes-All is implemented very much like the following pseudocode:

```
def kWTA(activation_values):
    sorted_values = activation_values.sort()  # ascending
    threshold = (sorted_values[k-2] + sorted_values[k-1]) / 2
    return filter(activation_values, threshold)
```

One addition being the edge case of when all activation values are the same value. Even though this should not occur - the algorithmic edge case is handles by setting k nodes to 1 at random, the remaining to 0. If only a subset of the nodes have the same value, yet having the number of nodes that are above the threshold exceed k; nodes are drawn at random from this subset and set to 0 until the number of nodes that are 1 correspond exactly to k. I.e. the non-optimized pseudo-code may be written the following way,

```
if(numpy.sum(kWTA_arr) > k):
    while numpy.sum(kWTA_arr) > k:
        kWTA_arr = remove_random_of_smallest(kWTA_arr)
```

Learning for *i* iterations

Following is the pseudocode implementing training of the hippocampal module on a set of training patterns for *i* iterations,

```
def learn_patterns_for_i_iterations_hpc_wrapper
    (hpc, patterns, num_of_iterations):

for i in range(num_of_iterations):
    if hpc._TURNOVER_MODE == 1:
        neuronal_turnover_helper(hpc)

    for [input_pattern, output_pattern] in patterns:
        hpc.setup_pattern(input_pattern, output_pattern)
        hpc.learn()

Tools.append_line_to_log("Learned for " + str(num_of_iters) +
    " iterations. Turnover: " + str(hpc._turnover_rate) + ".
    DG-weighting: " + str(hpc._weighting_dg))
```

Appendix D

Additional experiment results

Included in this appendix are figures created from the experiment results of the experiments on the DG-weighting, and the experiments on the neuronal turnover rate, τ, as referred to in chapter 4. Additionally, the figures of training on the second subset in the low-level demonstration are included.

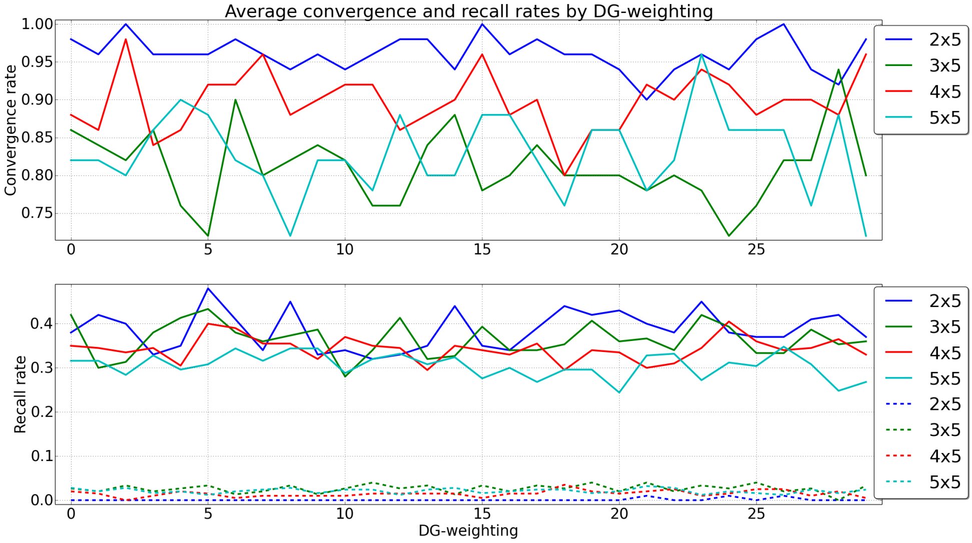

Figure 5.3: Showing the results attained for **convergence and recall rates by the DG-weighting** when using **synchronous** CA3-layer updating and neuronal turnover for **every training iteration**, $\tau = 0.04$.

