

Mismatch Detection Neural Circuit Applied to Navigation

ETH zürich



University of
Zurich^{UZH}

Eloy Parra Barrero

Advisor: Dr. Yulia Sandamirskaya

Institute of Neuroinformatics
ETH Zürich & University of Zürich

This dissertation is submitted for the degree of
Master of Science

August 2017

I dedicate this thesis to the peacefulness of Zürich's residential areas.

Acknowledgements

I would like to thank first and foremost my supervisors: Dr. Yulia Sandamirskaya, who has provided me with constant support and guidance, while at the same time allowing me to work on what I found most interesting; and Prof. Dr. Giacomo Indiveri, for acting as the official supervisor and examiner of the thesis. Second, I would like to thank my parents, for everything that they have taught me and for their support throughout my studies. Third, I am grateful to Laia Serratosa for reading the draft. Finally, I thank friends and colleagues at the NSC master's program for many interesting discussions, and the Institute of Neuroinformatics for providing such an intellectually stimulating environment.

Abstract

Detecting mismatches between the world and our model of it is a fundamental aspect of our cognitive capacities. Mismatches reveal the presence of elements and situations that we did not expect, and thus, have not prepared for, often demanding changes in behavior. Additionally, mismatches frequently expose errors in our model of the world, thus providing a useful learning signal. Furthermore, mismatches between the world and our motivations directly indicate situations in which actions should be taken. In spite of the importance and pervasive character of mismatch detection in cognition, its neural implementation is not well understood. This contributes to the lack of common sense often displayed by artificial intelligence systems that are based on neural networks. In this thesis, we present a novel neural network architecture that has mismatch detection as the basis of its operation. The network will signal the presence of unexpected elements (mismatch by excess) and the absence of expected ones (mismatch by deficit). The neurons of the network include biologically inspired dendritic nonlinearities that increase their discriminatory capacity. We introduce a rule for learning the associations between concepts which operates using local information and resembles the mechanism of synaptic tagging and capture of biological neurons. This rule will ensure that connections from different input combinations cluster onto different dendrites, and can also be used to learn the association between desires and the actions that are needed in order to fulfill them. The model is applied to a navigation task in order to demonstrate its capabilities. A navigation system based on the mismatch detection circuit is proposed, where the environment is encoded as a series of landmarks and their relative positions, inspired by findings in animal spatial cognition. The system is simulated in a simple virtual environment, where it detects the addition and removal of landmarks. The mismatch detection network could have applications in artificial intelligence and may also offer insights about how learning and mismatch detection happen in biological brains.

Table of contents

1	Introduction	1
1.1	Mismatch Detection	1
1.1.1	Mismatch Signals in the Brain	2
1.1.2	Predictive Coding	3
1.1.3	Other Models of Mismatch Detection	5
1.1.4	Mismatch and Learning	6
1.2	Spatial Cognition and Navigation	8
1.2.1	Spatial Cognition in the Brain	8
1.2.2	Spatial Cognition and Object Recognition	12
1.2.3	Spatial Cognition in Artificial Systems	14
1.2.4	Spatial Cognition in Biologically Inspired Systems	14
2	Models	17
2.1	Mismatch Detection Neural Circuit	17
2.1.1	Qualitative Overview	17
2.1.2	Detailed Description	24
2.2	Spatial Cognition and Navigation System	32
3	Results	37
3.1	Mismatch Detection Neural Circuit	37
3.1.1	Basic operation	37
3.1.2	Dendritic Clustering and Weight Homogenization	39
3.1.3	Capturing the Statistics of Associations	44
3.1.4	Blocking Learning during Transients	45
3.1.5	Learning to Act	50
3.2	Spatial Cognition and Navigation System	54

4 Discussion	65
4.1 Summary and Conclusions	65
4.2 Outlook	68
References	71

Chapter 1

Introduction

This thesis presents a novel neural mismatch detection circuit applied to a navigation task. The structure of the thesis is as follows. In this introductory chapter, the problems of mismatch detection and navigation are presented and the related literature is reviewed. In Chapter 2, the models developed for the mismatch detection circuit and the navigation system are described. Chapter 3 contains simulation results for both, and chapter 4 a general discussion. The reader can choose whether to proceed in this order or to read the sections pertaining to the mismatch detection circuit first, followed by the sections related to the navigation system.

1.1 Mismatch Detection

Imagine that upon returning home, you find that the floor has new tiles, your bed has shrunk in size and the lamps are missing. You will quickly notice these facts and find them puzzling. This example illustrates an activity we effortlessly engage in all the time, namely, checking whether what we experience matches with our models of the world and detecting when those models fail.

Why would we engage in such a burdensome task? One possible reason is that mismatches between the world and our model of it expose errors in our model that need to be corrected, thus providing a useful learning signal. However, not all mismatches require learning, at least not of a durable kind. For instance, the realization that a car is approaching the road that you are about to cross will produce a brief mismatch signal, since you were unaware of it. This mismatch will, nevertheless, disappear as soon as the activations in the model are updated to reflect the new state of affairs, without any learning being required since you already knew that cars go on the road. There is, however, a reason that makes all mismatches equally relevant, and that is the fact that they indicate the presence of elements and situations that

you did not predict, or could not have predicted, and that therefore you have not prepared for, often demanding changes in behavior. This becomes particularly relevant in the context of a system with limited cognitive resources that can not process all stimuli to the same extent and thus requires some mechanism to select the information that it will focus on.

Another case in which mismatch detection is fundamental is when the world is being compared to your desires. Here, mismatches directly indicate situations in which actions should be taken.

Despite the differences in the cause of the mismatch across the cases mentioned, the nature of the mismatches themselves seems to be essentially the same. At a fundamental level, mismatches can be classified into two categories. First, there can be some element in the world that is not active in the model, thus, the presence of this element is unexplained or unjustified. We will call this mismatch by *excess*, due to the presence of this "extra" element that is not supported by the model. Otherwise, we may have the opposite case; there can be an active element in the model that is missing in the world. We will call this mismatch by *deficit*, meaning that there is a lack of some element in the world with respect to our model. The apparently more complicated case of deformations can also be construed in terms of these two types of mismatch: shifted elements may elicit excess mismatches at their new positions and deficit mismatches at their reference ones; alternatively, if the pattern is encoded in a relational manner, excess and deficit mismatches may arise for the magnitudes that relate the elements to each other. Since mismatches reduce to these two simple categories, it seems plausible that a single mechanism, or at least similar mechanisms, could explain mismatch detection in general.

In spite of the relevance and pervasive character of mismatch detection in cognition, its neural implementation is not well understood. Perhaps because of this, artificial intelligence systems that are largely based on artificial neural networks generally also lack this capability. This fact accounts for the failure of most AI systems to react appropriately to errors and anomalies, which makes them appear to lack common sense and true understanding.

1.1.1 Mismatch Signals in the Brain

Perhaps the most obvious form of mismatch signal that has been measured in the human brain is the so called Mismatch Negativity (MMN). This is a component of the event-related potential that is measured in response to odd stimuli. In a common experiment, rare deviant sounds are interspersed among a series of repetitive standard sounds, provoking the MMN. The phenomenon has been observed for a wide range of auditory stimuli, including abstract changes such as grammar violations, as well as for stimuli in the somatosensory, olfactory

and visual domains. The MMN is also elicited by omissions of expected stimuli (SanMiguel et al. (2013), Salisbury (2012); and see Näätänen et al. (2007) for a review).

A related, if not the same, kind of signal has been recorded using two-photon imaging in behaving mice. Keller et al. (2012) report that responses in layer 2/3 of mouse primary visual cortex are strongly driven by mismatch between actual and expected visual feedback based on locomotion. Fiser et al. (2016) further report that a subset of neurons, again in layer 2/3 of mouse V1, exhibit responses that are predictive of the upcoming visual stimulus in a spatially dependent manner and that the omission of expected stimuli drives strong responses in the area.

The hippocampus is also believed to be involved in match/mismatch detection. It is hypothesized that area CA1 may act as a comparator between predictions arriving from CA3 and "reality" arriving from the cortex. This mismatch signal would be used to adapt its dynamics for learning or recall (Hasselmo and Schnell (1994), Lisman and Grace (2005)). Consistent with this hypothesis, fMRI studies by Kumaran and Maguire (2007), Chen et al. (2011) and Duncan et al. (2012) have found that area CA1 exhibits a pattern of activity consistent with that of an associative match/mismatch detector.

Midbrain dopaminergic neurons are also known to code for a signed error in prediction of reward. Unpredictable rewards reliably elicit neuronal activations. If the reward becomes predictable, the activations elicited in response to it decrease and shift towards the reward-predicting stimulus. If however, the reward fails to occur, dopamine neurons are depressed below baseline at the time when the reward was expected (Hollerman and Schultz, 1998)).

For an older but extensive review on neural coding of prediction errors, see Schultz and Dickinson (2000).

1.1.2 Predictive Coding

The Mismatch Negativity and related mismatch signals are often interpreted under the light of predictive coding. The basic idea of predictive coding is that higher-level areas constantly send predictions to lower-level areas producing prediction errors. These prediction errors are in turn sent back to the higher-level areas to update their hypothesis.

Rao and Ballard (1999) first proposed predictive coding as a model of cortical processing in order to explain neurons with extra-classical receptive fields in visual cortical areas. These are neurons, abundant in layers 2/3, that respond to an optimally oriented line segment of a certain (small) longitude, but whose response is reduced or eliminated when the longitude of the segment increases. Rao and Ballard suggested that these neurons coded for prediction errors which were caused by the line segment being too short to activate the corresponding higher-level representation that would explain away the error. A simplified version of their

neural circuit for predictive coding is illustrated in Fig. 1.1. Wacongne et al. (2012) have also proposed a model based on predictive coding that accounts for some critical features of the MMN in auditory cortex.

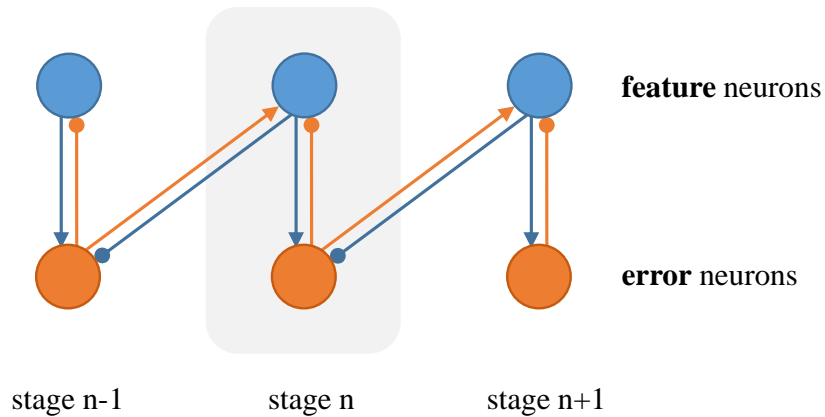


Fig. 1.1 Basic predictive coding neural circuit, adapted from Rao and Ballard (1999). The complete circuit is only drawn for the central feature, and only one feature per stage is shown. Connections ending in an arrow are excitatory, whereas those ending in a circle are inhibitory.

Predictive coding has been mathematically formalized and has been shown to approximate Bayesian inference based on the minimization of a function referred to in statistics as free-energy (Friston (2005), Bogacz (2017)). A further advantage of the model is that computation is local, with the rules describing the weight updates being Hebbian, proportional to the activation levels of the pre- and post-synaptic neurons. This adds to the model's biological plausibility. Bastos et al. (2012) have even proposed a mapping of the model onto the canonical microcircuit postulated by Douglas et al. (1989).

Predictive coding is thus an elegant model with plenty of explanatory success. However, some of its features are not fully satisfactory. For instance, the model does not seem to be well suited to explain the detection of mismatches where the expected element is found to be missing. In these cases, error neurons only receive the prediction input, but since predictions are inhibitory, this leads to negative activation levels. These can not be communicated in the brain by means of action potentials, unless the neurons have a high baseline rate and negative activation levels correspond to firing rates below baseline. However, this would be energetically inefficient, and would also require further circuitry in order to be used. Predictive coding therefore seems to account only for one of the two types of mismatch, namely, mismatch by excess but not by deficit.

Another possibly problematic feature of the model is that information to the next area in the hierarchy stems only from the error neurons, that is, information is only conveyed

when there is an error. This is typically advertised as the main virtue of predictive coding, since it would lead to less message passing and thus higher energetic efficiency. However, it would also mean that once a stimulus is explained, it virtually disappears for the system and would not be able to participate in any additional operation. This does not seem to be advantageous or match our introspective experience. According to predictive coding, when in a familiar setting, the world should literally disappear, and although it is true that attention will typically be allocated to mismatches, it is not the case that explained phenomena vanish and become inaccessible.

1.1.3 Other Models of Mismatch Detection

Besides predictive coding, other more specialized neural models have been proposed to address specifically the generation of match/mismatch judgments.

For example, Johnson et al. (2009) propose a dynamic neural fields model (Schöner et al., 2016) for working memory and change detection. The model consists of an excitatory *perceptual* field, an excitatory *working memory* field and a shared inhibitory field that is reciprocally connected to both excitatory fields. Inputs arrive to the *perceptual* field, where they form a peak of activation which then projects to the *working memory* field, forming a second peak of activation. This one, however, will remain active after the input is gone through stronger recurrent local excitation. When the same input is presented again, it will not be able to form a peak in the *perceptual* field since the field at that location is being strongly inhibited by the working memory peak (through the inhibitory layer). Thus, only new inputs can form peaks of activation in the *perceptual* field. Using this information, the model identifies the presence of an input that is different from the working memory prediction, that is, it signals mismatch by excess.

Engel and Wang (2011) present an original model composed of two types of neurons which they call match-enhancement and match-suppression neurons. Their circuit is illustrated in Fig. 1.2. Both types of neurons receive perceptual bottom-up input, inhibit each other (all to all), implementing a winner take all where one of the populations will tend to win, and self-excite, although the self-excitation in the match-enhancement neurons is weaker. Match-enhancement neurons, however, also receive top-down input from the reference pattern. When the bottom-up and top-down inputs match, match-enhancement neurons will receive more input and dominate over match-suppression neurons. However, when the inputs do not match, match-suppression neurons receiving the bottom-up input will win the competition simply because of their stronger self-excitation. Thus, match-enhancement neurons signal the match condition whereas match-suppression neurons indicate mismatch by excess.

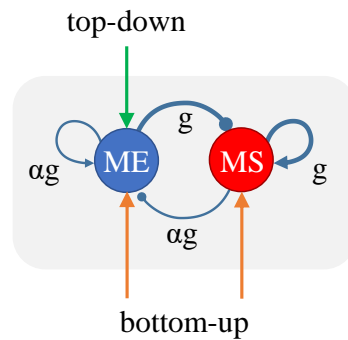


Fig. 1.2 Circuit mechanism composed of match-enhancement (ME) and match-suppression (MS) neurons, adapted from Engel and Wang (2011), Fig. 3.

Haikonen (2014) proposes a simple model where three conditions are detected, which he calls *match*, *mismatch* and *novelty*. The *match* condition is the result of performing the operation $\text{AND}(\text{input vector}, \text{feedback vector})$; *mismatch* corresponds to $\text{AND}(\text{NOT}(\text{input vector}), \text{feedback vector})$; and *novelty* results from $\text{AND}(\text{input vector}, \text{NOT}(\text{feedback vector}))$. Haikonen's *mismatch* signal is what in this thesis is called mismatch by deficit and his *novelty* signal is considered mismatch by excess. We prefer this choice of words since a mismatch by deficit can also be considered to be a novelty.

Notably, Haikonen's model detects both types of mismatch whereas the rest of the models reviewed focus only on mismatch by excess. However, one disadvantage of this model is that it requires both the input and reference patterns to be explicitly represented in order for them to be compared element by element using his proposed boolean logic. This means that more memory resources are needed in order to hold both patterns and that the computations become more complex, since the duplicity has to be managed and also some associative mechanism is needed in order to reproduce from memory the model patterns.

Regarding the match signals present in these models, we consider them secondary to mismatch signals. As suggested at the beginning, mismatch signals are very useful for detecting new and unexpected elements in order to allocate more processing resources to them. In contrast, the use of match signals seems to be much more limited and in many cases, detecting a match condition can be equivalent to detecting the absence of mismatch signals. Therefore, in this work we will focus exclusively on the latter.

1.1.4 Mismatch and Learning

As mentioned earlier, mismatches expose errors in our model of the world that often need to be corrected by means of learning, suggesting a tight link between these two phenomena.

Most of the early research on learning was performed in the context of animal conditioning, and in particular classical conditioning. Classical conditioning refers to the associative process by which an initially neutral stimulus, the (to-be) conditioned stimulus (CS), becomes associated with a naturally positive or aversive stimulus, the unconditioned stimulus (US), such that it starts to produce the same response as the US.

A prominent model of classical conditioning is the Rescorla-Wagner Model (Rescorla and Wagner (1972), reviewed in Miller et al. (1995)). The model famously succeeded in explaining the blocking effect. This is the observation that conditioning a response to stimulus X after $AX \rightarrow US$ pairings is impaired if A alone had previously been paired with the US. In essence, the model states that learning is proportional to the error in the prediction of the US, therefore, when the US is already predicted by a previously conditioned CS, conditioning of new CSs will be hindered. The model is formalized by the following equations indicating the change in associative strength (V) of a CS, X , after a pairing with US_1 on trial $n + 1$:

$$\begin{aligned}\Delta V_X^{n+1} &= \alpha_X \beta_1 (\lambda_1 - V_{total}^n) \\ V_X^{n+1} &= V_X^n + \Delta V_X^{n+1}\end{aligned}\tag{1.1}$$

where α_X is the associability of CS X (range 0 to 1 and related to the intensity of the CS), β_1 is the associability of US_1 (range 0 to 1 and related to the intensity of the US), λ_1 is the maximum associative strength that US_1 can support, and V_{total}^n is the sum of the associative strengths of all CSs (including X) that are present on trial $n + 1$.

