Network: Computation in Neural Systems 20(3):137-161 (2010)

Additive neurogenesis as a strategy for avoiding interference in a sparsely-coding dentate gyrus

Peter A. Appleby* and Laurenz Wiskott

Institute for Theoretical Biology
Humboldt Universität zu Berlin
Invalidenstrasse 43
10115, Berlin
Germany

*To whom correspondence should be addressed: Tel.: +49 302 0938633, Fax.: +49 302 0938801, E.-mail: p.appleby@biologie.hu-berlin.de.

Abstract

Recently we presented a model of additive neurogenesis in a linear, feedforward neural network that performed an encoding-decoding memory task in a changing input environment. Growing the neural network over time allowed the network to adapt to changes in input statistics without disrupting retrieval properties, and we proposed that adult neurogenesis might fulfil a similar computational role in the dentate gyrus of the hippocampus. Here we explicitly evaluate this hypothesis by examining additive neurogenesis in a simplified hippocampal memory model. The model incorporates a divergence in unit number from the entorhinal cortex to the dentate gyrus and sparse coding in the dentate gyrus, both notable features of hippocampal processing. We evaluate two distinct adaptation strategies; neuronal turnover, where the network is of fixed size but units may be deleted and new ones added, and additive neurogenesis, where the network grows over time, and quantify the performance of the network across the full range of adaptation levels from zero in a fixed network to one in a fully adapting network. We find that additive neurogenesis is always superior to neuronal turnover as it permits the network to be responsive to changes in input statistics while at the same time preserving representations of earlier environments.

1 Introduction

The production, maturation and integration of new neurons into existing circuits is known to occur in a variety of species and brain areas, including the HVC of song birds (Nottebohm, 1981; Alvarez-Buylla and Kirn, 1997; Nottebohm, 2002), the primate hippocampus (Gould *et al.*, 2001), and the medial cortex of lizards (Font *et al.*, 2001; Marchioro *et al.*, 2005). In rats particularly high levels of neurogenesis can be found in the olfactory bulb and the dentate gyrus of the hippocampus (Altman and Das, 1965; Kornack and Rakic, 2001; Ming and Song, 2005), although some studies suggest that the birth of new neurons also occurs in other areas (Chechneva *et al.*, 2005; Takemura, 2005). In neonatal rats it is estimated that around $10,000$ new neurons are born in the subgranular layer of the dentate gyrus each day, a number that declines rapidly with age (McDonald and Wojtowicz, 2005). New neu-

rons have been shown to integrate with existing circuits and acquire mature firing characteristics within a few weeks (Markakis and Gage, 1999; Paton and Nottebohm, 1984). Although the majority of these new neurons subsequently die, those that do survive can persist for a year or more (Bayer *et al.*, 1982; Boss *et al.*, 1985). Surviving neurons appear to be initially more excitable and plastic than the existing mature neurons, abilities which may facilitate their integration into the existing network (Snyder *et al.*, 2001; van Praag *et al.*, 2002; Schmidt-Hieber *et al.*, 2004). Adult neurogenesis is regulated in a number of ways, with voluntary exercise (van Praag *et al.*, 1999), access to enriched environments (Brown *et al.*, 2003), and age (McDonald and Wojtowicz, 2005) all having been shown to affect levels of neurogenesis in rats and mice (for review, see Lehmann, Butz, and Teuchert-Noodt, 2005 and Ming and Song, 2005). Regulation of neurogenesis can take place either at the proliferation or apoptosis stage, two processes that appear to be at least partially distinct and that can be regulated independently (Cameron *et al.*, 1995; Kempermann *et al.*, 1997; Petreanu and Alvarez-Buylla, 2002).

Given the apparent concentration of neurogenesis in particular brain regions and the variety of ways in which it can be regulated, it is natural to speculate as to whether the ability to grow a neural network over time lends some specific functional advantage compared to a static network of a fixed size. A number of explanations of neurogenesis have been put forward in the literature. Many theoretical studies focus on the role of neurogenesis in learning and the computational advantages of neurogenesis have been illustrated in a variety of networks and learning tasks (Gould *et al.*, 1999; Cecchi *et al.*, 2001; Chambers *et al.*, 2004; Becker, 2005; Crick and Miranker, 2005; Aimone *et al.*, 2006; Chambers and Conroy, 2007). Typically neurogenesis is implemented as part of a replacement process for units which die and are subsequently removed from the network, so that the neurogenesis actually takes place as part of a neuronal turnover mechanism. For example, Cecchi et al. (2001) present a model of olfactory bulb neurogenesis in which inhibitory granule cells are randomly deleted and replaced by cells with newly initialised, random connectivity. Survival of these new units is determined in an activity dependent manner. This kind of network is capable of developing odour selectivity and can also adapt to changes in the input patterns as the turnover permits a rewiring of the neuronal connections over time. Chambers et al. (2004) examine neuronal turnover in a very

similar three layer feed forward network but with plastic connections. A supervised learning rule is used to train the network on two different data sets which are presented sequentially. Introducing a random turnover of middle layer units speeds up learning of the input-output relations of the new data set by helping the network forget the old input-output relations of the original data set, an effect that could be accelerated if turnover was targeted to units that stored the most information. A similar result was obtained by Crick and Miranker (2005) using an unsupervised learning rule.

Neuronal turnover has also been examined in the context of hippocampal processing. In Becker (2005) an explicit hippocampal memory model is formulated in which entorhinal cortex input patterns are encoded by the dentate gyrus and then stored in CA3. In this model the dentate gyrus is responsible for generating distinct codes for each of the input patterns received from the entorhinal cortex. This reduces the overlap between very similar input patterns, an effect which can otherwise cause problems during the storage and recall stages. Randomised turnover in the dentate gyrus layer was shown to offer advantages over static networks by allowing more distinctive codes to be generated when encoding memory patterns of very similar events.

Although these studies have provided useful insights into the computational properties of neurogenesis when neurogenesis is part of either a general or targeted turnover of units (which appears to be the dominant process in the olfactory bulb) experimental work suggest that neurogenesis in the hippocampus is additive, with new neurons being added to an expanding network rather than being used as replacements for dead or dying cells. In rats, for example, it has been estimated that neurogenesis leads to a growth of around 30% of the dentate gyrus over the lifetime of the animal (Bayer *et al.*, 1982; Boss *et al.*, 1985). The interpretation of why neurogenesis in the hippocampus is additive and not a turnover depends on the function ascribed to the network. The precise role of the hippocampus is still a matter of some debate, but a variety of experimental results have indicated that the hippocampus is involved in some form of memory function. In humans, hippocampal damage leads to anterograde amnesia and graded amnesia of episodic memories while leaving procedural memory unimpaired (Scoville and Milner, 1957), while in rats, damage to the hippocampus adversely affects navigation in the Morris water maze (Czurkó *et al.*, 1997), a task likely

to involve at least some form of episodic memory. Several models of the hippocampus as a memory system have been put forward inspired by these results (See, for example, Treves and Rolls, 1994; McClelland et al., 1995; and Hasselmo and Wyble 1997). These models differ in a number of important ways, for example in architecture (Lisman, 1999), permanence of storage (Nadel *et al.*, 2000), or the storage and retrieval mechanisms (Kunec *et al.*, 2005), but are typically built around the same basic principle that the hippocampus acts as a memory store.

In a previous study Wiskott, Rasch, and Kempermann (2006) examined the effect of additive neurogenesis in a simple, linear, feed forward neural network that performed an encoding-decoding memory task. In this model, the middle layer of the network was required to form a compact code representation of the input patterns for subsequent storage. The patterns were later retrieved and decoded to reproduce an approximation to the original input pattern. The dual constraints that the statistics describing the incoming patterns depended on the current environment, and that the system was required to deal with changes in this environment, led to a form of interference where adapting to the new input statistics severely disrupted the retrieval of previously stored memory patterns. Their key finding was that, unlike conventional forms of adaptation where network connections change over time, additive neurogenesis allowed the network to adapt to new environments while at the same time avoiding a breakdown of retrieval properties. Note that this form of interference is a decoding issue that arises when changes in network connectivity disrupt the correct interpretation of activity patterns, and is distinct to the forms of interference considered in other models of hippocampal neurogenesis, such as Becker (2005) where the network has difficulty distinguishing between very similar input patterns, and to interference in a Hopfield network when too many patterns have been stored and the network is no longer able to recall any of the stored patterns. It was proposed that the process of adult neurogenesis might fulfil a very similar computational role in the dentate gyrus of the hippocampus, endowing it with the ability to adapt to new input statistics while at the same time preserving representations of earlier environments. However, in this previous study the network was linear and the middle layer had fewer units than the input layer, so that representations of input patterns in the middle layer took the form of a compact code. In the hippocampus there is a large di-

vergence in unit number between the input (entorhinal cortex) and middle (dentate gyrus) layers. The representation in the middle layer is also very sparse, with average activity levels as low as 0.5% (Barnes *et al.*, 1990; Jung and McNaughton, 1993). The resulting encoding is therefore highly non-linear, and the computation performed by such a network very different to the simple, linear case we examined previously. Furthermore, only a single, fixed level of neurogenesis was examined in this earlier study leaving open the question of whether neurogenesis is always a superior adaptation strategy compared to more conventional strategies, or if regimes exist where the network performance is actually worse.

Here we explicitly evaluate the hypothesis that neurogenesis might play a role in avoiding interference in the dentate gyrus by examining additive neurogenesis in a simplified memory model of the hippocampus. The model incorporates both a dimensionality increase from the entorhinal cortex to the dentate gyrus and sparse coding, both notable features of hippocampal processing. The system is required to first encode and store, then later retrieve and decode, input patterns in a changing input environment. We consider two distinct ways in which the network can adapt; in neuronal turnover the network is of fixed size but units can be deleted and replaced, while in additive neurogenesis the network starts out smaller in size and grows over time. We derive analytical expressions and perform accompanying simulations to quantify the severity of decoding interference in this model and evaluate the performance of this network under different adaptation strategies at biologically realistic levels of adaptation. Having done this we extend our simulations and compare network performance across the full range of neurogenesis and turnover levels.