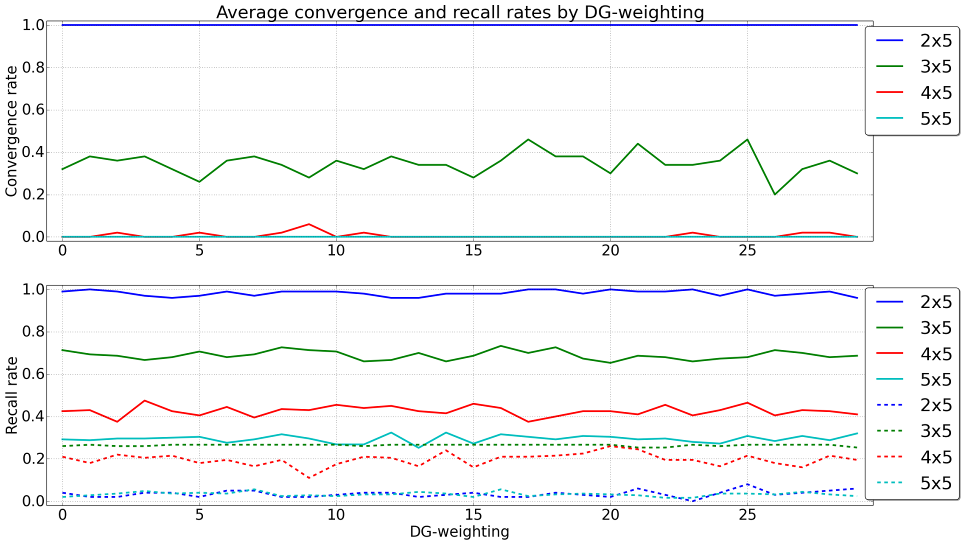

Figure 5.4: Showing the results attained for **convergence and recall rates by the DG-weighting** when using **asynchronous** CA3-layer updating and neuronal turnover for **every training iteration**, $\tau = 0.04$.

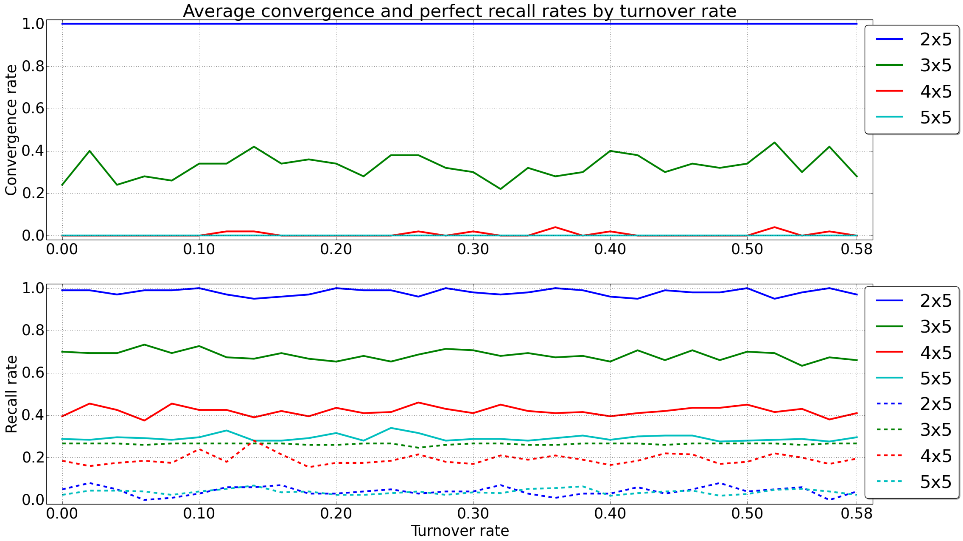

Figure 5.5: Displaying the average **convergence- and perfect and spurious recall rates by the neuronal turnover rate** for **asynchronous** CA3-updating, using a **DG-weighting of 1**, and turnover between **every learnt training subset**, with $\tau = 0.50$.