Learning is thus proportional to the surprise or prediction error term, $(\lambda_1 - V_{total}^n)$. In the model by Pearce and Hall (1980), this error term also appears, although it is considered to represent the associability of the CS.

Sutton and Barto (1981) expanded on this type of model, essentially converting the Rescorla-Wagner model into a continuous time version and adding eligibility traces. Later, they developed the Temporal-Difference (TD) algorithm of reinforcement learning, where learning is still proportional to errors in the prediction of reward, albeit in a slightly more complicated way. They relate this model back to classical conditioning in Sutton and Barto (1990). Their model has also been related to the activity of dopaminergic neurons, as reviewed in Glimcher (2011).

Given the rewarding or aversive nature of the stimuli used in animal conditioning experiments, it is not surprising that their results and models have often been seen through the lens of reinforcement learning. It seems plausible, however, that the dependency on prediction errors that these models postulate relates to learning in a more general sense; the rewarding or aversive nature of the stimuli in the experiments being simply needed to make the animals care about learning the associations. As already mentioned, in the framework of predictive

coding, weight updates are proportional to the activity of the pre- and post-synaptic neurons, with the latter representing prediction errors. In the mismatch detection circuit presented in the next chapter, learning will also be proportional to the mismatch signals.

1.2 Spatial Cognition and Navigation

Now we switch to the topic of spatial cognition. This was the initial objective of the thesis before the mismatch detection circuit was conceived. However, navigation is a case in which mismatch signals are very common. In fact, many of them are triggered by changes in our environment that we detect as we navigate through it. Thus, both subjects meet and a model for navigation based on the mismatch detection circuit will be proposed. This model will serve as a practical application of the circuit that will both stimulate its development and demonstrate its capabilities, while, of course, also serving to explore the very challenging topic of spatial cognition.

1.2.1 Spatial Cognition in the Brain

The ability to learn a representation of space and use it for navigation is with no doubt fundamental to any advanced creature. The last four decades have seen substantial progress in our understanding of the neural mechanisms responsible for spatial cognition. In particular, a number of cell types coding for different aspects of navigation have been found in the hippocampal formation. These findings are summarized next, and are extensively reviewed in Moser et al. (2008) and Hartley et al. (2014). The data that supports them comes from *in vivo* extracellular unit recordings in freely behaving animals, mainly rodents.

The experimental study of spatial representations in the brain was initiated by the discovery of place cells by O'Keefe and Dostrovsky (1971). These are neurons that present increased firing rates when the animal is at a specific location, referred to as the neuron's 'place field'. The firing rate is normally insensitive to the animal's direction. Place cells recorded at dorsal sites tend to have smaller place fields while those recorded at ventral sites are more broadly tuned. Place cells are not topographically organized and the same cells can participate in the representation of different environments occupying different positions in each of them. However, O'Keefe and Burgess (1996) reported that the location most place cells' fields across several boxes of different magnitudes appeared to be set in relation to the distances or proportions of distances to the box walls along the box axes directions. Besides geometric information, place cells are also sensitive to variables such as colors or odors (Anderson and Jeffery, 2003) and are more commonly found close to behaviorally

relevant landmarks, and close to the walls of the recording chambers, particularly if they have orienting cues (Hetherington and Shapiro, 1997).

An interesting property of these cells is the phenomenon known as phase precession. Place cells will fire at different times relative to the ongoing theta oscillation as the animal progresses through the cells' place field. Thus, at a given point in time, if the animal is exiting place cell X's place field, in the middle of Y's and entering Z's, X will fire at late phase, Y will do so at an intermediate phase and Z at an early one.

Another important cell type found in the navigation network are head direction cells, described by Taube et al. (1990). These are cells that fire as a function of the animal's head direction in the horizontal plane, independent of the animal's behavior, location, or trunk position. The firing rate of each cell is maximal for a single direction and decreases linearly with angular deviation from it. Different cells seem to be uniformly distributed over all possible directions. Their firing patterns are anchored to cues in the environment and can be made to rotate in unison when the orienting cues are rotated, also affecting the position of place cells. Recently, bidirectional head direction cells have been found that display two peaks separated by approximately 180° (Taube, 2017).

Grid cells (Hafting et al., 2005) display multiple firing fields which tessellate the environment forming a triangular pattern. Different grid cell groups will present patterns of different scale, orientation and spatial phase. Like head direction cells, the relative firing pattern of grid cells to one another is held constant across environments and is also anchored to landmarks, although it can continue for some time in the dark. Their function and formation is not well understood, but they are believed to be involved in path-integration, that is, the use of self-motion signals to estimate traveled distances and directions. Kraus et al. (2015) provides evidence for this from an experiment where rats run in a treadmill. They report that during running in place, grid cells signaled a non-linear, often multi-peaked representation of time or distance traveled, or a combination of both.

The dependency of place cells on boundaries suggested that these would also be represented in the brain. Indeed, boundary cells have been found that fire at short distances from the edges of the environment, for example, whenever a barrier is found at approximately 5cm to the south of the animal.

Kropff et al. (2015) have also reported speed cells characterized by a context-invariant, positive and linear response to running speed.

It is worth noticing that many of the spatial cell types show complex responses to combinations of locational, directional and sensory information. Qualitative reports of these cell's responses highlight this fact, for example, in O'Keefe and Dostrovsky (1971): "[the neuron] had no spontaneous activity and only fired when the rat was pointing in the directions

marked by A or B and was simultaneously lightly but firmly restrained by a hand placed over its back with thumb and index finger on its shoulder and upper arm. Both the particular orientation and tactile stimulus were necessary." This speaks of the complexity of the system and suggests that results presented should be taken with caution.

The activity of the cell types described above is often interpreted as evidence that the hippocampal formation supports a systematic, cohesive and allocentric representation of space known as the cognitive map (O'Keefe and Nadel, 1978). According to this view, as animals move through their environment, they maintain and update a representation of their own allocentric position and orientation within a map, from which they compute the current egocentric distance and direction to goals. Thus, mammalian navigation would resemble the processes that humans use when navigating by means of an actual map.

Eichenbaum et al. (1999) challenge this view by noting that there is no evidence of such a systematic representation of spatial loci bound together within a unified frame of reference. As noted earlier, place cells do not seem to sample space uniformly, being more common next to landmarks and boundaries. In addition, different place cells can encode locations relative to different frames of reference simultaneously, for example, relative to the fixed environment and to task-relevant, mobile objects (Gothard et al., 1996). Thus, instead of encoding a systematic cognitive map, Eichenbaum et al. propose that the hippocampus acts as a general memory space where events get linked into sequences, in agreement with the views that regard the hippocampus as crucial for episodic memory in humans. Place cells would simply correspond to the nodes of a particular type of episodic memory.

In agreement with this, Aronov et al. (2017) propose that the hippocampal formation serves as a general mechanism for encoding continuous, task-relevant variables. They performed an experiment in which they trained rats to manipulate sounds along the frequency axis and found that hippocampal and entorhinal neurons represented the entire behavioral task, including activity that formed discrete firing fields at particular frequencies. Interestingly, they showed that the neurons modulated by their task overlapped with neurons that acted as place and grid cells in a foraging task.

Wang and Spelke (2000) also challenge the idea of a cohesive allocentric map and propose instead that mammal navigation may be closer to that of insects. Collett et al. (1998) have described how desert ants navigate through the use of a global 'homing' vector that always points to the direction of their nest and local vectors between landmarks. Mammals are also capable of extracting these vector representations out of path integration, and humans can combine several of these vector representations to find novel shortcuts, as reviewed in Etienne and Jeffery (2004). It seems plausible, therefore, that humans and other mammals represent

space not as a cohesive map, but as a series of linked vectors indicating the distances and directions between landmarks.

Warren et al. (2017) defend a very similar idea, arguing that human's spatial representation is best understood as a 'labeled graph' where nodes represent landmarks and contain local information about path lengths and junction angles to neighboring landmarks without the nodes being embedded in globally coherent reference system. They provide evidence for this from a virtual reality experiment where they introduce wormholes. Klauss et al. (2015) also provide strong evidence for this hypothesis in another virtual reality experiment where participants learn to navigate through 'impossible worlds' with severe violations of euclidean geometry. Two examples of such worlds are shown in Fig. 1.3. When participants had to reproduce the learned patterns by walking "blindly", no angular distortions were observed, meaning that the spatial representations that they had learned were globally incoherent. Notably, the participants did not even realize that violations of geometry were present.

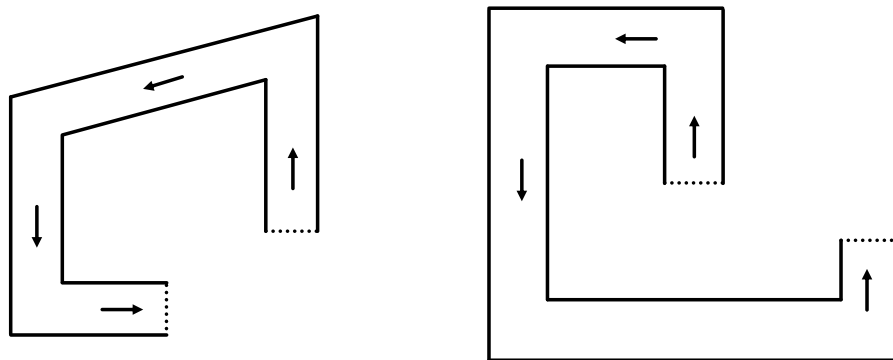


Fig. 1.3 Impossible triangle and impossible square used in the experiment by Klauss et al. (2015). Dotted lines represent topological continuity that can not be displayed.

Finally, Foo et al. (2005) and Poucet et al. (2013) provide experimental evidence for the high reliance of humans and rats respectively on landmarks for navigation, which relaxes the demands on metrically accurate representations of space.

Thus, the observations in animals and humans seem to favor the existence of graph-like representations of space. This is not surprising, since this kind of representation offers several advantages. First, it reduces the complexity of the system by sidestepping the problem of maintaining globally coherent maps. Second, it forms more compact representations since it only encodes relevant landmarks and not empty space. Finally, it facilitates navigation by directly providing the information that is necessary for going from one place to another,

whereas additional mechanisms would be required in order to extract this information from a Cartesian map.

1.2.2 Spatial Cognition and Object Recognition

Further support for a distributed, graph-like representation of space may come from recognizing the similarity between places and objects. Essentially, both places and objects are arrangements of features in space and often, the boundary between them is not clear (think, for example, of a boat, a bed, or the Eiffel Tower). In many cases, places and objects can be recognized based on vision in apparently the same way. Furthermore, both the recognition of places and objects often involves a concatenation of movements and sensory impressions; in the case of objects, the movements correspond to saccadic eye movements, whereas for large places, locomotive movements are also required.

The advantage of looking at spatial cognition through the lens of object recognition is that the latter seems to be more mature as a field, specially in terms of developing neurally inspired artificial systems. After several decades of trying to model vision, what can be recognized easily is that standard neural network approaches to object recognition have trouble with the recognition of objects invariant to translation, scale and rotation. In the popular deep convolutional neural networks (Lecun et al., 1998), translation and scale invariance are solved by repeating weight kernels at different positions and scales, whereas rotation invariance requires training the network with rotated samples. While these solutions provide good results, they are computationally expensive and the weight sharing scheme that they employ does not seem to be biologically plausible.

The solution that suggests itself is equivalent to the one proposed before for the case of places: objects are encoded in terms of a set of features and the relative distances and angles between them. Thus, when recognizing an object, its translation, scale and rotation could be estimated and then used to systematically correct the extracted distances and angles so that a comparison with the object's memorized representation can be made. This systematic correction would amount to additions and multiplications. Saccadic eye movements, which are a problem for other models of vision would in fact be helpful here, since they could be used to obtain the distances and angles between the elements of the pattern. Otherwise, these distances and angles could be calculated by some internal mechanism. If the angles are specified as an allowed range of values, instead of a single value, this model would also easily account for the recognition of objects with flexible parts, like joints.

This type of model has been proposed by Noton (1970), who suggested that patterns are encoded in feature networks like the one illustrated in Fig. 1.4. The feature network would be composed of the features of the pattern and the attention shifts required to pass from one

feature to another. The model has the further advantage that memorizing a pattern becomes closely analogous to memorizing and repeating a conventional sequence of behaviors, each being an alternating sequence of sensory and motor activities.

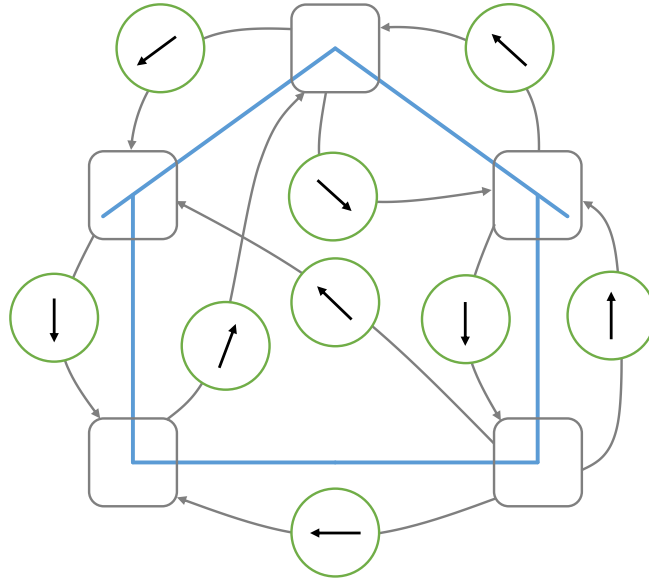


Fig. 1.4 Possible feature network for a house-pattern, following the model proposed by Noton (1970).

Rybak et al. (1998) build a similar model in which patterns are stored in terms of relative distances and angles between features. However, in contrast to Noton's proposal, where angles are all relative to a common reference frame, Rybak et al. use feature-based reference frames that are aligned with the features' gradient of brightness. Their model performed well in recognizing translated, scaled, rotated and partly occluded objects.

Zetsche et al. (2008) have proposed a similar model, where patterns are encoded in terms of 'sensorimotor features', composed of the triple (s_{i-1}, m_{i-1}, s_i) , that is, origin sensory feature, motor action and final sensory feature. Interestingly, they use the same scheme for controlling saccadic eye movements and performing object recognition (the micro-scale) and for controlling body movements and performing place recognition (macro-scale). The movements of the body are defined in terms of the initial angle that has to be turned from the starting view to the next place, the distance traveled, and the angle that orients the agent towards the target view. This is shown in Fig. 1.5. They integrate both systems in a simulated robot and test it in a large set of virtual rooms.

To our knowledge, Zetsche et al. are the only ones to have applied a strategy based on relative positions to both object and place recognition.

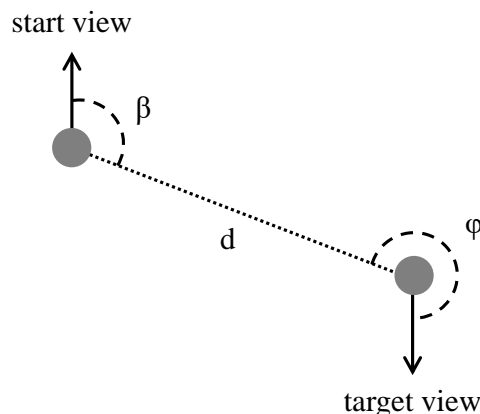


Fig. 1.5 Sensorimotor feature, adapted from Zetzsche et al. (2008) Fig. 3.

1.2.3 Spatial Cognition in Artificial Systems

Mobile robots face the same problems as animals, needing to learn a representation of space that they can then use for navigation. In robotics, this is known as the problem of simultaneous localization and mapping (SLAM) and is a very active area of research. Specifically, the problem is that of acquiring a spatial map of an environment while simultaneously localizing the robot relative to this model. Mathematically, it amounts to calculating the posterior probability, $p(x_t, m \mid Z_T, U_T)$, of the position of the robot, x_t , and the map, m , given the history of sensory observations, Z_T , and of odometric information, U_T .

Despite substantial progress in the field, robust mapping of unstructured, dynamic, and large-scale environments in an online fashion remains largely an open research problem. The majority of SLAM algorithms can be divided into three main families, reviewed in Stachniss et al. (2016). They are based, respectively, on extended Kalman filters, particle filters and graph-based optimization. These models will not be discussed here since they bear little relation to biological navigation. However, it is worth noticing that the challenges faced by these algorithms stem primarily from their commitment to producing metrically accurate Cartesian representations of space out of noisy sensory and self-motion estimates. As it has been suggested above, this may not be necessary, and the apparent difficulty of doing so supports the appeal of distributed graph-like representations.

1.2.4 Spatial Cognition in Biologically Inspired Systems

In this section we introduce a few biologically inspired navigation systems, sampling from a broad range of schemes that highlights the lack of consensus that exists in the field.

We start with a recent proposal made by Kanitscheider and Fiete (2016). They describe a neural model that self-localizes in polygonal environments based only on touch-based information upon contact with the boundaries. The system is very simple, consisting of an input layer, a recurrent layer and an output layer. The input layer encodes noisy self-motion cues like velocity and head direction change, as well as noisy boundary-contact information like relative angle and distance to boundary. The recurrent layer is comprised of 256 Long Short-Term Memory units (which corresponds to 512 dynamical units). The output layer performs a linear readout of the recurrent units providing the estimated location coordinates. The network is trained using a form of stochastic gradient descent.

The network can be trained to localize in 100 environments and it is reported to perform as well as a particle filter with 10^4 particles. Some hidden units exhibit stable place fields reminiscent of those found in the hippocampus. More strikingly, the network can localize in new environments without learning, suggesting that it can build and use dynamic representations of the environment based on a set of fixed weights. It is not clear that the problem solved by the network is the same as that being solved by animal brains, and it is definitely not clear that the solutions are similar, since arguably we create enduring representations of space whereas this network seems to rely to some extent on temporary dynamic representations. The results are interesting in any case.

