

Computational Modelling of Adult Hippocampal Neurogenesis

COMPUTATIONAL MODELLING OF ADULT HIPPOCAMPAL NEUROGENESIS

By Rory Finnegan,

*A Thesis Submitted to the School of Graduate Studies in the Partial Fulfilment of the
Requirements for the Degree Master of Science*

McMaster University September 26, 2016



This thesis is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

McMaster University

Master of Science (2016)

Hamilton, Ontario (McMaster Integrative Neuroscience & Discovery)

TITLE: Computational Modelling of Adult Hippocampal Neurogenesis

AUTHOR: Rory Finnegan (McMaster University)

SUPERVISOR: Dr. Suzanna Becker

NUMBER OF PAGES: x, 84

Abstract

The hippocampus has been the focus of memory research for decades. While the functional role of this structure is not fully understood, it is widely recognized as being vital for rapid yet accurate encoding and retrieval of associative memories. Since the discovery of adult hippocampal neurogenesis (AHN) in the dentate gyrus (DG) by Altman and Das in the 1960s, many theories and models have been formulated to explain the functional role it plays in learning and memory. These models postulate different ways in which new neurons are introduced into the DG and their functional importance for learning and memory. Few, if any, previous models have incorporated the unique properties of young adult-born dentate granule cells (DGCs) and their developmental trajectory. In this thesis, we propose a novel computational model of the DG that incorporates the developmental trajectory of these DGCs, including changes in synaptic plasticity, connectivity, excitability and lateral inhibition, using a modified version of the restricted boltzmann machine (RBM). Our results show superior performance on memory reconstruction tasks for both recent and distally learned items, when the unique characteristics of young DGCs are taken into account. The unique properties of the young neurons contribute to reducing retroactive and proactive interference, at both short and long time scales, despite the reduction in pattern separation due to their hyperexcitability. Our replacement model is subsequently extended to support learning dependent regulation of neurogenesis and apoptosis, using a convergence based approach to network growing and pruning. This hybrid additive and replacement model provides a more realistic and flexible approach to investigating the role of neurogenesis regulation in learning and memory. Finally, we incorporate the dentate gyrus model into a full hippocampal circuit to assess cued recall performance. Once again, our neurogenesis model shows decreased proactive and retroactive interference.

Preface

This thesis consists of five chapters. Chapter 1 provides a brief literature review and context for the experiments discussed in following chapters. Chapter 2 presents our initial AHN model in the DG using a static neural turnover method. Chapter 3 extends our model by simulating learning dependent regulation of neurogenesis and apoptosis. Finally, chapter 4 presents a full hippocampal model to explore the role of AHN on cued recall tasks.

Chapter 2 has been published in a special topics edition of the *Frontiers in Systems Neuroscience* journal series¹. The content from chapter 2 has been included in this thesis under the terms of the Creative Commons Attribution License (CC-BY 3.0). However, the introduction and discussion sections have been heavily modified in order to better fit the format of this thesis. The source code for all experiments presented in this document have been made publicly available under the terms of an MIT License². This thesis and its corresponding defence presentation are also available under a Creative Commons Attribution License (CC-BY 4.0).

Dr. Suzanna Becker originally proposed using the RBM as the base artificial neural network (ANN) model for this work and provided input on the design of the full hippocampal model described in chapter 4. Dr. Becker assisted in writing the original introduction and discussion sections for chapter 2, and provided a review and edits for the remainder of this thesis. The base RBM model was written in Julia³ using the Boltzmann.jl package⁴. I was responsible for writing the remainder of the manuscript, as well as designing, implementing and analyzing the results from each experiment.

¹Finnegan and Becker, 2015.

²Finnegan, 2015–2016.

³Bezanson et al., 2009–2016.

⁴Zhabinski, Finnegan, and contributors, 2015–2016.

Support for this research was provided by an NSERC Discovery grant awarded to Dr. Suzanna Becker, and a personal grant from Invenia Technical Computing.

Acronyms

AHN adult hippocampal neurogenesis

ANN artificial neural network

CD contrastive divergence

CRBM conditional restricted boltzmann machine

DG dentate gyrus

DGC dentate granule cell

DNC dynamic node creation

EC entorhinal cortex

FFNN feed forward neural network

LSM liquid state machine

LSTM long short-term memory

NPC neural progenitor cell

PP perforant pathway

RBM restricted boltzmann machine

RNN recurrent neural network

TA temporoammonic pathway

TRBM temporal restricted boltzmann machine

Acknowledgements

Thank you to the faculty and staff members in the MiNDS program for imparting the skills and knowledge necessary to complete this thesis. I am particularly appreciative of Dr. Sue Becker's constant support, guidance and above all patience over the past two years. It has been a pleasure working with you.

I am grateful for the supportive, friendly and stimulating work environment fostered in the Neurotechnology and Neuroplasticity Lab. Sincerest thanks to Kiret Dhindsa and Craig Hutton for always being around to provide suggestions and feedback.

To my family and friends, thank you for your continual support and confidence in my abilities throughout this endeavour.

Finally, to my partner and best friend Raelene Foisy, thank you for your wisdom, kindness and encouragement. This would not have been possible without you.

Contents

Abstract	iii
Preface	iv
Acronyms	vi
Acknowledgements	viii
1 Introduction	1
1.1 Hippocampal Structure & Function	1
1.2 Computational Models	8
1.2.1 Feed Forward Neural Networks	10
1.2.2 Recurrent Neural Networks	12
2 Neurogenesis paradoxically decreases both pattern separation and memory interference	19
2.1 Methods	20
2.1.1 Sparsity	21
2.1.2 Neuron Growth	22
2.1.3 Neural Turnover	24
2.1.4 Experiments	25
2.2 Results	30
2.3 Discussion	35

3	Learning Dependent Regulation of Neurogenesis and Apoptosis	39
3.1	Methods	40
3.1.1	Monitoring Learning	44
3.1.2	Convergence Method	44
3.1.3	Experiments	46
3.2	Results	47
3.3	Discussion	49
4	Neurogenesis in a full hippocampal model	52
4.1	Methods	53
4.1.1	conditional restricted boltzmann machines (CRBMs)	53
4.1.2	Stacking	55
4.1.3	Experiments	57
4.2	Results	59
4.3	Discussion	62
5	Conclusion	65
	Bibliography	70

Chapter 1

Introduction

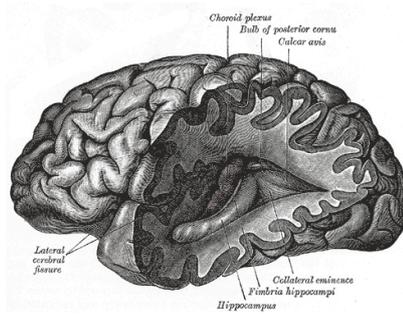
The role of the hippocampus in memory has been a subject of endless fascination for many decades. It is widely recognized that the hippocampus is crucial for rapid, accurate encoding and retrieval of associative memories. However, the neural mechanisms underlying these complex operations are still relatively poorly understood. In particular, despite the numerous theories and models put forward, many questions regarding the functional importance of adult hippocampal neurogenesis (AHN) remain unanswered. In this thesis, we will be using an artificial neural network (ANN) to explore the functional role AHN and young dentate granule cells (DGCs) play in learning and memory.

1.1 Hippocampal Structure & Function

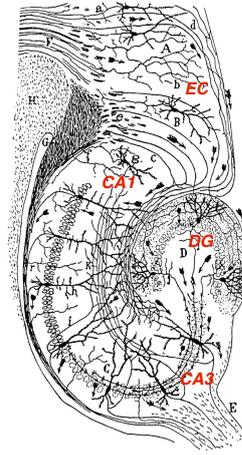
Located under the cerebral cortex, in the medial temporal lobe of the vertebrate brain, the hippocampal structure consists of the entorhinal cortex (EC), dentate gyrus (DG), and cornu ammonis sub-layers CA1 and CA3. Cortical sensory information from the perirhinal, parahippocampal, and prefrontal cortices enters the hippocampus via layers II and III of the EC. Layer II inputs are projected onto the DG and CA3 via the

perforant pathway (PP), while information from layer III is relayed to the CA1 via the temporoammonic pathway (TA). Information processed within the hippocampus is propagated from the CA1 layer back to the aforementioned cortical areas via deep-layer EC neurons. The DG receives information from the EC and projects onto the CA3 pyramidal cells via mossy fibres. Due to high levels of feedforward and feedback inhibition from local interneurons and extremely low firing rates among DGCs, it is believed that the DG serves to separate input patterns (Jung and McNaughton, 1993; Chawla et al., 2005; Rolls, 1987; O'Reilly and McClelland, 1994; Rolls and Treves, 1998). Despite the sparse activation of DGCs within the DG, evidence suggests that a single mossy fibre synapse is capable of activating many CA3 pyramidal cells, indicating that the DG has a significant influence on CA3 memory encoding (McNaughton and Morris, 1987; Treves and Rolls, 1992; O'Reilly and McClelland, 1994; McClelland, McNaughton, and O'Reilly, 1995; Myers and Scharfman, 2009). Along with receiving input from the EC and DG via the PP and mossy fibres respectively, the CA3 receives input from itself via many recurrent collateral connections. These are thought to help form the auto-associative activity needed for memory reconstruction and/or temporal encoding. Recent evidence indicates that reciprocal connection may exist between the CA3 and the DG, and between the CA3 and the EC (Scharfman, 2007). Despite a significant focus on modelling the trisynaptic pathway (EC → DG → CA3 → CA1 → EC), computational models utilizing these reciprocal connections have shown better pattern separation and recall capabilities (Myers and Scharfman, 2011). While Johnston and Amaral (1998) provide a more thorough overview of the hippocampal circuitry, figures 1.1a, 1.1b and 1.2 are provided as a visual summary of the hippocampal anatomy and circuitry. In particular, figure 1.2 covers the hippocampal layers modelled in this thesis.

Marr's theory of archicortex (Marr, 1971) was highly influential in setting the stage for subsequent computational theories of hippocampal function. At the core of his theory was the proposal that an associative memory system requires an initial sparse



(A) Posterior and inferior cornua of left lateral ventricle exposed from the side (Gray, 1918).



(B) Early drawing of neural circuitry in the hippocampal formation (Cajal, 1909). The original arrows show the path of excitation through the trisynaptic pathway, but additional labels for the EC, DG, CA3 and CA1 have been added for clarity.

FIGURE 1.1

coding stage followed by a subsequent processing stage that performs associative retrieval. While Marr's initial neural circuit proposed a 2 layer network consisting of an input layer and an associative layer with sparse coding, later this was revised into a 3 layer network consisting of an input layer, a sparse coding layer and an associative layer.

Subsequent modellers refined Marr's ideas and further suggested that these functions of coding and retrieval map onto the known anatomical and physiological properties of the DG and CA3 regions respectively. The assumption is that the sparse coding stage in Marr's model could represent the sparse activation of granule cells in the DG and the associative layer would represent the CA3 pyramidal cells with their dense recurrent connections (McNaughton and Morris, 1987; Treves and Rolls, 1992; O'Reilly and McClelland, 1994; McClelland, McNaughton, and O'Reilly, 1995; Myers

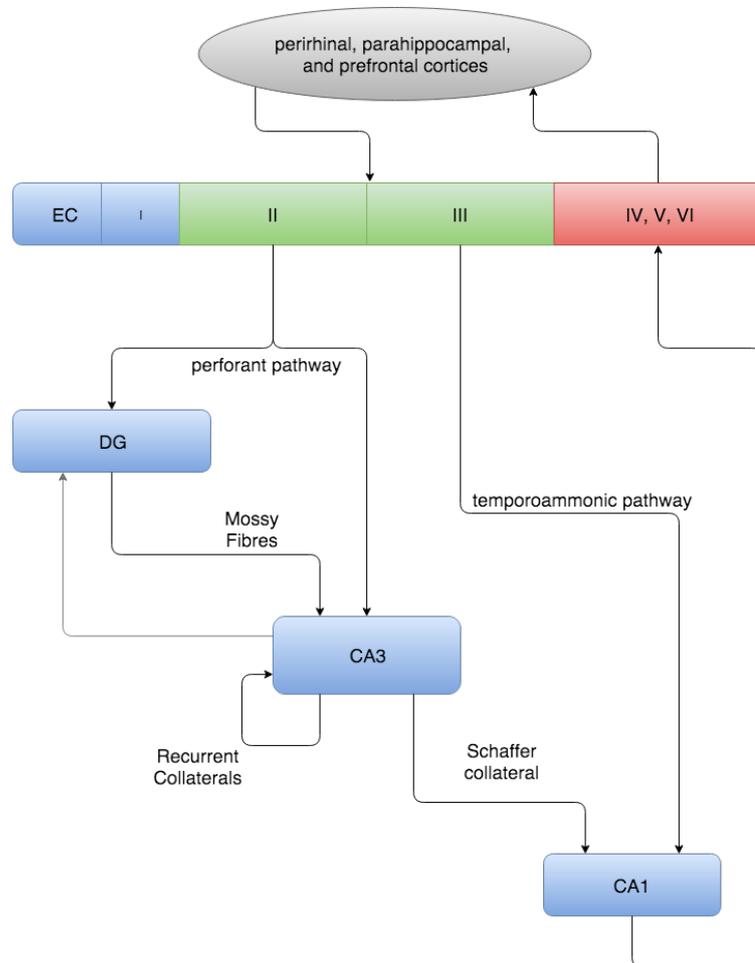


FIGURE 1.2: All cortical sensory information from the perirhinal, parahippocampal, and prefrontal cortices enters the hippocampus via layers II and III of the EC. The DG receives input directly from layer II EC axons via the PP, where it is believed that the DG performs pattern separation on the input through lateral inhibition. The CA3 layer receives input from both the EC via the PP, and the DG via mossy fibres. The dense recurrent connections among the CA3 pyramidal neurons are thought to be involved in associative retrieval of memories. These pyramidal neurons relay information to the CA1 through the Schaffer collaterals. Along with input from the CA3 layer, the CA1 receives information from the EC layer III axons via the TA. All excitatory output leaves the hippocampal formation via back-projections through CA1 layer to the deep-layer neurons of the EC and onto the aforementioned cortical areas.

and Scharfman, 2009). These models incorporate an important characteristic of the mature DGCs: they are heavily regulated by feedback inhibition, resulting in extremely

sparse firing and high functional selectivity (Jung and McNaughton, 1993; Chawla et al., 2005). Computer simulations demonstrate that the DG is thereby able to improve its capacity for storing overlapping memory traces by generating less overlapping neural codes, a process that has come to be known as pattern separation (Rolls, 1987; O'Reilly and McClelland, 1994; Rolls and Treves, 1998). Similarly, a key component of many full hippocampal models is the many recurrent connections among CA3 pyramidal cells. Simulations of memory recall have demonstrated the importance of these recurrent connections for accurate memory recall and pattern completion (McNaughton and Morris, 1987; Treves and Rolls, 1992; O'Reilly and McClelland, 1994).

The discovery of AHN, first in rodents (Altman and Das, 1965; Altman and Das, 1967) and subsequently in a wide range of mammalian species including humans (Eriksson et al., 1998), has forced theorists to reconsider the computational functions of the DG. Several computational models incorporating neurogenesis have been put forward. These models postulate a range of functional roles for neurogenesis, including mitigating interference (Chambers et al., 2004; Becker, 2005; Wiskott, Rasch, and Kempermann, 2006; Becker, MacQueen, and Wojtowicz, 2009; Cuneo et al., 2012), temporal association of items in memory (Aimone, Wiles, and Gage, 2006; Aimone, Wiles, and Gage, 2009) and clearance of remote hippocampal memories (Chambers et al., 2004; Deisseroth et al., 2004; Weisz and Argibay, 2009; Weisz and Argibay, 2012). While these different theories are not necessarily incompatible with one another, they make different predictions regarding the effect of temporal spacing.

When similar items are spaced closely in time, some models predict that neurogenesis should increase pattern integration (Aimone, Wiles, and Gage, 2006; Aimone, Wiles, and Gage, 2009). By the same token, the reverse should be true of animals with reduced neurogenesis: they should exhibit impaired pattern integration, and therefore,

enhanced pattern separation for closely spaced items. Thus factors that suppress neurogenesis such as stress and irradiation (Gould et al., 1998; Wojtowicz, 2006) should impair pattern integration, resulting in *superior* abilities to distinguish similar items that are learned within the same time period. That said, the opposite has been observed empirically. Rodents with reduced neurogenesis are impaired at spatial discriminations for closely spaced locations that are learned within the same session (Clelland et al., 2009), while rodents with running-induced elevated neurogenesis show enhanced performance on spatial tests of pattern separation (Creer et al., 2010). Consistent with these data, humans who have undergone several weeks of aerobic exercise training show superior performance on a within-session behavioural test of pattern separation while those with elevated stress and depression scores show a deficit on the same task (Déry et al., 2013).

When similar items are spaced widely in time, different predictions can be made regarding the fate of the item in remote memory versus the newly learned item. Most or all computational theories agree that neurogenesis should facilitate the encoding of new items, protecting against proactive interference from previously learned information. Empirical data support this notion. For example, animals with intact levels of neurogenesis are able to learn to discriminate olfactory odour pairs that overlap with pairs learned several days ago, whereas irradiated animals with reduced neurogenesis show greater proactive interference on this task (Luu et al., 2012). On the other hand, opposing predictions arise regarding the influence of neurogenesis on remote memories. Some theories predict that neurogenesis should promote clearance of remote memories (Chambers et al., 2004; Deisseroth et al., 2004; Weisz and Argibay, 2009; Weisz and Argibay, 2012). Other theories make the opposite prediction, that intact neurogenesis levels should protect against retroactive interference of new learning on remote memories (Becker, 2005; Becker, MacQueen, and Wojtowicz, 2009). Consistent with the latter

prediction, when animals with reduced neurogenesis learn overlapping visual discriminations in different sessions spaced several days apart, the more recently learned discrimination disrupts the retrieval of the earlier memory (Winocur et al., 2012). These data support a role for neurogenesis in minimizing retroactive interference between remote and recent memories. However, it is possible that neurogenesis plays dual roles in remote memory, protecting some hippocampal memories from interference while causing other memories to decay.

How is it that AHN can contribute to improved memory and reduced interference when similar items are learned within a single session as well as when items are learned across temporal separations of days or weeks? The following thesis set out to investigate whether a single computational model of hippocampal coding could accommodate the role played by neurogenesis across this wide range of time scales. We propose that the functional properties of a heterogeneous ensemble of young and mature DGCs contributes to this improved memory and reduced interference among similar items. Studies have shown that the presence and developmental trajectories of adult-generated neurons contributes to the functional heterogeneity among neurons within the granule layer (Wang, Scott, and Wojtowicz, 2000; McAvoy, Besnard, and Sahay, 2015). As such, our model attempts to take this trajectory into account during learning. In most if not all previous DG models, these characteristics have been ignored (Becker, 2005; Chambers et al., 2004; Weisz and Argibay, 2009; Weisz and Argibay, 2012). It is known that young adult-generated neurons in the DG are more plastic, have less lateral inhibition, have sparser connectivity and are more broadly tuned than their mature counter-parts (Schmidt-Hieber, Jonas, and Bischofberger, 2004; Snyder, Kee, and Wojtowicz, 2001; Temprana et al., 2015; Dieni et al., 2013; Piatti, Ewell, and Leutgeb, 2013; Marin-Burgin et al., 2012). All of these may affect how young DGCs learn in relation to the existing networks of mature DGCs.