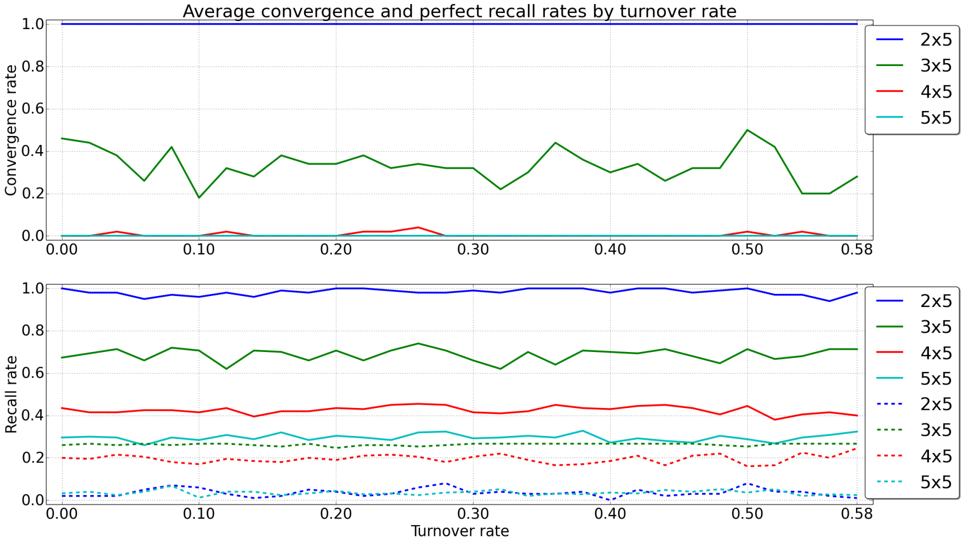

Figure 5.6: Displaying the average **convergence- and perfect and spurious recall rates by the neuronal turnover rate** for **asynchronous** CA3-updating, using a **DG-weighting of 1**, and turnover for **every training iteration**, $\tau = 0.04$.

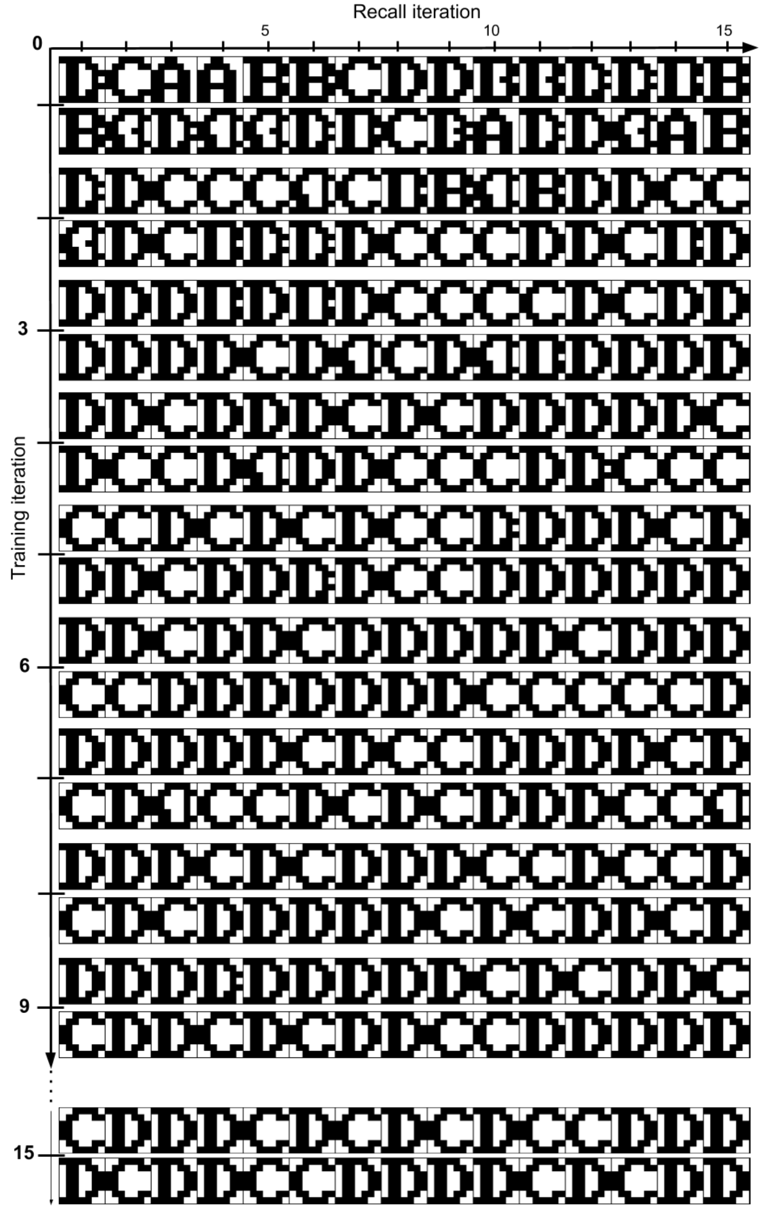

Figure 5.7: Displaying the **recall for the two patterns inputs of the training set for each training iteration**. Output is displayed in two rows per training iteration, corresponding to the training inputs 'C' and 'D', consecutively. Note that the origin is in the upper left corner. Furthermore, the model setup which is used is: **Asynchronous** CA3-layer updating, neuronal turnover between **every learnt subset**, i.e. not throughout the results displayed in this graph, a **DG-weighting of 1**, and a turnover rate $\tau = 0.50$.