2 Model

To examine the functional implications of additive neurogenesis in as wide a sense as possible we do not consider any one particular hippocampal model but explore neurogenesis in the context of a generalised, encoding and decoding memory model. Although it is possible that the hippocampus also plays a role in spatial processing, it is known that spatially structured activity exists outside of the hippocampus (Hafting *et al.*, 2005) so it is unclear to what extent the hippocampus actually performs spatial computation as

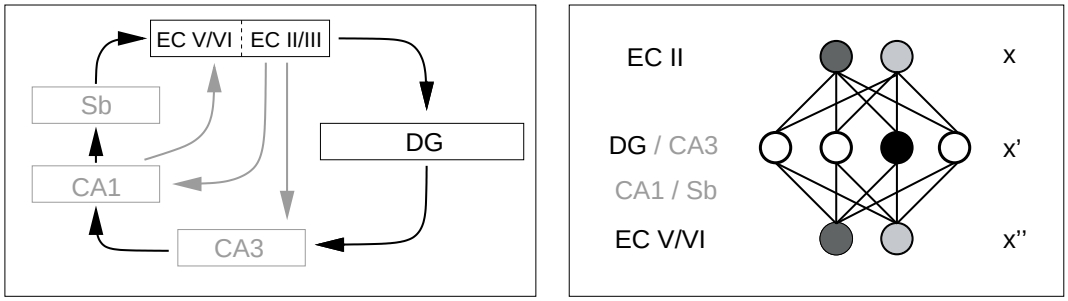

Figure 1: (Left panel) our simplified hippocampal model. We focus on the role of the EC and DG, while the remaining areas are modelled only implicitly and are shown as grey in the figure. Connectivity which does not play a role in our model is indicated by grey arrows. (Right panel) the autoencoding network we abstract from our simplified hippocampal model. A continuous N-dimensional EC input pattern, denoted $\mathbf{x}$, is encoded into a binary M-dimensional DG representation, denoted $\mathbf{x}'$. This pattern is stored and retrieved at some later time, then inverted to reproduce an approximation to the original pattern, $\mathbf{x}''$. We do not consider the details of storage and retrieval, and instead focus on the interaction between the EC and DG. In particular, we consider how the DG encoding can adapt to accommodate changes in the EC input statistics.

opposed to simply operating on spatially structured input from the EC. In such circumstances we have chosen to make the hypothesis that the primary function of the hippocampus is to act as a memory system where some useful information is stored and later retrieved. Incoming patterns from the neocortex arrive at the superficial layers of the entorhinal cortex (EC), which act as a gateway to the hippocampus. Input patterns are encoded by the dentate gyrus (DG) then stored downstream in the hippocampus proper, presumably in area CA3. Stored patterns are retrieved at some later time and decoded to reproduce the original input pattern in the output layers of the EC. We do not consider the details of how the hippocampus performs the storage and retrieval, nor the nature of the stored information. The stored patterns may be genuine memories or pointers to memories stored elsewhere, and storage can either be permanent or for a limited time, after which memories are transferred to another storage site via a process of consolidation. This simplified hippocampal memory model is illustrated in Figure 1.

It takes the form of a three-layer autoencoder network with N units in the input layer representing the superficial layers II/III of the EC, M units in the middle or hidden layer representing the DG, and N units again in the output layer representing the deep layers V/VI of the EC. The input and output layers have continuous units that can assume positive and negative real values while the hidden layer has binary units that can only assume the values 0 and 1. This reflects the fact that cells in the EC have a fairly continuous firing rate distribution while the granule cells of the DG show a more binary, bursting behaviour typified by place cells (O'Keefe and Dostrovsky, 1971). Because of the extreme sparseness of around 0.5% activity in the DG (Barnes *et al.*, 1990; Jung and McNaughton, 1993) we use a one-out-of-M representation in the hidden layer, so that exactly one unit is active and set to 1 and all others are inactive and set to 0. Input and output EC-activities are therefore N-dimensional vectors with real coefficients and the DG-activity is an M-dimensional binary vector, with $M \gg N$. This produces a reasonable level of sparseness for the network sizes we consider, and has the added advantage of permitting some degree of analysis of the network. The representational restriction in the hidden layer is compensated to some extent by the fact that the hidden (DG) layer has many more units than the other two layers (Boss *et al.*, 1985; Mulders *et al.*, 1997).

Each unit i in the hidden layer has an associated N-dimensional *encoding vector* $\hat{\mathbf{x}}_i$. An input vector $\mathbf{x}$ activates the unit i^* in the hidden layer with associated encoding vector $\hat{\mathbf{x}}_i$ lying closest to the input vector $\mathbf{x}$. Thus, for any given input vector $\mathbf{x}$ the *winning unit* is

$$i^* := \arg\min_i |\mathbf{x} - \hat{\mathbf{x}}_i| \,. \tag{1}$$

This activation rule induces a Voronoi tessellation of the input space into M Voronoi cells, with each cell containing one encoding vector, as illustrated in Figure 2.

Each unit in the hidden layer has an associated *decoding vector* $\tilde{\mathbf{x}}_i$, which determines the output vector $\mathbf{x}''$ if i is the winning unit in the hidden layer, so that $\mathbf{x}'' = \tilde{\mathbf{x}}_{i^*}$. Usually, en- and decoding vectors are identical, so that $\tilde{\mathbf{x}}_i = \hat{\mathbf{x}}_i$, but if the network adapts at any time between storage and retrieval, the decoding vectors used during retrieval might be different from the encoding vectors used during storage.

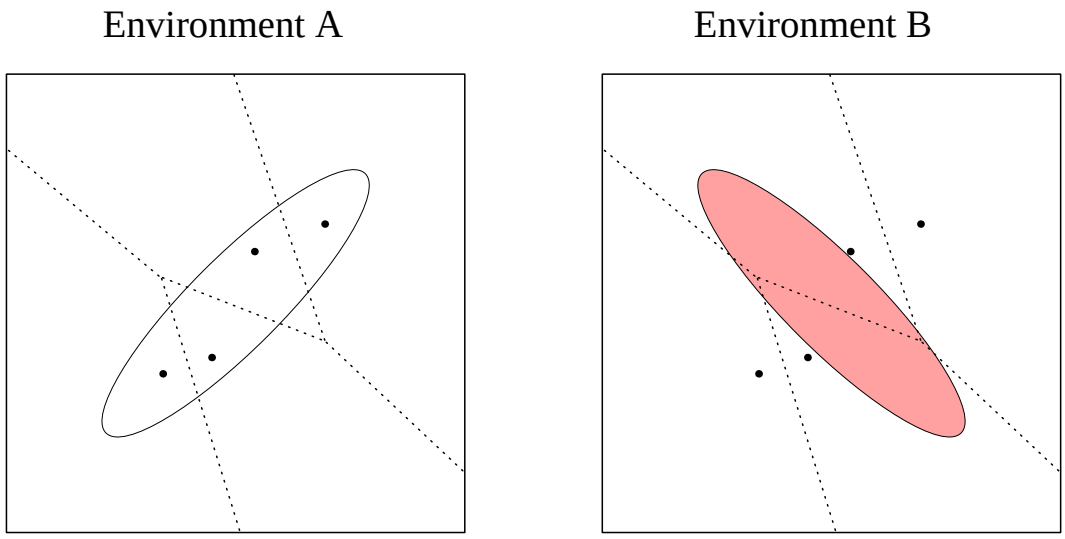

Figure 2: Encoding of the EC input patterns by the DG. Initially the system is in environment A (left panel), the statistics of which is represented by the ellipse. The set of encoding vectors (solid circles) representing the DG units are placed according to the same distribution, and input patterns are mapped to the nearest encoding vector according to a $1/M$ code. The partitioning of input space therefore takes the form of a Voronoi Tessellation with the (dotted) lines of partition falling equidistant between any two neighbouring encoding vectors. Upon moving to environment B (right panel) the input statistics may change (grey ellipse) so that the original encoding is no longer appropriate. There are several ways of adapting the network to this change in input statistics, and we are interested in finding the method that maximises network performance in the new environment while at the same time preserving its function as a memory store for the old environment.

2.1 Analysis

2.1.1 Recoding and retrieval error

The purpose of an autoencoder network, such as the one described above, is to reconstruct the input vector in the output layer as faithfully as possible under the constraint of a change of representation in the hidden layer. In our case the representation in the hidden layer is constrained to be a binary one-out-of-M code. A common measure of the performance of such a network, or rather the error it makes, is the mean squared Euclidean distance between input vectors $\mathbf{x}$ and output vectors $\mathbf{x}''$,

$$E := \langle |\mathbf{x} - \mathbf{x}''|^2 \rangle_{\mathbf{x},\{\hat{\mathbf{x}}_i,\tilde{\mathbf{x}}_i\}} \,, \tag{2}$$

where the averaging denoted by $\langle \cdot \rangle$ is over the distribution of the input vector $\mathbf{x}$, the sets of encoding vectors $\{\hat{\mathbf{x}}_i\}$ with $i = 1, 2, ..., M$, and the sets of decoding vectors, $\{\tilde{\mathbf{x}}_i\}$, if they differ from the encoding vectors. We refer to this error as the *recoding error* if the output vector $\mathbf{x}''$ is calculated directly from the input vector $\mathbf{x}$ by immediate en- and decoding. If there is a storage and retrieval process involved between en- and decoding, we speak of a *retrieval error*. It is important to make this distinction as the retrieval error is not always the same as the recoding error. This typically occurs when the network changes due to adaptation to a new environment between storage and retrieval, so that the en- and decoding effectively come from two different networks. Note that, in general, this means that there are two distinct contributions to the error. First, there is the error that arises due to the loss of information when input patterns are encoded by the network. Second, there is the error that arises from imperfect decoding due to the meaning of DG units changing in the time since the pattern was originally stored. The error due to loss of information when encoding patterns is a property of the whole network, and is an unavoidable consequence of of the change of representation between EC and DG layers. The magnitude of this error is a function of number of units in the DG layer and, on average, is the same for all networks with equal numbers of DG units. By contrast, the error due to incorrect decoding is a consequence of the way in which the network adapts to changes in the input statistics. The magnitude of this error can change quite dramatically, or even be reduced to zero, depending on the adaptation strategy used (see below).