Among existing computational hippocampal models, those that incorporate neurogenesis typically do so by either replacing existing neurons by re-randomizing their weights (Becker, 2005; Chambers et al., 2004) or introducing new neurons with random weights (Weisz and Argibay, 2009; Weisz and Argibay, 2012). Several additional models have looked at how regulation of neurogenesis can impact learning and plasticity by simulating dynamically regulated neural turnover and replacement (Deisseroth et al., 2004; Crick and Miranker, 2006; Chambers and Conroy, 2007). Studies by Butz and colleagues include a model of synaptogenesis, providing a framework for how neurogenesis regulation impacts synaptic rewiring and plasticity over varying time periods (Lehmann, Butz, and Teuchert-Noodt, 2005; Butz et al., 2006; Butz et al., 2008). However, none of these models have investigated how regulation of neurogenesis and apoptosis contribute to learning as a continually evolving temporal process.

1.2 Computational Models

Many computational approaches to modelling cognition and memory exist. For example, some approaches view the mind as a system that operates on abstract symbols to form complex behaviours (Turing, 1950; Searle, 1980). These models can be used to quickly formulate theories about how high level cognitive operations might interact. Alternatively, other models might focus on simulating the conductances of single neuron membrane potentials and voltage-gated ion channels (Hodgkin and Huxley, 1952). These models are useful when investigating the impact of changes to resting membrane potentials, or intra- and extra- cellular voltages. ANN models seek to simulate collections of neuron in order to learn an objective function (Hebb, 2002; Rosenblatt, 1962; Rumelhart, McClelland, and PDP Research Group, 1986; McClelland, Rumelhart, and PDP Research Group, 1986). These are particularly useful when exploring how the organization of neural networks impact learning and memory. ANN models can also vary

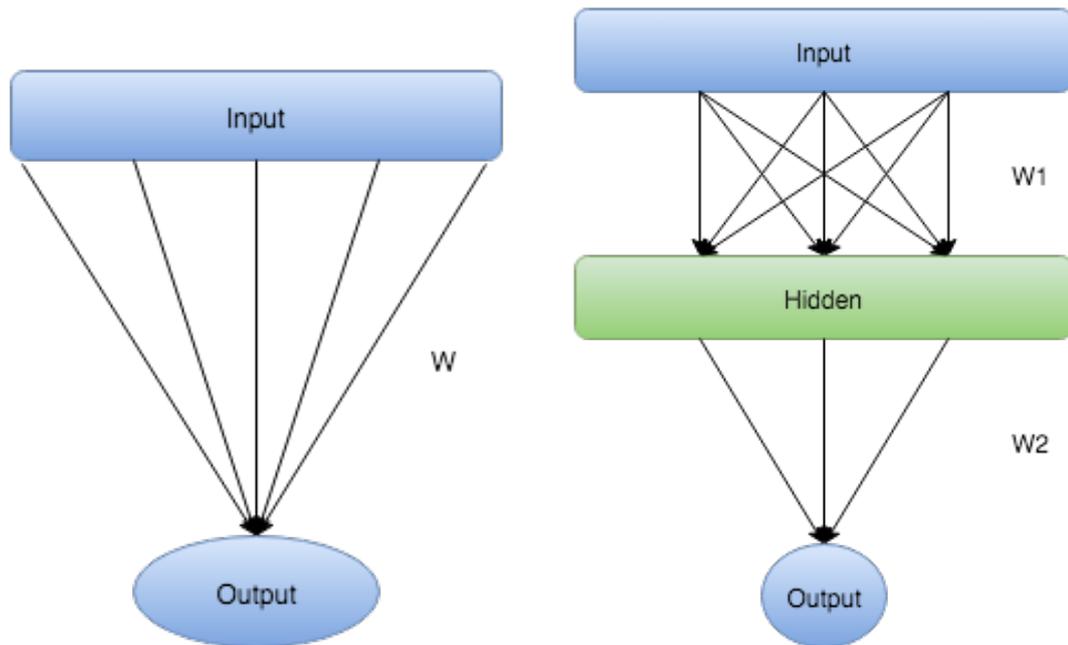
in their levels of abstraction. Spike time and firing based models focus on simulating the temporal firing patterns among neurons, which can be useful when investigating the functional role of oscillatory patterns such as theta rhythms observed in EEG studies (Brunel and Hakim, 1999). Alternatively, other ANNs simply learn a set of weights by applying a learning rule over a discrete set of independent patterns. Unfortunately, the more biologically plausible conductance and spiking network models require more resources to compute, which reduces the size of the networks that can be simulated. The perspective we have taken when selecting a base model to use is best summarized by George Box, who stated that "All models are wrong but some are useful" (Box and Draper, 1987). For our purposes, a non-spiking ANN should provide enough detail to explore the functional impact of the hippocampal structure and interactions between layers while remaining relatively computationally inexpensive to simulate, allowing us to build networks containing thousands to millions of neurons.

The architecture or topology of ANNs can be summarized as a graph where nodes represent neurons and edges, typically called weights, represent synapses. As will be discussed below these graphs can be acyclic or cyclic, with unidirectional or bidirectional edges. ANNs learn by presenting labelled (supervised) or unlabelled (unsupervised) data to the network and using a learning rule to update the weights to better fit the observed data. These learning rules update the weights between nodes in much the same way that synaptic plasticity modifies the synaptic strengths between neurons in biological neural systems.

Donald Hebb's theory of synaptic plasticity (Hebb, 2002) paved the way for the majority of modern learning rules, particularly unsupervised learning rules. The core idea was that if a neuron consistently takes part in firing another neuron then some growth in one or both neurons is required to make that behaviour more efficient. Carla Schatz summarized it best with the phrase, "Cells that fire together, wire together" (Schatz,

1992). This type of learning is often referred to as Hebbian learning in the ANN literature.

1.2.1 Feed Forward Neural Networks



(A) Single layer perceptron ANN with a set of inputs and a single output unit.

(B) Multilayer perceptron with a set of inputs, hidden layer and single output unit.

FIGURE 1.3

One of the first connectionist models developed by Frank Rosenblatt in the late 1950s was the perceptron. The perceptron model is what we would call a binary classifier today. This means that it attempts to learn an arbitrary function of the form $y = f(x)$, where y can only be 0 or 1, provided a large enough training set of corresponding x and y values. The perceptron learns this function by iteratively updating a set of weights W between the input vector and the single binary output unit, such that it minimizes the error between the observed and predicted values of y given x (Rosenblatt, 1962). A diagram of the network architecture is provided in figure 1.3a and the calculations for

estimating $f(x)$ and the weight updates can be seen in equations 1.1 and 1.2 respectively.

$$f(x) = \sum_i W_i x_i \quad (1.1)$$

$$\Delta W_i = \epsilon(y_{\text{data}} - y_{\text{pred}}) \times x_i \quad (1.2)$$

In order to expand the perceptron to multiple layers, a method for sending the error signal back through multiple layers was required. Backpropagation (Rumelhart, Hinton, and Williams, 1986b) is a widely used method for calculating this update for each layer of weights. The revised multilayer architecture and update rule are provided in figure 1.3b and equation 1.3 respectively. For the full derivation of the update rule please see (Rumelhart, Hinton, and Williams, 1986b).

$$\Delta W_{i,j} = -k \frac{\partial E}{\partial W_{i,j}} \quad (1.3)$$

However, in order for the derivation of the above update rule to work a differentiable activation function needs to be used instead of the threshold function from the single layer perceptron. The logistic function provided in equation 1.4 is one of the most common activation functions that satisfy this requirement and is used in other ANNs we will discuss.

$$f(x) = \frac{1}{1 + \exp(-x)} \quad (1.4)$$

Autoencoders are a special case of the multilayer perceptron, commonly used in modelling the hippocampus (Gluck and Myers, 1993; Becker, 2005). Autoencoders consist of at least an input layer, a hidden layer and an output layer, where the input layer and the output layer are the same size. In this case, rather than predicting an output variable y , the network is trying to find a latent representation of the input patterns such that it can encode and reconstruct them. This network has two key advantages relevant to hippocampal modelling. First, the network can be trained in an unsupervised fashion, meaning that we do not need any labelled data y , since the network is just encoding and decoding the input x . Also, since the network is simply learning to encode and decode the training patterns it can be used as a simple model of the associative memory in the hippocampus.

Several issues arise with modelling the hippocampus as a multilayer perceptron. First, the training of a large multilayered network using backpropagation from randomly initialized weights is often slow due to the number of differential calculations needed on each iteration through a training set. Second, passing the derivative of the error through each hidden layer results in most of the learning occurring in the last layer, and little change in the earlier layers. This is known as the vanishing gradient problem (Hochreiter et al., 2001). Put another way, the learning typically gets lost in the noise, and converges on a very poor set of weights. Finally, this method is considered to be less biologically plausible due the requirement of non-local computations (Stocco, Lebiere, and Anderson, 2011).

1.2.2 Recurrent Neural Networks

Recurrent neural networks (RNNs) are another class of ANNs which operate over cyclic graphs, unlike feed forward neural networks (FFNNs) which are acyclic. The cyclic

connections within the network provide a mechanism for modelling temporal dynamics and sequence learning (Lipton, Berkowitz, and Elkan, 2015). Since a full literature review of RNNs is outside the scope of this thesis we will only provide a very brief overview of some of the common architectures relevant to hippocampal modelling, including the restricted boltzmann machine (RBM) used in the remaining chapters.

The simplest method of learning in an RNN is to reuse the backpropagation learning rule. In this case, we store that unit’s activation at one or more previous time steps, and we simultaneously learn the weights from that unit to other units across all of these time steps. This is referred to as backpropagating through time (Werbos, 1988). Unfortunately, when learning many time steps, this has the same vanishing gradient issue found in large multilayer perceptrons (Hochreiter et al., 2001).

Hopfield networks (Hopfield, 2014) are an RNN which function as a type of content addressable or associative memory network. The network consists of a single layer of neurons which are all symmetrically interconnected with each other, except no neuron is connected to itself. The network learns by clamping patterns to the units and updating the weights according to equation 1.5. The Hebbian nature of this learning rule implies that units with the same state (active or inactive) for the majority of patterns will, on average, learn to attract each other with positive weights and repel differing units with negative weights.

$$\Delta W_{i,j} = \frac{1}{n} \sum_{p=1}^n \epsilon_i^p \epsilon_j^p \quad (1.5)$$

where n is the number of patterns and ϵ_i^p specifies the unit state at element i in pattern p .

The activation of individual units in the network is a thresholded sum over the activations, weighted by the corresponding weights of all incoming connections as can be

seen in equation 1.6.

$$s_i = \begin{cases} +1, & \text{if } \sum_j W_{i,j}s_j \geq \theta_i, \\ -1, & \text{otherwise} \end{cases} \quad (1.6)$$

where θ_i is the threshold for unit i , and s_i and s_j is the activation states for units i and j respectively.

The Hopfield network's binary threshold activation function, combined with appropriate assumptions about the order in which units' states are updated (updated sequentially in random order) can be shown to minimize the energy function in equation 1.7. This can be used to monitor the global state of the network at each step. When a pattern is presented to the network, determining its initial state, as units' states are repeatedly updated, the network's global state converges to a stable energy minimum, referred to as an attractor state. Furthermore, the Hebbian weight-update equation creates energy minima around the stored training patterns, thereby stabilizing each pattern as an attractor state. Thus, as weights in the network are updated the energy value of the network will decrease. The minimization of free energy within the network, combined with the unsupervised and local learning rule, provides a more biologically plausible model of associative memory in the hippocampus.

$$E = -\frac{1}{2} \sum_{i,j} W_{i,j}s_i s_j + \sum_i \theta_i s_i \quad (1.7)$$

Despite these advantages, the Hopfield network suffers from limited storage capacity. For a network with n units the asymptotic upper bound is $2n$ in the general case. While efforts to improve the storage capacity of the Hopfield network have been made, other networks such as perceptrons still have better performance (Wu et al., 2012).

The Boltzmann machine (Ackley, Hinton, and Sejnowski, 1985) is another type of RNN which learns a set of weights so as to form a probabilistic, generative model of the training data. The network consists of a set of fully and reciprocally connected stochastic units, partitioned into visible and hidden units. Weights in the network are updated based on the difference between the data-dependent expectations (distribution of the dataset) and the model’s expectations. Calculation of these expectations is intractable; however, they can be approximated through Gibbs sampling. In this approach a Markov chain is run for every training pattern to approximate the data-dependent expectation, while another chain is run to approximate the model’s expectation. Unfortunately, these Markov chains still take significant time to stabilize.

The RBM simplifies the Boltzmann machine by removing visible-to-visible and hidden-to-hidden connections, forming a bipartite graph as seen in figure 1.4a. This makes the sampling of the data-dependent and model expectations more tractable. Sampling time can be further reduced using a technique called contrastive divergence (CD) (Hinton, 2002; Carreira-Perpinan and Hinton, 2005). The CD learning rule is provided in equation 1.8. This equation includes the same positive and negative Hebbian learning terms representing the data-dependent expectation and the model’s expectation. Brief Gibbs sampling is still used to obtain the visible and hidden unit states for the positive and negative terms in the learning rule. While figure 1.4b shows a single step of Gibbs sampling, the visible and hidden units could be reconstructed for many steps to achieve a better approximation of the underlying distribution.

$$\Delta W_{ij} = \epsilon((v_i h_j)_{\text{data}} - (v_i h_j)_{\text{recon}}) \quad (1.8)$$

where v_{data} is the input vector and h_{data} is the data-driven hidden state generated by clamping the states of the visible units to v_{data} and sampling the hidden units’ states

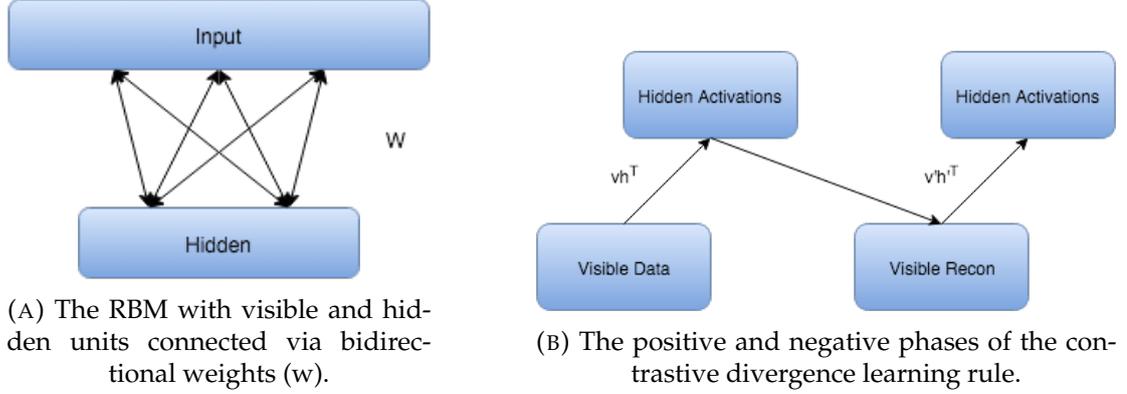


FIGURE 1.4

according to equation 1.12. v_{recon} is a reconstruction of the input vector generated by clamping the states of the hidden units to the data-driven pattern h_{data} and sampling the states of the visible units according to equation 1.11. h_{recon} is then created in the same way as h_{data} , but by clamping the visible units' states to v_{recon} .

$$\Delta a_i = \epsilon(v_{i\text{data}} - v_{i\text{recon}}) \quad (1.9)$$

$$\Delta b_j = \epsilon(h_{j\text{data}} - h_{j\text{recon}}) \quad (1.10)$$

In equations 1.12 and 1.11 below a_i and b_i represent biases which provide a mechanism for shifting the output of the sigmoid activation function, similar to thresholds in other neural network models. Equations 1.9 and 1.10 show that a and b are updated using the same positive and negative terms used in updating W . Figure 1.4b provides a visual representation of this learning procedure.

$$p(v_i = 1|h) = \sigma(a_i + \sum_j h_j w_{ij}) \quad (1.11)$$

$$p(h_j = 1|v) = \sigma(b_j + \sum_i v_i w_{ij}) \quad (1.12)$$

We can see from equation 1.8 that the positive Hebbian term associates data-driven input and hidden state vectors, while the negative Hebbian term tries to “unlearn” the association between the corresponding reconstructed visible and hidden state vectors. Theoretically, the learning procedure should converge when its internal reconstructions of the training patterns exactly match the corresponding data-driven states. In general, an RBM model’s reconstructions of the training patterns are obtained by alternately sampling hidden and visible unit states that are nearby data-driven states using the model’s bottom-up and top-down weights respectively.

Like the Hopfield network, the RBM utilizes a local and unsupervised learning rule, which also minimizes the free energy within the network (Barra et al., 2012). However, the presence of distinct visible and hidden units, along with the ability to stack RBMs, provides greater memory capacity. Furthermore, the ability to leave the RBM unclamped, in a generative state, may provide a way of simulating imagination and dreaming along with memory reconstruction. It is for these reasons that the RBM is used as the base ANN for our model.

Before concluding, we would like to mention long short-term memory (LSTM) networks as a type of RNN that has had significant success on sequence and time series learning problems (Graves, 2012; Schmidhuber, Wierstra, and Gomez, 2005). LSTMs use the concept of a memory cell, also called an LSTM block. This block feeds a set of inputs through a squashing function to read, write and keep gates, which control long and short term storage within the network. Once again, backpropagation can be used to send an error signal back through the memory cells. Interestingly, by continuously

feeding the error signal back through the gate weights within the same block, the vanishing gradient problem can be avoided (Hochreiter and Schmidhuber, 1997). While LSTMs are better designed for sequence learning discussed in the later chapters of this thesis, this was not our primary path of investigation and as such the simpler RBM model satisfied the requirements for our base associative memory model.

For the remainder of this thesis we will be using the RBM to explore the role of young DGCs in rapid encoding and recall within the hippocampus. Chapter 2 presents a novel model of the DG, which incorporates the developmental trajectory of adult-born DGCs. Chapter 3 adds a mechanism for modelling learning dependent regulation of neurogenesis and apoptosis. Finally, chapter 4 presents a combined DG and CA model in order to explore the role of young DGCs on full hippocampal encoding and recall.