A very different approach is followed by Conradt (2008). He developed a system which decomposes space into a distributed graphical network of behaviorally significant places. Each place is represented by a "place agent" that maintains the spatial and behavioral knowledge relevant for navigation in that place. Place agents know of their neighbors and how to reach them (distances and angles) relative to their egocentric reference frames. The collection of such place agents does not represent space in any coherent global structure. The system learned to navigate a crowded 60x23m space, creating about 150 place agents. The model is inspired by biological computation and seems amenable to neural implementation although that implementation was not carried out.

Moving towards systems that seek stronger biological fidelity, we introduce a family of neurally inspired navigation systems that employ continuous attractor networks. An early exponent of this is Samsonovich and McNaughton (1997). A more recent model is "RatSLAM", proposed by Milford and Wyeth (2009). Continuous attractors arise out of networks with local excitatory connectivity and broader or general inhibition, thus supporting local peaks of activation that can be easily moved. They can thus be used for representing a continuously varying position in 2D space (x and y coordinates), or, in RatSLAM, in a 3D space that also includes heading direction. As activity shifts in this space representation, neurons get associated to other neurons representing sensory input. On the next visit to

the place, sensory neurons will inject activity in the corresponding (previously associated) neurons representing that place, thus correcting for small errors accumulated from path integration. However, it is not clear that this system can perform properly when large errors accumulate. In RatSLAM, the authors end up introducing an "experience map" that also receives odometric and visual information and seems suspiciously close doing everything independent of the neural system. This family of models thus faces the problems of committing to Cartesian representations of space.

The last model we introduce is that by Cuperlier et al. (2007). They propose a model which includes place cells and transition cells. Place cells are activated by a visual system. A specialized population also encodes the last place visited. Transition cells then link past and current places to the motor commands required to go from one to the other. These motor commands are defined in terms of the distance and direction (with the use of a compass) traveled. At the same time, a graph-like cognitive map is built where the nodes are the transitions, and the edges indicate the frequency with which different transitions are linked. For generating goal oriented behavior, motivation is fed into the cognitive map and diffuses through the graph. At the same time, the currently active place predicts the possible transitions and the combination of both things triggers the next motor command.

The model introduced in this thesis will be most similar to those of Conradt and Cuperlier et al., where space will be explicitly modeled in terms of relative distances and directions between relevant places.

Chapter 2

Models

2.1 Mismatch Detection Neural Circuit

In the first part of the introduction we set the task of finding a neural implementation of mismatch detection. We proposed that mismatches can be of two types: mismatch by excess, when there are elements in the world not predicted or justified by the model; and mismatch by deficit, when there are elements in the model that are found to be missing in the world. Here, we provide qualitative and quantitative descriptions of a novel neural circuit that can account for these two types of mismatches.

2.1.1 Qualitative Overview

The circuit model that we propose is shown in Fig. 2.1 in its most basic form. It is composed of neurons whose activation is described by a real value and that output values in the range $[0,1]$. "Head" neurons, depicted in blue, signal the presence of some feature or concept (concepts are seen as higher-level features or vice versa), typically in an all-or-none fashion as has been reported by Quiroga et al. (2008). In the figure, the "head" neurons represent features A, B, C and Y, which form a pattern. That is, when features A, B and C are experienced, feature Y is also typically experienced. The neurons depicted in orange and green belong to the circuit motif responsible for feature Y. This circuit motif will be repeated for all features, but for clarity, only that corresponding to Y is shown here. Orange "should-not" neurons receive excitatory input from the "head" through a connection of weight 1, and receive inhibitory input from the rest of "head" neurons that compose the pattern. This inhibitory input adds to -1. When all the features of the pattern are active, excitatory and inhibitory inputs at the "should-not" neuron cancel each other out and the neuron outputs 0. When features A, B and C are present but not Y, the neuron is driven to a negative value and also

outputs 0. Only when feature Y is active but the rest of the features of the pattern are not, the "should-not" neuron will get activated with a value of 1. Thus, the activation of this neuron means that feature Y is active but it "should not" be (hence the name), given that no currently active pattern justifies its presence, that is, it signals mismatch by excess. Green "should" neurons play the complementary role. They receive inhibitory input from the head neuron and excitatory input from the head neurons of the rest of the pattern. They become active only when features A, B and C are present but not Y. The neuron therefore signals that feature Y is not active but "should" be, given that the rest of the pattern is present (mismatch by deficit).

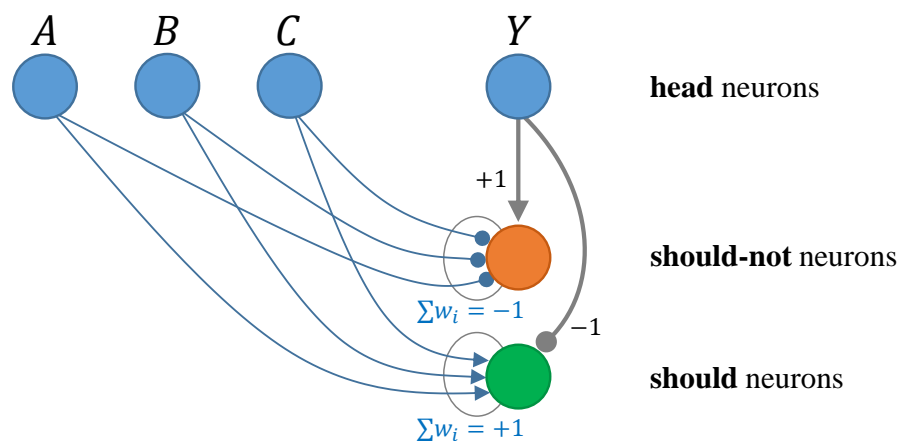


Fig. 2.1 Basic mismatch detection neural circuit. "Head" neurons signal the presence of features A, B, C and Y, which form a pattern. The "should-not" neuron corresponding to feature Y signals that Y is active but "should not" be, whereas its "should" neuron indicates that Y is not active but "should" be.

The connections from a "head" neuron to its "should-not" and "should" neurons, depicted in thicker gray lines are fixed and stereotypical for the circuit motif. The connections from other "head" neurons, depicted in thinner blue lines, are the ones that encode the pattern and will be learned by a mechanism which we address later.

What has been described forms the basis of the model, however, some elements need to be added. So far, each feature or concept has only one pair of "should" and "should not" neurons (henceforth referred to as "s-pair"). However, features or concepts should be able to receive several independent sources of information coming from different stages in the hierarchy (e.g. bottom-up, top-down or lateral connections) or from different sensory modalities. Each source of information should be capable of eliciting its own "should" and "should-not" signals. For example, we might experience something that tastes like salt, and thus activates

the concept neuron for salt, but does not look like it, eliciting a "should-not" signal for salt in the visual but not in the taste domain. Therefore, we expand the circuit to include several s-pairs per concept, where each s-pair interacts with a different source of information. This is illustrated in Fig. 2.2. If some source of information is not available, for example, if you are looking at salt but not tasting it, the s-pair corresponding to the unavailable source of information should be silenced, otherwise those neurons may incorrectly signal that the concept is active but should not be.

If a source of information is very reliable in indicating the presence of some feature, and it is signaling (through the activation of the corresponding "should" neuron) that the feature is not active but should be, then it would make sense that the system activates the feature. Similarly, if it is indicating that the feature is active but should not be, the feature should deactivate. This is what we see in the upward green and orange connections in the figure. Another source of information that is less reliable will do the same but with smaller weights, represented by dotted lines. Once the head neuron has been activated by some "should" neuron, it will remain active through self-excitation until it is inhibited by some "should-not" neuron. Thus, "should" and "should-not" neurons drive the "head" neurons and the basic mode of operation of the network is fully driven by these three types of neurons arranged in the stereotypical circuit motif, or microcircuit.

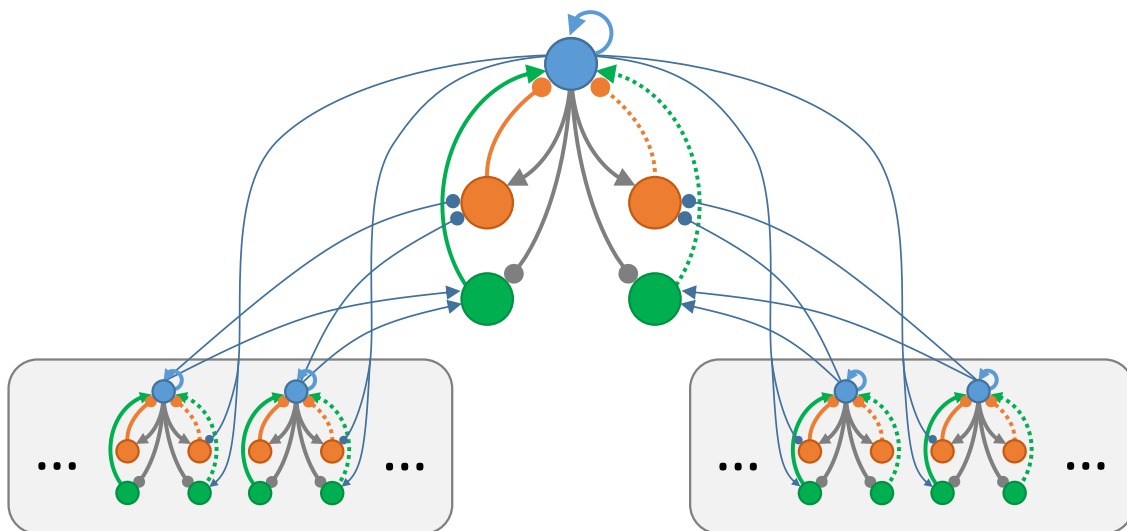


Fig. 2.2 Example of a complete mismatch detection neural circuit. The microcircuit in the center contains two pairs of "should" and "should-not" neurons to interact with two streams of information. The stream of information in the right is less reliable and thus, the mismatch neurons that receive that information drive the head less strongly (dotted lines). The information is coming from more microcircuits of the same type, and typically there will be reciprocal connections.

The hypothesis that is entertained in this thesis is that something like the proposed circuit could be the standard form of operation throughout much of the brain, in the same way that the predictive coding circuit is claimed to be by its proponents.

The cross-connections between microcircuits (depicted in blue) will be learned through a type of Hebbian rule. The basic idea is simple. The weight from a head neuron, H_i , to a "should" or "should-not" neuron of a different microcircuit, S_j , will be modified in proportion to the activation level of the "head" neuron and the negative value of the "should" or "should-not" neuron:

$$\Delta w_{H_i \rightarrow S_j} \propto H_i \cdot (-S_j) \quad (2.1)$$

The negative value is needed since we want the weights to take values equal but opposite to those of the fixed connections from the microcircuit's "head".

Let us walk through an example for weights getting learned onto a "should-not" neuron. Before learning, the "should-not" neuron only receives the excitatory input from its "head" neuron, thus, when the "head" neuron gets activated, the "should-not" neuron will do so too, signaling mismatch by excess (no currently active pattern justifies the presence of the feature). If there are other head neurons active at the same time, according to the learning rule, negative (inhibitory) connections will grow until they exactly cancel out the excitatory input and the "should-not" neuron reaches a value of zero. If, for some reason, the weights would grow too negative, the "should-not" neuron would go negative and the weights would increase until a balance is reached again. Thus, the learning rule is self-normalizing.

One last element needs to be introduced. Within the same stream of information, there might be different input combinations that need to be associated to the same concept. For example, both "a" and "A" need to trigger the same high-level representation of the letter through its s-pair corresponding to the visual domain. However, as the circuit has been presented, there is only one "should" and "should-not" neuron per s-pair. Thus, the attempt to register these two patterns would produce a linear superposition of them that could respond incorrectly to stimuli that are neither "a" nor "A" but a combination of both. The obvious solution would be to use different neurons to register each pattern, but this runs into the problem that then, most of the time, most "should-not" neurons would inconveniently be active. In the example, when presented with "a", the "should-not" neuron that recognizes "A" would activate and vice versa. We might postulate that if one "should-not" neuron is silent (meaning that the concept is being justified by one pattern), the rest should also remain silent. However, a more elegant solution comes from introducing dendritic nonlinearities and having each dendrite respond to a different pattern. This is illustrated in Fig. 2.3.

The introduction of dendritic nonlinearities is inspired by findings in biology. Polsky et al. (2004) found that, in thin dendrites of rat neocortical pyramidal neurons, nearby inputs

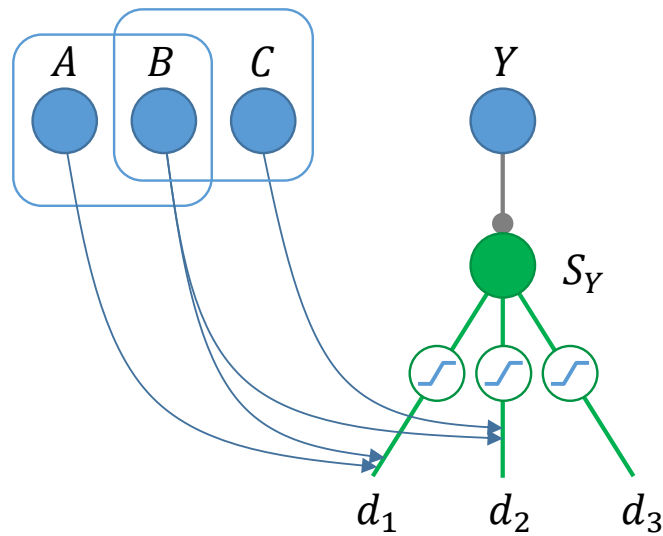


Fig. 2.3 Dendritic nonlinearities are used to increase the discriminatory capacity of "should" and "should-not" neurons. Here, "should" neuron S_Y responds to patterns (A, B) and (B, C) but not to (A, C) , since this combination, which is not one of the learned patterns, only causes subthreshold activation in the dendrites.

on the same branch summed sigmoidally, whereas widely separated inputs or inputs to different branches summed linearly. This provided the first experimental support for a two-layer neural network model of the pyramidal neuron. The active properties of neocortical pyramidal dendrites are reviewed in Major et al. (2013). The principal dendritic nonlinearity is supported by NMDA spikes, which have been found in all classes of thin dendrites of neocortical excitatory neurons and in all neocortical areas and layers examined, and probably also occur in hippocampal apical tufts.

In addition to the electrophysiological findings, Makino and Malinow (2011) report that sensory experience in mice preferentially produced clustered synaptic potentiation onto nearby dendritic synapses, supporting the hypothesis that dendrites are used by the brain for independently binding different input combinations.

Several models have been proposed that make use of these dendritic nonlinearities (Mel (1991), Wu and Mel (2009), Legenstein and Maass (2011), Urbanczik and Senn (2015)). Strictly speaking, dendritic nonlinearities seem to operate on a sliding window of integration across dendritic branches but for simplicity, we will assume that dendrites are the fundamental unit of integration.

A further advantage of introducing dendritic nonlinearities in our model concerns learning. According to the learning rule, when inputs arrive to the "should" and "should-not" neurons of a microcircuit in the absence of activity in its head neuron, the active connections will

get weakened. This makes sense, since the circuit has to be able to learn that certain stimuli no longer predict others. However, without dendritic nonlinearities, it would run into the following problem. Different concepts will typically respond to combinations of features that overlap partially, thus, when presented with the features corresponding to one concept, the "should" and "should-not" neurons of other concepts will also receive partial stimulation. In the absence of activity in their "head" neurons, this partial stimulation would weaken the connections that cause it. For instance, every time a yellow lemon is seen, the connection from yellow to banana would be reduced. I explored a solution involving sliding thresholds for learning, but dendritic non-linearities provide a much more elegant solution: partial combinations of features below some threshold simply do not cause activation in the "should" or "should-not" neurons and do not trigger learning.

The introduction of dendrites, however, makes the learning rule more complicated since now some additional mechanisms need to be introduced to ensure the automatic clustering of inputs onto different dendrites. This will be discussed in the next section.

Thus far, "should" and "should-not" neurons signal mismatches between the world and the currently active model of it. However, in an s-pair that does not drive its head neuron, or does it very weakly, the "should" and "should-not" neurons could also be used for the production of behavior. In this case, these neurons would signal the mismatch between some motivation and the current state of affairs. They could then be used to generate actions until the desired state is achieved, thus acting as a negative feedback control system. Furthermore, they could be linked in a hierarchy, where the motivational "should" or "should-not" signals of some higher-level state propagate to those of a lower-level state that is required for the activation of the first. For example, the desire to obtain hot water (its motivational "should" neuron) could propagate to the desire of having the tap turned to the left, whereas the desire to reduce the temperature (the motivational "should-not" neuron of the "hot-water" circuit) could be linked to turning the tap in the opposite direction. This type of arrangement is shown in Fig. 2.4 for the activity of a motivational "should" neuron being propagated through two levels until it is finally used to generate action.

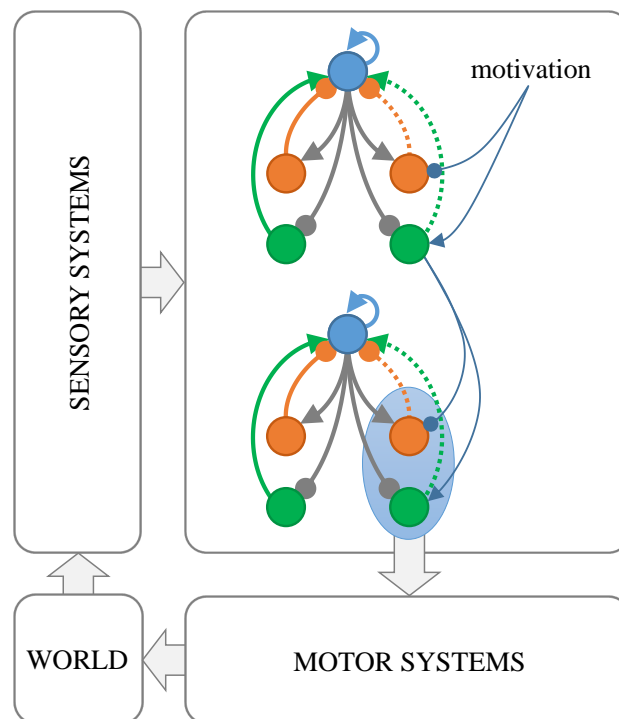


Fig. 2.4 The mismatch detection circuit used for the production of behavior. The motivation for a higher-level state propagates to that of a lower-level state that controls the motor systems, closing a negative feedback control loop through the world and sensory systems.