As we wish to focus on the interaction of the EC and DG, we do not model storage and retrieval explicitly. Instead, we note the index of the winning hidden unit in the storage phase and later use that unit for decoding in the retrieval phase. In other words, we assume that the storage and retrieval mechanism works sufficiently well so that no additional information is lost during storage and retrieval. The recoding error defined in Equation 2 therefore fully defines the performance metric of the network.

2.1.2 En- and decoding vectors

Input vectors $\mathbf{x}$ are drawn from the probability density function $p_X(\mathbf{x})$. Given any such distribution of input vectors, there is an optimal set of en- and decoding vectors which could be found by some vector quantisation algorithm. However, for analytical reasons we assume that the encoding vectors are distributed in a probabilistic manner according to the distribution $p_{\hat{X}}(\hat{\mathbf{x}})$, and that the decoding vectors always match the current set of encoding vectors. The encoding vectors are therefore statistically independent from each other. This simplifies the analysis of the network by permitting an immediate factorisation of the expressions for the recoding and retrieval errors (See section 2.2). Obviously, $p_{\hat{X}}(\hat{\mathbf{x}})$ should be similar to $p_X(\mathbf{x})$ for good encoding and we typically choose them to be identical. Provided that sufficiently many units in the hidden layer are used, the performance difference compared to the optimised case is by a constant factor, which does not affect the pattern of our results or the observations made here (see discussion).

2.1.3 Errors and adaptation strategies

To evaluate the performance of the network we consider a scenario where a virtual rat moves from one environment A to another environment B and back again. Whenever we refer to one of these environments, we replace X by A or B, respectively. For instance, $p_{\hat{A}}(\hat{\mathbf{a}})$ indicates the distribution of the encoding vectors, $\hat{\mathbf{a}}$, for environment A. The distributions defining environments A and B are assumed to differ by a random rotation with respect to each other, but are otherwise identical. Note that, if this rotation is small, environments A and B will be very similar. As we are interested in the average network performance, the network must be able to deal with very similar as well as very different environments. The network fully adapted to A is re-

ferred to as *network* A. After the network has completely adapted to B it is referred to as *network* B. To quantify the performance of the network we consider the following five errors.

(i) **Recoding error for network A in environment A:** The recoding error in environment A when the network is fully adapted to A.

(ii) **Recoding error for network A in environment B:** The recoding error in environment B when the network has not yet had time to adapt to B.

(iii) **Recoding error for network B in environment B:** The recoding error in environment B when the network has finished adapting to B.

(iv) **Retrieval error for network B in environment B:** The error for patterns from environment A stored with network A and later retrieved and decoded using network B. This is the error if the hypothetical rat retrieves and decodes memories from A after having adapted to B.

(v) **Recoding error for network B in environment A:** The recoding error when the rat has returned to environment A but has not yet readapted to it.

This set of five errors fully quantifies the performance of the network as the hypothetical rat moves between the two environments.

We evaluate two different adaptation strategies; neuronal turnover and additive neurogenesis. In neuronal turnover the network is fixed in overall size but units may be deleted and reinitialised according to the current input statistics, so that some of the DG units are then adapted to the current environment. The proportion of DG units that turn over is denoted by M_2 and ranges from zero in a fixed network to one in a completely reinitialising network. In neurogenesis the network starts out small in size and is then allowed to grow over time. In this case the degree of adaptation, M_2, is interpreted as the proportion of units that were added in environment B after the network has finished growing.

We compare the performance of the network under two distinct classes of adaptation strategy, in the form of neuronal turnover and additive neurogenesis. These two adaptation strategies yield the following four classes of network that are of particular interest.

(a) **Fixed network:** The network has M hidden units. They are initialised with encoding vectors $\hat{\mathbf{a}}_i$ and identical decoding vectors $\tilde{\mathbf{a}}_i = \hat{\mathbf{a}}_i$ randomly drawn from $p_{\hat{A}}(\hat{\mathbf{a}})$ and are thereafter kept fixed.

(b) **Partial turnover:** As for (a) but M_2 of the M hidden units are reinitialised with new encoding vectors $\hat{\mathbf{b}}_i$ and identical decoding vectors $\tilde{\mathbf{b}}_i = \hat{\mathbf{b}}_i$ randomly drawn from $p_{\hat{B}}(\hat{\mathbf{b}})$ when the rat adapts to environment B.

(c) **Full turnover:** As for (b), but with $M_2 = M$. In other words, all of the hidden units get reinitialised when the rat adapts to environment B.

(d) **Neurogenesis:** The network starts with a smaller set of (M_1) hidden units, where $M_1 < M$. The en- and decoding vectors of these M_1 hidden units are fixed. When the rat adapts to environment B, M_2 new units are added to the hidden layer which are initialised with encoding vectors $\hat{\mathbf{b}}_i$ and identical decoding vectors $\tilde{\mathbf{b}}_i = \hat{\mathbf{b}}_i$ randomly drawn from $p_{\hat{B}}(\hat{\mathbf{b}})$. After adaptation the total number of hidden units is the same as in the other networks, $M_1 + M_2 = M$.

Note that after growth is complete in the neurogenesis strategy the network is, in terms of the recoding error in environment B, indistinguishable from the partial turnover network. The *retrieval* error will, however, be lower because the units added in environment B will not affect the retrieval of patterns stored in A as they were not available when those patterns were originally stored. This decrease in retrieval error is at the expense of a loss of initial representative power in environment A due to a smaller initial network size. It is our goal here to measure the relative sizes of the loss of representational power in environment A compared to the gain in retrieval accuracy in environment B and to quantify the overall performance of the network under the different adaptation strategies outlined above.

Regardless of strategy, the system starts with a network fully adapted to environment A, so that all the hidden units are initialised with encoding vectors $\hat{\mathbf{a}}_i$ and identical decoding vectors $\tilde{\mathbf{a}}_i = \hat{\mathbf{a}}_i$ randomly drawn from $p_{\hat{A}}(\hat{\mathbf{a}})$. The number of hidden units in the initial network may vary depending on the adaptation strategy. In the neurogenesis strategy the initial number of units is typically smaller than for the turnover network because, if we add

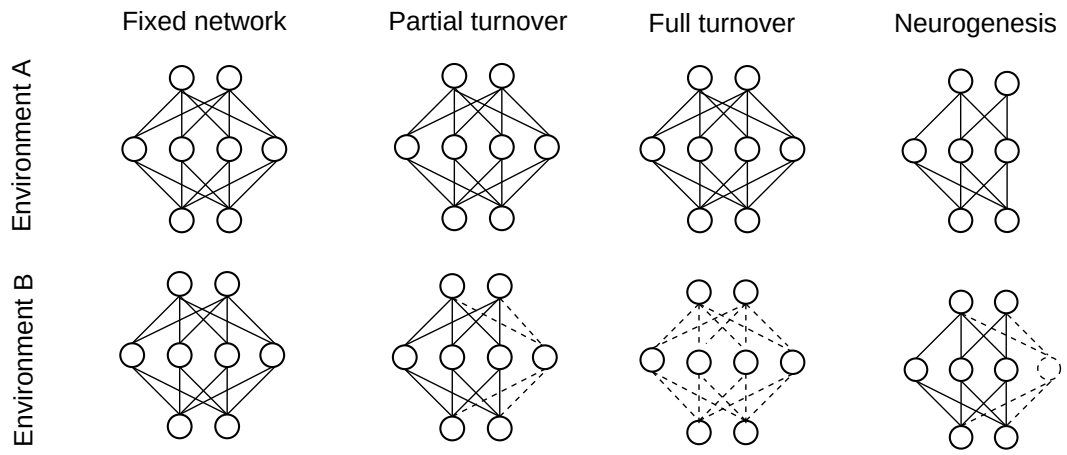

Figure 3: The state of a simple network in environments A and B for the four adaptation strategies we consider. For clarity we illustrate a fairly simple network with a total of four DG units. In neuronal turnover some of the DG units are fixed but a proportion are allowed to turn over and are reinitialised according to the current input statistics. This can range from no turnover (column 1), to partial turnover (column 2), to full turnover (column 3). In all three cases, the network starts out with a full set of DG units which are adapted to environment A (solid lines). On entry to environment B the subset of units undergo turnover and are reinitialised according to the input statistics of environment B (dotted lines) while the rest remain fixed. In neurogenesis (column 4) the network starts out with fewer units adapted to environment A and later adds an extra unit which adapts to environment B.

new units later, we do not want the network to have the advantage of having more hidden units than the other networks.

The state of a simple network in environments A and B, for each of the four adaptation strategies, is shown in Figure 3. The corresponding encoding of the EC input patterns by the DG is illustrated in Figure 4.

2.2 Formal analysis

As the encoding vectors that represent the optimal response locations of the DG units are drawn independently of each other, an analytical equation for the recoding error is straightforward to formulate. Evaluating it, however, requires further simplifications. Consider first the recoding error for network A, so that all encoding vectors are drawn from distribution A. Let