Chapter 2

Neurogenesis paradoxically decreases both pattern separation and memory interference

In this chapter, we present a novel computational model of the dentate gyrus (DG) incorporating the developmental trajectory of adult-born dentate granule cells (DGCs), using a modified version of the restricted boltzmann machine (RBM) to model the neural circuitry and learning equations of DGC. As discussed in chapter 1, an RBM is a type of neural network model consisting of 1 layer of visible and 1 layer of hidden units, with each visible unit reciprocally connected to each hidden unit. In our model, a single RBM (not stacked RBMs) will represent the entorhinal cortex (EC) input and DGCs with its visible and hidden units respectively. As the model DGCs undergo development, they become progressively less plastic, more sparse in their firing, and more densely connected to their entorhinal inputs. We demonstrate how these properties can explain the importance of adult-generated DGCs at both short and long time scales.

In the model described here, the maturational trajectory of adult born DGCs will be loosely based on mouse data, for DGCs from the third week of maturation onward. It is

at about age 3-4 weeks that adult born DGCs have established synaptic afferent and efferent connections and are able to fire action potentials (Zhao et al., 2006). As compared to more mature neurons, Schmidt-Hieber, Jonas, and Bischofberger (2004) have shown that these young neurons have a higher input resistance, lower capacitance, lower activation threshold and a slower membrane time constant. As a result, 3-4 week old DGCs can be described as being more excitable, while having smaller and slower action potentials (Schmidt-Hieber, Jonas, and Bischofberger, 2004; Snyder, Kee, and Wojtowicz, 2001). Moreover, the young neurons are more sparsely connected to their perforant pathway (PP) inputs from the EC relative to mature DGCs (Piatti, Ewell, and Leutgeb, 2013). From weeks five through eight the young neurons undergo a gradual decline in synaptic plasticity and are increasingly regulated by feedback inhibition (Temprana et al., 2015). By the eighth week, the physiological properties of the adult-generated DGCs are largely indistinguishable from that of existing mature DGCs (Temprana et al., 2015; Piatti, Ewell, and Leutgeb, 2013).

2.1 Methods

In this section, we propose a novel approach to expressing neurogenesis in an artificial neural network (ANN) model of the DG. While several replacement and additive models of neurogenesis have looked at how new neurons affect learning (e.g. Becker, 2005; Weisz and Argibay, 2009), few models have considered the full range of unique properties of adult hippocampal neurogenesis (AHN) including the developmental trajectory of of adult-generated neurons: changes in plasticity, connectivity, excitability and survival versus apoptosis. The primary contribution of this work is to provide a computational framework within which all of these factors can be manipulated, differentiating the role of young versus mature DGCs in memory, and the progression from one to

the other. In the computational model described here, we use the RBM (Hinton, 2002; Smolensky, 1986; Freund and Haussler, 1992) architecture and learning procedure.

As discussed in chapter 1, RBMs are a type of generative, associative neural network model commonly used in deep learning applications (see e.g. Hinton and Osindero, 2006; Nair and Hinton, 2009). Our approach to expressing the neural trajectory of young DGCs in an RBM is to incorporate additional constraints into the learning equation, such as a dynamic learning rate and sparsity penalties. While there are several advantages to RBMs as discussed in chapter 1, it is important to note that the use of these constraints is not limited to RBMs and could easily be applied to other types of neural network models (eg. multilayer perceptrons, autoencoders, recurrent neural networks (RNNs), etc).

2.1.1 Sparsity

In our simulations of neurogenesis, we take into consideration both sparse coding and sparse connectivity. Sparse coding means that very few strongly activated neurons respond to a given event. This helps to improve pattern separation as it minimizes the probability of overlap in the model’s internal representation of highly similar input patterns. As noted in chapter 1, extreme sparse coding is observed in mature DG granule cells, but not in less mature adult-generated neurons. In our model, we simulate sparse coding by incorporating a sparsity cost constraint into the learning objective. Our sparse coding cost term is the average squared difference between each hidden unit’s average activation and its target probability of activation (Nair and Hinton, 2009). By taking the derivative of this cost term with respect to the weights, we obtain an added component to the learning equation that adjusts the weights so as to penalize units whose activation deviates from a target level of sparseness. The relative importance of this sparse coding term increases with the age of the neurons, to simulate

the increased degree of connectivity with inhibitory interneurons of mature DGCs. In the updated learning equation below, q_j is the mean of our sampled hidden activation for hidden unit j from equation 1.12 and p is our target activation probability.

$$\Delta W_{ij} = \epsilon((v_i h_j)_{\text{data}} - (v_i h_j)_{\text{recon}}) - \text{cost}(q_j - p) \quad (2.1)$$

Sparse connectivity describes the level of interconnectedness between the visible and hidden layers. As mentioned earlier, the degree of inter-connectivity is another property that changes as the young DGCs mature.

We simulate the maturational evolution of increased sparse coding and decreased sparse connectivity as follows. In the case of sparse coding we vary the weight on the sparsity cost for each hidden unit so that it is smaller for young neurons and larger for their mature counterparts. To impose a sparse connectivity constraint, a binary matrix is used as a connectivity mask for the weight matrix. For young DGCs, only 30% percent of their connections were randomly unmasked (non-zero), to simulate low connectivity. Thus, a young DGC is initially connected to relatively few ECCs. As the hidden units mature, the number of non-zero visible-to-hidden connections in the connectivity matrix for that hidden unit is increased probabilistically. At the end of each weight update, the weight matrix is multiplied by this connectivity mask in order to maintain the “disconnected” links and weights of zero.

2.1.2 Neuron Growth

Our model makes the assumption that young neurons are more plastic, have less lateral inhibition (simulated via our sparse coding cost rather than lateral connections) and are more sparsely connected than their mature counterparts, in accordance with biological

data (Schmidt-Hieber, Jonas, and Bischofberger, 2004; Oswald and Reyes, 2008; Marin-Burgin et al., 2012; Wang, Scott, and Wojtowicz, 2000). For simplicity, we assume that each of these characteristics follows a temporal growth curve that can be described with some permutation of the Gompertz function (Gompertz, 1832). The Gompertz function has been used to model growth in a variety of applications ranging from modelling bacterial growth in biology to product demand in economics (Zwietering et al., 1990; Towhidul Islama, 2002).

$$g(t) = e^{-e^{-st}} \quad (2.2)$$

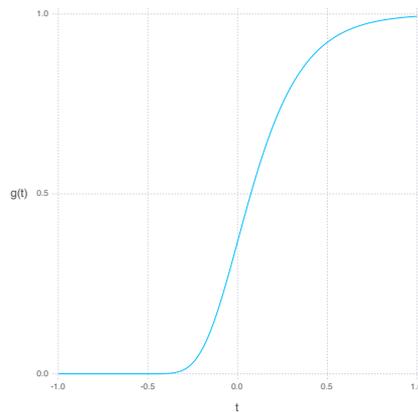


FIGURE 2.1: Gompertz function where s is set to 5 and t is between -1 and 1 .

The Gompertz function in equation 2.2 defines a sigmoid-like growth curve, where t describes the time step and s describes the shape or steepness of the function as can be seen in figure 2.1. For our purposes, t is bounded between -1 and 1 and the s is always set to 5. To model young DGC growth characteristics in the RBM, each hidden neuron has its own set of parameters defining its current learning rate and sparsity constraints. Additionally, each hidden unit has a time parameter representing its age. At each simulated unit time interval, the age of a hidden unit is increased, and its

constraint parameters are updated as follows. The learning rate, which can be thought of as a neuron’s plasticity level, is defined as $1 - g(t)$ normalized to lie between 0.0025 and 0.1. Inversely, our sparsity cost can simply be taken from $g(t)$ and normalized to lie between 0 and our initial sparsity cost of 0.9. Given these variable properties, the learning rule can be redefined as

$$\Delta W_{ij} = \epsilon_j((v_i h_j)_{\text{data}} - (v_i h_j)_{\text{recon}}) - (\lambda_j W_{ij}) - \text{cost}(q_j - p) \quad (2.3)$$

where the learning rate ϵ , weight decay λ and sparsity cost terms are now each weighted by dynamically changing vectors of values rather than static hyperparameters.

2.1.3 Neural Turnover

It is difficult to estimate the rate at which adult-generated neurons undergo apoptosis versus survival and maturation into adult DGCs. These processes are governed by many factors (see, e.g., Elmore, 2007; Hutchins and Barger, 1998; Cecchi et al., 2001; Cameron and McKay, 2001) and are not completely understood. Generally, apoptosis among healthy neurons tends to be activity and age dependent, such that the older a neuron is, the more likely it is to undergo apoptosis, whereas greater involvement in neural coding protects a neuron from cell death (Hutchins and Barger, 1998; Cecchi et al., 2001) and a significant number of new DGCs survive to adulthood (Cameron and McKay, 2001). Using these observations, we formulate a rule for determining whether a given neuron will survive or undergo apoptosis based on its age and its contribution to learning and memory. To assess a unit’s contribution to learning and memory, we define two terms: its specificity and average synaptic strength. To assess stimulus specificity, we calculate the standard deviation of each hidden unit’s incoming weights, a quantity we refer to hereafter as its “differentiation”. The justification is that hidden units with equal weight to all visible units will be less effective at differentiating

input patterns. Similarly, we calculate the average absolute value of the those incoming weights, to assess synaptic strength. Combining the differentiation and synaptic strength penalty terms, we are penalizing hidden units with incoming weights that are all very similar and close to zero. We rank each hidden neuron based on a weighted average of its synaptic strength, differentiation and age with equation 2.4. Neurons within the lowest 5% of this ranking undergo simulated apoptosis by having their age reset to 0 and weights reset to random initial values (or set to 0 in the case of bias weights).

$$Z_i = (\alpha \text{Strength}_i + \beta \text{Differentiation}_i + \gamma(1 - \text{Age}_i)) / (\alpha + \beta + \gamma) \quad (2.4)$$

where

- Strength_i is the average of the weights from all visible units to a given hidden unit i .
- Differentiation_i is the standard deviation of the visible weights to hidden unit i
- Age_i is our recorded age for the hidden unit i
- α , β & γ are coefficients for modifying the relative importance of the Strength, Differentiation and Age terms. For our simulations these are set to 0.2, 0.65 and 0.15 respectively.

2.1.4 Experiments

Returning to our primary thesis in this chapter, what role does the developmental trajectory of young DGCs have on learning and memory in the DG? To investigate this

we designed a set of experiments to monitor proactive and retroactive memory interference over short and long time scale. This was achieved by training our models iteratively on highly similar patterns with the expectation that new similar patterns would be more difficult to learn (proactive interference) and similar distally learned patterns would be more easily forgotten (retroactive interference). Noisy versions of 5 prototype classes were used to represent the highly similar (but different) patterns, intended to cause interference. Interference was measured by using the Hamming distance between the input and the reconstructed patterns. We began by comparing an RBM with and without sparse coding to confirm that the sparsity constraint successfully reduces both proactive and retroactive memory interference. In Simulation 2, our neurogenesis model with and without sparse connectivity was compared with the base RBM with a static sparsity constraint to observe how our development trajectory impacts memory interference.

All models simulated in the experiments reported here used contrastive divergence with 1 step Gibbs sampling on a single layer RBM as described in chapter 1. A learning rate of 0.0025 was used for all models lacking neurogenesis and a value between 0.0025 and 0.1 was used for all models that included neurogenesis. For all sparse coding models the expected probability of activation for each hidden unit (representing the target sparseness of mature DGCs) was set to 0.05. This is a very conservative constraint as previous models and empirical studies have this set at around an order of magnitude lower, 0.004 or 0.4% (Barnes et al., 1990; Jung and McNaughton, 1993). All models had 200 visible units and 1000 hidden units in order to roughly match the relative numbers of EC and DG neurons respectively observed in rodents, as in previous models (O'Reilly and McClelland, 1994). For all experiments, each model was trained on mini-batches of 5 training patterns at a time, with 1 sample from each parent class as described below. In order to simulate rapid one-shot learning, only 1 iteration through the training set was taken. Similar to Orielly and McClelland (1994), we set the expected

probability of activation of each unit in the training and test patterns (representing the activation level of each EC input unit) to be 0.1.

Each simulated model was trained on a set of binary patterns representing input from the EC. These patterns were randomly generated, with ten percent of the elements of each pattern being active (set to 1.0) and the remainder inactive (set to 0.0). The patterns were created as random variations on a base set of prototypes, so as to create patterns that had varying degrees of similarity. Initially, five binary seed patterns were created, representing prototype patterns from 5 different classes. For each of these classes, 10 additional overlapping prototypes were generated by randomly resetting 20% percent of the original pattern. From these 55 prototypes (representing 5 classes and 11 subclasses per class), 1200 patterns were generated and partitioned into 1000 training patterns and 200 test patterns. Each pattern was created by randomly resetting another 5% of the elements in one of the subclass patterns. By generating our own dataset in this way, we were able to control the similarity between patterns and the subsequent levels of interference produced between training sessions.

While the training and testing scenarios varied between experiments, our evaluation of performance remained the same. As an estimate of the model's ability to recognize a given test pattern, the test pattern was presented to the model and the Hamming distance between the input pattern and the model's reconstruction of that test pattern was calculated. The Hamming distance was used to measure reconstruction accuracy because of its simplicity, as can be seen in equation 2.5. From there the percent match was calculated using equation 2.6, where l is the length of the V_{data} and V_{recon} . This metric serves as an approximation of the formal log-likelihood cost function for the Boltzmann model; however, it is appropriate to use an approximation to the true cost function as there are several other approximations such as brief gibbs sampling and

small mini-batches inherent to the RBM model.

$$D(V_{\text{data}}, V_{\text{recon}}) = \sum_{i=1}^n |(V_{\text{data}_i} - V_{\text{recon}_i})| \quad (2.5)$$

$$M(V_{\text{data}}, V_{\text{recon}}) = 1 - (D(V_{\text{data}}, V_{\text{recon}})/l) \quad (2.6)$$

In Simulation 1, we evaluated the contribution of sparse coding (without neurogenesis) to associative memory in the DG model. Thus, we compared the accuracy of the sparse coding RBM with the base RBM lacking a sparse coding constraint. We hypothesized that the sparse coding RBM would perform better, particularly for encoding highly similar patterns. We evaluated this and all other models on both proactive and retroactive interference. Learning a pattern that is highly similar to one the model previously learned is a source of proactive interference, potentially making it more difficult to encode the current pattern. Additionally, learning the current pattern could interfere retroactively with the model’s ability to retrieve a previously learned overlapping pattern. Thus each model was trained on groups of patterns, consisting of all training patterns from 5 of the 55 prototypes (90 patterns for a training set of 1000), one from each class, and immediately tested with the corresponding test patterns on its accuracy at reconstructing these patterns. As mentioned above, these patterns were presented to the model in mini-batches of 5 (1 example per class), and the training and test patterns had noise added to them from their prototypes by randomly resetting 5% of the elements. It was then trained on another group of 90 patterns with one prototype selected from each class, with each successive group of 90 patterns overlapping with previously learned patterns. After learning the entire set of 1000 patterns consisting of 11 groups of 90, the model was finally tested on its ability to reconstruct all test patterns from all previously learned groups to test retroactive interference.

In Simulation 2, the sparsely coded RBM with neurogenesis, with and without sparse connectivity, was compared to the sparse RBM. We were particularly interested in how the neurogenesis model would perform at encoding and recognizing similar patterns when they were encountered within the same learning session versus across different learning sessions spaced more widely in time. We therefore compared the performance of the various models across 2 conditions: 1) same-session testing in which the neurogenesis models had no neural turnover or growth, 2) multi-session testing which had both neural growth and neural turnover. The same-session testing condition was created with no simulated passage of time after training on each successive group of 90 patterns. For multi-session training, the passage of time between each block of 90 patterns was simulated by incrementing the neuron age parameter for all hidden units. As discussed previously, neural growth was simulated by incrementing the age parameter and recomputing the learning rate and sparsity cost using the Gompertz function for each hidden unit. Similarly, to simulate neural turnover, we ranked the performance of each hidden unit based on the weighted average of the synaptic strength, differentiation and age as described earlier, and reinitialized the lowest 5%. Both neural turnover and growth were performed between sessions (or groups of 90 patterns) when we incremented the age parameter of the hidden units.

Our hypothesis for same-session testing was that the neurogenesis models would perform better than the sparsely coded RBM without neurogenesis due to the presence of a few young more plastic neurons. Further, because the available pool of young excitable neurons would be constant for same-session learning, making it difficult for the model to generate distinctive traces for similar items experienced within the same context, we predicted that sparse connectivity would be particularly important for same-session learning. For multi-session testing, given that a new pool of young neurons would be available at each learning session, we hypothesized that the neurogenesis models would perform even better than they did for same-session testing. Further,

allowing some of the young neurons to mature and forcing less useful neurons to be replaced was predicted to lead to improved reconstruction accuracy with lower proactive and retroactive interference.

2.2 Results

The results from initial tests comparing the sparse coding RBM with the base RBM show a significant improvement in overall reconstruction accuracy, as can be seen in both the during and post training tests shown in figures 2.2A and 2.2B respectively, as well in the summary graph in figure 2.2D. Similarly, the sparse coding was shown to be effectively helping to increase pattern separation, as can be seen by the reduced pattern overlap of the hidden unit activations in figure 2.2C. It is noteworthy that the overlap for the base RBM was less than 30% and the slow increase in performance during training suggests that it was able to learn the sparse representation of the dataset to some extent, but not as quickly as its sparsely constrained counterpart.