2.1.2 Detailed Description

This section contains a detailed description of the circuit including the equations that govern its behavior. The equations are for a single microcircuit composed of one "head" neuron and one "should" and "should-not" neurons, but the modifications needed to include several linked microcircuits or multiple s-pairs are straightforward. The system is modeled in discrete time.

The activation level of the "head" neuron, H , is defined by

$$\begin{aligned}\Delta H &= \frac{1}{\tau_H} (-H + H_{out} + w_S S - w_N N + H_{ext}) \\ H_{out} &= f(H)\end{aligned}\tag{2.2}$$

where τ_H is the time constant; w_S and w_N are fixed weights from the "should" and "should-not" neuron respectively, both positive; and H_{ext} is an unmodeled external input. This external input will be used to activate the neurons for the first time until connections are learned and the neuron is driven by the s-pair. The activation function of the neuron, f , is the piecewise linear function plotted in Fig. 2.5.

$$f(x) = \begin{cases} 0 & \text{if } x \leq \theta_{H,l} \\ \frac{x}{\theta_{H,h} - \theta_{H,l}} - \frac{\theta_{H,l}}{\theta_{H,h} - \theta_{H,l}} & \text{if } \theta_{H,l} < x \leq \theta_{H,h} \\ 1 & \text{if } x > \theta_{H,h} \end{cases}\tag{2.3}$$

Due to the self-excitation of the "head" neuron and the form of its activation function, the neuron shows a bistable behavior with equilibrium points in 0 and 1. The attraction towards the high equilibrium point after a threshold has been exceeded can be seen as a restorative force that ensures that the signal does not disappear after several stages of processing.

The activation of the "should" neuron, S , is defined by the following equations.

$$\begin{aligned}\Delta S &= \frac{1}{\tau_S} (-S - H_{out} + \sum_{i=0}^n g_S(d_i) + S_{ext}) \\ S_{noisy} &= -H_{out} \cdot U(1 - a, 1 + a) + \sum_{i=0}^n g_S(d_i) \\ S_{out} &= \max(0, \min(1, S))\end{aligned}\tag{2.4}$$

where τ_S is the time constant, which should be smaller than τ_H to avoid oscillations; H_{out} is the output of the head neuron; $\sum_{i=0}^n g_S(d_i)$ is the sum of the activation values of the neuron's n dendrites, d , after the dendritic nonlinearity, g_S ; and S_{ext} is an external input that will be used in some occasions to simulate unmodeled inputs. S_{noisy} is an instantaneous and noisy version

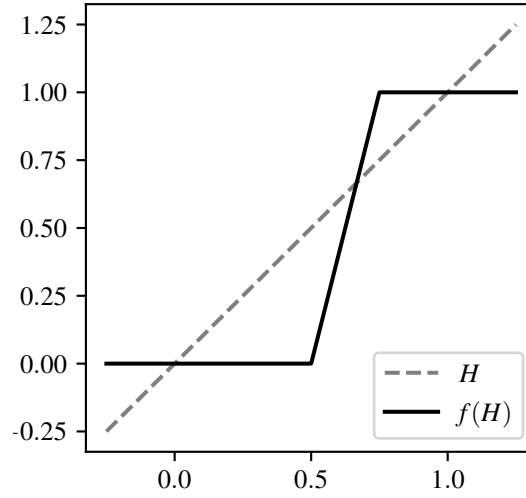


Fig. 2.5 Activation function of the "head" neuron with $\theta_{H,l} = 0.5$ and $\theta_{H,h} = 0.75$, which are the parameters used in the simulations.

of S that will be backpropagated to the dendrites and act as a learning signal. $U(1 - a, 1 + a)$ is a uniform noise distribution that draws samples between $1 - a$ and $1 + a$. The purpose of this noise term in learning will be explained later. S_{noisy} has been introduced for convenience, since the noise is required for learning but it is preferable to have a clean signal in S .

The value of dendrite i is the dot product of an input vector, I , and a weight vector for that dendrite, w_i .

$$d_i = I \cdot w_i \quad (2.5)$$

The dendritic nonlinearity is a piecewise linear function which only saturates from below and has value of 1 for $x = 1$.

$$g_S(x) = \begin{cases} 0 & \text{if } x \leq \theta_d \\ \frac{x}{1-\theta_d} - \frac{\theta_d}{1-\theta_d} & \text{if } x > \theta_d \end{cases} \quad (2.6)$$

Similarly, the behavior of the "should-not", N , is defined by

$$\begin{aligned} \Delta N &= \frac{1}{\tau_S} (-N + H_{out} + \sum_{i=0}^n g_N(d_i) + N_{ext}) \\ N_{noisy} &= H_{out} \cdot U(1 - a, 1 + a) + \sum_{i=0}^n g_N(d_i) \\ N_{out} &= \max(0, \min(1, N)) \end{aligned} \quad (2.7)$$

where the only differences with respect to "should" neurons are in the sign of the input from the "head" neuron, and in the dendritic activation function, g_N , which is the antisymmetric function of g_S (a dendritic nonlinearity for the inhibitory inputs).

Now we will discuss the learning rule. The basic idea was already introduced in the previous section. The weight update will be proportional to the pre-synaptic neuron and the negative value of the post-synaptic "should" or "should-not" neuron. However, we need to solve two problems. First, some mechanism has to be introduced that ensures that different input combinations cluster onto different dendrites. We want the mechanism to be biologically plausible, so synapses will only have access to local information and will not know directly the value of other synapses. Second, we would like the learning rule to allow for the integration of new elements into a learned pattern. For example, the system may have learned that features A and B predict feature Y, and thus connections with a weight of 0.5 will have been established from the "head" neurons of A and B to the "should" neuron of Y. If now the pattern changes to A, B, C and Y, the system should learn this and modify the weights so that now A, B and C contribute 1/3. However, as the rule stands right now, this will not happen since the weight change is proportional to the "should" neuron, and this is already 0 after the first association is learned.

Let us first discuss this second problem. If a connection is to be established from C to the "should" neuron of Y, this neuron should have a value different from 0, at least some of the time. This is where the noise term in S_{noisy} comes in. Even if the connections from A and B add up to cancel on average the fixed connection from the microcircuit's head, there will be moments in time where its value will become negative, which is when new connections can grow. But this is not enough; new (and old) connections will grow by a certain small amount when the noise brings S below 0, but they will subsequently decrease by the same amount on average when the noise makes S go above 0. We need a mechanism that breaks this symmetry and allows new connections to grow on average. A solution could be to make smaller synapses increase faster and/or larger synapses decrease faster. Thus, when the noise makes the connections grow, new connections (which are initially smaller) will grow more than the old ones (which are larger); and when the noise makes them decrease, old connections will decrease by a larger amount, eventually leading to a homogenization of the weights. The problem with this approach is that the different rates for increasing and decreasing the weights, added to the uniform noise distribution will make the weights stabilize around a value that is not exactly the desired one and that does not make the value of S equal to 0 on average. What we would like is that smaller weights increase faster and larger weights decrease faster only when the weights have different value, but otherwise have equal increase and decrease rates so that they stabilize around the correct value. The way in which

we achieve this is by dividing the weight update process into two stages. First, weights get tagged with a "desired" weight update that is the same for all weights regardless of their size, and that has the same rate for increases and decreases. Second, a certain "available" weight update is distributed to the weights according to their "desired" update and a priority score. It is in this priority score that the asymmetry is introduced by making smaller weights have priority in capturing the available positive weight updates and larger weights have priority in capturing the available negative updates. This solves the problem.

Very conveniently, it also provides the basis for solving the dendritic clustering problem. In essence, we can make synapses at different dendrites compete for a limited amount of weight update available from the soma in such a way that weights in dendrites with larger tagged weights have priority in the capture of positive updates and/or weights in dendrites with smaller tagged weights have priority to capture the negative updates. This will lead to weights from a particular input pattern clustering onto a single dendrite. However, nothing would prevent another input combination from clustering onto the same dendrite, thus rendering the clusters useless. In order to avoid this, we limit the maximum amount of weight that a dendrite can hold to that corresponding to one cluster. If this quantity is exceeded, the weights compete with each other based on a measure of fitness. The fitness measure of weights that have already succeeded in clustering in a dendrite will be high, thus, new weights from different input combinations will not be able to compete for resources or space in this dendrite and will cluster in an unoccupied one. This mechanism is not biologically implausible. Gray et al. (2006) have found that PSD-95, a protein thought to determine the size and strength of synapses, is retained by individual spines only for short periods of time (with median retention times in the order of tens of minutes) and is constantly exchanged between neighboring synapses by diffusion. The authors suggest that spines are competing for a limited shared pool of PSD-95. Fonseca et al. (2004) also provide experimental evidence for competitive interactions between synapses in regimes of reduced protein synthesis.

The division between tagging and distribution of weight update will also make it easier to learn behaviors as was shown in Fig. 2.4. Learning a behavior consists of learning the association between the motivational s-pair of the desired state and that of a state or action that when activated brings about the desired state. In order to achieve this, we can allow tagging to happen constantly and low-pass filter it, thus, in the exploratory phase, tags will grow for all the actions performed, and then decay exponentially. Only when the desired state is fulfilled, the capture process is allowed. This way, only the actions that were performed right before the state came about will get learned, since they are the only ones with sufficiently large tags at the time. The low-pass filter of the tags could also be used to learn behaviors where there is a delay between the execution of the action and the fulfillment of the desire.

This general mechanism that we have just proposed, where weights first get tagged with a "desired" weight update and then compete for an "available" update was conceived of out of necessity, and is useful in solving three different problems. Interestingly, it bears a striking resemblance to what is known in neuroscience as the synaptic tagging and capture hypothesis (Redondo and Morris, 2011). According to this hypothesis, synapses are first tagged for potentiation or depression and then capture "plasticity products" that are necessary for producing enduring changes in synaptic efficacy. In this capture process, cooperative and competitive interactions can take place between synapses (Sajikumara et al., 2014).

In fact, Govindarajan et al. (2006) have already proposed a biological model for clustered plasticity that is partly based on the synaptic tagging and capture hypothesis and exploits a cooperative effect in the capture process.

The last element of the learning rule that needs to be introduced is a "flexibility" variable. This variable will be used for decreasing the learning rate as the neurons "ages" so that patterns can get learned very quickly at the beginning and then only modified slowly. This is very roughly modeled in the circuit and the reset functionality that would be needed if the neuron gets recruited for a new pattern is not implemented.

With this we can go back to describing the equations. The "age", A , just mentioned will simply increase every time step that the "head" neuron is active.

$$\Delta A = H_{out} \quad (2.8)$$

The "flexibility", Φ , will be an exponentially decaying function of the "age", plus a constant, Φ_∞ , so that learning never fully stops:

$$\Phi = k_\Phi \cdot \exp(-A/\tau_{aging}) + \Phi_\infty \quad (2.9)$$

where τ_{aging} is the aging time constant.

The rest of the equations will focus on the weights onto a "should" neuron, but the case for "should-not" neurons is equivalent, except for a few trivial sign changes.

The weight fitness, F , corresponding to the weight from input j onto dendrite i , is given by

$$\Delta F_{ij} = \eta_{tagging} \cdot \eta_{capture} \cdot \Phi \cdot g(d_i) \cdot I_j \quad (2.10)$$

where $\eta_{tagging}$ and $\eta_{capture}$ are the tagging and capture rates respectively, Φ is the flexibility, $g(d_i)$ is the value of the dendrite after the nonlinearity, and I_j is the input. The weight fitness will thus increase every time the input drives the activation of the dendrite above threshold.

Learning will be blocked for a certain number of time steps when the derivative of the "should" neuron exceeds a certain threshold. The reason for this will become clear with the help of simulations in the next chapter. This feature is implemented by means of a "block count" that gets reset to an initial value every time the derivative exceeds a threshold, θ_{block} , and otherwise is decremented by 1. Learning is enabled through a boolean variable, E that is 1 when the "block count" is not positive.

$$blockCount = \begin{cases} blockCount_0 & \text{if } S(t) - S(t-1) > \theta_{block} \\ blockCount - 1 & \text{otherwise} \end{cases} \quad (2.11)$$

$$E = \begin{cases} 1 & \text{if } blockCount \leq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.12)$$

The instantaneous value of the update tag ("desired" update) corresponding to the weight onto dendrite i , and input j is given by

$$T_{wij}^{inst} = E \cdot \eta_{tagging} \cdot \Phi \cdot (-B(S_{noisy}, d_i)) \cdot I_j \quad (2.13)$$

where B is a function that handles the backpropagation of the learning signal, S_{noisy} , to the dendrites. In order to understand the need for this function, suppose that a "should" neuron with two dendrites has learned input patterns (A,B) and (B,C,D), on each of the dendrites. After learning, when presented with a pattern, say, (A,B), due to the noise in S_{noisy} , the weights will go up and down slightly. The problem is that the weight from B onto the dendrite that recognizes (B,C,D) will also go up and down, but since there is less amount of weight tag on that dendrite (only that from B), the dendritic clustering rule will make that weight from B capture preferentially the negative updates, making it decrease on average. The solution is thus to block backpropagation of the learning signal to dendrites that are below threshold, when the value of the learning signal is due only to noise.

$$B(x, d) = \begin{cases} 0 & \text{if } d \leq \theta_d \text{ and } x > -\omega \\ x & \text{otherwise} \end{cases} \quad (2.14)$$

where $-\omega$ is the largest negative value of S_{noisy} that can be due to noise alone.

The instantaneous value of the tag is then low-pass filtered. However, we do not want to filter out the noise, since the noise was introduced with the purpose of allowing new elements to get integrated into learned patterns. Therefore, we divide the tag into two components, one to keep track and low-pass filter the positive weight updates, and the other to do the same

with the negative weight updates. The negative tag has to be clipped if it exceeds the weight, since otherwise the weight would try to become negative (and it is supposed to be excitatory). When the only source of input to the instantaneous tag is the noise, the "up" and "down" tags will have equal values and the equilibrium will be a dynamic one, where weights are tagged to increase and decrease by the same amount.

$$T_{w_{ij}}^{up} = \begin{cases} \alpha \cdot T_{w_{ij}}^{up} + (1 - \alpha) \cdot T_{w_{ij}}^{inst} & \text{if } T_{w_{ij}}^{inst} > 0 \\ \alpha \cdot T_{w_{ij}}^{up} & \text{otherwise} \end{cases}$$

$$T_{w_{ij}}^{down} = \begin{cases} \alpha \cdot T_{w_{ij}}^{down} + (1 - \alpha) \cdot (-T_{w_{ij}}^{inst}) & \text{if } T_{w_{ij}}^{inst} < 0 \text{ and } \alpha \cdot T_{w_{ij}}^{down} + (1 - \alpha) \cdot (-T_{w_{ij}}^{inst}) \leq w_{ij} \\ w_{ij} & \text{if } T_{w_{ij}}^{inst} < 0 \text{ and } \alpha \cdot T_{w_{ij}}^{down} + (1 - \alpha) \cdot (-T_{w_{ij}}^{inst}) > w_{ij} \\ \alpha \cdot T_{w_{ij}}^{down} & \text{otherwise} \end{cases} \quad (2.15)$$

Now we need to calculate the "available" weight updates. These will be produced by the soma independently, based on a "soma tag" that is equivalent to the weight tags except that it does not depend on the values of pre-synaptic neurons and is not modulated by the backpropagation function.

$$T_{soma}^{inst} = E \cdot \eta_{tagging} \cdot \Phi \cdot (-S_{noisy}) \quad (2.16)$$

$$T_{soma}^{up} = \begin{cases} \alpha \cdot T_{soma}^{up} + (1 - \alpha) \cdot T_{soma}^{inst} & \text{if } T_{soma}^{inst} > 0 \\ \alpha \cdot T_{soma}^{up} & \text{otherwise} \end{cases}$$

$$T_{soma}^{down} = \begin{cases} \alpha \cdot T_{soma}^{down} + (1 - \alpha) \cdot T_{soma}^{inst} & \text{if } T_{soma}^{inst} < 0 \\ \alpha \cdot T_{soma}^{down} & \text{otherwise} \end{cases} \quad (2.17)$$

The "available" increments and decrements of weights that can be captured at a time step will then simply be the multiplication of the "soma tags" by a parameter that defines the rate of the capture process.

$$\Delta w_{available} = \eta_{capture} \cdot T_{soma}^{up}$$

$$\nabla w_{available} = \eta_{capture} \cdot T_{soma}^{down} \quad (2.18)$$

Finally, we need the priority scores with which the "available" weight update is distributed. There will be a priority score for the dendrites, which will produce the dendritic clustering, and one for the weights, that will ensure the weight homogenization and integration of new elements. For the dendrites, the priority for capturing positive weight updates is given by

the sum of the weights onto that dendrite multiplied by their positive tags. The priority for capturing negative weight updates is the sum of the weights' negative tags divided by the weights' values. A small constant, ε is added to the weights so that the priorities do not go to 0 or infinity. In addition, the priorities are multiplied by a uniform noise distribution, $U_{slow}(1 - b, 1 + b)$, that varies slowly (with a time constant of about half of the time it takes to learn a pattern). In the simulation this is implemented by drawing a sample every T time steps. This noise is necessary to break the initial symmetry and allow one dendrite to win the competition.

$$\begin{aligned}
 P_{d_i}^{up} &= \sum_{j=0}^m (T_{w_{ij}}^{up} \cdot (w_{ij} + \varepsilon)) \cdot U(1 - b, 1 + b, T) \\
 P_{d_i}^{down} &= \sum_{j=0}^m \left(\frac{T_{w_{ij}}^{down}}{(w_{ij} + \varepsilon)} \right) \cdot U(1 - b, 1 + b, T)
 \end{aligned} \tag{2.19}$$

The priority for a weight to capture positive updates is given by a negative exponential of its weight, whereas the priority to capture negative updates is a positive exponential of the weight.

$$\begin{aligned}
 P_{w_{ij}}^{up} &= \exp(-k_P \cdot w_{ij}) \\
 P_{w_{ij}}^{down} &= \exp(k_P \cdot w_{ij})
 \end{aligned} \tag{2.20}$$

Now the "available" weight update has to be distributed according to the "desired" weight updates given by the weight tags and the priority scores. This process is not trivial. We could imagine that it would be enough to multiply the dendrite and weight priorities to get a global priority of each weight and then simply distribute the weight update proportional to the priorities. This has two problems. First, the dendrite priorities give advantage to larger weights, but the weights priorities do the opposite, so multiplying the two priority scores counteracts their effect. The second problem is that distributing the updates in proportion to the priorities may assign to a weight more weight update than what it is asking for, effectively "wasting" this excess of weight update that could have been used by another weight. This is not just a problem of economy; it actually introduces an error in the average value of the weights. Thus, the effect of the two priority scores has to be kept independent, and weights should not be assigned more weight update than what they "desire", but all the weight update should be distributed if possible.