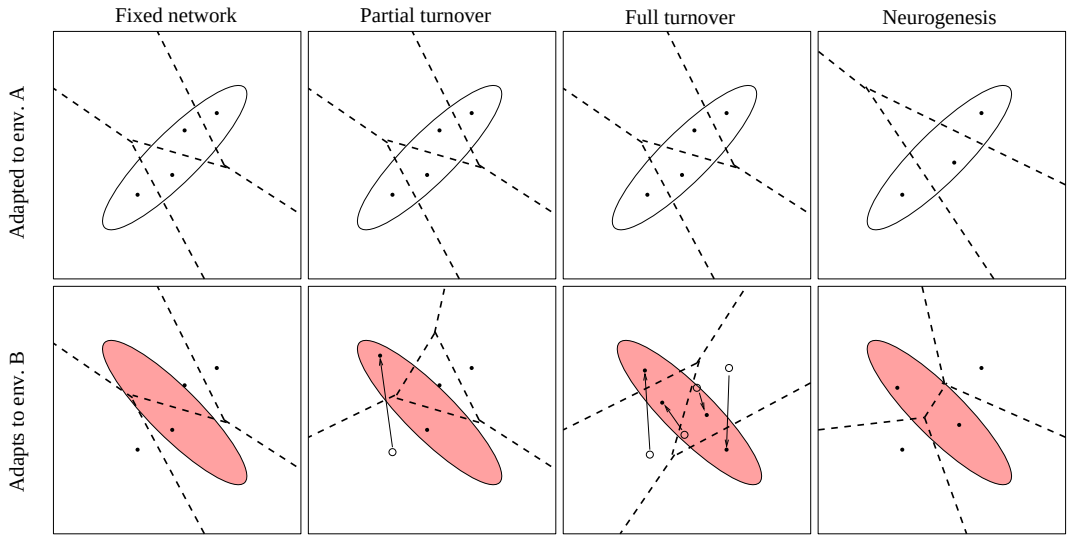

Figure 4: Encoding of EC input patterns by the DG for the four basic types of adaptation strategy we consider. Again, for clarity we illustrate a fairly simple network with a total of four DG units. With no turnover (column 1) the partitioning of phase space is fixed and well suited to environment A and all four units are used. Upon moving to environment B, we find that the network is not well suited to the new input statistics and can typically only recruit two of the DG units. This leads to a higher recoding error in environment B compared to environment A. With partial turnover (column 2) the network again uses all four units in environment A but now, upon moving to environment B, one randomly selected unit can adapt to the new input statistic, a change indicated by the arrow. This network can recruit three units in environment B and the recoding error is therefore lower than for the fixed network. With full turnover (column 3) all of the units can adapt and the network can use all four units in both environments. The recoding error is therefore approximately the same in environments A and B. In neurogenesis (column 4) the network starts out with three units adapted to environment A. Upon entry to environment B an additional unit is added which adapts to the input statistics of environment B. Thus, the network can encode patterns from environment B fairly well without having to change any of the existing units. The key feature of the neurogenesis strategy is that the network is able to encode both environments reasonably well without the need to change the meaning of existing DG units, indicated by the absence of arrows in column 4. This has significant consequences when we consider, in addition to the recoding error for storage of new patterns, the *retrieval* error for patterns that were previously stored in environment A.

$P_{\hat{A}\oslash}(\mathbf{x}, r)$ be the probability that an encoding vector $\hat{\mathbf{a}}$ randomly drawn from distribution A has a distance from $\mathbf{x}$ greater than radius r

$$\begin{aligned} P_{\hat{A}\oslash}(\mathbf{x}, r) &:= P(|\mathbf{x} - \hat{\mathbf{a}}| > r) && (3) \\ &= \langle H(|\mathbf{x} - \hat{\mathbf{a}}| - r) \rangle_{\hat{\mathbf{a}}} && (4) \\ &= 1 - \langle H(r - |\mathbf{x} - \hat{\mathbf{a}}|) \rangle_{\hat{\mathbf{a}}} \ , && (5) \end{aligned}$$

where $H(\cdot)$ is the Heaviside function. An analogous definition follows for distribution B, and we will make use of this below. If we have M_A units in the hidden layer with encoding vectors $\hat{\mathbf{a}}_i$ randomly drawn from $p_{\hat{A}}$, and associated decoding vectors $\tilde{\mathbf{a}}_i = \hat{\mathbf{a}}_i$, the recoding error for input vectors $\mathbf{x}$ randomly drawn from p_X is

$$E = M_A \left\langle |\mathbf{x} - \hat{\mathbf{a}}_1|^2 \left[P_{\hat{A}\oslash}(\mathbf{x}, |\mathbf{x} - \hat{\mathbf{a}}_1|) \right]^{M_A - 1} \right\rangle_{\mathbf{x}, \hat{\mathbf{a}}_1} . \qquad (6)$$

$|\mathbf{x} - \hat{\mathbf{a}}_1|^2$ is the recoding error for particular vectors $\mathbf{x}$ and $\tilde{\mathbf{a}}_1 = \hat{\mathbf{a}}_1$. $\left[P_{\hat{A}\oslash}(\mathbf{x}, |\mathbf{x} - \hat{\mathbf{a}}_1|) \right]^{M_A - 1}$ is the probability that a given $\hat{\mathbf{a}}_1$ is used for encoding for a particular input vector $\mathbf{x}$, because the other $(M_A - 1)$ encoding vectors, $\hat{\mathbf{a}}_{i \neq 1}$, are farther away from $\mathbf{x}$ than $\hat{\mathbf{a}}_1$. If $\mathbf{x}$ and $\hat{\mathbf{a}}_1$ are close to each other this probability is high. If they are far apart this probability is low, as it is likely that one of the other encoding vectors is closest to $\mathbf{x}$ and will therefore be used for encoding instead of $\hat{\mathbf{a}}_1$. Since we have M_A encoding vectors, which are all independent of each other and therefore contribute identically to the total error, we multiply the result by M_A to get the total error. Equation (6) can be used for recoding errors A and B for network A if we set $\mathbf{x}$ to $\mathbf{a}$ or $\mathbf{b}$, respectively. For a fixed network it can be used for all errors, because the network does not adapt at all. Finally, equation (6) can also be used for the retrieval error A for network B in the neurogenesis strategy, because in that case the added hidden units are irrelevant and the old hidden units are still adapted to A.

An equation analogous to equation (6) holds for environment B.

$$E = M_B \left\langle |\mathbf{x} - \hat{\mathbf{b}}_1|^2 \left[P_{\hat{B}\oslash}(\mathbf{x}, |\mathbf{x} - \hat{\mathbf{b}}_1|) \right]^{M_B - 1} \right\rangle_{\mathbf{x}, \hat{\mathbf{b}}_1} . \qquad (7)$$

This equation can be used for recoding errors A and B for network B in the full turnover strategy.

Consider now if the encoding vectors are drawn from two different distributions, for example if M_A encoding vectors are drawn from distribution

A and M_B are drawn from distribution B. In a direct extension of (6) and (7) we get

$$\begin{aligned} E &= M_A \left\langle |\mathbf{x}-\hat{\mathbf{a}}_1|^2 \left[P_{\hat{A}\oslash}(\mathbf{x},|\mathbf{x}-\hat{\mathbf{a}}_1|)\right]^{M_A-1} \left[P_{\hat{B}\oslash}(\mathbf{x},|\mathbf{x}-\hat{\mathbf{a}}_1|)\right]^{M_B} \right\rangle_{\mathbf{x},\hat{\mathbf{a}}_1} \\ &+ M_B \left\langle |\mathbf{x}-\hat{\mathbf{b}}_1|^2 \left[P_{\hat{A}\oslash}(\mathbf{x},|\mathbf{x}-\hat{\mathbf{b}}_1|)\right]^{M_A} \left[P_{\hat{B}\oslash}(\mathbf{x},|\mathbf{x}-\hat{\mathbf{b}}_1|)\right]^{M_B-1} \right\rangle_{\mathbf{x},\hat{\mathbf{b}}_1}, \end{aligned} \tag{8}$$

where now the probability of all other encoding vectors $\hat{\mathbf{a}}_{i\neq 1}$ and $\hat{\mathbf{b}}_j$ being farther away from input vector $\mathbf{x}$ than encoding vector $\hat{\mathbf{a}}_1$ leads to the product term $\left(P_{\hat{A}\oslash}(\mathbf{x},|\mathbf{x}-\hat{\mathbf{a}}_1|)\right)^{M_A-1} \left(P_{\hat{B}\oslash}(\mathbf{x},|\mathbf{x}-\hat{\mathbf{a}}_1|)\right)^{M_B}$ and a similar term emerges for $\hat{\mathbf{b}}_1$. For $M_B = 0$ equation (8) goes over to (6); for $M_A = 0$ it goes over to (7). We encounter encoding vectors from both distributions in the hidden layer in the partial turnover strategy and the neurogenesis strategy. Thus, equation (8) can be used in these cases for the recoding errors A and B for network B.

In the considerations above, we assumed that the decoding vectors are identical to the encoding vectors and simply set $\tilde{\mathbf{a}}_1 = \hat{\mathbf{a}}_1$ and $\tilde{\mathbf{b}}_1 = \hat{\mathbf{b}}_1$. However, when we consider retrieval of stored patterns the en- and decoding vectors might be different as the network might have changed between storage and retrieval. For instance, when a pattern has been stored with a network completely adapted to A and is then retrieved after the network has partially adapted to B, some of the decoding vectors, say M_{AB} of them, will have changed from $\tilde{\mathbf{a}}_i = \hat{\mathbf{a}}_i$ to new vectors $\tilde{\mathbf{a}}_i = \hat{\mathbf{b}}_i$ (in reality the encoding vectors have changed as well from $\hat{\mathbf{a}}_i$ to $\hat{\mathbf{b}}_i$, but that is not relevant here, because we consider retrieval and not recoding) and only M_{AA} decoding vectors are still intact with $\tilde{\mathbf{a}}_i = \hat{\mathbf{a}}_i$. In that case equation (6) becomes

$$\begin{aligned} E &= M_{AA} \left\langle |\mathbf{x}-\hat{\mathbf{a}}_1|^2 \left[P_{\hat{A}\oslash}(\mathbf{x},|\mathbf{x}-\hat{\mathbf{a}}_1|)\right]^{M_A-1} \right\rangle_{\mathbf{x},\hat{\mathbf{a}}_1} \\ &+ M_{AB} \left\langle \left\langle |\mathbf{x}-\hat{\mathbf{b}}_1|^2 \right\rangle_{\hat{\mathbf{b}}_1} \left[P_{\hat{A}\oslash}(\mathbf{x},|\mathbf{x}-\hat{\mathbf{a}}_1|)\right]^{M_A-1} \right\rangle_{\mathbf{x},\hat{\mathbf{a}}_1}. \end{aligned} \tag{9}$$

$\left\langle |\mathbf{x}-\hat{\mathbf{b}}_1|^2 \right\rangle_{\hat{\mathbf{b}}_1}$ is the actual error made for a particular input vector $\mathbf{x}$ averaged over all possible decoding vectors $\tilde{\mathbf{b}}_1 = \hat{\mathbf{b}}_1$. This equation provides the retrieval error A for network B for the partial and the full turnover adaptation strategies.

Although equations (6–9) are fairly easy to formulate they are, in practice, rather difficult to evaluate directly. We therefore take the limiting case of the N-dimensional input distribution in which only one dimension has

significant variance and the remainder are set to zero. In other words, we approximate the N-dimensional distributions with 1D distributions embedded in an N-dimensional space. This preserves our central assumption about the input distributions, that they are structured in some way with some dimensions carrying more of the variance than others, while at the same time allowing us to make further analytical progress.