The same session tests showed improved accuracy for both neurogenesis models, even without neural aging or turnover. This was expected since the initial ages of the hidden units were randomly selected, allowing the encoded characteristics of our young neurons to provide the necessary advantage. The sparse connectivity appears to provided a further advantage for same session testing as we can see in figure 2.3D. Interestingly, figure 2.3C shows that the neurogenesis models have more overlap among hidden unit activation than the normal sparse RBM, which demonstrates that the neurogenesis models are providing an opportunity to have slightly less sparse activations due to the young neurons. Another interesting pattern can be seen in figure 2.3B, which shows a kind of recency effect found in numerous memory studies (e.g., Murdock, 1962). At the same time, figure 2.3A shows the neurogenesis models have reduced

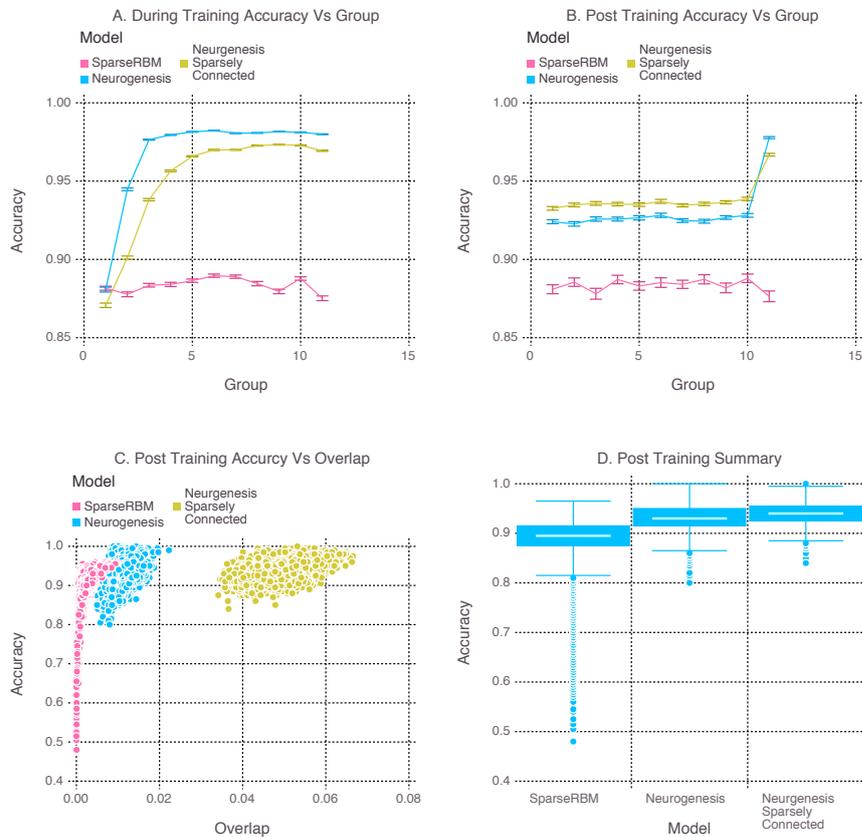


FIGURE 2.3: Simulation 2: performance of the models with and without neurogenesis and sparse connectivity on within-session pattern reconstruction tests. The models were trained sequentially on 11 groups of 90 patterns, and tested on noisy versions of these training patterns after each group to test proactive interference and after all groups had completed to test retroactive interference. **(A)** Shows proactive interference for input reconstruction accuracies during training. **(B)** Shows retroactive interference for input reconstruction accuracies on each group after training to test retroactive interference. **(C)** Shows the relationship between post training reconstruction accuracy with hidden unit activation overlap. **(D)** Shows the distribution of post training accuracy over all groups.

again shows the neurogenesis models outperforming the sparse RBM models. We can also see from figure 2.4B a recency effect and reduced proactive interference from the neurogenesis models. However, the use of neural maturation and turnover in the multi

session tests provided less benefit to overall performance than expected. While the non-sparsely connected neurogenesis model did see about a 1% increase in performance over the same session tests, the sparsely connected neurogenesis model saw no improvement and did about the same as its non-sparse counterpart. Interestingly, figure 2.4C shows that the increased overlap for the sparsely connected model is no longer present for our multi session tests and instead the overlap for the non-sparsely connected neurogenesis model has increased. This latter point suggests that the sparse connectivity and neural turnover work in equilibrium with each other depending on the learning demands required.

Simulation	Models	Means	Confidence Interval	Significant
1 - SameSession	RBM vs SparseRBM	(0.844, 0.884)	(0.03, 0.054)	*
2 - SameSession	SparseRBM vs Neurogenesis	(0.883, 0.938)	(0.035, 0.057)	*
	SparseRBM vs Neurogenesis Sparsely Connected	(0.883, 0.938)	(0.04, 0.065)	*
	Neurogenesis vs Neurogenesis Sparsely Connected	(0.93, 0.938)	(0.006, 0.01)	*
2 - MultiSession	SparseRBM vs Neurogenesis	(0.883, 0.934)	(0.04, 0.06)	*
	SparseRBM vs Neurogenesis Sparsely Connected	(0.883, 0.932)	(0.037, 0.058)	*
	Neurogenesis vs Neurogenesis Sparsely Connected	(0.934, 0.932)	(-0.004, 0.0)	

TABLE 2.1: Post training summary statistics for the 3 simulations. Mean accuracies of each pair of models and 99% bootstrapped confidence intervals around the difference between means are shown; *s indicate statistically significant differences (those with confidence intervals which do not include 0). The confidence intervals were generated by calculating the difference in mean performance of pairs of models across 20 repeated simulations with different randomly generated training and test sets. From these 20 repeated simulations, we generated 10,000 bootstrapped resamples, to obtain bootstrapped estimates of the distributions of the mean differences

In summary, the results from the neurogenesis tests showed an improvement over the sparse coding RBM in all cases with and without sparse connectivity. That being said, while the models with sparse connectivity did show better performance on the same session scenario, they showed no significant improvement for multisession tests. This suggests that the sparse connectivity of young neurons provides improved performance on pattern separation and completion tasks in the short term, but provide little

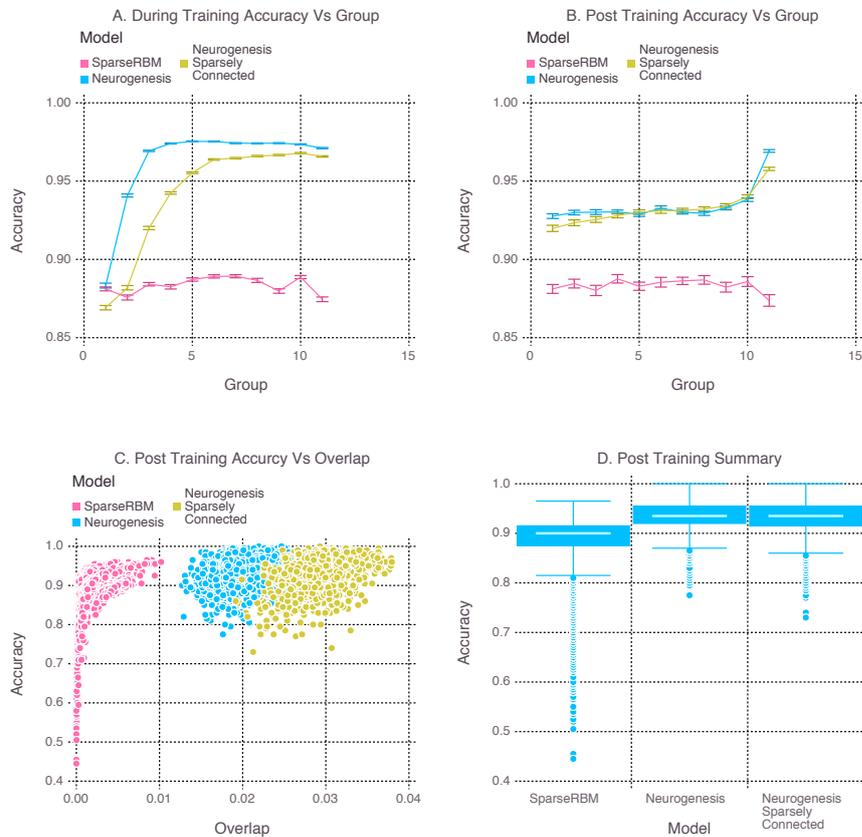


FIGURE 2.4: Simulation 2: performance of the models with and without neurogenesis and sparse connectivity on across-session pattern reconstruction tests. The models were trained sequentially on 11 groups of 90 patterns, and tested on noisy versions of these training patterns after each group to test proactive interference and after all groups had completed to test retroactive interference. **(A)** Shows proactive interference for input reconstruction accuracies during training. **(B)** Shows retroactive interference for input reconstruction accuracies on each group after training to test retroactive interference. **(C)** Shows the relationship between post training reconstruction accuracy with hidden unit activation overlap. **(D)** Shows the distribution of post training accuracy over all groups.

benefit for longer term applications. Table 4.1 shows the mean values and confidence intervals from the post training tests for each simulation.

2.3 Discussion

The main goal of this study was to investigate whether the unique characteristics of young adult-born DGCs during their maturation period, such as increased synaptic plasticity and reduced lateral inhibition (Schmidt-Hieber, Jonas, and Bischofberger, 2004; Marin-Burgin et al., 2012), contribute to learning novel, highly overlapping patterns. We were particularly interested in the potential contribution of these various properties of young neurons to interference reduction when similar patterns are encountered at short vs. long time spacings.

Previous modelling studies have shown that the sparse coding caused by lateral inhibition within the DG results in improved pattern separation (O’Reilly and McClelland, 1994) which is useful for distinguishing highly similar patterns. We reaffirmed this in simulation 1, where we compared the reconstruction of highly similar patterns for an RBM with and without a sparse coding constraint. Similar to previous studies, we found significantly better performance for the RBM using a sparse coding constraint.

Our main finding is that the models with a mixture of young and old neurons did not learn a neural code that maximized pattern separation, and yet they outperformed models with sparser, less overlapping codes but lacking neurogenesis. This may seem counter-intuitive in light of the findings of simulation 1: for models lacking neural turnover, those with a sparse coding constraint were superior. An alternative explanation for these results is that the degree of pattern separation achieved by the control model (sparsely coded RBM lacking neurogenesis) was so high (less than 0.05% pattern overlap in some cases; see figure 2.3C) that it would be impossible for models without such a sparseness constraint on the young neurons to achieve the same degree of pattern separation. However, a closer examination of the distribution of pattern separation

scores versus model performance makes this explanation seem unlikely. The RBM has the flexibility to learn any neural code that is optimal for pattern reconstruction, ranging from a sparse code to a highly distributed code. In fact, the sparse RBM and the RBM with neurogenesis produced codes with varying degrees of pattern separation in different cases (see figure 2.3C), and there was considerable overlap in the distributions of pattern separation scores for the two models. In cases where the sparse RBM achieved the highest degree of pattern separation (the bottom tail of the distribution in figure 2.3C), the sparse RBM actually performed most poorly. In other cases where the sparse RBM converged to somewhat less sparse codes, performance appeared to be asymptotically approaching about 95% (the top end of the distribution in figure 2.3C). On the other hand, models with neurogenesis achieved performance approaching 100%, in spite of a wide range of pattern separation scores; in some situations the neurogenesis models achieved comparable pattern separation to the sparse RBM but still produced superior performance. These results support our main conclusion that a heterogeneous model with a balance of mature more sparsely firing neurons and younger neurons with higher firing rates achieves superior pattern encoding relative to a purely sparse code. While our simulations suggest that the addition of younger, more hyperactive neurons strictly leads to reduced pattern separation, McAvoy et al (2015) suggest that young neurons may counter this effect via potent feedback inhibition of mature granule cells. The latter mechanism could thus compensate for the increased activity in the young neuronal population by inducing greater sparsity in the mature population. The net result of this could be a homeostatic maintenance of the overall activity level in the dentate gyrus (McAvoy, Besnard, and Sahay, 2015). In either case, pattern separation is obviously not a strict requirement for accurate neural coding. The more distributed code learned by the models with a pool of younger neurons seems to offer a good compromise between high pattern separation and high plasticity.

Sparse connectivity was found to be critical when the model attempted to encode

similar patterns encountered within a single training session. In this case, the model would not have the opportunity to generate a set of new neurons between encoding of one similar pattern and the next, and it therefore had to rely on sparse connectivity of the young neurons to generate distinct responses to similar patterns. Across a longer temporal separation, some of the young neurons would have matured while there would be additional young, more plastic neurons available to encode successive similar patterns. Thus, these additional properties of greater plasticity and higher activation were more important for separating patterns that were encountered across longer time scales. While these results shed light on the ways in which different features of young neurons may contribute to memory, there are several limitations to our models that will be addressed in the remaining chapters.

While our results are relatively robust to changes in the chosen training and evaluation methods, several limitations exist. First, the values of our hyperparameters were largely selected based on Geoffrey Hinton’s “A Practical Guide to Training Restricted Boltzmann Machines” (2012). While our results are robust to minor variations in the learning rate, decay and sparsity parameters, we do not expect changes over an order of magnitude to yield the same results. For example, changing the learning rates for DGCs from (0.0025-0.1) to (0.3-0.5) would likely not produce the same results presented here. Second, since our experiments were explicitly designed to produce interference between training sessions, we would not expect to find the same results in other real-world datasets without appropriate preprocessing. In particular, groups of highly similar patterns would need to be identified and organized into training sessions appropriately, so as to produce the same interference properties present in our synthetic dataset.

The current model using the RBM requires reciprocal connectivity between the input and output layers, whereas the known anatomy of the dentate gyrus does not support

this architecture; dentate granule cells do not project back to the EC. However, in an elaborated version of this model that will be developed in chapter 4 (Becker and Hinton, 2007), we incorporate the reciprocal connections between the CA3 and the DG (Myers and Scharfman, 2011), and between the CA3 and the EC, thus providing a natural fit of the stacked RBM architecture as described in chapter 1 to that of the hippocampal circuit. This full hippocampal circuit model will be required to explore the functional impact of young vs mature DGCs on hippocampal learning, particularly when investigating the performance changes on memory recall (pattern completion) and sequence replay tasks. Similarly, the generative characteristics of the RBM combined with this stacked architecture provide a method of simulating imagination and dreaming, along with memory reconstruction.

Finally, we modelled neurogenesis and apoptosis as one operation with the simplified replacement approach. In chapter 3, our model will be extended to treat neurogenesis and apoptosis as two independent processes for regulating the population of DGCs. We propose creating a hybrid additive and replacement model in which neurogenesis can be up or down regulated in order to better investigate the role of neurogenesis in pattern separation and completion tasks over varying time spans.

In summary, our results suggest that the developmental trajectory of adult-born DGCs may be important in explaining the role of young neurons in interference reduction at both short and long time scales. Interestingly, even though the young neurons decrease sparseness and pattern separation, they play a critical role in mitigating both retroactive and proactive interference. In order to address the limitation of the current model chapter 3 will expand it into a hybrid additive & replacement model and chapter 4 will explore the functional impact of DGC maturation on full Hippocampal learning tasks.

Chapter 3

Learning Dependent Regulation of Neurogenesis and Apoptosis

As discussed in chapter 1, computational hippocampal models that incorporate neurogenesis typically do so by either replacing existing neurons by re-randomizing their weights (e.g., Becker, 2005; Chambers et al., 2004) or by introducing new neurons with random weights (e.g., Weisz and Argibay, 2009; Weisz and Argibay, 2012). Several additional models have looked at how regulation of neurogenesis can impact learning and plasticity by simulating dynamically regulated neural turnover and replacement (Deisseroth et al., 2004; Crick and Miranker, 2006; Chambers and Conroy, 2007). However, none have modelled neurogenesis and apoptosis as independent operations. Such a model could prove extremely useful in exploring the results of recent studies examining the potential role of neurogenesis in human memory at both short and long time scales. Studies have shown that alcohol, stress & depression, age and environmental enrichment all help to regulate rates of neurogenesis (Altman and Das, 1965; Brown et al., 2003; Kempermann, Kuhn, and Gage, 1997) Likewise, a study by Déry, Goldstein, and Becker (2015) showed that lower stress and depression scores were associated with improved item recognition over larger time spans (two weeks). While the

stress & depression scores were presumed to negatively correlate with neurogenesis levels, it remains unclear as to what extent neurogenesis contributed to performance on item recognition tasks (Déry, Goldstein, and Becker, 2015). A model that can up and down regulate neurogenesis on memory encoding and cued recall tasks could be useful in testing these assumptions. Furthermore, dynamic regulation of both neurogenesis and apoptosis could help control the network size relative to changes in the input datasets; such a model could have benefits to artificial neural network (ANN) and machine learning research as well. In this chapter, we will expand our replacement neurogenesis model from chapter 2 into a more dynamic model by separating apoptosis and neurogenesis into separate processes, allowing neurogenesis and apoptosis to be a up and down regulated appropriately.

It is difficult to estimate exact rates of apoptosis and neurogenesis in the dentate gyrus as many factors govern these complex processes. However, it is generally accepted that among healthy cells, apoptosis is activity and age dependent (Hutchins and Barger, 1998; Cecchi et al., 2001). Likewise, studies have shown that alcohol, stress & depression, age and environmental enrichment all help to regulate rates of neurogenesis (Altman and Das, 1965; Brown et al., 2003; Kempermann, Kuhn, and Gage, 1997). Given these regulator mechanisms, how can we cohesively model them in an ANN so as to benefit learning? In this chapter, we will demonstrate how existing methods of hidden layer growing and pruning in ANNs can be leveraged to create such a hybrid model.

3.1 Methods

To review, ANNs learn datasets by minimizing some cost function. While different objective functions can be used depending on the type of ANN, in all cases the cost

function represents how well the network has been fit to the desired data. Similarly, the gradient of the cost function can be used to monitor learning in a ANN. While many hyperparameters in an ANN can be tuned to improve performance or find the minimum more quickly, changing the size of an ANN's hidden layer is one of the most common and effective. While increasing the size of a hidden layer will usually improve the model's fit to the training set, this has a diminishing return relative to the computational cost of running the network, and can even contribute to overfitting (Baum and Haussler, 1989; Denker et al., 1987; LeCun, 1989). As figure 3.1 demonstrates, the restricted boltzmann machine (RBM) model used in chapter 2 has exactly this problem. The computational complexity increases at a linear rate, while the performance is only increasing sublinearly.

Since the optimal hidden layer size depends on other hyperparameters as well as the dataset being learned, it is typically left to the network architect to decide what the appropriate hidden layer size should be. Unfortunately, this task is often tedious and time consuming for even the most experienced network architects. As a result, several automated methods have been proposed for determining the optimal hidden layer size, which can be grouped into two primary classes. The first class starts with a small hidden layer size and gradually adds neurons, while the second starts with a large network and prunes off neurons.

Network growing involves starting with a small hidden layer, often containing 0 or 1 neurons, and gradually adding new nodes. There are two common approaches to network growing, the cascade-correlation learning architecture (Fahlman and Lebiere, 1990) and dynamic node creation (DNC) (Ash, 1989). Cascade-correlation learning uses a special kind of feedforward network where each new neuron is trained on the network input and also receives input from all previously trained hidden neurons. Once a new hidden unit is trained, it is added to the network, and its input weights are frozen.

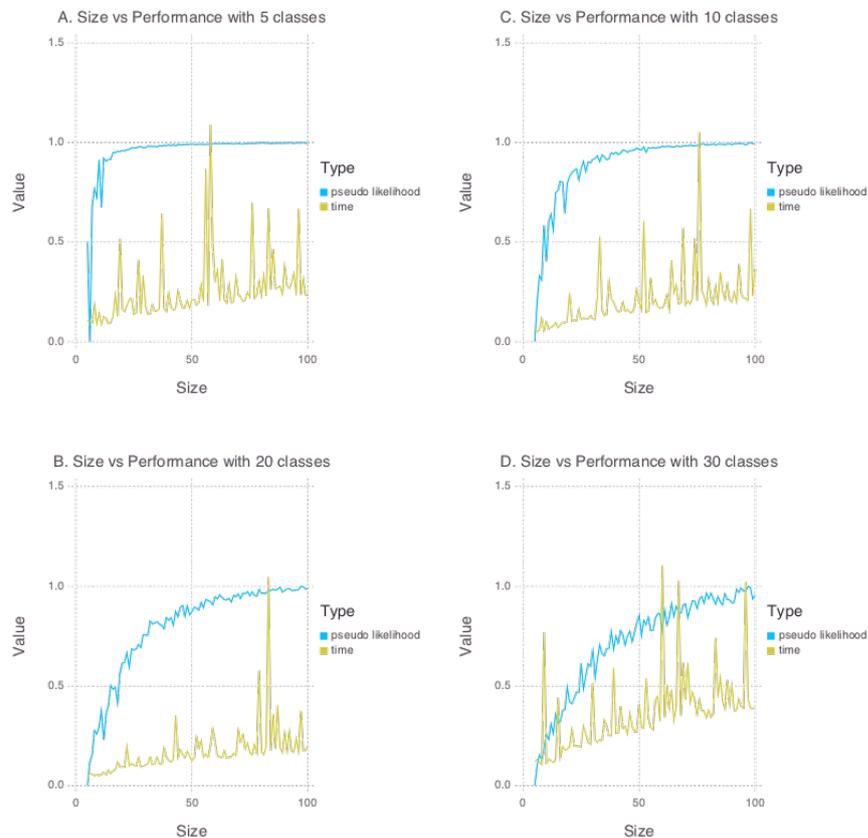


FIGURE 3.1: The pseudo-likelihood score and computational cost relative to hidden layer size.