It seems plausible that this mechanism could be achieved by having the weight update diffuse in and out of the dendrites with a probability based on the dendrite priority, while the capture process within the dendrites happens at a faster rate, effectively dissociating the two effects. Alternatively, the weight update could be shipped to the dendrites based on

their priorities and total "desired" update in that dendrite, and then be distributed within the dendrites based on the weight priorities and their "desired" updates. We follow this second approach and solve it algorithmically. We use the same procedure for distributing the weight updates to the dendrites first and to the weights later. The algorithm is implemented by the following Python code:

```
import numpy as np
def distribute(available , desired , priorities ):
    assigned = np.zeros(desired.shape)
    while(available > 1e-5 and (desired > 1e-5).any()):
        priorities = np.where(desired < 1e-5, 0, priorities)
        canAssign = (available * priorities / np.sum(priorities))
        assigns = np.minimum(canAssign , desired)
        assigned += assigns
        desired -= assigns
        available -= np.sum(assigns)
    return assigned
```

Finally, if the weights onto a dendrite exceed the amount corresponding to one cluster (plus a small margin), $1 + \varepsilon$, a competitive mechanism kicks in. The amount by which the limit has been exceeded, $\sum_j w_{ij} - (1 + \varepsilon)$, will be distributed with negative sign among the weights. The proportion that will be assigned to each weight is calculated as follows. For each (non-zero) weight, we calculate a score, λ , by dividing the maximum fitness in the dendrite by each weight's fitness.

$$\lambda_{ij} = \frac{\max(F_i)}{F_{ij}} \quad (2.21)$$

and then distribute subtractively the excess of weight proportional to this:

$$\Delta w_{ij} = - \left(\sum_j w_{ij} - (1 + \varepsilon) \right) \cdot \frac{\lambda_{ij}}{\sum_j \lambda_{ij}} \quad (2.22)$$

This finalizes the description of the circuit. Simulation results are shown in Chapter 3.

2.2 Spatial Cognition and Navigation System

In this section a navigation system based on the mismatch detection circuit is presented.

First, we need to define the higher-level architecture of the system. How will space be represented? We believe that the arguments and experiments presented in the literature review are sufficiently strong to rule out a Cartesian map. We will thus represent space as a

series of landmarks and the relative distances and directions between them. Two possibilities are considered, which are shown in Fig. 2.6.

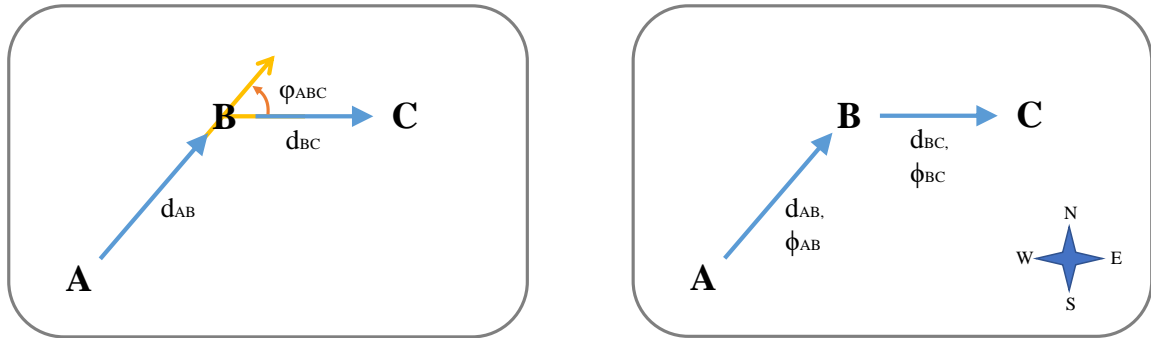


Fig. 2.6 Two possibilities for representing space, explained below.

In both schemes, the information that links one landmark to the next includes the distance between them. The difference between the two proposals is in how the directions are handled. In scheme depicted at the left, the direction which has to be pursued in order to get from B to C is defined relative to the previous direction, A to B. On the right, all directions are specified with respect to a local reference frame.

The first proposal agrees with behavioral experiments in humans that expose a lack of global coherence in our cognitive maps. It is also similar to the vector-based navigation system of insects. Furthermore, the representation of directions in purely relative terms relaxes the demands on the path integration system and eliminates the need for loop closure mechanisms. A final advantage is that the representations learned are very easy to use to generate behavior; the relative angles directly tell you how much you have to turn to go from one place to the next.

However, it has two important disadvantages. First, it is at odds with the well established finding of head direction cells in rodents that signal direction with respect to some local but coherent reference frame. Second, it does not generalize well. For example, if we now learn a new path to B, from, say, D, after traveling this path, the system will not know how to go to C, since it can not apply the relative angle ϕ_{ABC} , instead, it will have to learn the new relative angle ϕ_{DBC} .

The second model does not have these two problems since it includes the local heading direction, which allows new paths to be followed. For example, C is to the (relative) east of B, so as long as you are oriented within this reference frame, you will know how to go from B to C regardless of where you are coming from. Therefore, this is the representation of space that we will use. However, it should be noted that the two representations are perhaps

complementary, such that places that are relatively open and/or well explored develop a local reference frame, whereas those that are less well known or more restrictive in the movements that they allow, rely on pure relative angles.

We now propose an implementation of this model composed almost exclusively of the mismatch detection circuit. The system will be capable of learning an environment composed of series of landmarks identified by their odor. Mismatch signals will be useful both for detecting changes in the environment and for driving behavior (going to landmarks). The choice of odors as landmarks is motivated by the fact that odors are local stimuli, which simplifies the system. The model is shown in Fig. 2.7. It is composed of three groups of microcircuits, where the microcircuits in one group code for different directions and distances, in another group code for different places and in the last, code for different odors.

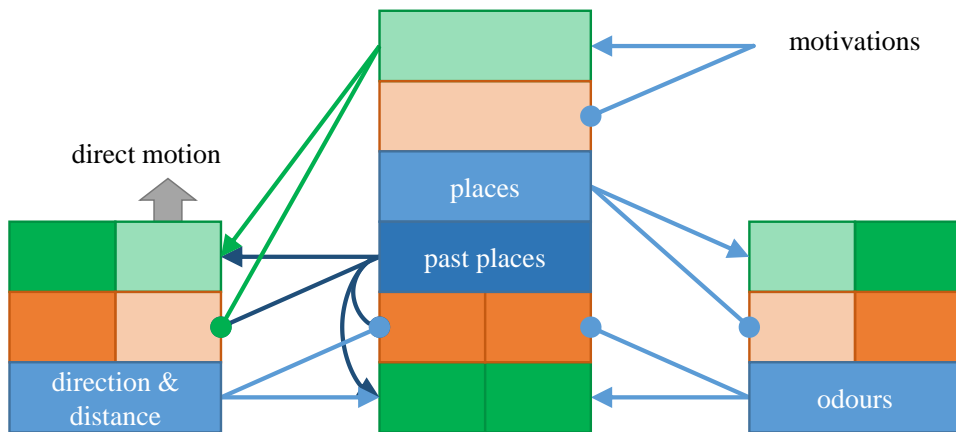


Fig. 2.7 Navigation system.

The color coding is consistent with the figures shown previously: blue rectangles represent the population of "head" neurons, orange rectangles represent "should-not" neurons and green ones, "should" neurons. In the groups that code for directions & distances, and for odors, each microcircuit contains two s-pairs. This is represented by the two pairs of green and orange rectangles. In the group that codes for places, each microcircuit has three s-pairs. The rectangles shown in light green and orange represent s-pairs that do not drive the head neuron whereas those in dark green and orange do. The rectangle in dark blue is a new element; it can be regarded as a working memory population of neurons that will code for the place that was last visited. This is implemented in a straightforward way: the "past places" neurons copy the activation values of the "place" neurons at certain moments in time and then maintain these values. How this maintenance actually occurs is not modeled explicitly, but it could happen through recurrent excitatory connections or by some internal mechanism

(Egorov et al. (2002), for example, report neurons in the entorhinal cortex that could sustain different levels of firing frequency in the absence of input).

The "head" neurons of direction & distance and odors will be driven by s-pairs that receive simulated odometric and sensory input respectively. The "head" neuron of places will be driven simultaneously by s-pairs receiving metric and odor information. The s-pair that receives metric information from directions and distances also receives information from the "past-places" such that it can learn associations of the type: B-past + 5m + East \rightarrow C. The non-driving s-pair of the odors receives information from places and will thus signal mismatches when a certain odor associated to the current place is missing or when an odor is present that is not justified by the place. Finally, a non-driving s-pair of places receives motivational input and, together with the "past-places", projects to the non-driving s-pair of directions & distances. This way, associations of the following form can be learned: B-past + C-should \rightarrow (5m, East), which can be used to travel to the place. Specifically, if the current distance and direction is not the one asked for or supported by where you are coming from and where you want to go, mismatch signals will appear (e.g. East-should, North-should-not) that can be used to steer the agent towards its goal.

The last issue that needs to be discussed is how the places will get activated for the first time, so that connections from the distances, directions and odors can be learned that will drive the activation of the place in the future. For this, we have introduced a system (very rough for now) that activates a place's "head" neuron when no other one is active and there is an unexplained odor. The neuron that will get recruited to represent this new place will be the one that has shown the least amount of activity, that is, probably, the least useful neuron for the system. If, however, there is a place neuron that is partly activated (above some threshold) when the unexplained odor occurs, the system will fully activate this place neuron. This will happen when the odor of a place changes. The place neuron will get partly activated because of the conflicting positive input from the metric s-pair (that tells it that it has arrived at the place) and negative input from the odor s-pair (that tells it that the place does not have the correct odor). The system fully activating this neuron means that it is judging that the place it represents has been reached despite the mismatch in the odor and is thus committing to having that neuron represent the place. This mechanism bears some resemblance to the "Adaptive Resonance Theory (ART)" proposed by Carpenter and Grossberg (1988). In ART, input patterns activate hypothesis in a winner-take-all fashion. If the match between the input pattern and the pattern predicted by the hypothesis is good enough, learning happens, allowing for small corrections in the pattern. If, on the contrary, no hypothesis matches the current input, a previously uncommitted node is selected to represent the new input pattern.

Simulation results for the navigation system are shown in the next chapter.

Chapter 3

Results

3.1 Mismatch Detection Neural Circuit

3.1.1 Basic operation

The circuit's operation will be first demonstrated with the help of an example where two different sources of information come into conflict. The example network is illustrated in Fig. 3.1. A microcircuit that represents the concept of salt contains two s-pairs for receiving low-level input from taste and visual features respectively. The s-pair that receives taste information drives the "head" neuron more strongly, accounting for the fact that taste is a more reliable indicator of the presence of salt. In turn, the "head" neuron of salt sends projections back to s-pairs of the taste and visual features in order to predict or justify their presence. The "head" neuron of the taste and visual features is however only driven by simulated sensory input arriving at a different s-pair.

A simulation of this network is shown in Fig. 3.2. The network contains microcircuits for representing three taste features, three visual features and the concept of salt. The activation levels of the neurons are represented in a heat map, where the activation of the "should" and "should-not" neurons is superimposed to that of the "head" neurons.

First, note how the activation of the taste and visual features is driven by the "should" and "should-not" neurons in their first s-pair, which receives simulated sensory input (not shown). When the input corresponding to a feature first arrives, it indicates that the feature "should" be active but is not, thus eliciting a response in its "should" neuron. Subsequently, the "should" neuron triggers the activation of its "head" and the mismatch disappears. When the input is removed, the "should-not" neuron gets activated, since the "head" is active but "should-not" be. This signal in turn suppresses the activation of the "head" and the mismatch

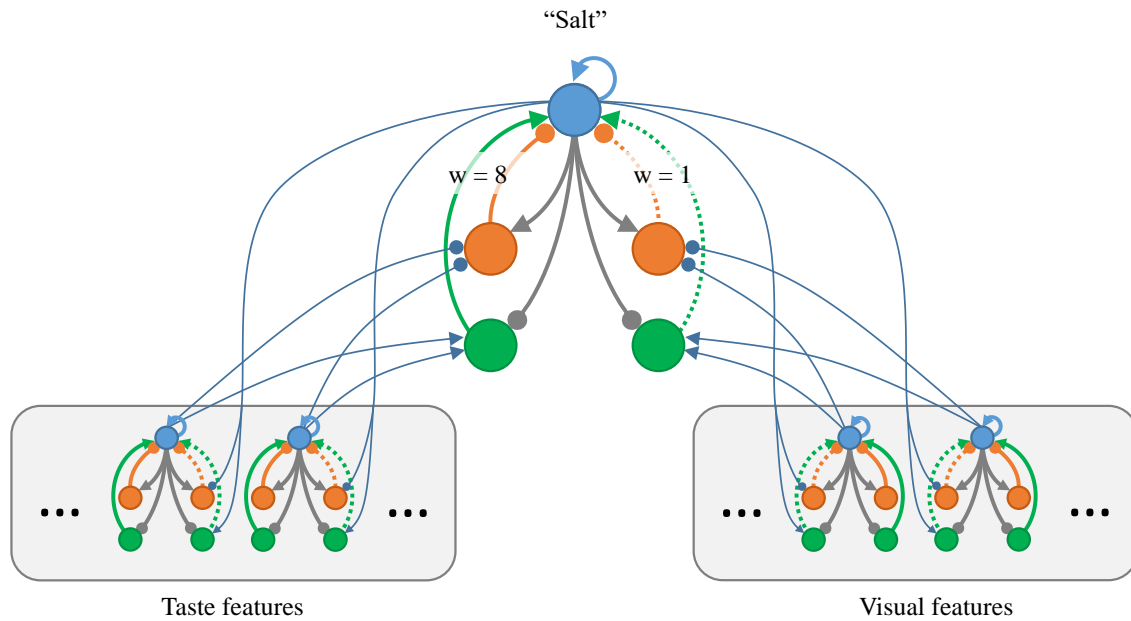


Fig. 3.1 Example network illustrating a microcircuit driven by two sources of information.

disappears again. The activation of the s-pair is thus signaling the onset and offset of the features and could be used to attract attention to them.

The simulation is divided into several stages. In (A), the network learns for the first time that salt is accompanied by taste feature 1 and visual features 2 and 3. We can see the activation of the "should-not" neurons until the associations are learned that justify the presence of those elements. Learning is switched off for the rest of the simulation. In (B), the network sees something that looks like salt and thus activates the concept. The s-pair of salt that receives taste features is silenced because nothing is being tasted yet. In (C), the substance is tasted but the taste feature found is not the one corresponding to salt, thus it switches-off the concept. The odd taste feature and the visual features are unjustified, with their "should-not" neurons switched on. The visual features keep indicating that the substance "should" be salt. In (D) the process happens in reverse; the network tastes something in the absence of visual stimuli and identifies it as salt. In (E), it sees it and finds the wrong visual features, however, since taste drives the "head" neuron of salt more strongly, it remains mostly active. The missing visual features of salt "should" be active; the odd visual feature "should-not", and according to the visual input, the substance "should-not" be salt.

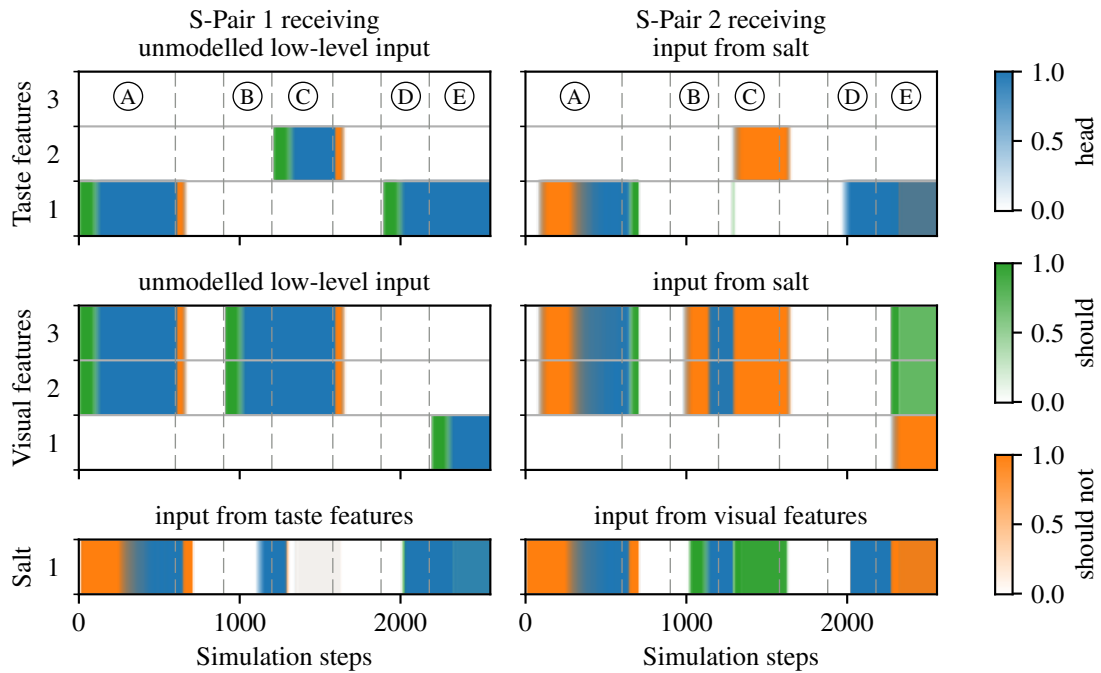


Fig. 3.2 Simulation of the network shown in Fig. 3.1. See text for explanations.