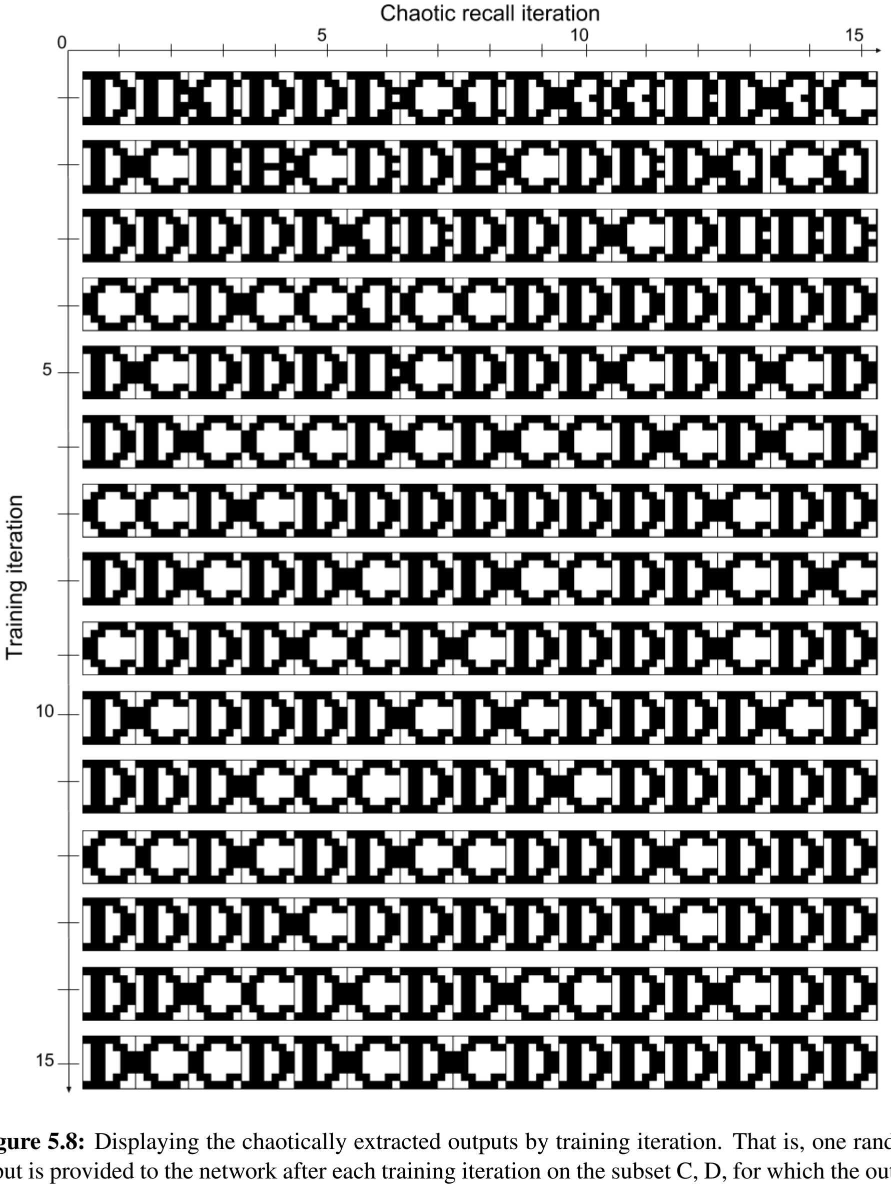

Figure 5.8: Displaying the chaotically extracted outputs by training iteration. That is, one random input is provided to the network after each training iteration on the subset C, D, for which the output is displayed after every recall iteration. The model scheme corresponds to that above in figure 5.7, using **asynchronous** CA3-layer updating.

Appendix E

Can a mechanistic neuron-level understanding of some aspects of cognition be attained?