This process is repeated until some satisfactory error rate threshold is reached. This architecture has the benefit of allowing new neurons to be added to a network without impacting existing hidden units, which eliminates the need for existing network weights to be re-adjusted and speeds up training times (Fahlman and Lebiere, 1990). DNC more intuitively trains a standard feedforward network with a single hidden unit, until the squared error converges, another hidden unit is added and the network is re-trained. Similar to cascade-correlation learning, this is repeated until some error rate threshold is reached. DNC has the benefit of being more general, in that we could easily apply it to our RBM model. DNC also tends to lead to smaller network architectures because it can utilize existing weights when re-training on new hidden units (Ash, 1989), which

better aligns with how new neurons impact existing neural connections in the dentate gyrus (DG) (McAvoy, Besnard, and Sahay, 2015). Furthermore, by adjusting our learning rates and weight decays based on neuron age, as discussed in chapter 2, we are already reducing the degree to which existing neural connections must change relative to the new neurons. More specifically, by having a higher learning rate and weight decay for young dentate granule cells (DGCs) and lower values for mature DGCs, we can ensure that the re-training steps impact the new neurons more than existing ones.

Network pruning involves starting with a large hidden layer and gradually removing neurons. While network growing attempts to add neurons until the additional neurons do not improve performance, pruning tries to remove unnecessary neurons until their removal degrades performance. The goal with network pruning is always to remove nodes with minimal negative impact on network performance. While more complex methods exist, the simplest approach is to use a metric for neural saliency, or how well a given unit is contributing to the learning in the entire network (Le Cun, Denker, and Solla, 1990). In chapter 2, we used the magnitude of the weights, the standard deviation between weights, and the neural age to rank neurons by their saliency. Essentially, if the average magnitude of the weights for a given hidden unit is low, then it should have less impact on the output. Similarly, if the standard deviation between the weights is relatively small, this is a sign that the neuron is not differentiating inputs as well. A nice property of this simplistic saliency metric is that we can clearly prioritize neurons to remove based on their stimulus specificity, synaptic strength and age.

While methods for automated hidden layer size selection exist, they only perform either network growing or pruning, but not both. In this chapter we propose a new method that can perform both network growing and pruning to model regulation of neurogenesis and apoptosis. This allows us to model the learning dependent regulation of neurogenesis and apoptosis observed in the existing literature.

3.1.1 Monitoring Learning

We can see that monitoring learning performance in both network growing and pruning are key to determining a stopping criterion. However, in order to regulate both mechanisms we will need a way of dynamically adjusting the amount to grow and prune. This method will need to adapt to new patterns unlike existing methods.

The first step in regulating growing and pruning of the hidden layer will be to determine a metric for evaluating and monitoring learning performance. While it is not unreasonable to use the reconstruction error of the RBM on the training dataset, the preferred method of monitoring learning is to calculate the pseudo-likelihood at the end of each epoch (iteration through the training set). The pseudo-likelihood in this situation is an approximation of how closely the representation of the dataset in the RBM fits the actual training set.

$$E = -a'v - \sum \log(1 + e^{b+W'v}) \quad (3.1)$$

where v is the input vector, W are the weights, and a and b are biases.

$$\text{PL}(v) = \frac{e^{-E(v_i)}}{(e^{-E(v_i)} + e^{-E(v_{i'})})} \quad (3.2)$$

where v_i is the input vector and $v_{i'}$ is the same input vector, but with a random element i flipped.

3.1.2 Convergence Method

We will be using the convergence of the pseudo-likelihood as a stopping condition, unlike the growing and pruning methods described above that used an arbitrary error

rate. This has two main benefits. First, by using convergence, we do not require any expectation of what the resulting pseudo-likelihood should be, making the method more robust to changes in the input data. Second, the convergence calculation will give us a method for dynamically deciding how many neurons to add or remove at a given time, rather than always adding or removing a single neuron.

To monitor convergence, we simply use the ratio test, also referred to as the D’Alembert’s criterion (d’Alembert, 1768).

$$r = \left| \frac{\text{PL}(v)_{n+1}}{\text{PL}(v)_n} \right| \quad (3.3)$$

when:

$r < 1$ | pseudo-likelihood is converging

$r = 1$ | pseudo-likelihood cannot converge anymore

$r > 1$ | pseudo-likelihood is diverging

So how does this relate to growing or pruning the hidden layer? If the pseudo-likelihood is still converging, adding more hidden units may still help. Conversely, if the pseudo-likelihood is not converging or even diverging, simply adding more hidden units likely will not help. However, pruning the existing layer may help by compressing the current network and making room for new neurons when the dataset changes. With these assumptions, we can formulate two simple calculations to give us the number of units to add and remove.

$$C = \lceil n \times \epsilon(1 - r) \rceil \quad (3.4)$$

$$D = ||n \times \epsilon(r)|| \quad (3.5)$$

In equations 3.4 and 3.5 n is the hidden layer size, r is our convergence ratio from 3.3, and ϵ is a maximum percentage with which to grow or prune the network.

3.1.3 Experiments

In order to evaluate our convergence based method, we first needed to demonstrate that it was successfully able to determine an appropriate layer size on different static datasets. In our first test, we repeatedly trained our neurogenesis model with identical settings on different static datasets, where each dataset had the same number of observations, but some datasets had more classes to learn than others. Between each training session we used our convergence method to determine how many neurons to add or remove from the hidden layer; the model was then recreated with the appropriate hidden layer size. We expected that networks being trained on datasets with fewer classes would require smaller hidden layers and would plateau more quickly, while the RBMs being trained on the datasets with more classes would require larger hidden layers and plateau more slowly.

In order to demonstrate that our convergence based method could model learning dependent regulation of neurogenesis and apoptosis, we also needed to demonstrate that the hidden layer size appropriately changed in relation to learning demand. In our second experiment, we followed the same aforementioned training procedure, but instead of just training on the same dataset for the entire time, we periodically changed it to observe how the convergence method adapts. We expected to see the same initial pattern as in the previous test, but with a sudden pruning followed by growing when the dataset changed.

For these experiments the models used a turnover of 10%, a learning rate of 0.1 and a momentum of 0.9, with no weight decay or sparsity constraints. The higher learning rate and momentum values were used to help speed up training and ensure the network had fully learned the dataset prior to resizing. The models had a starting hidden layer size of 150 and a visible layer size of 100. Datasets were generated by creating 5, 10 and 15 prototype patterns, which were then used to seed 1000 observations for training. The training data was repeatedly fit to the model and after each fitting session the dynamic hidden layer scaling described earlier was applied. This was performed for 100 repetitions to observe how the hidden layer size changed relative to the complexity of the input data. During each fit session the training was terminated when either the pseudo-likelihood calculation converged or when training exceeded 100 epochs. These stopping conditions were chosen to constrain each training session, while providing enough training time to minimize noise between hidden layer resizing. While the same settings were used for the second experiment, the process was repeated on 3 different datasets, without reinitializing our model in between. These dataset changes were intended to represent a novel environment with increased learning demands.

3.2 Results

Our preliminary results from the static dataset test showed appropriate hidden layer growth relative to the complexity of the input data. Specifically, figure 3.2 shows that training on fewer classes results in less hidden layer growth and a quicker plateau, while training on a dataset with more classes takes longer to plateau and leads to larger hidden layer sizes. While this very clearly fits our initial hypothesis, it is interesting to note that for the first 25 repetitions the network trained with 10 classes required a larger hidden layer than the network trained with 15 classes. This demonstrates that our method may still be sensitive to the learning rate, momentum and stopping conditions

used in training the network; however, this should be thoroughly investigated prior to these results being finalized.



FIGURE 3.2: Changes in size per iteration are shown for three different training sets with either 5, 10 or 15 different pattern classes per dataset for a single static dataset.

While our preliminary results from the dynamic dataset test showed the sudden pruning and growth we expected, it appears that the apoptosis and neurogenesis are not balanced in this experiment. We can see in figure 3.3 that for each new dataset introduced, the total number of neurons required is significantly increased. While in many circumstances this seems appropriate, it could once again suggest that our stopping conditions may be too strict.

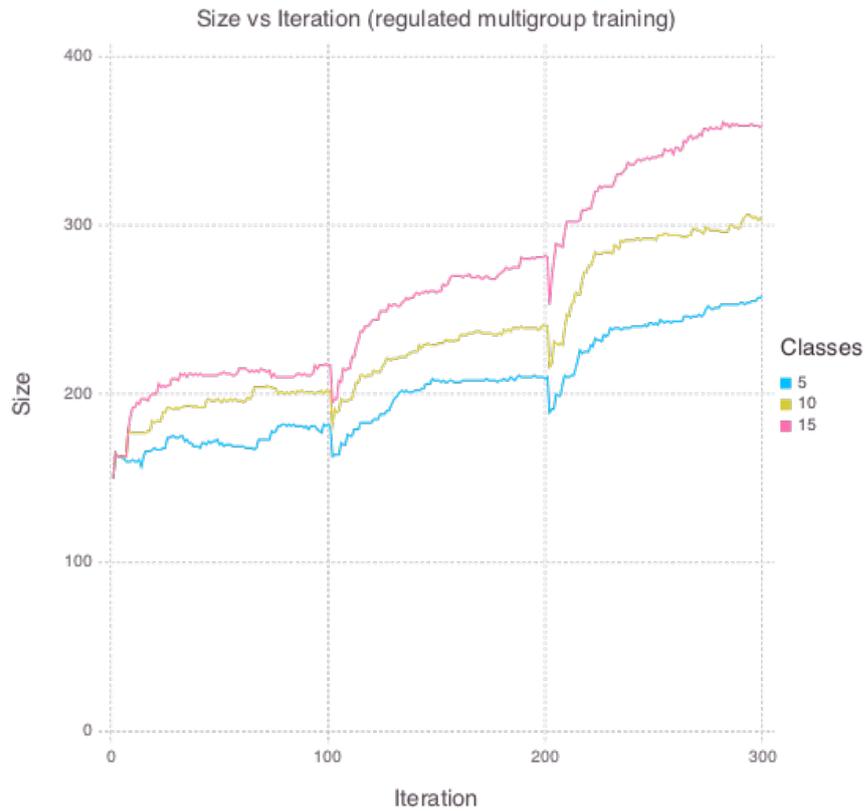


FIGURE 3.3: Changes in network size per iteration are shown for three different training sets with either 5, 10 or 15 different pattern classes per dataset. Over the 300 iterations the training dataset is changed twice (after 100 and 200 iterations) to observe how the existing network adapts to new data.

3.3 Discussion

Existing models have used either an additive or replacement method for introducing new neurons in a neurogenesis model. In this chapter, we proposed that a hybrid approach, with neurogenesis and apoptosis as independent operations, could provide a more biologically plausible model. We were particularly interested in showing whether such a method could 1) regulate neurogenesis based on the complexity of the input

data, and 2) allow the network to adjust its rates of neurogenesis and apoptosis in response to dataset changes.

We began by drawing on existing literature for growing and pruning ANN hidden layers. This revealed a common theme of using a minimum threshold over a cost function to determine when to add or remove neurons in the network. By replacing the threshold with a convergence metric as we approach the minimum, we were able to produce a method that can both grow and prune a network.

While our first experiment shows that our method is sensitive to the complexity of the input dataset, the relationship between input complexity and the resulting hidden layer size does not always hold, particularly for early repetitions. This may indicate that we needed to further tweak our hyperparameters or make the stopping criteria more flexible. Our second experiment showed that our method is also adaptable to changes in the input datasets. Once again, we noted that the rates of neurogenesis and apoptosis were not balanced, often resulting in significantly higher rates of neurogenesis over apoptosis when presented with new datasets. Again, this indicates that further tweaking of our hyperparameters or stopping conditions may be necessary.

While the current method simply uses the convergence ratio to determine how many neurons to add or remove, this can be particularly problematic in networks where the learning has already plateaued. In these cases our method would not create or destroy any neurons, despite being the correct choice. As such, our method could benefit from a stochastic offset parameter that could promote exploration once the network size has already converged. This parameter, along with the max percentage change, could be useful when examining other external neurogenesis factors such as exercise, depression and alcohol. For example, it is generally believed that exercise increases the metabolic rate, which can increase the number of neural progenitor cells (NPCs), but the learning demand is what determines whether those cells are recruited or die off (Olson et

al., 2006). In order to model this behaviour, future experiments could adjust the max percentage change parameter and include a convergence ratio offset to work collaboratively in much the same way.

In summary, we presented a novel approach to modelling learning dependent regulation of neurogenesis and apoptosis, and demonstrated how it successfully adapts to relative complexity and changes in the input dataset. Future work in this area should address the issue of exploration vs exploitation once the network has converged.

Chapter 4

Neurogenesis in a full hippocampal model

As discussed in chapter 2, a full hippocampal circuit model will be required to explore the functional impact of young vs mature dentate granule cells (DGCs) on hippocampal learning, particularly when investigating the performance changes on memory recall (pattern completion) and sequence replay tasks. Similarly, the generative characteristics of the restricted boltzmann machine (RBM) combined with this stacked architecture provide a method of simulating imagination and dreaming along with memory reconstruction. Using an existing stacked RBM approach to represent the dentate gyrus (DG) and CA layers in a full hippocampal model (Becker and Hinton, 2007; Fox and Prescott, 2010), we will investigate how our neurogenesis model performs on cued recall tasks.

4.1 Methods

4.1.1 conditional restricted boltzmann machines (CRBMs)

Recall from chapter 1, that the CA3 layer in the hippocampus has many recurrent collaterals which is believed to help with associative and temporal learning. While we are primarily investigating the impact of adult hippocampal neurogenesis (AHN) on cued recall tasks, any model of the CA3 will require a way of encoding sequences of data. While recurrent neural networks such as long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) and liquid state machines (LSMs) (Maass and Markram, 2004) have proven effective for learning such data (Graves, 2012; Schmidhuber, Wierstra, and Gomez, 2005), several techniques already exist for our base RBM model.

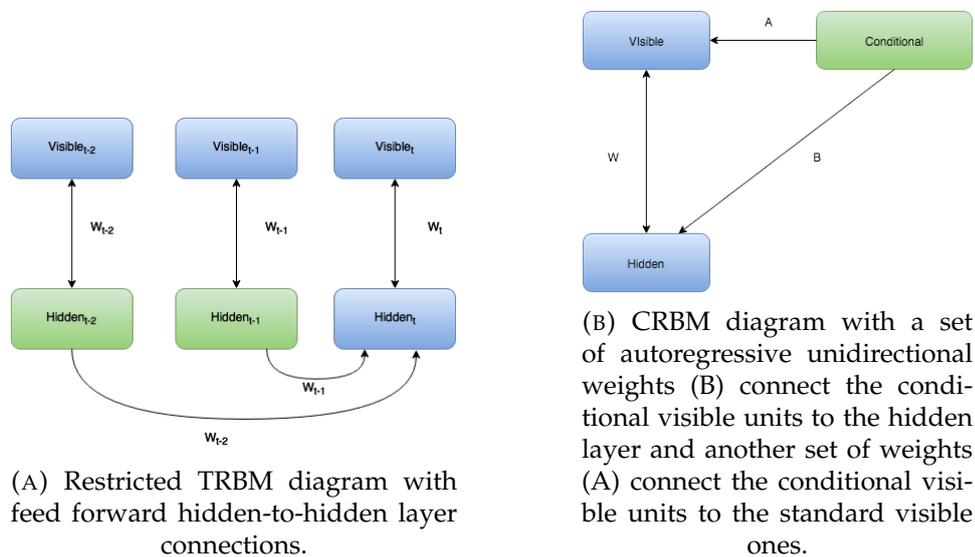


FIGURE 4.1

The TRBM extends the RBM by training a sequence of RBMs, one for each time step in a lookback, using feed forward visible-to-hidden and hidden-to-hidden connections from previous RBM time steps (Sutskever and Hinton, 2007). A common restriction

on the TRBM involves using only hidden-to-hidden temporal connections to speed up contrastive divergence (Sutskever, Hinton, and Taylor, 2008). A diagram of the TRBM architecture is provided in Figure 4.1a

The CRBM extends the RBM by adding visible-to-visible and visible-to-hidden autoregressive weights from other (or conditional) visible inputs (Taylor, Hinton, and Roweis, 2007). The idea is that the RBM’s visible units can be conditioned on other known data. This approach has proven useful in modelling timeseries data, such as video processing, where the visible input can be conditioned on the same input from previous timesteps (Taylor, Hinton, and Roweis, 2007). That being said, the CRBM is not limited to conditioning on these historical observations. For example, an electricity provider may want to model the conditional dependence between weather and load, to better predict load requirements from weather predictions. This could be achieved with a CRBM by conditioning the visible load observations on weather forecasts for temperature, humidity, wind speed & direction, etc. The flexibility of our conditional inputs will prove useful later in this chapter. A diagram of the CRBM is in figure 4.1b. Subsequently, the updated learning rule is provided in equation 4.1, and the update rules for the autoregressive weights A and B can be seen in equations 4.2 & 4.3 respectively.

$$\Delta W_{ij} = \sum_k \epsilon((v_{i,t}h_{j,t})_{\text{data}} - (v_{i,t}h_{j,t})_{\text{recon}}) \quad (4.1)$$

$$\Delta A_{k,i} = \sum_t \epsilon((v_{i,t}v_{k,<t})_{\text{data}} - (v_{i,t}v_{k,<t})_{\text{recon}}) \quad (4.2)$$

$$\Delta B_{k,j} = \sum_t \epsilon((h_{j,t}v_{k,<t})_{\text{data}} - (h_{j,t}v_{k,<t})_{\text{recon}}) \quad (4.3)$$

For our purposes, the CRBM is the most flexible and simplest method for learning sequence data with little computational overhead. While the CRBM has been an effective method of learning sequence data, it is generic enough that we can also use it to describe other conditional relationships. At the end of chapter 2 we acknowledged that the bidirectional weights of the RBM are less biologically plausible, given that there is no evidence that the DG has backprojections to the entorhinal cortex (EC). By making the DG layer a CRBM we can avoid this issue. If we invert our DG layer such that our bidirectional weights represent the mossy fibres and backprojections between the DG and CA3, then we can use the autoregressive visible-to-hidden weights to represent EC to DG connections. By doing so, the DG will be learning patterns of activation in the CA3 by conditioning on the EC. This provides an RBM based DG model that correctly accounts for the directionality of the connectivity within the hippocampal structure. Since we will not be simulating sequence learning in our experiments, our model will not be conditioning on previous timesteps. However, this would be a promising addition for future studies.