2.3 Approximation with 1D distributions

To proceed with the analysis we assume that all distributions are inherently one-dimensional, so that all input-, encoding-, and decoding-vectors drawn from any single environment lie on a common line. This permits us to confine the analysis to the 2D plane spanned by these two lines, and, without loss of generality, we assume that the abscissa is the principal axis of environment A. The principal axis of B then lies at an angle ϕ relative to line A. If a and b indicate local coordinates on lines A and B, respectively, we can formulate everything with the 1D pdfs, denoted by $p_A(a)$, $p_B(b)$, $p_{\hat{A}}(\hat{a})$ etc. Within the 2D plane we have the vector relations

$$\mathbf{a} = (a, 0)\,, \tag{10}$$

$$\mathbf{b} = (b\cos\phi, b\sin\phi)\,. \tag{11}$$

The probability that an encoding vector $\hat{\mathbf{a}}$ randomly drawn from distribution A lies farther away from $\mathbf{a}$ or $\mathbf{b}$ than radius r is given by

$$P_{\hat{A}\oslash}(\mathbf{a}, r) = 1 - \int_{a-r}^{a+r} p_A(\hat{a})\,\mathrm{d}\hat{a}\,, \tag{12}$$

$$P_{\hat{A}\oslash}(\mathbf{b}, r) = 1 - \int_{\alpha_-(b,r)}^{\alpha_+(b,r)} p_A(\hat{a})\,\mathrm{d}\hat{a} \tag{13}$$

$$\text{with} \quad \alpha_\pm(b, r) := b\cos\phi \pm \Re\left(\sqrt{r^2 - (b\sin\phi)^2}\right)\,, \tag{14}$$

where $\Re(\cdot)$ indicates the real part. An analogous definition can be made for distribution B by interchanging A and B, a and b, $\hat{a}$ and $\hat{b}$, etc.

Inserting equations (10–13) into (6–9) and averaging over the scalar variables a, $\hat{a}$, etc. instead of over the vector variables $\mathbf{a}$, $\hat{\mathbf{a}}$, etc. yields equations that can be integrated numerically. For instance, in this approximation the

recoding error B for a network fully adapted to A becomes

$$E \overset{(6)}{=} M_A \left\langle |\mathbf{b} - \hat{\mathbf{a}}_1|^2 \left[P_{\hat{A}\oslash}(\mathbf{b}, |\mathbf{b} - \hat{\mathbf{a}}_1|) \right]^{M_A - 1} \right\rangle_{\mathbf{b}, \hat{\mathbf{a}}_1} \quad (15)$$

$$\overset{(13)}{=} M_A \left\langle r^2 \left(1 - \int_{\alpha_-(b,r)}^{\alpha_+(b,r)} p_A(\hat{a})\, \mathrm{d}\hat{a} \right)^{M_A - 1} \right\rangle_{b, \hat{a}_1} \quad (16)$$

with (14) and $r := |\mathbf{b} - \hat{\mathbf{a}}_1|$ (17)

$$\overset{(10,11)}{=} \sqrt{(b\cos\phi - \hat{a}_1)^2 + (b\sin\phi)^2}\,. \quad (18)$$

For the analytical results below we will use equations (6–9) with this 1D-approximation (10–13). Table 1 presents an overview of the equations used to calculate the different errors in the different environments.

3 Results

First we examine network performance with a particular, approximately biological, level of neurogenesis. We present analytical results for the cases $N = 2$ and large N under the simplifying assumption that all distributions are inherently one-dimensional. We then perform simulations using full N-dimensional distributions for the input and encoding vectors with $N = 60$.

Having done this we extend our simulations to examine network performance across the entire range of turnover and neurogenesis levels.

3.1 Analytical results

First consider the case where the input is two-dimensional and there are three to four units in the hidden layer, i.e. $N = 2$, $M_1 = 3$, $M_2 = 1$, and $M = 4$ (see Figure 3). All distributions are assumed to be one-dimensional Gaussian distributions with zero mean and variance one. Distribution A lies along the abscissa, while distribution B is rotated by an angle ϕ relative to A. For symmetry reasons we assume that all angles are equally likely, and we therefore average over all angles ϕ with equal probability to yield the total expected error.

Results for the small $N = 2$ case are shown in table 2. The five errors that quantify the performance of the network are described on the left hand side of the table. The first two errors are recoding errors for the network

adapted to environment A, and quantify network performance in environments A and B. The remaining three errors refer to when the network has subsequently adapted to environment B, and quantify the recoding error in B, the retrieval error for patterns that were stored by network A but are decoded by network B, and the recoding error in A (representing the situation where the rat has once again reentered A and not had time to readapt).

Column one shows these five errors for a fixed network. As expected, we see that network A performs well in environment A, as it has a full set of DG units that are adapted to the input statistics of that environment. Also as expected network A performs initially poorly in environment B as the network has not yet had time to adapt. This poor performance does not improve after the network adapts and becomes network B as, with fixed unit numbers and connections, the network is unable change in any way. Thus, the third and fifth errors (recoding errors in environments B and A with network B) are identical to the corresponding recoding errors for network A. Finally, the retrieval error for patterns stored under network A is the same as the original recoding error as, in a fixed network, the meaning of the encoding and decoding vectors does not change.

Column two shows the same set of five errors for the partial turnover strategy. Here, one unit of the four total units is deleted from the network and reinitialised according to the current input statistics, and the remaining three are fixed. The first two errors are the same as for a fixed network, as we still have a network with a full set of DG units all adapted to A. However, on entry to environment B the network can now adapt by changing the meaning of one of the DG units, reducing the recoding error for network B in environment B (third row). This increase in performance comes at the expense of a corresponding increase in the recoding error for new patterns from environment A as the network is now partially adapted to the statistics of A and partially adapted to the statistics of B (fifth row). Crucially, we also see a large increase in the retrieval error for patterns stored under network A as the meaning of the encoding and decoding vectors associated with the reinitialised unit have changed in the time since the patterns were stored. Decoding of memories which used this unit during storage is therefore disrupted.

Column three shows the five errors for the full turnover strategy. Here, all four DG units turn over and reinitialised according to the current input

statistics. Again, the first two errors are the same as for a fixed network, as the network has a full set of DG units adapted to A. On entry to environment B the entire DG layer adapts to the new input statistics. This produces a recoding error for network B in environment B (third row) that is identical to network A in environment A (first row). As expected, in the process the network loses the ability to successfully recode new patterns from environment A (in fact, the recoding errors for environments A and B are simply interchanged; fifth row). We also see that the retrieval properties of the network have now been completely disrupted as the meaning of all the encoding and decoding vectors have changed since the patterns were stored. This is an extreme example of the problem of interference, and the network is unable to decode retrieved patterns in a meaningful way.

Column four shows the five errors for the neurogenesis strategy. In this case, we start with an initially smaller network with three DG units and later add a fourth unit that is adapted to environment B. The first two errors follow a similar pattern to the fixed network, in that the network recodes patterns from A better than patterns from B. However, the absolute value of the error is, in both cases, higher as we have a network with fewer DG units and therefore a slightly lower representational power. On entry to environment B the network grows a new DG unit that is adapted to the statistics of environment B. This produces a hybrid network that partially adapted to the statistics of A and partially adapted to the statistics of B. In fact, in terms of recoding errors for environments A and B this network is indistinguishable to the partial turnover network discussed above. Thus, the recoding errors for network B in environment A and B are identical to the partial turnover network (third and fifth rows). In contrast to the partial turnover network, however, we see that the retrieval properties of the network have not been disrupted as the new unit added for environment B was not used in the original encoding of these patterns. The error is still non-zero due to the unavoidable loss of information from the change in representation between the EC and DG layers. In fact, the error in the original environment A is slightly higher in the neurogenesis network compared to the full turnover network due to the lower representational power of the initially smaller DG. Crucially, however, the increase in recoding error in A is much smaller than the corresponding gain from eliminating the retrieval error in B. Thus, by growing the network instead of introducing neuronal

turnover in a network of fixed size we arrive at a network that produces a reasonable level of performance in both environment A and B.

Now consider the high-dimensional case with large N. For consistency with earlier work (Wiskott *et al.*, 2006) we choose $N = 60$, but any number of the same order of magnitude would suffice. Since the distributions are still inherently one-dimensional the two distributions A and B span a two-dimensional subspace within the N-dimensional input space. Thus, we can treat the $N = 60$ case in the same manner as the $N = 2$ case but with a greater number of units in the hidden layer. We use $M_1 = 225$, $M_2 = 75$, and $M = 300$, so that the ratio between input and hidden units is the same as the five-to-one ratio seen in biology and that the increase in number of units is one quarter as in the two-dimensional case. In a high-dimensional space two randomly rotated vectors are almost certainly orthogonal. Thus, for large N the angle ϕ is not evenly distributed but typically lies very close to $\pi/2$. For the high-dimensional case we therefore simply set $\phi = \pi/2$ and do not need to average over some distribution of angles.

Results for the large $N = 60$ case are also shown in table 2. We see the same pattern of errors as for the small N case. In every case, adaptation strategies derived from a network of fixed size with some level of neuronal turnover suffer from at least one large error, representing a breakdown of network function in the corresponding task. For the fixed network, this task is dealing with new patterns from environment B, while for the partial or full turnover networks this task is correctly decoding retrieved patterns that were originally stored in A. Additive neurogenesis, on the other hand, suffers from no such problems and can deal with the full range of retrieval and recoding tasks. Indeed, the pattern of errors is even clearer in this large N case, as we have a much greater representational power in the DG so a well adapted network produces an error very close to zero (indicated by entries of < 0.01 in the table).

Thus, for both the small and large N cases we see a similar pattern of results. A fixed network successfully recodes and retrieves patterns from environment A but cannot deal with environment B. A partial turnover network deals more successfully with environment B but, as a direct result of the turnover, suffers significant errors when retrieving previously stored patterns from A. A full turnover network creates an extreme version of this problem and destroys the retrieval properties of the network entirely.

In contrast, the neurogenesis strategy offers good all-round performance as the network can accommodate the change in input statistics when moving from A to B without disrupting the retrieval properties of previously laid down memories.