3.1.2 Dendritic Clustering and Weight Homogenization

The learning rule ensures that different input combinations will cluster onto separate dendrites. The simulations will be based on the circuit depicted in Fig. 3.3.

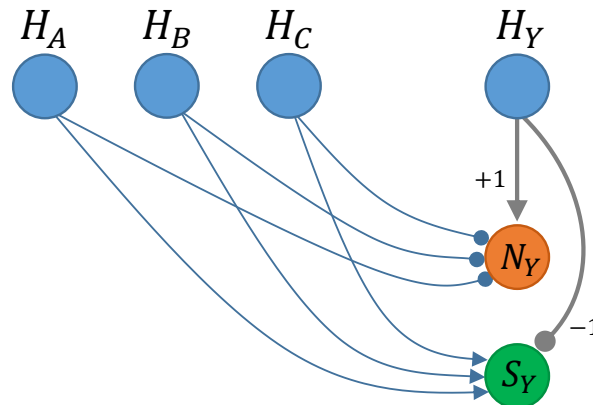


Fig. 3.3 Basic mismatch detection circuit.

The network will be presented with patterns (A,B,Y) and (B,C,Y). Only circuit Y is learning the associations, and the simulation shows the weights onto two dendrites of its

"should" neuron (the case for the "should-not" neuron is equivalent). The "should" neuron of Y, S_Y , thus has to establish connections from A and B on one dendrite and B and C on the other. The simulation is shown in Fig. 3.5.

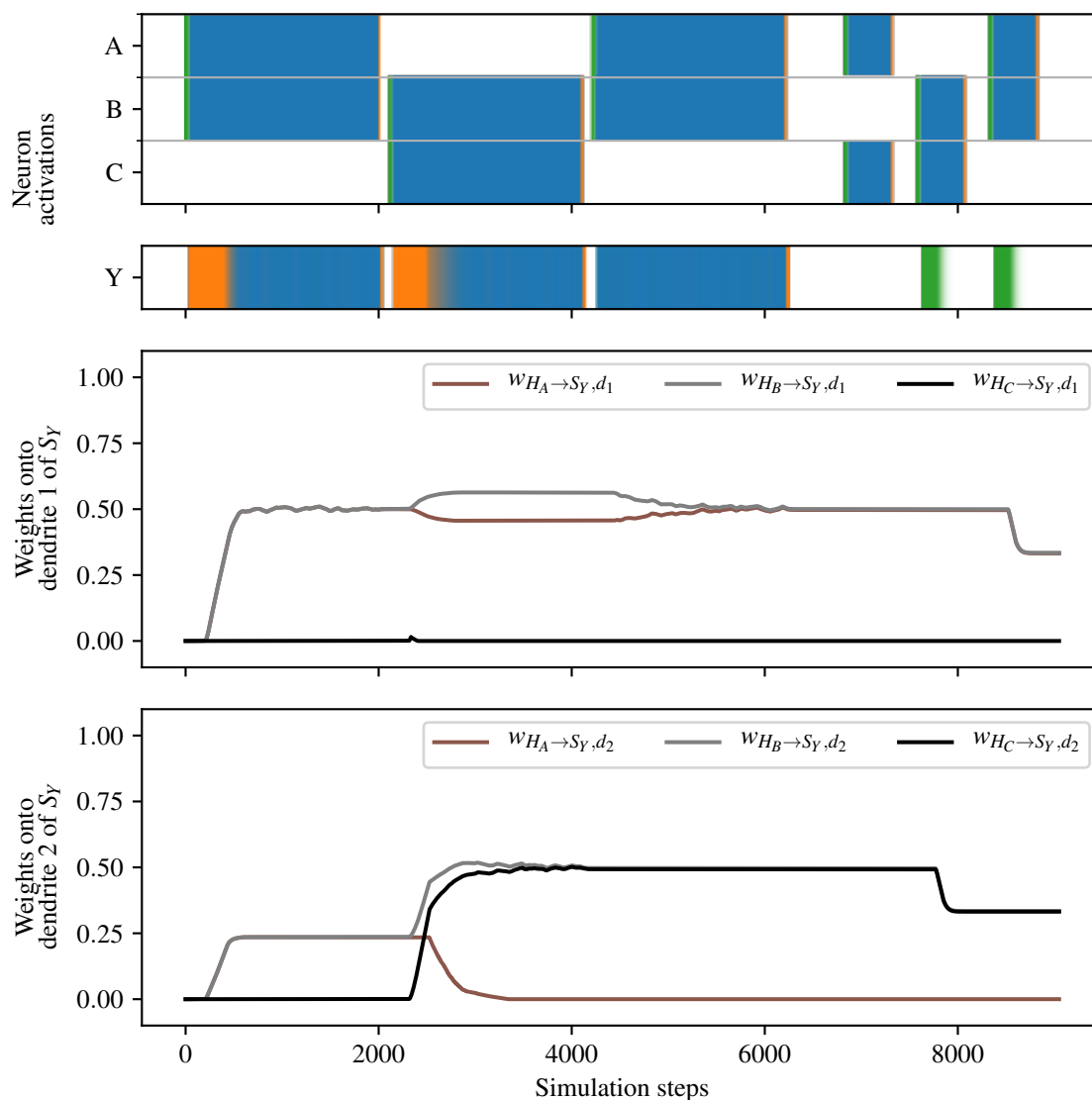


Fig. 3.4 Dendritic clustering when patterns (A,B,Y) and (B,C,Y) are presented for sufficiently long intervals. Weights from A and B cluster onto the first dendrite by chance, and weights from B and C are subsequently forced into the second one.

Weights from A and B cluster by chance on the first dendrite and their measure of fitness increases as they successfully activate the dendrite above threshold. When pattern (B,C,Y) is presented next, weights from B and C must grow. The first dendrite is already full, so weights have to grow at the expense of others, competing based on their fitness. The weight

from B, which has the same fitness as A, manages to grow slightly, but the weight from C, with no fitness, can not compete. The second dendrite is partly full, but the weights there do not have any fitness since they did not manage to drive the dendrite above threshold. Thus, weights from B and C can easily grow there, expelling the residual weight from A.

At the end of the simulation, it is shown how input combinations (A,B) and (B,C) cause the activation of the "should" neuron, but not the combination (A,C) which is not one of the learned patterns and causes only subthreshold activation in each dendrite. The dendrites thus successfully separate the two patterns. Since learning is still enabled and with a high learning rate, the presentation of (A,B) and (B,C) without Y causes the network to unlearn the association. Note, however, that the weights only decrease until they no longer cause above-threshold activation in the dendrites. As explained earlier, this is necessary so that the presentation of a pattern does not cause the depression of weights corresponding to other partially overlapping patterns. This can also be seen in the simulation: when (B,C) is presented, the weight from B onto the dendrite that recognizes (A,B) is not affected because B is causing below-threshold activation on the dendrite. This feature also allows learned patterns to become ineffective in a way that is easy to revert, specially if the dendritic threshold is high and thus making patterns effective or ineffective amounts to small changes in the weights.

In the next simulation (Fig. 3.4), the same procedure is shown, but the patterns are alternating at a higher frequency. Since weights have less time to increase their fitness, they are not as effective in keeping other weights from growing on the same dendrites and the initial stage of the clustering is more competitive.

Finally, we use the same configuration to illustrate the mechanism that allows new elements to be integrated into learned patterns. For this, we present pattern (A,Y), followed by (A,B,Y) and (A,B,C,Y). The simulation can be seen in Fig. 3.6.

The weights have clustered by chance onto the second dendrite. When the patterns change to incorporate new elements, the weights are modified slowly to reflect this. The time scale for the integration of new elements is slower than for learning the associations for the first time, since it depends entirely on noise. This is in accordance with the behavior observed in animal conditioning experiments.

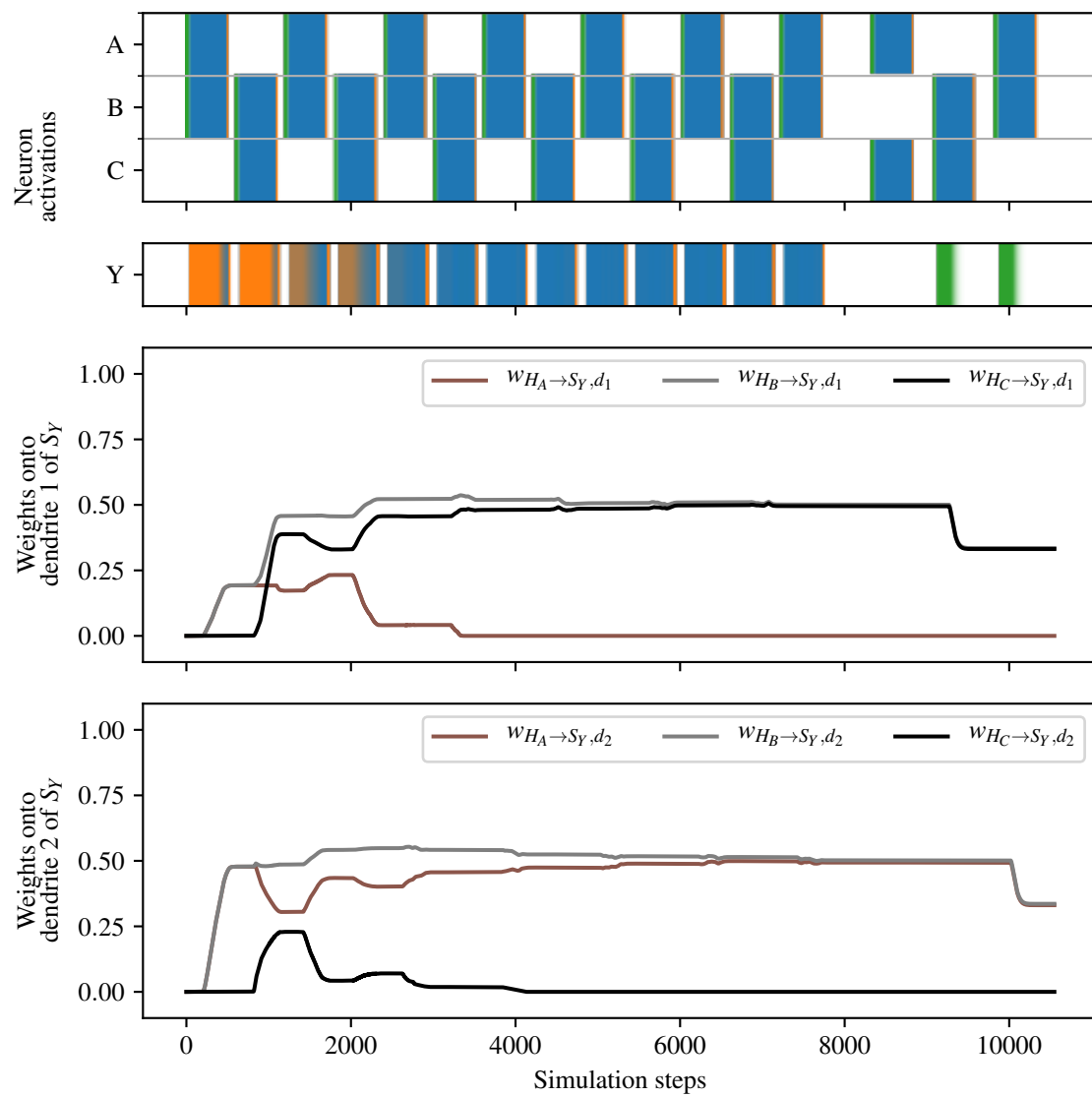


Fig. 3.5 Dendritic clustering when the patterns alternate at a higher frequency.

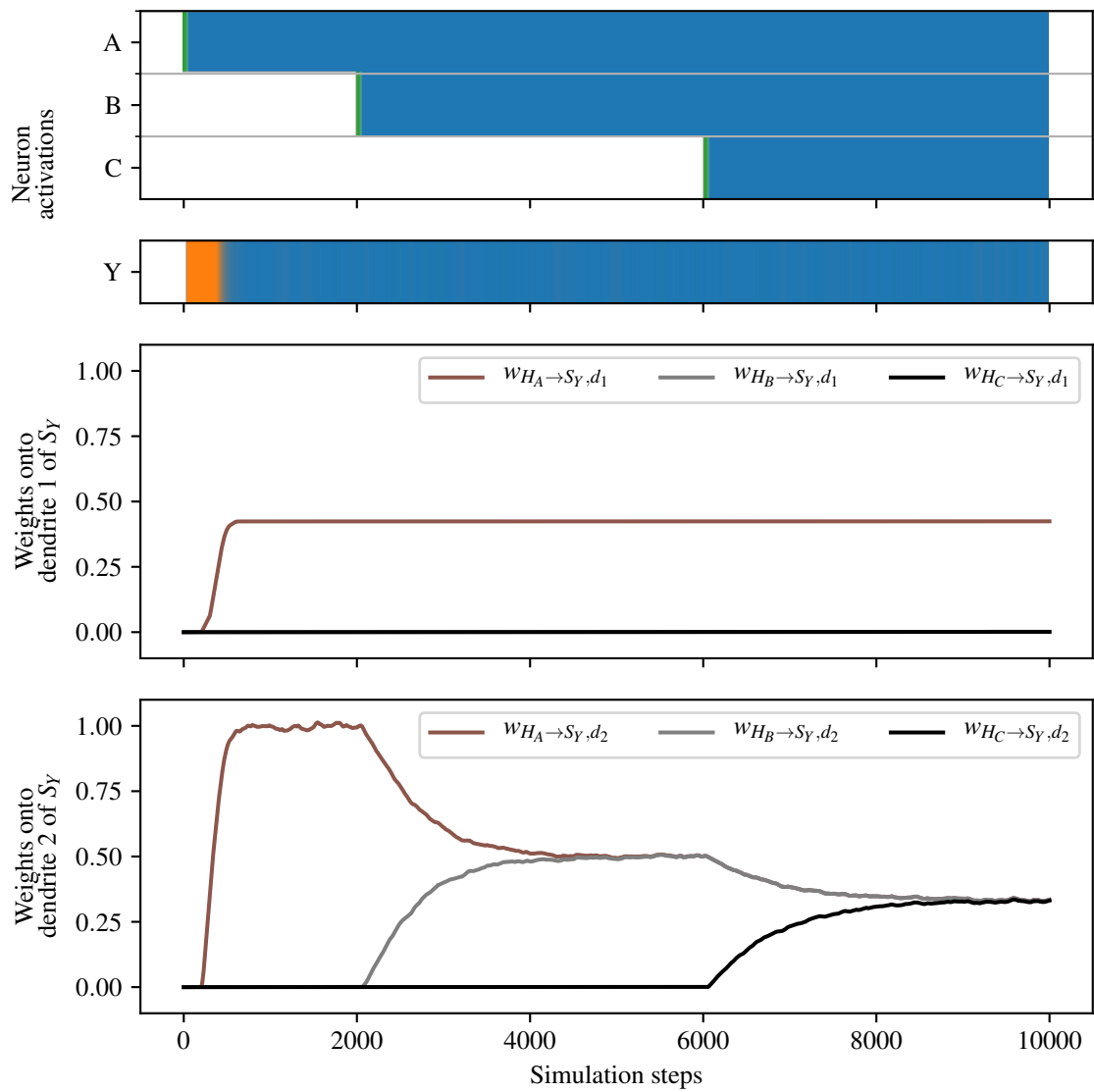


Fig. 3.6 Weight homogenization.

3.1.3 Capturing the Statistics of Associations

The learning rule translates the statistics of the associations into the strength of the weights that encode them. We show this in an example where two circuits become associated. Circuit 1 will be active some percentage of the time, whereas Circuit 2 (not shown) is always active. The strength of the connection from Circuit 2 to Circuit 1 will depend on the percentage of the time that Circuit 1 is active, which is the percentage of the time that Circuit 2 successfully predicts its presence.

In the first simulation (Fig. 3.7), the Circuit 1 is active half of the time. Thus, the weight from the second circuit to the first one will converge to a value such that its "should" and "should-not" neurons have a value of 1/2 when the circuit is inactive or active respectively. Since the dendritic threshold in the simulation is 0.66, the weight needed for causing an activation of 1/2 in "should" and "should-not" neurons is 0.83 (halfway through 0.66 and 1).