4.1.2 Stacking

Training of the multilayer model depicted in 4.2 begins by training the CA3 & CA1 layer on the EC input. The EC input is then transformed through this layer and clamped, along with the initial EC patterns, as input to the DG layer. The DG layer then learns the CA output conditioned on the initial EC patterns. Similarly, cued recall testing is triggered by transforming the degraded EC input to the CA3 & CA1 layer and passing that through to the DG layer, which generates a new activation to the CA3 & CA1 layer. Finally, the CA3 & CA1 generates the completed patterns from those activations.

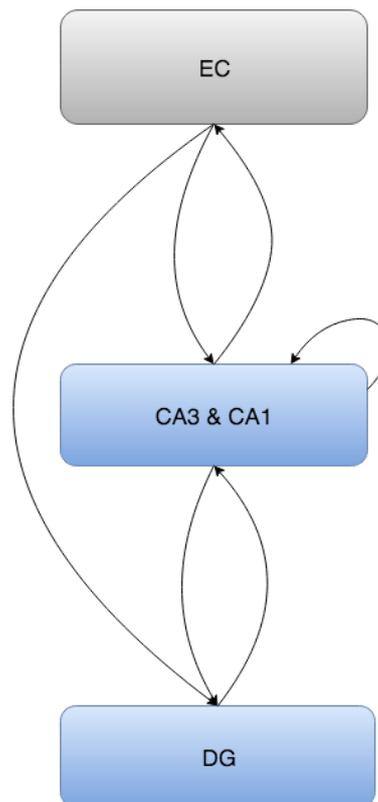


FIGURE 4.2: As a simplification of the circuitry presented in chapter 1, the DG, modelled as a CRBM, receives conditional input from the PP and visible input from CA3 backprojections. Likewise, the bidirectional weights from the backprojections to represent the mossy fibres. The CA3 & CA1 have been collapsed into 1 CRBM with visible units representing input from the EC and optional conditioning on previous timesteps. This architecture is very similar to one proposed by Becker and Hinton (2007) using TRBMs rather than CRBMs.

4.1.3 Experiments

Returning to our primary thesis in this chapter, what role does the developmental trajectory of young DGCs have on full hippocampal learning and memory? To investigate this, we used a similar approach to the one from chapter 2. We designed a set of experiments to monitor proactive and retroactive memory interference over short and long time scales. This was achieved by training our models iteratively on highly similar patterns with the expectation that new similar patterns would be more difficult to learn (proactive interference) and distally learned similar patterns would be more easily forgotten (retroactive interference). Noisy versions of 5 prototype classes were used to represent the highly similar sequences, intended to cause interference. Unlike in chapter 2, where the Hamming distance between the input and reconstruction was used to measure encoding, the distance between the source prototype and the reconstruction was used to measure cued recall. If the model is able to reconstruct the prototype with a high degree of accuracy, despite having been trained on many variations of the prototype, we can infer that it has learned the general features of the training set rather than just memorizing exemplars. We compare our hippocampal model with and without neurogenesis to observe how the developmental trajectory discussed in chapter 2 impacts cued recall tasks in our full hippocampal model. Our hypothesis is that our neurogenesis model will have better performance on cued recall tasks, with reduced proactive and retroactive memory interference across short and long time spans.

The models simulated in this experiment used contrastive divergence with 1 step Gibbs sampling on each RBM layer in the stack. A learning rate of 0.0025 was used for all layers lacking neurogenesis and a value between 0.0025 and 0.1 was used for DG layer of model that included neurogenesis. For all sparse coding models, the expected probability of activation for each hidden unit (representing the target sparseness of mature DGCs) was set to 0.05. This is a very conservative constraint as previous models

and empirical studies have this set at around an order of magnitude lower, 0.004 or 0.4% (Barnes et al., 1990; Jung and McNaughton, 1993). The initial network started with 200 EC inputs, 200 CA hidden units and 1000 DG hidden units in order to roughly match the relative numbers of EC, CA and DG neurons observed in rodents, as in previous models (O'Reilly and McClelland, 1994). However, for models that included neurogenesis, the DG hidden layer was allowed to grow and shrink according to our regulated neurogenesis and apoptosis method described in chapter 3. Since our dataset does not directly represent temporal sequences, the recurrent connections in the CA layer are ignored. For all experiments, each model was trained on mini-batches of 5 training patterns at a time, with 1 sample from each parent class as described below. In order to simulate rapid one-shot learning, only 1 iteration through the training set was taken. Similar to Orielly and McClelland (1994), we set the expected probability of activation of each unit in the training and test patterns (representing the activation level of each EC input unit) as 0.1

Each simulated model was trained on a set of binary patterns representing input from the EC. These patterns were randomly generated, with ten percent of the elements of each pattern being active (set to 1.0) and the remainder inactive (set to 0.0). The patterns were created as random variations on a base set of prototypes, so as to create patterns that had varying degrees of similarity. Initially, five binary seed patterns were created, representing prototype patterns from 5 different classes. For each of these classes, 10 additional overlapping prototypes were generated by randomly resetting 20% percent of the original pattern. From these 55 prototypes (representing 5 classes and 11 subclasses per class), 1200 patterns were generated and partitioned into 1000 training patterns and 200 test patterns. Each of these patterns were created by randomly resetting another 5% of the elements in one of the subclass patterns.

4.2 Results

The same session tests showed improved cued recall performance for models with neurogenesis. Even without neural aging or turnover, we can reduce interference in both the during and post training tests shown in Figures 4.3A and 4.3B respectively, as well as the summary graph in Figure 4.3D. Again, this was expected since the initial ages of the hidden units were randomly selected, allowing the encoded characteristics of our young neurons to provide the necessary advantage. Unsurprisingly, figure 4.3C shows higher DG hidden unit overlap for models with neurogenesis, as the more active young DGCs are less selective in their firing patterns. Interestingly, the improved performance for the neurogenesis models appears to be magnified relative to the single EC-DG layer network in chapter 2.

The multi session tests showed similar improvement to cued recall performance. Once again, figure 4.4D shows the model with neurogenesis outperforming the model without, and figure 4.4B shows a recency effect and reduced proactive interference from the neurogenesis model. However, the use of neural maturation and turnover in the multi session tests provided less benefit to overall performance than expected. Again, the improved performance for the neurogenesis models appears to be magnified relative to the single EC-DG layer network in chapter 2. Interestingly, Figure 2.4C shows a further overlap in DG hidden layer activation. This is likely due to the increased population of young DGCs relative to their mature counter parts, using our regulated neurogenesis method from chapter 3.

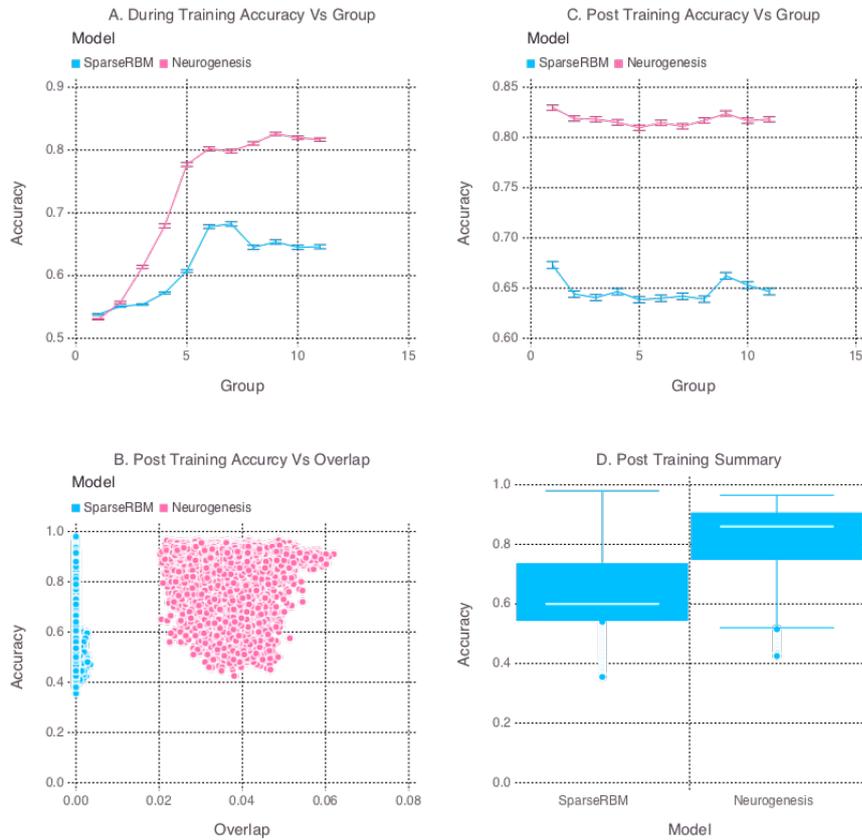


FIGURE 4.3: Performance of the models with and without neurogenesis on within-session cued recall tests. The models were trained sequentially on 11 group of 90 patterns, and tested on noisy versions of these training patterns after each group to test proactive interference and after all groups had completed to test retroactive interference. **(A)** Proactive interference for cued recall accuracies during training. **(B)** Retroactive interference for cued recall accuracies on each group after training to test retroactive interference. **(C)** The relationship between post training recall accuracy with DG hidden unit activation overlap. **(D)** The distribution of post training accuracy over all groups.

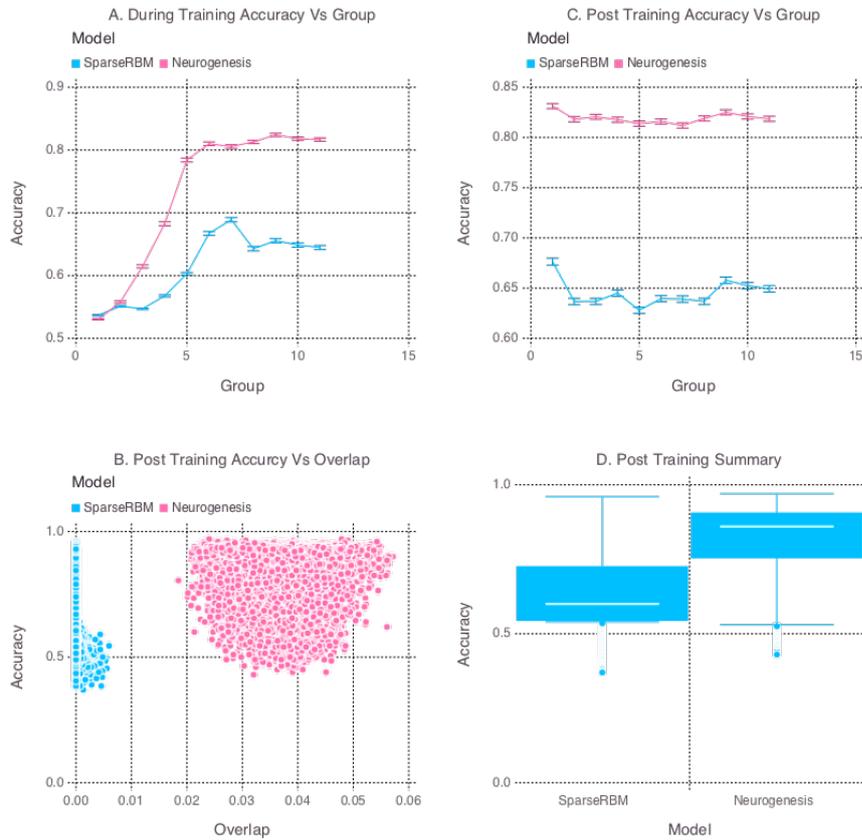


FIGURE 4.4: Performance of the models with and without neurogenesis on cross-session cued recall tests. The models were trained sequentially on 11 group of 90 patterns, and tested on noisy versions of these training patterns after each group to test proactive interference and after all groups had completed to test retroactive interference. **(A)** Proactive interference for cued recall accuracies during training. **(B)** Retroactive interference for cued recall accuracies on each group after training to test retroactive interference. **(C)** The relationship between post training recall accuracy with DG hidden unit activation overlap. **(D)** The distribution of post training accuracy over all groups.

Simulation	Models	Means	Confidence Interval	Significant
SameSession	SparseRBM vs Neurogenesis	(0.635, 0.81)	(0.148, 0.203)	*
	MultiSession	SparseRBM vs Neurogenesis	(0.645, 0.811)	(0.144, 0.19)

TABLE 4.1: Post training summary statistics for both simulations. Mean accuracies of each pair of models and 99% bootstrapped confidence intervals around the difference between means are shown; *s indicate statistically significant differences (those with confidence intervals which do not include 0). The confidence intervals were generated by calculating the difference in mean performance of pairs of models across 20 repeated simulations with different randomly generated training and test sets. From these 20 repeated simulations, we generated 10,000 bootstrapped resamples, to obtain bootstrapped estimates of the distributions of the mean differences

4.3 Discussion

In this chapter we investigated the functional impact of AHN on cued recall tasks within the hippocampal structure. To begin, we built a full hippocampal model by stacking two CRBMs. The first CRBM layer represented the CA3 & CA1 regions by accepting input from the EC. While not utilized in our experiments, this CA3 & CA1 layer can be conditioned on previous EC input, representing the recurrent collateral connections in the CA3, and allowing for learning of sequence data. The second CRBM layer represented the DG using the same neurogenesis model developed throughout this thesis, but extended to a CRBM, which is trained off the CA3 & CA1 hidden layer output and conditioned on the EC input. This modification to our neurogenesis model from the previous chapters addresses 1 of the 3 problems discussed at the end of chapter 2.

While the same evaluation method from chapter 2 was used, we measured the Hamming distance between the reconstruction and the source prototype rather than the reconstruction of the input pattern itself in order to test cued recall. Given that the evaluation and training methods from chapter 2 were largely reused, our simulation

suffers the same limitations as previously discussed. Specifically, changing the RBM hyperparameters by more than an order of magnitude is likely to yield different results. Similarly, since our experiments were explicitly designed to produce interference between training sessions, we would not expect to find the same results in other real-world datasets without appropriate preprocessing.

The primary finding from these experiments is that our neurogenesis model, specifically with the presence of young DGCs, helps with cued recall tasks in a full hippocampal model, in much the same way that we found they helped with rapid encoding in chapter 2. Myers and Scharfman (2011) argue that the backprojections from the CA3 to the DG are vital for learning within the DG. These backprojections are represented in our model by our bidirectional weights between the CA layer and the DG layer, which is simply conditioned on EC input. We believe it is these bidirectional connections which allow the young DGCs to interact with the full memory encoding, storage and recall cycle and contributing the improved cued recall performance seen in figures 4.3 and 4.4.

While we synchronously propagated the training and test patterns through the CA layer to the DG, future experiments should explore continuously training each layer, reconstructing and even generating input as asynchronous processes. In the hippocampal circuitry, the EC sends information via the PP to both the DG and CA3 concurrently, which should produce a kind of race condition between the DG and the CA3 layers. Our model simplifies this by requiring information from the EC to be processed in the CA layer in order to send the teaching signal, via backprojections, to the DG. In reality, these backprojections from the CA3 to the DG are likely being activated concurrently with the mossy fibres from the DG to the CA3. This would fit with existing theories of sequence and reverse sequence replay within the hippocampal structure (Lisman, 1999). While outside the scope of this thesis, a promising use for such a model would

be to simulate sequence replay using real place cell, grid cell and time cell recordings (O'Keefe, 1976; O'Keefe and Burgess, 2005; Burgess and O'Keefe, 2011; Eichenbaum, 2014).

Finally, we demonstrated in this chapter that our neurogenesis model from chapter 2 shows the same improved performance on cued recall tasks in a full hippocampal circuit. However, we did not demonstrate what advantage the CA3 & CA1 layer provides in memory encoding and recall. Does it help in reducing proactive and retroactive interference even without the presence of neurogenesis and young DGCs? Along with conditioning on previous input in order to model the CA3 recurrent collaterals, future experiments should identify the independent role the CA3 & CA1 layer plays on learning and memory.

In summary, we extended our neurogenesis model, described in the previous chapters, to include the full hippocampal circuit. We found models with neurogenesis had better cued recall performance than models without. These results indicate that AHN in the DG may play an important role in recall, as well as rapid encoding. Future work in this area should address the following questions: 1) How does this model behave on sequence data by conditioning on previous input patterns in CA3 & CA1 layer? 2) What advantage does the CA3 & CA1 layer play in memory encoding and recall tasks within this full hippocampal architecture? 3) How well does this model simulate real-world datasets?

Chapter 5

Conclusion

In this thesis we investigated the functional impact of adult hippocampal neurogenesis (AHN) on rapid memory encoding and recall in the hippocampal structure, focusing on the developmental trajectory of adult-generated neurons. Young dentate granule cells (DGCs) are more plastic, have less lateral inhibition, sparser connectivity and are more broadly tuned than their mature counter-parts (Schmidt-Hieber, Jonas, and Bischofberger, 2004; Snyder, Kee, and Wojtowicz, 2001; Temprana et al., 2015; Dieni et al., 2013; Piatti, Ewell, and Leutgeb, 2013; Marin-Burgin et al., 2012). However, it is unclear what impact these unique neurophysiological properties have on learning and memory. Do these neurons contribute to learning novel and highly overlapping patterns, or do they help in forgetting old ones?

We chose to use restricted boltzmann machine (RBM) based methods for our dentate gyrus (DG) and full hippocampal models. As previously discussed, despite other common neural network models, the RBM has several useful properties which require little computational overhead. Unlike most other types of artificial neural network (ANN) models, RBMs can be stacked and trained sequentially to form deep multilayer networks without relying on back-propagation. In contrast, deep networks trained by the error back-propagation learning procedure (LeCun, 1985; Rumelhart, Hinton, and

Williams, 1986a) suffer from the vanishing gradient problem (Hochreiter et al., 2001). Furthermore, these models are considered to be less biologically plausible due the requirement of non-local computations (Stocco, Lebiere, and Anderson, 2011). While deep multilayer networks can be pretrained using stacked autoencoders (Erhan et al., 2010), bypassing the vanishing gradient problem, the autoencoder still relies on back-propagation. The RBM has the additional advantage of forming a generative model of the data, allowing it to generate novel input patterns from the same data distribution that it was trained on. It thereby has the potential to simulate cognitive processes such as memory reconstruction and consolidation (Kali and Dayan, 2002), as well as imagining the future and prospective memory. Given that our objective was to see how the variability in plasticity, lateral inhibition and connectivity among a heterogenous pool of young and mature DGCs impacts memory and interference, the RBM satisfied our requirements.

We added additional constraints to the RBM learning rule to simulate the unique properties of young DGCs as they mature. The learning rule modifications that we introduced are not specific to the RBM and could easily be combined with other neural network learning rules. For example, autoencoders, multilayer perceptrons and recursive neural networks can all use the same variability in learning rate, weight decay and sparsity constraints based on the age of the neurons in the DG layer.