The included essay was written by me in association with the course "Neural Computation: Models of Brain Function", at University College London, UCL, the autumn term of 2014.

The brain exists in the physical reality, and cognition emerges from the physical processes occurring in the brain. If a sound and complete framework describing the physical processes down to a minuscular level can be attained, then it is theoretically possible to derive cognition from the axioms of this framework. An analogy to this is how chemistry can be derived from physics. Now, the axioms of the physical processes of the brain may not be determined with complete certainty, but they can be hypothesized, as in physics. It is not yet known what level of detail is required in order to fully encompass cognition. So can a mechanistic neuron-level understanding of some aspects of cognition be attained? Some claim that this level of abstraction does not encompass the essential aspects of brain function. This paper argues that at least some aspects of cognition can be understood on a mechanistic neuron-level, and that there is clear evidence of progression in the field towards more sophisticated models that encompass greater detail of the emerging cognition. To support this, aspects of spatial orientation and navigation are elaborated on by using models of place cells and grid cells, further explaining how the neuronal level processes may enable us to perform cognitive tasks of spatial processing. Following is a paragraph on associative memory and the hippocampus, explaining how pattern-completion and error correction, content-addressable memory and auto-associative memory might be constituted at the physical level. Thus linking them to among other things how different memories may be associated with one another, and how the model is compatible with episodic memory. Lastly, a more detailed model for short-term memory is considered, along with the compatibility with long-term memory. To exemplify this, an analogy of several processes related to walking to the office will be outlined from a mechanistic and computational perspective. Finally, an alternative view of computational neuroscience is introduced; a framework regarding one of the higher aspects of cognition intelligence purely as a processes of prediction.

Computational models of place cells allow for a prediction of firing rate patterns, and thus an understanding of how spatial cognition may be constituted in the brain. Beginning with the computer simulation of place cells in the rats hippocampus by Sharp in 1991[1], building on the ideas reported by OKeefe in 1976[2], Sharp found that firing rate maps for simulated place cells correlated with the firing rate maps in vivo. The simulated hippocampal pyramidal cells were adapted from Rumelhart and Zisper (1986)[3], modified into two types of input-cells simulating the sensory cells, and subsequent layers with a lateral inhibition type of an architecture, i.e. a one winner-takes-it-all in every cluster. These findings illustrate that the hippocampal pyramidal cells fire relative to where you are, and that a mechanistic understanding of neurons can be used to simulate such cells. By analogy, when youre standing on the pavement and walk in the direction of your office, your place cells will fire accordingly. Sharps competitive learning model sufficed to explain the robustness of place cells firing when moving. However, simplifications were made, and the aspect of spatial navigation such as removal of environmental space field cues were not explained. In other words; the predictions yielded by the model[1] might render it valuable with regards to prediction and explanation under certain constraints, providing ground for further research and understanding. Addressing the analogy; how you notice that a car drives by fairly close to you, and what cues you receive from the environment is omitted. The former can be explained by OKeefe and Burgess (1996)[4], who modelled place cells using so-called boundary vectors cells (BVCs) oriented at a certain angle to one another. These may in turn be approximated by a thresholded sum of Gaussians for the distance from the walls of the environment for the rat. This provides a model that illuminates the representation of space fields in place cells, predicting a firing rate peak when an object, such as the car in the previous example, is at a fixed distance from you (ranging with the BVCs thresholds). This further illucidates how the cognitive map may be constituted on a mechanistic neuron-level. While the mechanistic models outlined so far do not encompass all aspects of spatial cognition, it is important to emphasise that they still provide a demonstration of how spatial cognition can be constituted at a physical level.