3.2 Simulation results

In order to show that our assumption that all distributions are inherently one-dimensional does not qualitatively affect our results we have performed a numerical simulation using full, N-dimensional Gaussian distributions. We use $N = 60$, $M_1 = 225$, $M_2 = 75$, and $M = 300$ as used in the larger network considered above. Standard deviations, which we denote by σ_i with $i = 1, 2, ..., N$, describing the 60-dimensional input distribution are assigned according to the approximately exponential distribution shown in Figure 5. Dimensions with larger standard deviation are more important than dimensions with smaller standard deviation as they carry more of the variance of the input. This introduces an element of structure into the input in a simple, biologically plausible manner. In order to make a meaningful comparison to the analytical results we constrain the total variance to be unity, $\sum_{i=1}^{N} \sigma_i^2 = 1$.

We draw 1,000 input vectors from the distribution defined above and encode them using either 225 or 300 encoding vectors, depending on the environment and the strategy. We repeat this 100,000 times to average over different sets of input vectors, encoding vectors, and over different angles between environments A and B. The results are set out in Table 3. Although the errors are generally larger than for the 1-dimensional analytical case presented in Table 2, a result of the much greater occupation of phase space by the input distributions, we see a similar pattern of results as for both the $N = 2$ and large N analytical cases (Table 2). Again, each of the three turnover strategies suffers from at least one large error while neurogenesis offers a good all-round performance in both of the environments.

3.2.1 Effect of changing the number of adapting units

The degree of adaptation in the network can be described by an adaptation parameter, p, which we define to be the proportion of units in the dentate gyrus layer that can adapt in some way. With neuronal turnover p is simply the proportion of units that turnover. With neurogenesis p is the proportion

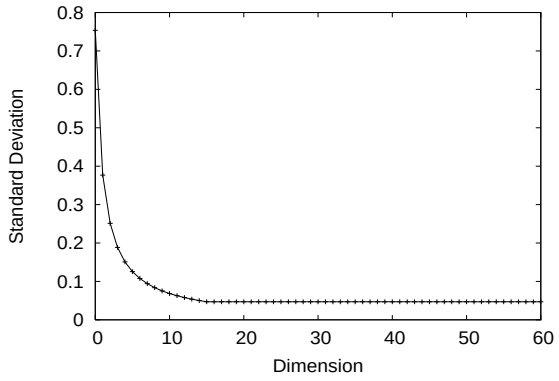

Figure 5: Distribution of standard deviations describing the 60-dimensional Gaussian input distribution used in the full simulations described in the Results section. The first 15 dimensions are interpreted as carrying useful or interesting information, while the remaining 45 dimensions are interpreted as noise. Standard deviations are assigned according to $\sigma_i = (\alpha/i)$ for $i = 1, 2, ..., 15$ and $\sigma_i = 0.1$ for $i = 16, 17, ..., 60$, then normalised such that $\sum_i \sigma_i^2 = 1$. We use $\alpha = 1.6$. The distribution is largely insensitive to this parameter choice, and varying α does not qualitatively affect our results.

of units that, after the network has finished growing, were added in environment B. Up to now we have used an adaptation parameter of $p = 0.25$, so that one quarter of the units in the network either grew or were able to adapt to changes in input statistics. We chose this value as it approximates the degree of neurogenesis observed over the lifetime of a real animal (Bayer *et al.*, 1982; Boss *et al.*, 1985). The performance of the network using the neurogenesis strategy at $p = 0.25$ is significantly better compared to partial turnover. However, it is important to understand the effect of changing p, and, in particular, to show whether regions exist where neuronal turnover is actually better than neurogenesis or whether neurogenesis is always superior. We therefore examine network performance for the full range of p between zero and one. We use the same 60-dimensional Gaussian distribution defined above, and perform simulations, as before, by drawing 1,000 input vectors and encoding them using a network of 300 units with the adaptation parameter, p, ranging from 0 to 1. We repeat this 100,000 times to average over all input vectors, encoding vectors, and angles between environments A and B.

Figure 6 is a plot of the recoding and retrieval errors for the neuronal

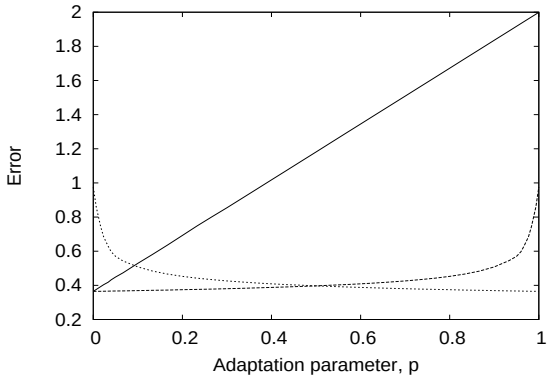

Figure 6: Effect of changing the adaptation parameter, p, on the recoding and retrieval errors for a network using neuronal turnover. We plot the recoding error in environment A (dashed line), the recoding error in environment B (dotted line), and the retrieval error for patterns previously laid down in environment A (solid line), for a network adapted to environment B. $p = 0$ corresponds to a fixed network, and $p = 1$ to the full turnover network. As p increases the recoding error in environment B falls, and there is a corresponding increase in the recoding error in environment A. However, as we increase p the retrieval error increases linearly as retrieved patterns are incorrectly decoded by the network. This increase is unavoidable and stems from the remapping of DG units that were previously used to encode memory patterns in environment A.

turnover strategy when the network is adapted to environment B. Three curves are shown; the recoding error in environment A, the recoding error in environment B, and the retrieval error of patterns laid down previously in environment A. These three errors form the lower set of entries in Table 3. $p = 0$ corresponds to a completely fixed network, $p = 1$ to the full turnover network, and values in between to partial turnover networks. For a fixed network the recoding and retrieval errors for environment A are identical. The exact value of this error is determined by the number of dentate gyrus units we provide the network with and we can, at least in principal, make this error arbitrarily small. For a fixed network the recoding error in environment B is higher than for environment A as the network can, on average, reuse only a small number of units that are adapted for A. As p increases there is a reduction in the recoding error in environment B derived from the increasing number of units which have turned over and are adapted to the new input statistics, and a corresponding increase in the recoding error for new patterns from environment A as not all of the units in the DG are now adapted to A. At the same time, the retrieval error for patterns previously laid down in A increases. This occurs because the meaning of some of the units in the DG has changed in the time since the patterns were originally stored, and, as a result, these patterns are incorrectly decoded by the network. Note that the increase is linear because any randomly chosen dentate gyrus unit is, in the averaged sense, of equal importance to the network as any other unit. The maximum retrieval error is reached for a fully adapting network at $p = 1$. In such a network, all of the dentate gyrus units have adapted to accommodate environment B. In other words, we first randomly initialise the dentate gyrus units using the statistics of environment A then randomly reinitialise them using the statistics for environment B. As the retrieval error is a comparison, in the mean square sense, of the initial encoding vector (adapted to A) and the new decoding vector (adapted for B), we therefore have a comparison of randomly drawn vectors from two distributions, A and B, which is simply the sum of the total variance of the two underlying distributions. We defined each distribution to have a total variance of one, thus, at $p = 1$, the retrieval error reaches a maximum value of two.

¡Figure 7 shows the corresponding curves for the neurogenesis strategy. $p = 0$ corresponds to a network in which all of the units are present from

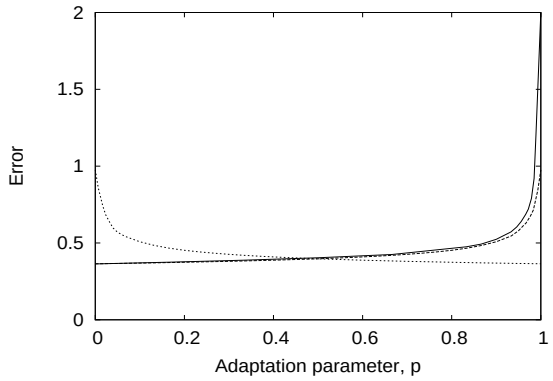

Figure 7: Effect of changing the adaptation parameter, p, on the recoding and retrieval errors for a network using a neurogenesis strategy. We plot the recoding error in environment A (dashed line), the recoding error in environment B (dotted line), and the retrieval error for patterns previously laid down in environment A (solid line), for a network adapted to environment B. As for the conventional adaptation strategy, as p increases the recoding error in environment B falls and there is a corresponding increase in the recoding error in environment A. This time, there is only a very gradual rise in the retrieval error as neurogenesis does not interfere with the decoding of previously stored and retrieved patterns.

the start and the network is not allowed to grow, in effect such a network is identical to the fixed network discussed above. As $p \to 1$ we have a network in which very few of the units are present from the start and almost all are allowed to grow on entry to environment B. At $p = 0$ the turnover and neurogenesis strategies are equivalent, and the errors are the same. As p increases from zero we see the same reduction in the recoding error in environment B, and increase in the recoding error for new patterns from environment A, as we saw for the neuronal turnover adaptation strategy. Again, this is derived from the increasing number of units which have adapted to the new input statistics. However, in contrast to the neuronal turnover adaptation strategy, the retrieval error for patterns previously laid down in environment A increases only very slowly. This gradual increase is due to the reduction in the number of units used to initially encode patterns from environment A. There is no disruption of decoding as the new units which have been added to the DG do not interfere with the interpretation of previously laid down patterns.

Comparing the curves shown in Figures 6 and 7, we see that, once the network has adapted to environment B, for every value of p the neurogenesis strategy performs better than the neuronal turnover adaptation strategy. This observation is also true if we examine the mean of the recoding and retrieval errors, which may be viewed as one method of conveniently assessing overall network performance, as a function of p. Interestingly, we find that this averaged error is minimised for a p-value around 0.3, which is very similar to experimental observations of the amount of neurogenesis that occurs over the lifetime of an animal.