While our findings from chapter 2 showed that models with a mixture of young and old neurons did not learn a neural code that maximized pattern separation, they did outperform models with sparser, less overlapping codes, but lacking neurogenesis. While these results may seem counter-intuitive given that our sparse coding model performed better than a base RBM, it may suggest that a heterogeneous model with a balance of mature more sparsely firing neurons and younger neurons with higher firing rates achieves superior pattern encoding relative to a purely sparse code. McAvoy et al

(2015) suggest that young neurons may counter their increased activity via potent feedback inhibition of mature granule cells. The latter mechanism could thus compensate for the increased activity in the young neuronal population by inducing greater sparsity in the mature population. The net result of this could be a homeostatic maintenance of the overall activity level in the DG (McAvoy, Besnard, and Sahay, 2015). Furthermore, Neher et al (2015) claim that Hebbian learning in the DG does not fully support its function as a pattern separator. In either case, pattern separation is obviously not a strict requirement for accurate neural coding, and the standard model that the DG and AHN only function to help with pattern separation during memory encoding should be revisited. For now, we can say that the more distributed code learned by the models with a pool of younger neurons seems to offer a good compromise between high pattern separation and high plasticity.

In order to address the limitations of our replacement approach to neurogenesis, discussed in chapter 2, we presented a novel method for modelling learning dependent regulation of neurogenesis in chapter 3. Using changes in the pseudo-likelihood, a metric for monitoring learning in RBMs, the number of DGCs to add or remove over time was easily regulated. We demonstrated how this method adapts to the relative complexity of the dataset being learned and how introduction of new novel patterns can successfully trigger apoptosis and neurogenesis functions in order to adapt. While these are only preliminary results, we believe that this approach is both more realistic and clearly reconciles the existing additive and replacement approaches to modelling neurogenesis.

In order to investigate how our neurogenesis model behaves on cued recall tasks, we built a full hippocampal model as described in chapter 1 to include the CA3 & CA1 for associative memory. In chapter 4, we showed how our entorhinal cortex (EC)-DG

RBM could be stacked below a CA3-CA1 layer, creating a multilayer model, and extended it to handle this requirement. Interestingly, by including the CA3 sublayer, we were forced to extend our base RBM network to condition on other visible inputs with the conditional restricted boltzmann machine (CRBM). The CRBM extension allowed us to rephrase the connections from the EC to the DG as unidirectional autoregressive connections rather than our original bidirectional ones, which better fit the existing biological evidence. This stacked architecture is very similar to one proposed by Becker and Hinton (2007) which used temporal restricted boltzmann machines (TRBMs) instead of CRBMs. Fox and Prescott (2010) extended their TRBM model to use a learning method that resembles particle filtering. They focused on sequence learning on a maze tasks but did not use a real-world dataset. Furthermore, they used hand set EC-DG weights and did not account for the CA3-DG backprojections or neurogenesis. In order to test the updated neurogenesis model on cued recall, we re-ran the same experiment from chapter 2. However, instead of testing the accuracy of the model to reconstruct the presented patterns, we presented degraded patterns and asked the network to produce the original source. Our preliminary results showed a significant improvement on cued recall tasks, specifically, the properties identified in chapter 2 appear to be magnified in the stacked architecture.

While this thesis has presented a novel model of AHN in a full hippocampal network, which in turn has provided several insights about how young DGCs contribute to rapid memory encoding and recall within the hippocampus, there are still many more avenues of investigation that can be taken. The model of the young adult-born DGC maturation presented here looked specifically at changes in synaptic plasticity and lateral inhibition during the cell's developmental trajectory; however, it does not take into account temporal changes in action potential kinetics (Schmidt-Hieber, Jonas, and Bischofberger, 2004; Marin-Burgin et al., 2012). This temporal component would be a valuable contribution for future work, particularly when modelling spatio-temporal

learning and sequence replay (Karlsson and Frank, 2009). Furthermore, while the full hippocampal model presented in chapter 4 supported recurrent CA3 connections by conditioning on previous time steps, we did not utilize them in our experiments. Modelling of these recurrent CA3 connections would also be useful when simulating spatio-temporal learning and sequence replay (Karlsson and Frank, 2009). Throughout this thesis, we have noted the ability of our RBM based model to simulate imagination and dreaming by running the network in an unclamped generative mode. While these generative properties were not explored in our experiments, interesting questions arise around the impact of young DGCs on memory reactivation. For example, recent studies have shown that targeted stimulation of DGCs can induce context-specific fear expression (Liu et al., 2012; Ramirez et al., 2013), but it remains unclear how the presence of young DGCs impact this expression. We believe the generative properties of our full hippocampal neurogenesis model could provide insights on these types of interactions. Finally, it would be interesting to see how this model performs at simulating real-world datasets such as place cell, grid cell and time cell recording (O’Keefe, 1976; O’Keefe and Burgess, 2005; Burgess and O’Keefe, 2011; Eichenbaum, 2014)

In summary, we have developed a novel hybrid additive & replacement neurogenesis model that accounts for the developmental trajectory of adult-born DGCs. Our results suggest that this developmental trajectory may be important in explaining the role of young neurons in reducing memory interference at both short and long time scales. Interestingly, even though the young neurons decrease sparseness and pattern separation, they play a critical role in mitigating both retroactive and proactive interference. Future work in this area should address the following important questions: 1) How does our model perform on temporal sequence learning? 2) How do changes in the temporal dynamics of action potentials between young and mature DGCs impact these results? 3) How well does this model fit real-world recordings such as place and grid cell firing behaviour?

Bibliography

- Ackley, D. H., Hinton, G. E., and Sejnowski, T. J. (1985). "A learning algorithm for Boltzmann machines". In: *Cognitive science* 9.1, 147–169.
- Aimone, J., Wiles, J., and Gage, F. (2006). "Potential role for adult neurogenesis in the encoding of time in new memories". In: *Nature Neuroscience* 9.6, pp. 723–727.
- (2009). "Computational influence of adult neurogenesis on memory encoding". In: *Neuron* 61.2, pp. 187–202. DOI: [10.1016/j.neuron.2008.11.026](https://doi.org/10.1016/j.neuron.2008.11.026). URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2670434/>.
- Altman, J. and Das, G. (1967). "Postnatal Neurogenesis in the Guinea-pig". In: *Nature* 214.5093, pp. 1098–1101. DOI: [10.1038/2141098a0](https://doi.org/10.1038/2141098a0). URL: <http://dx.doi.org/10.1038/2141098a0>.
- Altman, J. and Das, G. D. (1965). "Post-Natal Origin of Microneurons in the Rat Brain". In: *Nature* 207.5000, pp. 953–956. DOI: [10.1038/207953a0](https://doi.org/10.1038/207953a0). URL: <http://www.nature.com/nature/journal/v207/n5000/abs/207953a0.html>.
- Ash, T. (1989). "Dynamic Node Creation in Backpropagation Networks". In: *Connection Science* 1 (4), pp. 365–375. DOI: [10.1080/09540098908915647](https://doi.org/10.1080/09540098908915647).
- Barnes, C., McNaughton, B., Mizumori, S., and Leonard, B. (1990). "Comparison of spatial and temporal characteristics of neuronal activity in sequential stages of hippocampal processing." In: *Prog Brain Res* 83, 287?300.
- Barra, A., Bernacchia, A., Santucci, E., and Contucci, P. (2012). "On the equivalence of Hopfield networks and Boltzmann Machines". In: *Neural Networks* 34, pp. 1–9. ISSN: 0893-6080. DOI: <http://dx.doi.org/10.1016/j.neunet.2012>.

BIBLIOGRAPHY

- 06.003. URL: <http://www.sciencedirect.com/science/article/pii/S0893608012001608>.
- Baum, E. and Haussler, D. (1989). "What Size Net Gives Valid Generation?" In: *Neural Computation* 1, pp. 151–160.
- Becker, S. and Hinton, G. (2007). "Caching and replay of place sequences in a Temporal Restricted Boltzmann Machine model of the hippocampus". In: Society for Neuroscience.
- Becker, S., MacQueen, G., and Wojtowicz, J. (2009). "Computational modeling and empirical studies of hippocampal neurogenesis-dependent memory: Effects of interference, stress and depression". In: *Brain Research* 1299, pp. 45–54.
- Becker, S. (2005). "A computational principle for hippocampal learning and neurogenesis". In: *Hippocampus* 15.6, pp. 722–738. ISSN: 1098-1063. DOI: [10.1002/hipo.20095](https://doi.org/10.1002/hipo.20095). URL: <http://dx.doi.org/10.1002/hipo.20095>.
- Bezanson, J., Karpinski, S., Shah, V., and contributors, other (2009–2016). *The Julia Language: A fresh approach to technical computing (MIT License)*. <https://github.com/JuliaLang/julia/>.
- Box, G. E. P. and Draper, N. R. (1987). *Empirical Model Building and Response Surfaces*. New York, NY: John Wiley & Sons, p. 424.
- Brown, J., Cooper-Kuhn, C., Kempermann, G., Van Praag, H., Winkler, J., Gage, F., and Kuhn, H. (2003). "Enriched environment and physical activity stimulate hippocampal but not olfactory bulb neurogenesis". In: *Eur J Neurosci* 17, 2042–2046.
- Brunel, N. and Hakim, V. (1999). "Fast Global Oscillations in Networks of Integrate-and-Fire Neurons with Low Firing Rates". In: *Neural Computation* 11, pp. 1621–1671.
- Burgess, N. and O'Keefe, J. (2011). "Models of place and grid cell firing and theta rhythmicity". In: *Current Opinion in Neurobiology* 21.5, pp. 734–744. ISSN: 0959-4388. DOI: <http://dx.doi.org/10.1016/j.conb.2011.07.002>. URL: <http://www.sciencedirect.com/science/article/pii/S0959438811001255>.

BIBLIOGRAPHY

- Butz, M., Lehmann, K., Dammasch, I., and Teuchert-Noodt, G. (2006). "A theoretical network model to analyse neurogenesis and synaptogenesis in the dentate gyrus." In: *Neural Netw* 19.10, pp. 1490–505.
- Butz, M., Teuchert-Noodt, G., Grafen, K., and Ooyen, A. van (2008). "Inverse relationship between adult hippocampal cell proliferation and synaptic rewiring in the dentate gyrus." In: *Hippocampus* 18.9, pp. 879–98.
- Cajal, S. Ramón y (1909). *Histologie du système nerveux de l'homme & des vertébrés*. Vol. 1. Maloine, p. 1012. URL: <http://www.biodiversitylibrary.org/item/103261>.
- Cameron, H. and McKay, R. (2001). "Adult neurogenesis produces a large pool of new granule cells in the dentate gyrus". In: *J Comp Neurol* 435, pp. 406–417.
- Carreira-Perpinan, M. A. and Hinton, G. E. (2005). *On Contrastive Divergence Learning*.
- Cecchi, G., Petreanu, L., Alvarez-Buylla, A., and Magnasco, M. (2001). "Unsupervised Learning and Adaptation in a Model of Adult Neurogenesis". In: *Journal of Computational Neuroscience* 11.2, pp. 175–182. ISSN: 0929-5313. DOI: [10.1023/A:1012849801892](https://doi.org/10.1023/A:1012849801892).
- Chambers, R. and Conroy, S. (2007). "Network modeling of adult neurogenesis: shifting rates of neuronal turnover optimally gears network learning according to novelty gradient." In: *J Cogn Neurosci* 19.1, pp. 1–12.
- Chambers, R., Potenza, M., Hoffman, R., and Miranker, W. (2004). "Simulated apoptosis/neurogenesis regulates learning and memory capabilities of adaptive neural networks". In: *Neuropsychopharmacology* 29.4, pp. 747–58.
- Chawla, M., Guzowski, J., Ramirez-Amaya, V., Lipa, P., Hoffman, K., Marriott, L., Worley, P., McNaughton, B., and Barnes, C. (2005). "Sparse, environmentally selective expression of Arc RNA in the upper blade of the rodent fascia dentata by brief spatial experience". In: *Hippocampus* 15, 579–586.
- Clelland, C., Choi, M., Romberg, C., Clemenson, G. J., Fragniere, A., Tyers, P., Jessberger, S., Saksida, L., Barker, R., Gage, F., and Bussey, T. (2009). "A Functional Role

BIBLIOGRAPHY

- for Adult Hippocampal Neurogenesis in Spatial Pattern Separation". In: *Science* 325.5937, pp. 210–213. DOI: [10.1126/science.1173215](https://doi.org/10.1126/science.1173215). URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2997634/>.
- Creer, D. J., Romberg, C., Saksida, L. M., Praag, H. van, and Bussey, T. J. (2010). "Running enhances spatial pattern separation in mice". In: *Proc Natl Acad Sci U S A* 107.5, pp. 2367–2372. DOI: [10.1073/pnas.0911725107](https://doi.org/10.1073/pnas.0911725107). URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2836679/>.
- Crick, C. and Miranker, W. (2006). "Apoptosis, neurogenesis, and information content in Hebbian networks". English. In: *Biological Cybernetics* 94.1, pp. 9–19. ISSN: 0340-1200. DOI: [10.1007/s00422-005-0026-8](https://doi.org/10.1007/s00422-005-0026-8). URL: <http://dx.doi.org/10.1007/s00422-005-0026-8>.
- Cuneo, J., Quiroz, N., Weisz, V., and Argibay, P. (2012). "The computational influence of neurogenesis in the processing of spatial information in the dentate gyrus". In: *Scientific Reports* 2.735.
- d'Alembert, J. (1768). *Opuscules*. Vol. 5, pp. 171–183.
- Deisseroth, K., Singla, S., Toda, H., Monje, M., Palmer, T., and Malenka, R. (2004). "Excitation-neurogenesis coupling in adult neural stem/progenitor cells". In: *Neuron* 42.4, pp. 535–52.
- Denker, J., Schwartz, D., Wittner, B., Solla, S., Howard, R., Jackel, L., and Hopfield, J. (1987). "Large Automatic Learning, Rule Extraction and Generalization". In: *Complex Systems* 1, pp. 877–922.
- Déry, N., Goldstein, A., and Becker, S. (2015). "A role for adult hippocampal neurogenesis at multiple time scales: A study of recent and remote memory in humans". In: *Behavioral Neuroscience (in press)*.
- Déry, N., Pilgrim, M., Gibala, M., Gillen, J., Wojtowicz, J., MacQueen, G., and Becker, S. (2013). "Adult hippocampal neurogenesis reduces memory interference in humans:

BIBLIOGRAPHY

- opposing effects of aerobic exercise and depression". In: *Frontiers in Neurogenesis* 7:66. DOI:10.3389/fnins.2013.00066 10.3389 pp.66.
- Dieni, C., Nietz, A., Panichi, R., Wadiche, J., and Overstreet-Wadiche, L. (2013). "Distinct determinants of sparse activation during granule cell maturation". In: *J. Neurosci.* 33, pp. 19131–19142.
- Eichenbaum, H. (2014). "Time cells in the hippocampus: a new dimension for mapping memories". In: *Nat Rev Neurosci* 15.11, pp. 732–744. DOI: [10.1038/nrn3827](https://doi.org/10.1038/nrn3827). URL: <http://dx.doi.org/10.1038/nrn3827>.
- Elmore, S. (2007). "Apoptosis: A Review of Programmed Cell Death". In: *Toxicol Pathol* 35.4, pp. 495–516. DOI: [10.1080/01926230701320337](https://doi.org/10.1080/01926230701320337). URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2117903/>.
- Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., and Bengio, S. (2010). "Why Does Unsupervised Pre-training Help Deep Learning?" In: *Journal of Machine Learning Research* 11, pp. 625–660.
- Eriksson, P., Perfilieva, E., Bjork-Eriksson, T., Alborn, A., Nordborg, C., Peterson, D., and Gage, F. (1998). "Neurogenesis in the adult human hippocampus". In: *Nature Medicine* 4.11, pp. 1313–1317. DOI: [10.1038/3305](https://doi.org/10.1038/3305).
- Fahlman, S. and Lebiere, C. (1990). "The Cascade-Correlation Learning Architecture". In: *Advances in Neural Information Processing Systems* 2. Morgan Kaufmann, pp. 524–532.
- Finnegan, R. (2015–2016). *Masters Thesis - Computational Modelling of Adult Hippocampal Neurogenesis* (MIT "Expat" License). <https://github.com/Rory-Finnegan/thesis>.
- Finnegan, R. and Becker, S. (2015). "Neurogenesis paradoxically decreases both pattern separation and memory interference". In: *Frontiers in Systems Neuroscience* 9.136. ISSN: 1662-5137. DOI: [10.3389/fnsys.2015.00136](https://doi.org/10.3389/fnsys.2015.00136). URL: <http://www.>

BIBLIOGRAPHY

- [frontiersin.org/systems_neuroscience/10.3389/fnsys.2015.00136/abstract](https://www.frontiersin.org/systems_neuroscience/10.3389/fnsys.2015.00136/abstract).
- Fox, C. and Prescott, T. (2010). "Hippocampus as unitary coherent particle filter". In: *The International Joint Conference on Neural Networks (IJCNN)*. DOI: [10.1109/IJCNN.2010.5596681](https://doi.org/10.1109/IJCNN.2010.5596681).
- Freund, Y. and Haussler, D. (1992). "Unsupervised learning of distributions on binary vectors using two layer networks". In: *Advances in Neural Information Processing Systems 4*. San Mateo, CA.: Morgan Kaufmann, pp. 912–919.
- Gluck, M. and Myers, C. (1993). "Hippocampal mediation of stimulus representation: a computational theory". In: *Hippocampus* 3, 491–516.
- Gompertz, B. (1832). "On the Nature of the Function Expressive of the Law of Human Mortality, and on a New Method of Determining the Value of Life Contingencies". In: *Phil. Trans. Roy. Soc. London* 123, pp. 513–585.
- Gould, E., Tanapat, P., McEwen, B. S., Flgge, G., and Fuchs, E. (1998). "Proliferation of granule cell precursors in the dentate gyrus of adult monkeys is diminished by stress". In: *Proc Natl Acad Sci U S A* 95.6, 3168–3171. URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC19713/>.
- Graves, A. (2012). *Supervised sequence labelling with recurrent neural networks*. Vol. 385. Springer.
- Gray, H. (1918). *Anatomy of the human body*. Lea & Febiger. ISBN: 1-58734-102-6. URL: www.bartleby.com/107/.
- Hebb, D. (2002). *The Organization of Behavior: A Neuropsychological Theory*. Taylor & Francis. ISBN: 9781410612403. URL: <https://books.google.ca/books?id=gUtwMochAI8C>.
- Hinton, G. (2002). "Training products of experts by minimizing contrastive divergence". In: *Neural Computation* 14.8, pp. 1771–1800.