Grid cells further illuminate the mechanistic picture of spatial cognition. Hafting et al. reported in 2005[5] that a neural map of the spatial environment was found in the dorsocaudal medial entorhinal cortex. The model and metric of grid cells explains how the map of an environment along with a subjects position can be represented internally, additionally suggesting that grid cells might constitute path integration. Hafting et al. (2005)[5] emphasized that these cognitive functions were, however, poorly understood on a mechanistic level; constituting ground for their later research where Sargolini et al. (2006)[6] found that the representation of position, direction and velocity is strongly facilitated by the observed grid and head-direction cells. Their results imply that all layers of the neocortex act together as an integrated unit, where interactions occur between grid cells in the principal layers and head-direction cells in the higher layers. It should be noted that these findings may be significant in terms of the memory system of the brain, and that the theory implies the existence of a common cortical structure, such as outlined by Mountcastle, V. (1978)[7]. The report on grid cells[6] raises the question of how the integration in cortex spans various cells. Fyhn et al. 2007[8] reported on rate remapping and stable grid fields and global remapping in the medial entorhinal cortex. Their findings show that grid cells maintain a locally constant spatial phase structure, and that representations from multiple environments could be globally coded by place cells. Thus providing a further explanation for how translation of position and maintenance of a path-vector, and thus the update of the internal model of the environment, might be constituted in the brain. Analogously, this explains how you might walk straight back home, in case you walked in circles to visit various nearby locations. Some observations, however, might raise doubt of the predictive power of the models as a whole. One such example is navigation in non-horizontal environments (i.e. three dimensions), which has not been tested in the reports given so far. That being said, Jeffery et al. reported in 2012[9] that evidence for a different coding when it comes to a three-dimensional environment is weak. And even though they found that the ability of processing information in a third spatial dimension was present, it still seems to be coded in a metrically flat dimension, suggesting the same underlying functions are used in the brain, thus explaining the processing in this dimension too. In a recent study by Bush, Barry and Burgess (2014)[10], the contribution of grid cells to place cell firing is considered. They suggest an alternative model for place cell firing, in which the role of grid cells as outlined by Sargolini et al. (2006)[6] is questonioned. Bush et al. (2014)[10] reported that it might not be the input modules of the entorhinal cortex that decide the spatial scales, as commonly assumed - but the synthesised input from various sensory channels, implementing information related to spatial cues. This is further underlined in viewing the firing patterns of place and grid cells as intertwined and complementary, which may be more biologically plausible, rather than working in a hierarchical manner. It is worth noting how the former models of Fyhn et al. (2007)[8] may be elaborated on. By further expanding on previous models, it may be possible to attainin an even more detailed understanding of the actual mechanistic processes underlying spatial cognition.

Memory is a central aspect of cognition, and mechanistic models have shown how neural networks might implement associative memory. In 1982 Hopfield[11] reported that a Hopfield network, a fairly simple artificial neural network, had several of the properties of associative memory. The network is a fully connected recurrent and symmetric binary artificial neural network, updating its weights using Hebbian learning. This results in a network that is content-addressable, i.e. which completes a previously learned pattern, having only a partial input. The model also explains how spurious memories may arise; one memory may evoke another auto-associatively due to the recurrent connections in the network, thus corrupting old memories in the sense of associating the new with the evoked old ones. Nonetheless, this model is indeed simple, and does not include the connections to episodic memory, switching from learning to recall, nor how more complex cognition may emerge from associative memory. This can be explained by a computational model for fast and slow learning in memory consolidation by Hasselmo et al. (1995)[12], where the learning dynamics of the medial septum and switching between learning and recall for novel experiences is simulated. This computational model is shown to be more biologically realistic, using cholinergic suppression to achieve stable excitatory connections and a regulation of learning and recall. Note that this model also allows for episodic memory by auto-association of one memory followed by another, but only a serial form of memories one at a time, which does not explain how the associations are stored temporarily in short-term memory. It is important to emphasise that this model explains how learning and recall may be regulated, and furthermore that it is compatible with associative memory and episodic memory in the hippocampus. To return to the recurring analogy of this paper, we can now associate cues in the environment with the path to the office. Furthermore, the peculiar function of suddenly remembering a particular memory, can be explained by this model. You unconsciously remember it because the content which you are experiencing now trigger that memory.