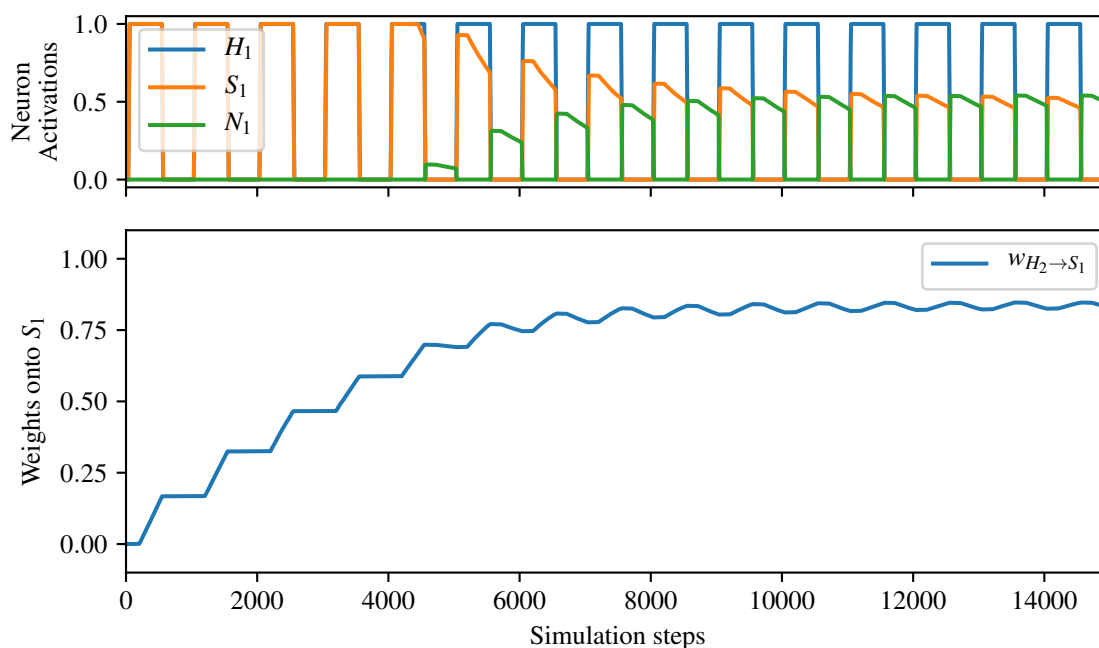


Fig. 3.7 The learning rule captures the statistics of the associations. Circuit 2 (not shown) is active always and successfully predicts the presence of Circuit 1 half of the time, thus, "should" and "should-not" neurons activate with an average value of 1/2.

In the second simulation (Fig. 3.8), Circuit 1 is active one third of the time. Thus, when it is not active, its "should" neuron will have a value of 1/3, and when it is active, its "should-not" neuron will have a value of 2/3.

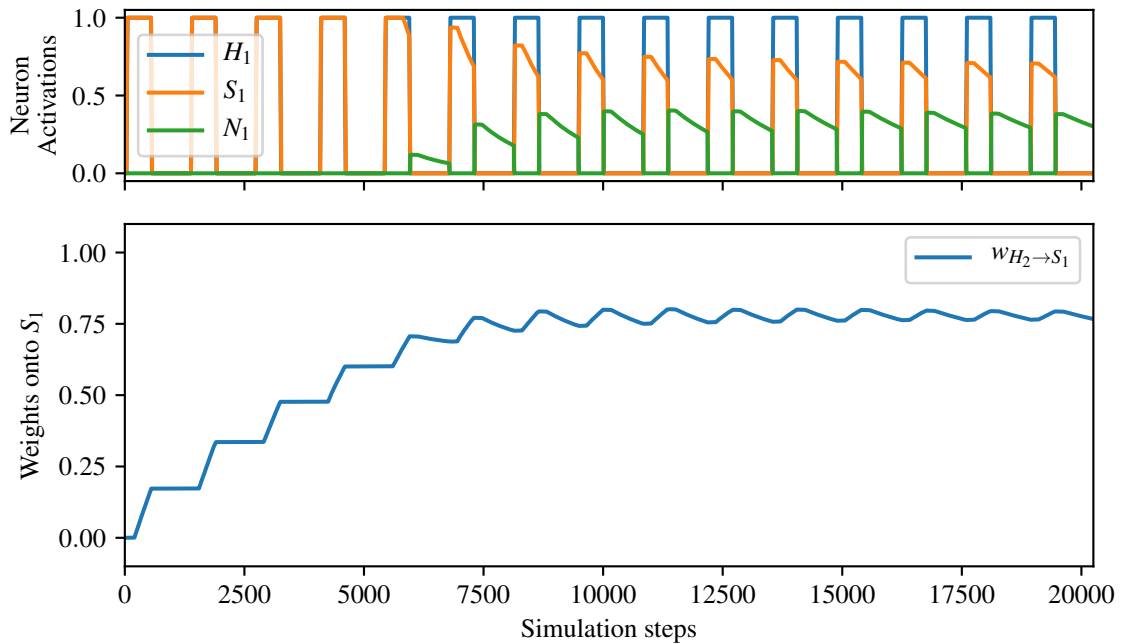


Fig. 3.8 Circuit 2 (not shown) is active always and successfully predicts the presence of Circuit 1 one third of the time, thus, Circuit 1 "should" neuron will activate with a value of $1/3$ and its "should-not" neuron with a value of $2/3$.

3.1.4 Blocking Learning during Transients

In this section, we illustrate the need for blocking learning during transients. The simulations are performed for the simple network shown in Fig. 3.9. This is the standard case where a lower-level microcircuit (Circuit 1) drives the activity of a higher-level one (Circuit 2). We only show the activities for the central s-pairs, that are the relevant ones for this example, and the weights onto the "should" neurons of these s-pairs, although the case for the "should-not" is symmetrical.

At the beginning of the simulation shown in Fig. 3.10, the association between Circuit 1 and 2 is learned. Then, Circuit 1 is switched off, causing Circuit 2 to deactivate too. However, because the learning rate is very high relative to the neuronal dynamics, in the time that it takes for Circuit 2 to switch off, the connection onto Circuit 1 is partly unlearned (since Circuit 2 is erroneously predicting the presence of Circuit 1). Furthermore, when Circuit 1 is presented again, it causes the activation of the "should" neuron of Circuit 2, but it unlearns the association before Circuit 2 has had time to activate.

Even when the learning rate is reduced, as in Fig. 3.11, the weights also degrade. As Circuit 2 activates, the weights onto Circuit 1 increase, because larger weights are needed to

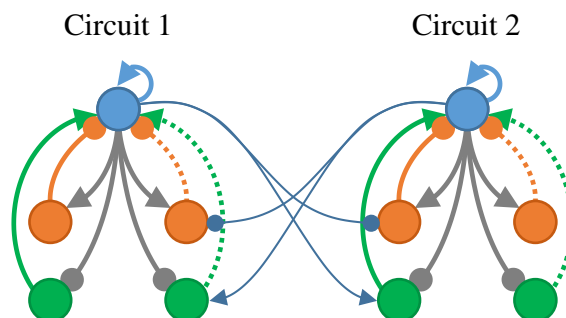


Fig. 3.9 Simple network where Circuit 1 strongly drives the activation of Circuit 2. Connections depicted in dotted lines are much weaker.

compensate for the smaller value in the activation of the "head" neuron in Circuit 2. This weight saturates in the upper bound of weight allowed for the dendrite. At the same time, the weight onto Circuit 2 diminishes as before, although now the "head" neuron manages to activate. However, if the presentation time is not enough to relearn the association that has been lost, the weight will continue to degrade on repeated presentations.

Thus, weights degrade during transients. In order to avoid it, we will simply block learning for some time when a transient is detected. The way in which a "should" or "should-not" neuron can identify these situations is by looking at the value of its own derivative. Onsets and offsets of the "head" neuron will cause high derivatives in the activations of "should" and "should-not" neurons. When the derivative exceeds the threshold that separates these transients from other slower changes, learning is disabled for some time. A simulation of this mechanism, showing the derivative of a "should" neuron and the enable signal for learning can be seen in Fig. 3.12. Weights are protected from the transients and the circuit operates as expected.

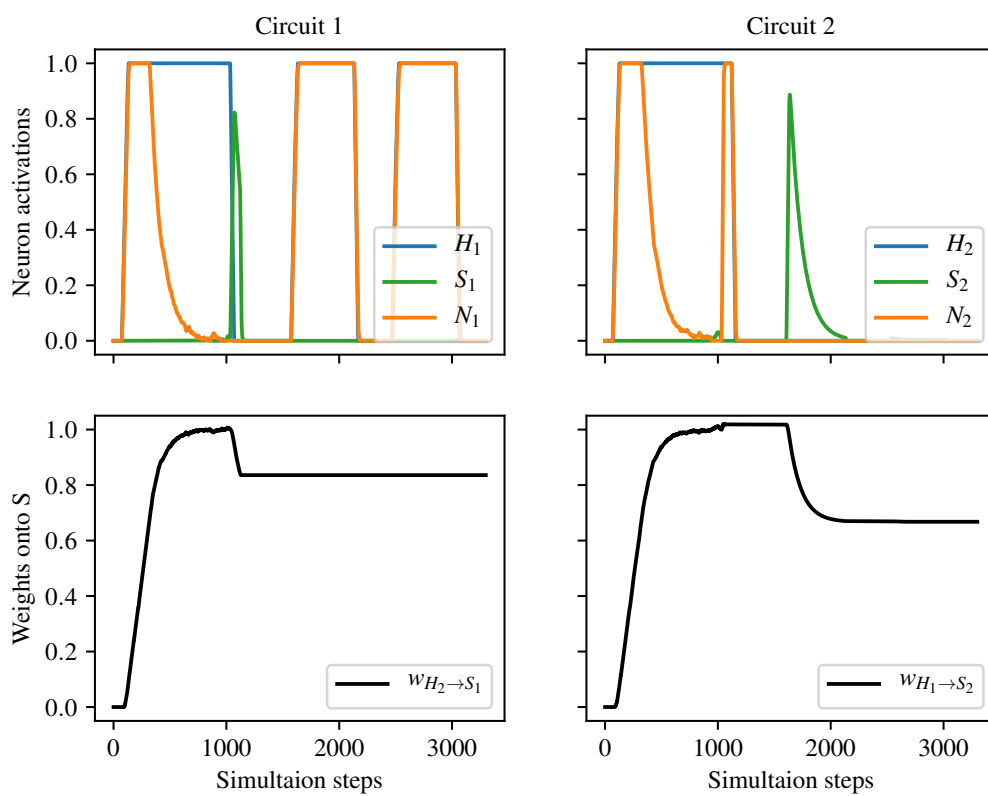


Fig. 3.10 Circuit 1 should trigger the activation of Circuit 2 but fails to do so because it unlearns the association before it succeeds in activating it.

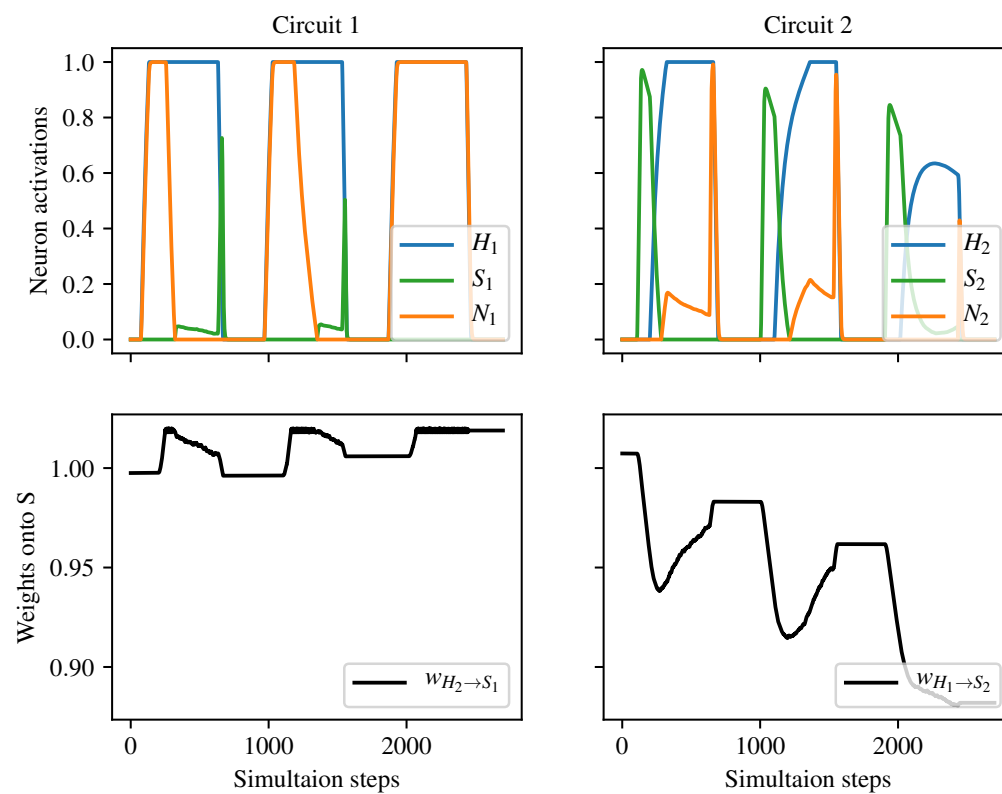


Fig. 3.11 With a reduced learning rate, Circuit 1 triggers the activation of Circuit 2 but partly unlearns the association, which does not have time to recover, eventually leading to the association being lost.

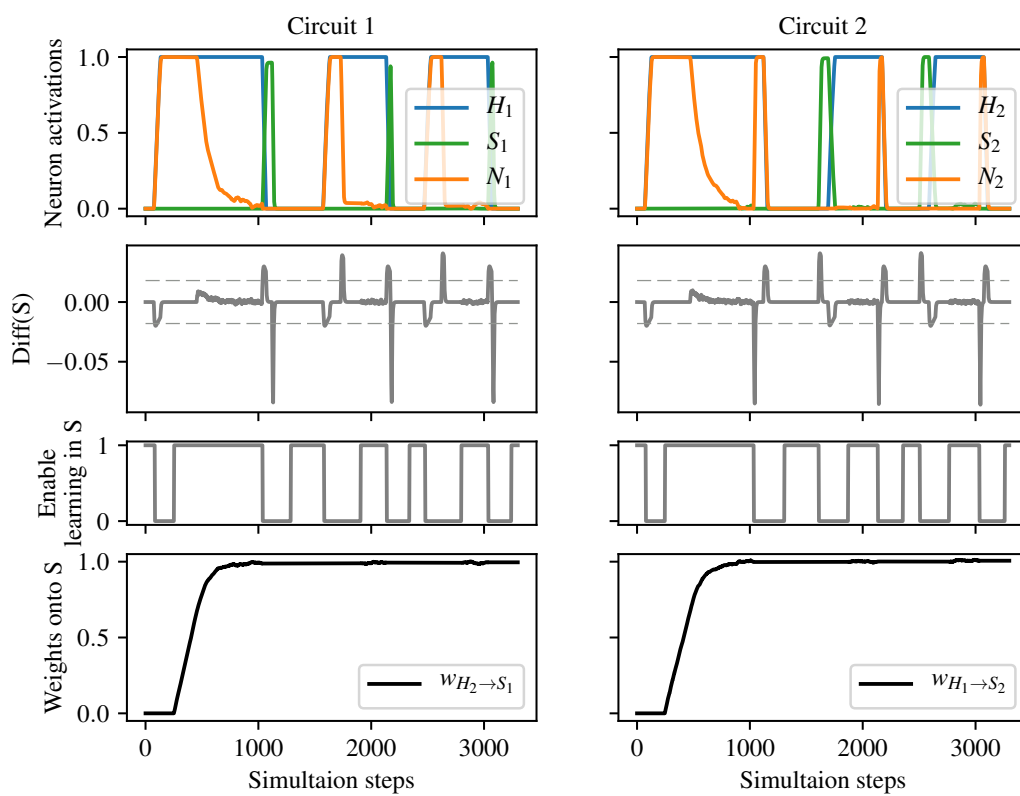


Fig. 3.12 When the derivative of the "should" neuron ($\text{Diff}(S)$) is above a certain threshold, learning is disabled for some time. This protects the weights from transients.

3.1.5 Learning to Act

Finally, we show how the circuit can learn to produce behavior. The example that we simulate is illustrated in Fig. 3.13. The system will receive motivational input for state A in its motivational s-pair. If A is desired but not currently present, the "should" neuron in this s-pair will activate. The system has to learn the association from this "should" neuron to whatever is needed to make A happen. That will be B in the example. Thus, the system will establish a connection between the motivational "should" neuron of A and that of B. This chain could go on for several stages of the hierarchy until at the end some motor system uses the mismatch signals to generate action.

In the first simulation (Fig. 3.14) we show a series of cycles where first the motivation for A arrives, then B gets activated (and remains activated) and after a delay, A is presented. An example of this could be that of learning that hot water is produced by leaving the tap turned to the left.

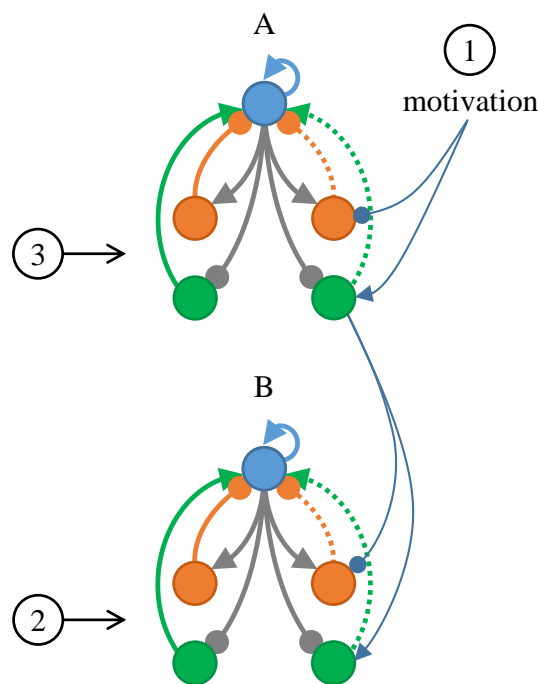


Fig. 3.13 Learning to act. The network will establish a connection between the motivational s-pair of the desired state A, and that of the state or action B that is required in order to bring about A. In the simulations (figures 3.14 and 3.15), the motivation input for A will arrive first, followed by the activation of B and then A.

Weight tagging is performed while the "should" neuron of A is active. The distribution of weight update then happens once A has been achieved. This way, the system learns to

associate the motivational "should" neuron to what was active right before the fulfillment of the desire (in this case, B).