BIBLIOGRAPHY

- Hinton, G. (2012). "A Practical Guide to Training Restricted Boltzmann Machines". In: *Neural Networks: Tricks of the Trade*. Ed. by G. Montavon, G. Orr, and K.-R. Müller. Vol. 7700. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 599–619. ISBN: 978-3-642-35288-1. DOI: [10.1007/978-3-642-35289-8_32](https://doi.org/10.1007/978-3-642-35289-8_32). URL: http://dx.doi.org/10.1007/978-3-642-35289-8_32.
- Hinton, G. E. and Osindero, S. (2006). "A fast learning algorithm for deep belief nets". In: *Neural Computation* 18.7, pp. 1527–1554.
- Hochreiter, S. and Schmidhuber, J. (1997). "Long short-term memory". In: *Neural Computation* 9.8, 1735–1780. DOI: [doi:10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735).
- Hochreiter, S., Bengio, Y., Frasconi, P., and Schmidhuber, J. (2001). *Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies*.
- Hodgkin, A. L. and Huxley, A. F. (1952). "A quantitative description of membrane current and its application to conduction and excitation in nerve". In: *J Physiol* 117.4, pp. 500–544. ISSN: 0022-3751.
- Hopfield, J. (2014). "Neural networks and physical systems with emergent collective computational abilities". In: *Spin Glass Theory and Beyond*. WORLD SCIENTIFIC. Chap. 43, pp. 411–415. DOI: [10.1142/9789812799371_0043](https://doi.org/10.1142/9789812799371_0043). eprint: http://www.worldscientific.com/doi/pdf/10.1142/9789812799371_0043. URL: http://www.worldscientific.com/doi/abs/10.1142/9789812799371_0043.
- Hutchins, J. B. and Barger, S. W. (1998). "Why neurons die: Cell death in the nervous system". In: *The Anatomical Record* 253.3, pp. 79–90. ISSN: 1097-0185. DOI: [10.1002/\(SICI\)1097-0185\(199806\)253:3<79::AID-AR4>3.0.CO;2-9](https://doi.org/10.1002/(SICI)1097-0185(199806)253:3<79::AID-AR4>3.0.CO;2-9). URL: [http://dx.doi.org/10.1002/\(SICI\)1097-0185\(199806\)253:3<79::AID-AR4>3.0.CO;2-9](http://dx.doi.org/10.1002/(SICI)1097-0185(199806)253:3<79::AID-AR4>3.0.CO;2-9).
- Johnston, D and Amaral, D. (1998). "Hippocampus". In: *The Synaptic Organization of the Brain*. 4th ed. New York: Oxford University Press.

BIBLIOGRAPHY

- Jung, M. and McNaughton, B. (1993). "Spatial selectivity of unit activity in the hippocampal granular layer". In: *Hippocampus* 3.2, pp. 165–182.
- Kali, S. and Dayan, P. (2002). "Replay, Repair and Consolidation". In: *Advances in Neural Information Processing Systems* 15. Ed. by S. Thrun and K. Obermayer. Cambridge, MA: MIT Press, pp. 19–26. URL: <http://books.nips.cc/papers/files/nips15/CS03.pdf>.
- Karlsson, M. P. and Frank, L. M. (2009). "Awake replay of remote experiences in the hippocampus". In: *Nature Neuroscience* 12.7, pp. 913–918. DOI: [10.1038/nn.2344](https://doi.org/10.1038/nn.2344). URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2750914/>.
- Kempermann, G., Kuhn, H., and Gage, F. (1997). "More hippocampal neurons in adult mice living in an enriched environment". In: *Nature* 386, pp. 493–495.
- Le Cun, Y., Denker, J. S., and Solla, S. A. (1990). "Optimal Brain Damage". In: *Advances in Neural Information Processing Systems*. Morgan Kaufmann, pp. 598–605.
- LeCun, Y. (1985). "A learning scheme for asymmetric threshold networks". In: *Proceedings of Cognitive 85*. Paris, France, pp. 599–604.
- (1989). *Generalization and Network Design Strategies*. Tech. rep. University of Toronto.
- Lehmann, K., Butz, M., and Teuchert-Noodt, G. (2005). "Offer and demand: proliferation and survival of neurons in the dentate gyrus." In: *Eur J Neurosci* 21.12, pp. 3205–16.
- Lipton, Z. C., Berkowitz, J., and Elkan, C. (2015). "A Critical Review of Recurrent Neural Networks for Sequence Learning". In: *ArXiv e-prints*. arXiv: [1506.00019 \[cs.LG\]](https://arxiv.org/abs/1506.00019).
- Lisman, J. (1999). "Relating hippocampal circuitry to function: Recall of memory sequences by reciprocal dentate-CA3 interactions". In: *Neuron* 22, 233–242. URL: <http://www.ncbi.nlm.nih.gov/pubmed/10069330>.
- Liu, X., Ramirez, S., Pang, P. T., Puryear, C. B., Govindarajan, A., Deisseroth, K., and Tonegawa, S. (2012). "Optogenetic stimulation of a hippocampal engram activates fear memory recall". In: *Nature* 484.7394, pp. 381–385. DOI: [10.1038/nature11028](https://doi.org/10.1038/nature11028).

BIBLIOGRAPHY

- Luu, P., Sill, O., Gao, L., Becker, S., Wojtowicz, J., and Smith, D. (2012). "The Role of Adult Hippocampal Neurogenesis in Reducing Interference". In: *Behavioural Neuroscience* 126.3, pp. 381–391.
- Maass, W. and Markram, H. (2004). "On the computational power of circuits of spiking neurons". In: *Journal of Computer and System Sciences* 69.4, pp. 593–616. ISSN: 0022-0000. DOI: <http://dx.doi.org/10.1016/j.jcss.2004.04.001>. URL: <http://www.sciencedirect.com/science/article/pii/S0022000004000406>.
- Marin-Burgin, A., Mongiat, L. A., Pardi, M. B., and Schinder, A. F. (2012). "Unique processing during a period of high excitation/inhibition balance in adult-born neurons". In: *Science* 335, pp. 1238–1242.
- Marr, A. (1971). "Simple memory: A theory for archicortex". In: *Philosophical Transactions of the Royal Society of London* 262 (Series B), pp. 23–81.
- McAvoy, K., Besnard, A., and Sahay, A. (2015). "Adult hippocampal neurogenesis and pattern separation in DG: a role for feedback inhibition in modulating sparseness to govern population-based coding." In: *Front. Syst. Neurosci.* 9.120. DOI: [10.3389/fnsys.2015.00120](https://doi.org/10.3389/fnsys.2015.00120). URL: <http://journal.frontiersin.org/article/10.3389/fnsys.2015.00120/abstract>.
- McClelland, J. L., McNaughton, B. L., and O'Reilly, R. C. (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". In: *Psychological Review* 102.3, pp. 419–457.
- McClelland, J., Rumelhart, D., and PDP Research Group, the (1986). Cambridge, MA: MIT Press.
- McNaughton, B. L. and Morris, R. G. M. (1987). "Hippocampal synaptic enhancement and information storage within a distributed memory systems". In: *Trends in Neurosciences* 10, pp. 408–415.

BIBLIOGRAPHY

- Murdock, B (1962). "The serial position effect of free recall". In: *Journal of Experimental Psychology* 64.5, pp. 482–488. DOI: [10.1037/h0045106](https://doi.org/10.1037/h0045106). URL: http://www.researchgate.net/publication/232580580_The_serial_position_effect_of_free_recall.
- Myers, C. E. and Scharfman, H. E. (2011). "Pattern Separation in the Dentate Gyrus: A Role for the CA3 Backprojection". In: *Hippocampus* 21.11, pp. 1190–1215. DOI: [10.1002/hipo.20828](https://doi.org/10.1002/hipo.20828). URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2976779/>.
- Myers, C. and Scharfman, H. (2009). "A role for hilar cells in pattern separation in the dentate gyrus: a computational approach". In: *Hippocampus* 19, pp. 321–337.
- Nair, V. and Hinton, G. E. (2009). "3-d object recognition with deep belief nets". In: *Advances in Neural Information Processing Systems* 22, pp. 1339–1347.
- Neher, T., Cheng, S., and Wiskott, L. (2015). "Memory Storage Fidelity in the Hippocampal Circuit: The Role of Subregions and Input Statistics". In: *PLoS Computational Biology* 11.5. DOI: [10.1371/journal.pcbi.1004250](https://doi.org/10.1371/journal.pcbi.1004250).
- Olson, A. K., Eadie, B. D., Ernst, C., and Christie, B. R. (2006). "Environmental enrichment and voluntary exercise massively increase neurogenesis in the adult hippocampus via dissociable pathways". In: *Hippocampus* 16.3, pp. 250–260. ISSN: 1098-1063. DOI: [10.1002/hipo.20157](https://doi.org/10.1002/hipo.20157). URL: <http://dx.doi.org/10.1002/hipo.20157>.
- O'Reilly, R. C. and McClelland, J. L. (1994). "Hippocampal conjunctive encoding, storage, and recall: Avoiding a trade-off". In: *Hippocampus* 4.6, pp. 661–682. ISSN: 1098-1063. DOI: [10.1002/hipo.450040605](https://doi.org/10.1002/hipo.450040605). URL: <http://dx.doi.org/10.1002/hipo.450040605>.
- Oswald, A.-M. M. and Reyes, A. D. (2008). "Maturation of Intrinsic and Synaptic Properties of Layer 2/3 Pyramidal Neurons in Mouse Auditory Cortex". In: *Journal of*

BIBLIOGRAPHY

- Neurophysiology* 99.6, pp. 2998–3008. ISSN: 0022-3077. DOI: [10.1152/jn.01160.2007](https://doi.org/10.1152/jn.01160.2007).
- O'Keefe, J. (1976). "Place units in the hippocampus of the freely moving rat". In: *Exp Neurol* 51, 78–109.
- O'Keefe, J. and Burgess, N (2005). "Dual phase and rate coding in hippocampal place cells: Theoretical significance and relationship to entorhinal grid cells". In: *Hippocampus* 15, 853–866.
- Piatti, V. C., Ewell, L. A., and Leutgeb, J. K. (2013). "Neurogenesis in the dentate gyrus: carrying the message or dictating the tone". In: *Frontiers in Neuroscience* 7.50. DOI: [10.3389/fnins.2013.00050](https://doi.org/10.3389/fnins.2013.00050). URL: <http://journal.frontiersin.org/article/10.3389/fnins.2013.00050/abstract>.
- Ramirez, S., Liu, X., Lin, P.-A., Suh, J., Pignatelli, M., Redondo, R. L., Ryan, T. J., and Tonegawa, S. (2013). "Creating a False Memory in the Hippocampus". In: *Science* 341.6144, pp. 387–391. ISSN: 0036-8075. DOI: [10.1126/science.1239073](https://doi.org/10.1126/science.1239073). eprint: <http://science.sciencemag.org/content/341/6144/387.full.pdf>. URL: <http://science.sciencemag.org/content/341/6144/387>.
- Rolls, E. T. and Treves, A. (1998). *Neural Networks and Brain Function*. Oxford University Press.
- Rolls, E. T. (1987). "Information representation, processing and storage in the brain: analysis at the single neuron level". In: *The Neural and Molecular Bases of Learning*, pp. 503–540.
- Rosenblatt, F. (1962). *Principles of neurodynamics: perceptrons and the theory of brain mechanisms*. Report (Cornell Aeronautical Laboratory). Spartan Books.
- Rumelhart, D., Hinton, G., and Williams, R. (1986a). "Learning internal representations by back-propagating errors". In: *Nature* 323, pp. 533–536.

BIBLIOGRAPHY

- Rumelhart, D., Hinton, G., and Williams, R. (1986b). "Learning representations by back-propagating errors". In: *Nature* 323.6088, pp. 533–536. DOI: [10.1038/323533a0](https://doi.org/10.1038/323533a0). URL: <http://dx.doi.org/10.1038/323533a0>.
- Rumelhart, D., McClelland, J., and PDP Research Group, the (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*. Cambridge, MA: MIT Press.
- Scharfman, H. E. (2007). "The CA3 "Backprojection" to the Dentate Gyrus". In: *Progress in brain research* 163, pp. 627–637. DOI: [10.1016/S0079-6123\(07\)63034-9](https://doi.org/10.1016/S0079-6123(07)63034-9).
- Schatz, C. J. (1992). "The Developing Brain". In: *Scientific American* 267.3, pp. 60–67. ISSN: ISSN-0036-8733. URL: <http://cognitron.psych.indiana.edu/busey/q551/PDFs/MindBrainCh2.pdf>.
- Schmidhuber, J., Wierstra, D., and Gomez, F. (2005). "Evolino: Hybrid Neuroevolution / Optimal Linear Search for Sequence Learning". In: Edinburgh: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI), 853?858.
- Schmidt-Hieber, C., Jonas, P., and Bischofberger, J. (2004). "Enhanced synaptic plasticity in newly generated granule cells of the adult hippocampus". In: *Nature* 429, pp. 184–187. DOI: [10.1038/nature02553](https://doi.org/10.1038/nature02553). URL: <http://www.nature.com/nature/journal/v429/n6988/full/nature02553.html>.
- Searle, J. (1980). "Minds, Brains, and Programs". In: *The Behavioral and Brain Sciences* 3, pp. 417–457.
- Smolensky, P. (1986). "Information processing in dynamical systems: Foundations of harmony theory". In: *Parallel Distributed Processing*. Ed. by D. E. Rumelhart and J. L. McClelland. Vol. 1. Cambridge: MIT Press. Chap. 6, pp. 194–281.
- Snyder, J., Kee, N., and Wojtowicz, J. (2001). "Effects of adult neurogenesis on synaptic plasticity in the rat dentate gyrus". In: *J Neurophysiol.* 85.6, pp. 2423–2431.
- Stocco, A., Lebiere, C., and Anderson, J. R. (2011). "Conditional Routing of Information to the Cortex: A Model of the Basal Ganglia's Role in Cognitive Coordination". In:

BIBLIOGRAPHY

- Psychol Rev.* 117.2, pp. 541–574. DOI: [10.1037/a0019077](https://doi.org/10.1037/a0019077). URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3064519/>.
- Sutskever, I. and Hinton, G. (2007). “Learning multilevel distributed representations for highdimensional sequences”. In: Eleventh International Conference on Artificial Intelligence and Statistics, pp. 544–551.
- Sutskever, I., Hinton, G., and Taylor, G. (2008). “The recurrent temporal restricted boltzmann machine”. In: *Advances in Neural Information Processing Systems*, p. 21.
- Taylor, G., Hinton, G., and Roweis, S. (2007). “Modeling human motion using binary latent variables”. In: *NIPS 19*, pp. 1345–1352. URL: http://www.cs.toronto.edu/~fritz/absps/fcrbm_icml.pdf.
- Temprana, S., Mongiat, L., Yang, S., Trincherro, M., Alvarez, D., Kropff, E., Giacomini, D., Beltramone, N., Lanuza, G., and Schinder, A. (2015). “Delayed Coupling to Feedback Inhibition during a Critical Period for the Integration of Adult-Born Granule Cells”. In: *Neuron* 85.1, pp. 116–130.
- Towhidul Islama Denzil G. Fiebigb, N. M. (2002). “Modelling multinational telecommunications demand with limited data”. In: *International Journal of Forecasting* 18.4, pp. 605–624. DOI: [10.1016/S0169-2070\(02\)00073-0](https://doi.org/10.1016/S0169-2070(02)00073-0). URL: <http://www.sciencedirect.com/science/article/pii/S0169207002000730>.
- Treves, A. and Rolls, E. T. (1992). “Computational constraints suggest the need for two distinct input systems to the hippocampal CA3 network”. In: *Hippocampus* 2, pp. 189–200.
- Turing, A. (1950). “Computing Machinery and Intelligence”. In: *Mind* 59.236, pp. 433–460.
- Wang, S., Scott, B. W., and Wojtowicz, J. M. (2000). “Heterogenous properties of dentate granule neurons in the adult rat”. In: *Journal of Neurobiology* 42.2, pp. 248–257. ISSN: 1097-4695. DOI: [10.1002/\(SICI\)1097-4695\(20000205\)42:2<248::AID-](https://doi.org/10.1002/(SICI)1097-4695(20000205)42:2<248::AID-)

BIBLIOGRAPHY

- NEU8>3.0.CO;2-J. URL: [http://dx.doi.org/10.1002/\(SICI\)1097-4695\(20000205\)42:2<248::AID-NEU8>3.0.CO;2-J](http://dx.doi.org/10.1002/(SICI)1097-4695(20000205)42:2<248::AID-NEU8>3.0.CO;2-J).
- Weisz, V. and Argibay, P. (2012). "Neurogenesis interferes with the retrieval of remote memories: forgetting in neurocomputational terms". In: *Cognition* 125.1, pp. 13–25.
- Weisz, V. I. and Argibay, P. F. (2009). "A putative role for neurogenesis in neurocomputational terms: Inferences from a hippocampal model". In: *Cognition* 112.2, pp. 229–240. DOI: [10.1016/j.cognition.2009.05.001](https://doi.org/10.1016/j.cognition.2009.05.001). URL: <http://www.sciencedirect.com/science/article/pii/S0010027709001061>.
- Werbos, P. J. (1988). "Generalization of backpropagation with application to a recurrent gas market model". In: *Neural Networks* 1.4, 339–356. DOI: [10.1016/0893-6080\(88\)90007-X](https://doi.org/10.1016/0893-6080(88)90007-X).
- Winocur, G., Becker, S., Luu, P., Rosenzweig, S., and Wojtowicz, J. (2012). "Adult Hippocampal Neurogenesis and Memory Interference". In: *Behavioural Brain Research* 277.2 Special Issue S1. doi:10.1016/j.bbr.2011.05.032, pp. 464–469.
- Wiskott, L., Rasch, M., and Kempermann, G. (2006). "A functional hypothesis for adult hippocampal neurogenesis: avoidance of catastrophic interference in the dentate gyrus". In: *Hippocampus* 16.3, pp. 329–343.
- Wojtowicz, J. M. (2006). "Irradiation as an experimental tool in studies of adult neurogenesis". In: *Hippocampus* 16.3, pp. 261–266. ISSN: 1098-1063. DOI: [10.1002/hipo.20158](https://doi.org/10.1002/hipo.20158). URL: <http://dx.doi.org/10.1002/hipo.20158>.
- Wu, Y., Hu, J., Wu, W., Zhou, Y., and Du, K. (2012). "Storage Capacity of the Hopfield Network Associative Memory". In: *Intelligent Computation Technology and Automation, International Conference on* 0, pp. 330–336. DOI: <http://doi.ieeecomputersociety.org/10.1109/ICICTA.2012.89>.
- Zhabinski, A., Finnegan, R., and contributors, other (2015–2016). *Boltzmann.jl - Restricted Boltzmann Machines in Julia* (MIT "Expat" License). <https://github.com/dfdx/Boltzmann.jl/>.

BIBLIOGRAPHY

Zhao, C., Teng, E. M., Summers, R. G. J., Ming, G. L., and Gage, F. H. (2006). "Distinct morphological stages of dentate granule neuron maturation in the adult mouse hippocampus". In: *J. Neurosci.* 26, pp. 3–11.

Zwietering, M., Jongenburger, I., Rombouts, F., and Riet K., van 't (1990). "Modeling of the bacterial growth curve". In: *Applied and Environment Microbiology* 56.6, pp. 1875–81. URL: <http://www.ncbi.nlm.nih.gov/pubmed/16348228?dopt=Abstract>.