Short-term memory, long-term memory, and memory consolidation are other important aspects of cognition. Modelling these may be required in order to acquire a more comprehensive understanding of the aspects of cognition that are related to memory. In 1990, McNaughton and Nadel[13] used Hopfield-like networks to construct memory matrices using both Hebbian learning, and memory matrices using feed-forward back-propagation to encode parts of the memory. These matrices could be used as models for short-term and long-term memory, respectively, and consecutively. McClelland et al. reported in 1995[14] that the neocortex learns slowly, which matches the fact that a feed-forward network converges slower than a network using Hebbian learning. These findings suggest that the hippocampus may teache the neocortex ensemble-structures over time, whilst learning novel patterns itself rapidly. This raises issues about how information is transferred, at what timescale, and how it might generalise in terms of semantic memory. These issues may be addressed by introducting the location of neuron ensembles in separate cortices, interlocking neuronal ensembles reciprocally in the different cortices, Damasio (1989)[15]. Note that this may explain how the two different types of memory networks can interact on a mechanistic level. Considering that the recurrent networks will fire at certain rates, and that we have several cortices in the hippocampus, this may suggest that short-term memories are stored in different frequencies of oscillatory sub-cycles. Lisman and Idiart (1995)[16] reported that oscillations are a mechanism for time processing of short-term memories. They also showed that several short-term memories could be stored in the same neural network, and found evidence that is compatible with the amount of short-term memories humans can store. To summarise, the mechanistically described models now allow us to remember who you just walked past, and yet maintain a line of thought. You are able to, after repeatedly having experienced a particular pattern at the same time as a road-sign, extrapolate that the sign probably means that there is a dangerous crossing ahead. Short-term memory is essential for high-level cognition, and long-term memory and memory consolidation for remembering what you have learned, in addition to seeing more complex patterns.

Whether a complete mapping on a neural level encompasses all aspects of cognition, remains an open question. Such a model might have to encode the intertwined processes occurring at a refined level of our physical existence, such as at the molecular or even quantum-mechanical level. That being said when a level at which it is believed to be possible to capture all that is required to constitute a particular aspect of cognition is found, it might be possible to extract the properties required to constitute the particular cognitive functions. I.e. an algorithm working on different principles, but still capturing the emerging cognitive functionality.

One of the high level aspects of cognition is intelligence. In his book On Intelligence [17], Jeff Hawkins outlines a framework with which he seeks to capture the so-called common cortical algorithm which he believes might suffice to constitute intelligence. Continuing along the same train of thought; it might be possible to simulate solely the quintessence of the processes constituting intelligence, providing for a more efficient simulation than the exact biological match of the desired processes, which are intractable. Successfully simulating an entity from which high-level cognition emerges might even allow for us to embed effective and sophisticated information processing algorithms directly into the entity. Furthermore, alternative forms of sensory input could be provided to possibly give the entity in a 'sense' a natural feel for various processes that are complex or abstract to the human mind. Imagine if doing statistical analysis was an as natural part of the mind as seeing different shapes and objects. One fact supporting this very fictional line of thought is the fact that human being are biological beings. In being so, not only do we still use older parts of the brain that might evolve to become more efficient with time, but we also need certain aspects of them to support our biological survival (hunger, reproductive need, etc.). This is not to mention the molecular function that is necessary to facilitate the information processing. Simulating biological evolution might be difficult or even impossible, but omitting the implementation of biological aspects that are solely needed for survival purposes, is theoretically plausible. For instance, who says that in the event of successfully creating an intelligent entity, it needs to have a goal? And does it necessarily need to have emotion? In Jeff Hawkins framework from On Intelligence, he assumes that intelligence is based entirely on prediction, which is constituted by mechanistic neuronal processes. It may appear that such a framework seems incompatible with some of the models outlined in this paper. However, with a closer examination, the models and observations can actually be considered to generally be compatible with Hawkins model. The major difference being that he regards the hippocampus as being on top of the cortical hierarchy. Regarding the findings of McNaughton and Nadel (1990)[13], and Hasselmo et al. (1995)[12], this might seem very intriguing, as memories are consolidated from the hippocampus to the neocortex. It is interesting to emphasise, however, that these findings can generally speaking be unified with the framework. There is evidently no strong observations in the two aforementioned studies implying that the hippocampus is on top of the cortical hierarchy. And just as Bush et al. (2014)[10] put grid cells in doubt by viewing the processes of grid and place cells as intertwined and working in a complementary fashion, the hippocampus and neocortex can be viewed as working in a complementary fashion. And it is worth noting that memory consolidation could still be outlined in a fairly similar way as in (Hasselmo et al. (1995)[12]). Even though information is first passed through the neocortex, the hippocampus can work as a filter in choosing what to relay back, and the neocortex will still need several recurrent signals of the same pattern to consolidate the memory. Therefore, further research on the framework should be conducted. One possible topic being how a novel computational model enclosing a circle of information processing within simulated aspects of the hippocampus, the prefrontal cortex, and the thalamus, would behave.