4 Discussion

The occurrence of adult neurogenesis in a variety of species is now a well accepted experimental result, but a theoretical analysis of the computational implications has only recently started to receive attention. Existing studies have explored neurogenesis in a variety of networks, using a variety of different learning rules (Gould *et al.*, 1999; Cecchi *et al.*, 2001; Chambers *et al.*, 2004; Becker, 2005; Crick and Miranker, 2005; Aimone *et al.*, 2006; Chambers and Conroy, 2007). These studies have been useful in illustrating some of the computational properties of neurogenesis, in particular when neurogenesis

acts as a general or targeted turnover of neurons and new units are generated as replacements for dying units in the network. Although neuronal turnover is the dominant process in the olfactory bulb, where it appears to act as a replacement mechanism for worn-out receptor cells (Ming and Song, 2005), hippocampal neurogenesis is apparently an additive process, so that new neurons are added to an existing population which grows over time (Bayer *et al.*, 1982; Boss *et al.*, 1985).

Here, we have sought to determine the functional consequences of additive neurogenesis in a simplified memory model of the hippocampus. In earlier work, it was shown that additive neurogenesis can play a useful role in avoiding a form of interference in a simple, linear feedforward neural network (Wiskott *et al.*, 2006). This form of interference occurs when a network that stores and retrieves memory patterns is required to adapt its encoding and decoding to deal with a changing input environment, and is a graded effect distinct from other forms of interference such as that occurring in a Hopfield net when too many patterns are stored and network performance breaks down (Hopfield, 1982). In Wiskott, Rasch and Kempermann (2006) it was hypothesised that additive neurogenesis might also be used to avoid this form of interference in biology, specifically in the dentate gyrus of the hippocampus. Here we have explicitly evaluated this hypothesis by examining additive neurogenesis in a simplified hippocampal memory model. This model incorporates both a divergence in dimensionality from the EC to the DG and sparse coding within the DG, leading to a highly non-linear computation that is very different in nature to that of the previous, linear model. The network received input that is structured in a simple way that depends on the current environment, and we examined the performance of the network when this environment changes.

We have shown that fixing the encoding for the initial environment produces poor performance in the new environment. Allowing neuronal turnover in the network, so that some of the units may be deleted and reinitialised according to the statistics of the current environment, remedies this but introduces significant errors when retrieving previously stored patterns from the initial environment due to a mismatch of encoding and decoding vectors in the network. Allowing the whole network to turn over creates an extreme version of this problem and destroys the retrieval properties of the network entirely. Thus, in our model, adaptation strategies derived from

neuronal turnover are inadequate when encoding and decoding takes place as part of a memory function. In contrast, an adaptation strategy based on additive neurogenesis, where the network starts with a smaller population of neurons adapted to the initial environment and then adds a small number of additional units adapted to the new environment eliminates entirely the problem of interference, and produces a good level of performance in both environments while at the same time preserving the retrieval properties of the network.

In this paper we have focused on the interaction of the EC and DG layers and the implications that the transformation of representation between these two layers has for network function. We have left implicit the parts of the network responsible for the actual storage and recall of input patterns. It has been suggested that CA3 could be the site of storage and recall in the hippocampus, with the highly recurrent CA3 pyramidal cells perhaps forming a Hopfield network. A Hopfield network possesses a number of useful properties that make it appropriate for memory storage, for example pattern completion, where a complete stored activity pattern is recalled from a partial cue. Our model is fully consistent with this view, and it would be relatively straightforward to include an explicit CA3 layer that operates as a Hopfield network. However, we have assumed only that some kind of storage and recall mechanism exists and that it works sufficiently well so that the additional sources of error such as partial or complete failure of retrieval are eliminated. This allows us to focus entirely on the role of the DG, as the recoding error defined in Equation 2 is due solely to the success or failure of the DG to deal with the encoding and decoding problem that the network faces with changing environments.

In our model, new units are added to the DG over time, but we should point out that a net growth in the size of the DG does not also imply that there is no cell death. It is certainly true that cell death can occur in the hippocampus in addition to new cell growth. However, in the model we present here, provided that cell death occurs on a time scale longer than that involved in consolidation of memories to locations outside of the hippocampus, cell death does not play a role in the encoding and decoding function of the network, and would therefore not affect our results in any way. Thus, we do not claim that there is no cell death in the DG only that it does not play any role in our model provided it takes place over a sufficiently long

time scale.

An important assumption in our model is that the EC input is structured in some way, and that this structure is different in environments A and B. For simplicity, we modelled the input patterns as multi-dimensional Gaussian distributions and we introduced structure into this input in a simple and biologically plausible way by choosing standard deviations for A and B according to an approximately exponential distribution. Other methods of structuring the input exist but, in practice, there is little experimental data with which to constrain them. If the distributions possess no structure, so that environments A and B are identical, then there exists a single, universally optimised encoding which eliminates the problem of interference and the need for neurogenesis. We speculate, however, that, provided there is some kind of structure in the input that differs between A and B, the problem of interference likely exists independently of the exact choice of input statistics. Indeed, with the Gaussian distributions we have used there is always considerable overlap between environments A and B in the region of common high density around the origin. With a different distribution this region of overlap could be reduced or vanish entirely. The recoding error for environment B when the system is adapted for environment A would therefore be much higher, and the need for neurogenesis presumably even stronger.

We assumed that the encoding of the EC patterns by the DG was optimised only in the sense that, having placed the encoding vectors, the encoding error was minimised by mapping the input vectors to the nearest encoding vector. We did not consider true optimisation, which fixes the location of the encoding and decoding vectors as a function of the input statistics, nor did we consider how plasticity of new DG units could optimise their location to maximise their usefulness in reducing the output error. Our probabilistic approach has the advantage of being analytically tractable, while at the same time still matching regions of high input and encoding vector density. With optimal encoding we would also expect to see regions of high input density populated by a high density of encoding vectors, although the exact placement of the encoding and decoding vectors would be different and the resulting recoding and retrieval errors lower. However, as this would reduce all of the errors we have calculated we expect that the overall pattern of our results would remain unchanged. Thus, optimisation of the

encoding and decoding would not alter our central result, that additive neurogenesis performs better than neuronal turnover when a network performs an en- and decoding as part of a memory task. In a similar way, we would expect plasticity in the DG to have minimal effect on our model. Indeed, we have performed additional simulations incorporating a gradient of plasticity into new DG units using a neural gas algorithm, so that new DG units have a period of time in which they are highly plastic and can better adapt to the current input statistics, and found that this has very little influence on our results, typically reducing the errors of Table 3 by less than one percent. We may conclude that plasticity (and therefore the optimisation) of new DG cells actually plays a very small role in our model and that it is the neurogenesis (be it additive or otherwise) that is the driving force behind network adaptation.

We have assumed that a $1/M$ code operates in the DG. This produces an appropriate level of sparseness for the network sizes we consider. As we increase the network size we expect more units to be active, until we reach biological levels of around $5,000$ active DG units out of a population of $1,000,000$. We are thus motivated to consider more general K/M codes, where $K = 1, 2, ..., M$. A K/M code may be interpreted in various ways. Overall network activity could be fixed, so that we have a K-winner-take-all mechanism or, alternatively, any number of units from 1 to K may be active at any one time. In either case, the M DG units will partition the input phase space into regions corresponding to different combinations of activity. The details of this partitioning will depend heavily on the activation rule used, and it is difficult to predict how such a change will affect network performance. However, we know that the input phase space will still be divided into a number of states, each of which corresponds to some combination of active DG units. Although the partitioning may be complicated, we may still represent these states using a $1/M$ code with M equal to the number of states. Thus, any K/M code therefore has an analog with the simple $1/M$ code presented here. As the problem of interference exists because there is an encoding from the EC to the DG that changes with time, it is likely that it is independent of the exact nature of the encoding and we therefore expect that qualitatively similar results would be produced with a K/M code. Indeed, in an earlier study a linear neural network with a very different encoding, in the form of a compact code, was used and similar re-

sults were found (Wiskott *et al.*, 2006). Thus, if we were to repeat our present study with a larger network using a K/M DG code we would expect to see the same trends as presented here for the $1/M$ network, and our conclusions would continue to apply.

In conclusion, we have considered additive neurogenesis in a simplified hippocampal memory model that is required to encode and decode patterns in a changing input environment. This hippocampal model differs from the simple, linear, feed-forward network we considered in earlier work as it incorporates both a divergence in dimensionality from the EC to the DG and sparse coding within the DG, both of which alter quite dramatically the nature of the computation in the network. Provided the network receives input that is structured in some way that depends on the current environment, we have shown that a a form of interference exists in the model. We find that adaptation strategies derived from fixed size networks with neuronal turnover are inadequate and produce large errors in either recoding or retrieval tasks. In contrast to neuronal turnover, an adaptation strategy based on additive neurogenesis, where the network starts with a smaller population of neurons adapted to the initial environment and then adds a small number of additional units adapted to the new environment, produces a good level of performance in both environments while at the same time preserving the retrieval properties of the network. This observation holds across the full range of adaptation levels available to the network but, interestingly, we see close to optimal performance at biologically realistic levels of neurogenesis.

Acknowledgement: LW has been partially supported by a grant from the Volkswagen Foundation for a junior research group and PAA has been supported by a grant from the German Federal Ministry of Education and Research (BMBF) for Project C3 of the Bernstein Centre for Computational Neuroscience Berlin.

Type of error	Adaptation strategy			
Network A	Fixed	Partial TO	Full TO	NG
Recoding A	(6) with $M_A = M$ $X = A$	(6) with $M_A = M$ $X = A$	(6) with $M_A = M$ $X = A$	(6) with $M_A = M_1$ $X = A$
Recoding B	(6) with $M_A = M$ $X = B$	(6) with $M_A = M$ $X = B$	(6) with $M_A = M$ $X = B$	(6) with $M_A = M_1$ $X = B$
Network B	Fixed	Partial TO	Full TO	NG
Recoding B	(6) with $M_A = M$ $X = B$	(8) with $M_A = M_1$ $M_B = M_2$ $X = B$	(7) with $M_B = M$ $X = B$	(8) with $M_A = M_1$ $M_B = M_2$ $X = B$
Retrieval A	(6) with $M_A = M$ $X = A$	(9) with $M_{AA} = M_1$ $M_{AB} = M_2$ $X = A$	(9) with $M_{AA} = 0$ $M_{AB} = M$ $X = A$	(6) with $M_A = M_1$ $X = A$
Recoding A	(6) with $M_A = M$ $X = A$	(8) with $M_A = M_1$ $M_B = M_2$ $X = A$	(7) with $M_B = M$ $X = A$	(8) with $M_A = M_1$ $M_B = M_2$ $X = A$

Table 1: Equations and parameters used to calculate the errors. See section 2.1.3 for a definition of the different types of errors and adaptation strategies considered and section 2.2 for the full equations referenced here by numbers.

Type of error	Adaptation strategy			
Network A	Fixed	Partial TO	Full TO	NG
Recoding A	0.39/<0.01	0.39/<0.01	0.39/<0.01	0.55/<0.01
Recoding B	0.74/ 1.00	0.74/ 1.00	0.74/ 1.00	0.85/ 1.00
Network B	Fixed	Partial	Full	NG
Recoding B	0.74/ 1.00	0.51/<0.01	0.39/<0.01	0.51/<0.01
Retrieval A	0.39/<0.01	0.79/ 0.51	2.00/ 2.00	0.55/<0.01
Recoding A	0.39/<0.01	0.47/<0.01	0.74/ 1.00	0.47/<0.01

Table 2: Analytical results for the recoding and retrieval errors for $N = 2/N = 60$. Errors indicated by <0.01 were very close to zero.

Type of error	Adaptation strategy			
Network A	Fixed	Partial TO	Full TO	NG
Recoding A	0.36	0.36	0.36	0.38
Recoding B	0.99	0.99	0.99	1.00
Network B	Fixed	Partial	Full	NG
Recoding B	0.99	0.44	0.36	0.44
Retrieval A	0.36	0.77	2.00	0.38
Recoding A	0.36	0.38	0.99	0.38

Table 3: Simulation results for the recoding and retrieval errors in environments A and B, under various different adaptation strategies for $N = 60$. The rows and columns are the same as for Table 2. Simulation parameters are given in the main body of text and standard deviations describing the input distribution are shown in Figure 5. We set $M_1 = 225$ and $M_2 = 75$, so that $M = 300$.

References

Aimone, J., Wiles, J., and Gage, F. (2006). Potential role for adult neurogenesis in the encoding of time in new memories. *Nat. Neurosci.*, **9**, 723–727.

Altman, J. and Das, G. (1965). Autoradiographic and histological evidence of postnatal hippocampal neurogenesis in rats. *J. Comp. Neurol.*, **124**, 319–335.

Alvarez-Buylla, A. and Kirn, J. (1997). Birth, migration, incorporation, and death of vocal control neurons in adult songbirds. *J. Neurobiol.*, **33**, 585–601.

Barnes, C., McNaughton, B., Mizumori, S., Leonard, B., and Lin, L. (1990). Comparison of spatial and temporal characteristics of neuronal activity in sequential stages of hippocampal processing. *Prog. Brain Res.*, **83**, 287–300.

Bayer, S., Yackel, J., and Puri, P. (1982). Neurons in the rat dentate gyrus granular layer substantially increase during juvenile and adult life. *Science*, **21**, 890–892.

Becker, S. (2005). A computational principle for hippocampal learning and neurogenesis. *Hippocampus*, **15**, 722–738.

Boss, B., Peterson, G., and Cowan, W. (1985). On the number of neurons in the dentate gyrus of the rat. *Brain Res.*, **338**, 144–150.

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. *Eur. J. Neurosci.*, **17**, 2042–2046.

Cameron, H., McEwen, B., and Gould, E. (1995). Regulation of adult neurogenesis by excitatory input and nmda receptor activation in the dentate gyrus. *J. Neurosci.*, **15**, 4687–4692.

Cecchi, G., Petreanu, L., Alvarez-Buylla, A., and Magnasco, M. (2001). Unsupervised learning and adaptation in a model of adult neurogenesis. *J. Comp. Neurosci.*, **11**, 175–182.

Chambers, A., Potenza, M., Hoffman, R., and Miranker, W. (2004). Simulated apoptosis/neurogenesis regulates learning and memory capabilities of adaptive neural networks. *Neuropsychopharmacology*, **29**, 747–758.

Chambers, R. and Conroy, S. (2007). Network modeling of adult neurogenesis : Shifting rates of neuronal turnover optimally gears network learning according to novelty gradient. *J. Cognit. Neurosci.*, **19**, 1–12.

Chechneva, O., Dinkel, K., Schrader, D., and Reymann, K. G. (2005). Identification and characterization of two neurogenic zones in interface organotypic hippocampal slice cultures. *Neuroscience*, **136**, 343–355.

Crick, C. and Miranker, W. (2005). Apoptosis, neurogenesis, and information content in hebbian networks. *Biol. Cybern.*, **94**, 9–19.

Czurkó, A., Czóh, B., Seress, L., Nadel, L., and Bures, J. (1997). Severe spatial navigation deficit in the morris water maze after single high dose of neonatal x-ray irradiation in the rat. *Proc. Natl. Acad. Sci. USA*, **94**, 2766–2771.

Font, E., Desfilis, E., Pérez-Cañellas, M., and Garca-Verdugo, J. (2001). Neurogenesis and neuronal regeneration in the adult reptilian brain. *Brain Behav. Evol.*, **58**, 276–295.

Gould, E., Tanapat, P., Hastings, N., and Shors, T. (1999). Neurogenesis in adulthood: a possible role in learning. *Trends Cogn. Sci.*, **3**, 186–192.

Gould, E., Vail, N., Wagers, M., and Gross, C. (2001). Adult-generated hippocampal and neocortical neurons in macaques have a transient existence. *PNAS*, **98**, 10910–10917.

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

Hasselmo, M. and Wyble, B. (1997). Free recall and recognition in a network model of the hippocampus: Simulating effects of scopolamine on human memory function. *Behav. Brain Res.*, **89**, 1–34.

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA*, **79**, 25542558.

Jung, M. and McNaughton, B. (1993). Spatial selectivity of unit activity in the hippocampal granular layer. *Hippocampus*, **3**, 165–182.

Kempermann, G., Kuhn, H., and Gage, F. (1997). More hippocampal neurons in adult mice living in an enriched environment. *Nature*, **386**, 493 – 495.

Kornack, D. and Rakic, P. (2001). The generation, migration, and differentiation of olfactory neurons in the adult primate brain. *PNAS*, **98**, 4752–4757.

Kunec, S., Hasselmo, M., and Kopell, N. (2005). Encoding and retrieval in the ca3 region of the hippocampus: A model of theta-phase separation. *J. Neurophysiol.*, **94**, 70–82.

Lehmann, K., Butz, M., and Teuchert-Noodt, G. (2005). Offer and demand: proliferation and survival of neurons in the dentate gyrus. *Eur. J. Neurosci.*, **21**, 3205–3216.

Lisman, J. (1999). Relating hippocampal circuitry to function: recall of memory sequences by reciprocal dentate-ca3 interactions. *Neuron*, **22**, 233–242.

Marchioro, M., Nunes, J., Ramalho, A., Molowny, A., Perez-Martinez, E., Ponsoda, X., and Lopez-Garcia, C. (2005). Postnatal neurogenesis in the medial cortex of the tropical lizard tropidurus hispidus. *Neuroscience*, **134**, 407–413.

Markakis, E. and Gage, F. (1999). Adult-generated neurons in the dentate gyrus send axonal projections to field ca3 and are surrounded by synaptic vesicles. *J. Comp. Neurol.*, **406**, 449–460.

McClelland, J., McNaughton, B., and O'Reilly, R. (1994). Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychol. Rev.*, **102**, 419–457.

McDonald, H. and Wojtowicz, J. (2005). Dynamics of neurogenesis in the dentate gyrus of adult rats. *Neurosci. Lett.*, **385**, 70–75.

Ming, G.-L. and Song, H. (2005). Adult neurogenesis in the mammalian central nervous system. *Ann. Rev. Neurosci.*, **28**, 223–250.

Mulders, W., West, M., and Slomianka, L. (1997). Neuron numbers in the presubiculum, parasubiculum, and entorhinal area of the rat. *J. Comp. Neurol.*, **385**, 83–94.

Nadel, L., Samsonovich, A., Ryan, L., and Moscovitch, M. (2000). Multiple trace theory of human memory: computational, neuroimaging, and neuropsychological results. *Hippocampus*, **10**, 352368.

Nottebohm, F. (1981). A brain for all seasons: cyclical anatomical changes in song control nuclei of the canary brain. *Science*, **214**, 1368–1370.

Nottebohm, F. (2002). Neuronal replacement in adult brain. *Brain Res. Bull.*, **57**, 737749.

O'Keefe, J. and Dostrovsky, J. (1971). The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat. *Brain Res.*, **34**, 171–175.

Paton, J. and Nottebohm, F. (1984). Neurons generated in the adult brain are recruited into functional circuits. *Science*, **225**, 1046–1048.

Petreanu, L. and Alvarez-Buylla, A. (2002). Maturation and death of adult-born olfactory bulb granule neurons: Role of olfaction. *J. Neurosci.*, **22**, 6106–6113.

Schmidt-Hieber, C., Jonas, P., and Bischofberger, J. (2004). Enhanced synaptic plasticity in newly generated granule cells of the adult hippocampus. *Nature*, **429**, 184–187.

Scoville, W. and Milner, B. (1957). Loss of recent memory after bilateral hippocampal lesions. *J. Neurol. Neurosurg. Psychiatry*, **20**, 11–21.

Snyder, J., Kee, N., and Wojtowicz, J. (2001). Effects of adult neurogenesis on synaptic plasticity in the rat dentate gyrus. *J. Neurophysiol.*, **85**, 2423–2431.

Takemura, N. (2005). Evidence for neurogenesis within the white matter beneath the temporal neocortex of the adult rat brain. *Neuroscience*, **134**, 121–132.

Treves, A. and Rolls, E. (1994). Computational analysis of the role of the hippocampus in memory. *Hippocampus*, **4**, 374–391.

van Praag, H., Christie, B., Sejnowski, T., and Gage, F. (1999). Running enhances neurogenesis, learning, and long-term potentiation in mice. *PNAS*, **96**, 13427–13431.

van Praag, H., Schinder, A., Christie, B., Toni, N., Palmer, T., and Gage, F. (2002). Functional neurogenesis in the adult hippocampus. *Nature*, **415**, 1030–1034.

Wiskott, L., Rasch, M., and Kempermann, G. (2006). A functional hypothesis for adult hippocampal neurogenesis: avoidance of catastrophic interference in the dentate gyrus. *Hippocampus*, **16**, 329–343.