There is undoubtedly cumulative evidence of that some aspects of cognition can be understood at a neural level. The physical processes can be hypothesized and simulated experimentally if not fully, then at least in a simplified manner. Different processes may be combined, and in either way the conceptual step can be taken to higher levels. This can give rise to an emerging description, understanding and prediction of specific cognitive processes in the hypothesized environment, which might correlate with cognitive processes in living organisms. It is noticeable that by the course of time, computational and mechanistic models of the brain have evolved. By becoming increasingly sophisticated, they explain a larger part of the brains processes on a mechanistic neuron-level, simultaneously yielding a better understanding of the processes that constitute the cognitive map. In the same sense, it can be predicted how and why the underlying processes, such as the firing rate maps postulated by Sharp (1991)[1] approximately will fire in a mammalian brain, and partly how the resulting cognitive view will look, bearing in mind that the model is restrictive to a certain extent. Grid cells can further elaborate on the previous computational models of place cells and boundary vector cells, and further explain how for instance path integration is constituted in the brain. A mechanistic neuron-level understanding of how cognition enables you to spatially represent and navigate the environment, remember the way, recognise the office building, be reminded of thoughts from the past on the way, and learn patterns of related thoughts and events, can be attained. Further research on novel frameworks is a natural aspect both in the evolution of older models, as well as in the construction of novel models, potentially replacing the old ones. With the continued development of the multidisciplinary fields involved in neural computation, it will be interesting to see when most of the high-level cognitive concepts of cognition will be understood from a mechanistic neuron-level perspective. Perhaps even more enthralling is the possible applications that such a sophisticated technology could give rise to.

References and bibliography

[1] Sharp, Patricia E. 1991. Computer simulation of hippocampal place cells. *Psychobiology*, **19** (2): 103-115.

[2] O'Keefe, J. 1976. Place units in the hippocampus of the freely moving rat. *Experimental neurology*, **51** (1): 78109.

[3] Rumelhart and Zipser. 1986. Feature discovery by competitive learning. In: Rumelhart et al. 1986. *Parallel Distributed Processing, vol. 1.* MA, USA: MIT Press Cambridge. Pp. 151-193.

[4] OKeefe and Burgess. 1996. Geometric determinants of the place fields of hippocampal neurons. *Nature*, **381**: 425-428.

[5] Hafting et al. 2005. Microstructure of a spatial map in the entorhinal cortex. *Nature*, **436**: 801-806.

[6] Sargolini et al. 2006. Conjunctive Representation of Position, Direction, and Velocity in Entorhinal Cortex. *Science*, **312** (no. 5774): 758-762.

[7] Vernon Mountcastle. 1978. An Organizing Principle for Cerebral Function: The Unit Model and the Distributed System. In: Mountcastle and Edelman. 1978. *The Mindful Brain*. Cambridge, MA: MIT Press.

[8] Fyhn et al. 2007. Hippocampal remapping and grid realignment in entorhinal cortex. *Nature*, **446**: 190-194.

[9] Jeffery et al. 2012. Navigating in a 3D world. In: Menzel and Fisher. 2011. *Animal Thinking: Contemporary Issues in Comparative Cognition.* Strngmann Forum Reports, vol. 8. Cambridge, MA: MIT press.

[10] Bush. Barry. Burgess. 2014. What do Grid Cells Contribute to Place Cell Firing?. *Trends in Neuroscience*, **37**: 136-145.

[11] Hopfield. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, vol. **79** (no. 8): 2554-2558.

[12] Hasselmo et al. 1995. Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *The Journal of Neuroscience*, **15** (7): 5249-5262.

[13] McNaughton and Nadel. 1990. Hebb-Marr Networks and the Neurobiological Representation of Action in Space. In: Rumelhart et al. 1990. *Neuroscience and Connectionist Theory*. Lawrence Erlbaum Associates, Inc., Publishers.

[14] McClelland et al. 1995. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, vol. **103** (3), 419-457.

[15] Damasio. 1989. Time-locked multiregional retroactivation: A systems-level proposal for the neural substrates of recall and recognition. *Cognition*, vol. **33** (1-2), 25-62.

[16] Lisman and Idiart. 1995. Short-Term Memories in Oscillatory Subcycles. *Science, New Series*, vol. **267**, no. 5203, 1512-1515.

[17] Hawkins and Blakeslee. 2004. *On Intelligence*. 1^{st} ed. Times Books: Henry Holt and Company, LLC.