We plot the weights from the motivational "should" neuron of A onto two dendrites of the corresponding neuron in B. Once the association has been learned, the unfulfilled desire of A causes the activation of the motivational "should" neuron of B.

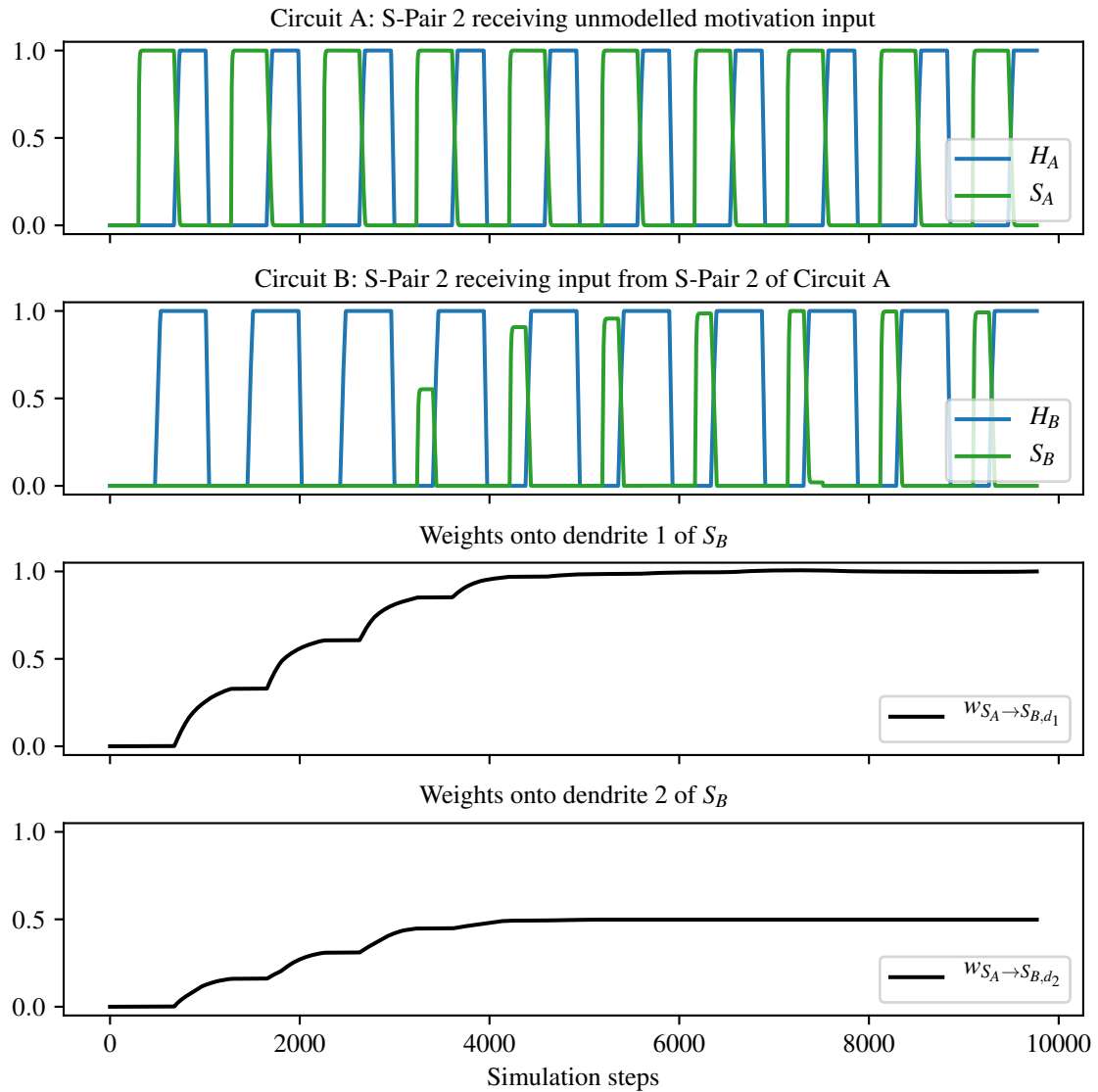


Fig. 3.14 The circuit in Fig.3.13 learns the association from the motivational "should" neuron of A to that of B.

In the simulation displayed in Fig. 3.15, the fulfillment of the desire and the action or state that produces it do not overlap. An example of this could be learning that people come and open the door after you ring the bell. In this case, we also show the evolution of the tags.

After the action has been performed, its motivational s-pair is silenced so that the system would not try to perform the action again in the small delay period.

At the beginning, the co-occurrence of the desire and the action increases the value of the positive tag ("tag up"). After the action is performed, the tag decays exponentially. When the desired state is achieved, the distribution of weight update is allowed and takes place with the value that remains in the tag at that time. When the system learns the association, it activates the motivational "should" neuron of B ahead of its occurrence. This causes the negative tag to grow in the periods leading to the action, since positive values in the "should" neuron always lead to depression of the weights that are causing it. However, the negative tag will decay while the action is actually performed. At that point, positive tags can grow if necessary. An equilibrium is reached when negative tags, larger but which start to decay earlier, match positive tags, smaller but closer to the point where the tags are used.

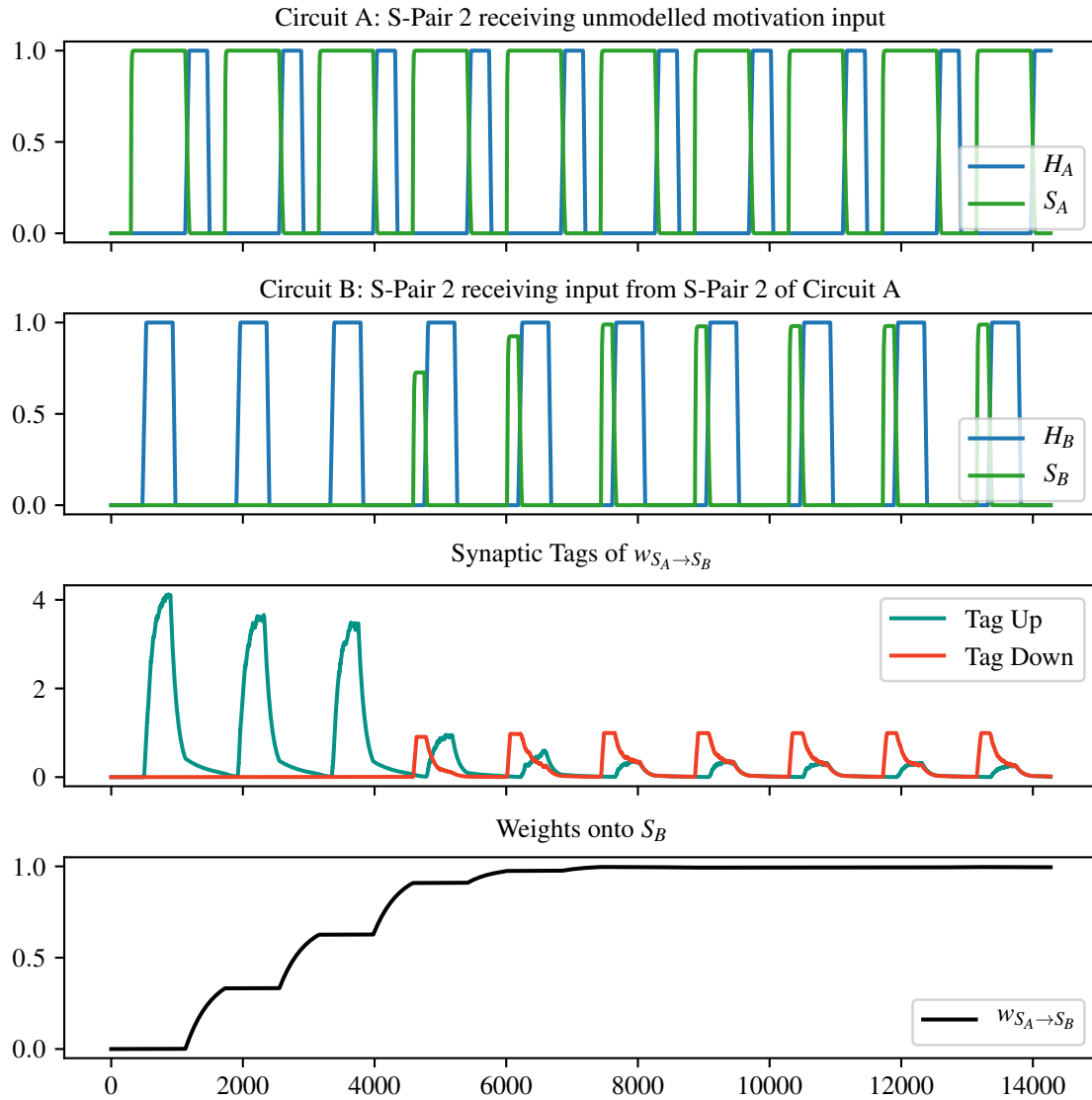


Fig. 3.15 The circuit in Fig.3.13 learns the association from the motivational "should" neuron of A to that of B, even when the activations of A and B do not overlap and there is a small delay between them. The third plot shows the weight tags. See text for explanations.

3.2 Spatial Cognition and Navigation System

Here we present simulation results for the navigation system presented in the previous chapter and reproduced again for convenience in Fig. 3.16, now labeling the s-pairs with the same index that is used in the simulations.

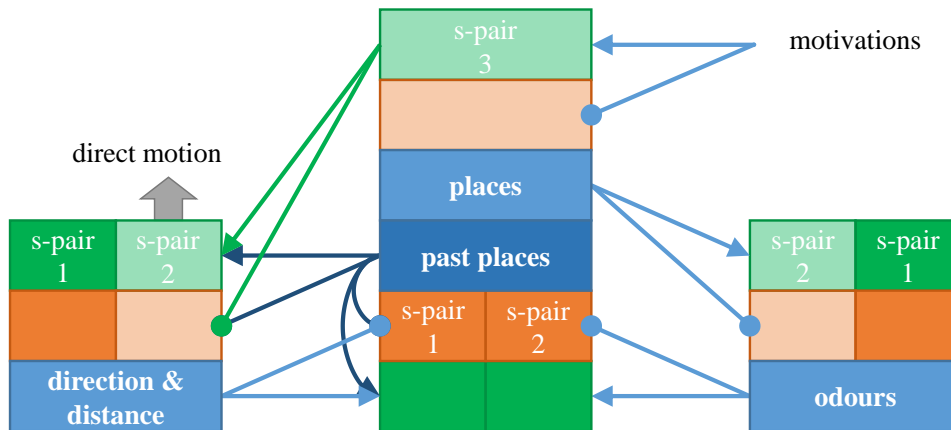


Fig. 3.16 Navigation system.

The system will learn the "pixelated" 1D environment shown in Fig. 3.17. The space is composed of 6 positions. Three of them have a characteristic odor and thus become represented as relevant places. The simulated agent will walk back and forth and learn the relative distances and directions between these places, as well as their odors.

S-Pair 1 of odors receives simulated sensory input that drives the activation of the odors at the corresponding places. S-Pair 1 of directions & distances receive simulated path integration input that drives their activation. When the agent is about to leave a place, the distance is reset to 0, the direction is modified if necessary, and the "past" neurons for places are reset to the current value of the "head" neurons for places.

The learning process is shown in figures 3.18 and 3.19. All simulations will be presented in this format. The first figure displays the activation of all neurons in a heat map, as shown previously, but now also including the "past" neurons in beige. The second figure shows all the weights onto "should" neurons, again through a heat map. The first plot in that figure, for example, represents all the weights onto dendrite number 1 of "should" neurons in s-pair 2 of the group of microcircuits coding for odors. The labels on the left indicate the number of the microcircuit whose "should" neuron is receiving the connection with the weight displayed, whereas the labels on the right indicate where the connections are coming from. "p" stands for places, thus "p1" means that the connection is from the first place. "p1p" means that the

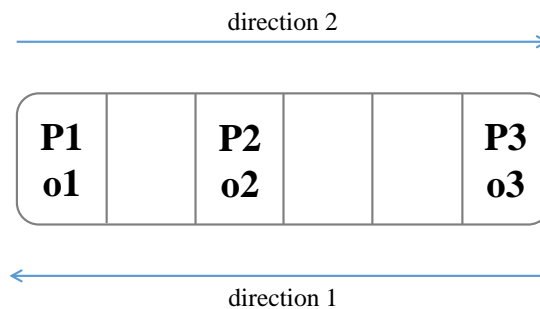


Fig. 3.17 "Pixelated" 1D environment used in the simulations.

connection is from the past place cell coding for the place. "p1s" means that the connection is from the (motivational) "should" neuron of the place. "o" stands for odors, "h" for heading direction and "d" for distance.

In the simulation, the agent first encounters the second odor. Since the odor is unexplained, it will recruit an uncommitted place neuron to represent it. The recruited place neurons will have the same index as the odor because it makes the visualization easier to follow. At a distance of 3, it will encounter odor number 3 and will recruit a new place neuron. Now it will also learn that this new place, place 3, is at a distance of 3 in direction 2 from place 2. This proceeds until it has associated a place to each odor and has learned the relative positions between them. Note that place number 2 can be reached in two ways ($p1p + h2 + d2$ or $p3p + h1 + d3$). It will learn each combination in one dendrite. This is why "should" neurons for places have two dendrites.

After learning, when the simulated agent is moved across the space, the corresponding places get activated and the correct odors are not surprising. This can be seen at the end of the simulation and at the beginning of the next one (Fig. 3.20). In this second simulation, the agent first localizes itself upon finding odor number 2. The odor by itself (in the absence of the correct metric information) is not enough to fully activate the place, and the odor is thus partly unexplained. This triggers the mechanism that recruits neurons, but, since the neuron for place 2 is already partly activated, the mechanism will "help" this neuron to become fully active instead of recruiting a new one. After one pass, the odor landmarks are removed. In the next passes, the network gradually gets lost. This happens because the lack of odors partially inhibits the activation of the places and the effect builds up: since the network is not sure that it has arrived to place 2, it will be even less sure that it has arrived to place 3 when the odometric information indicates it. In as much as it is certain that it is in a place, it will trigger the activation of the "should" neurons for the odors associated to the place.

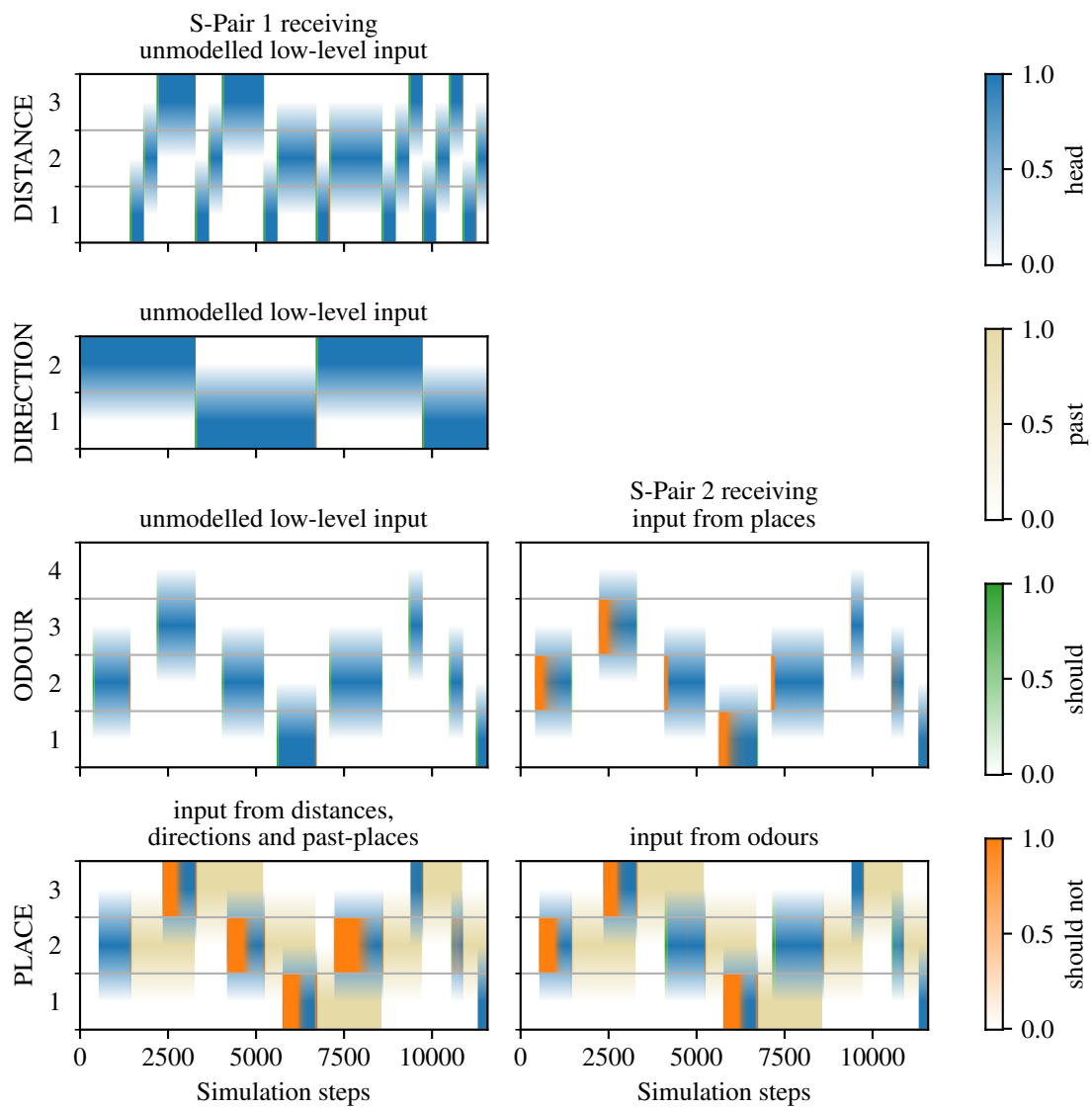


Fig. 3.18 The network learns the environment for the first time.

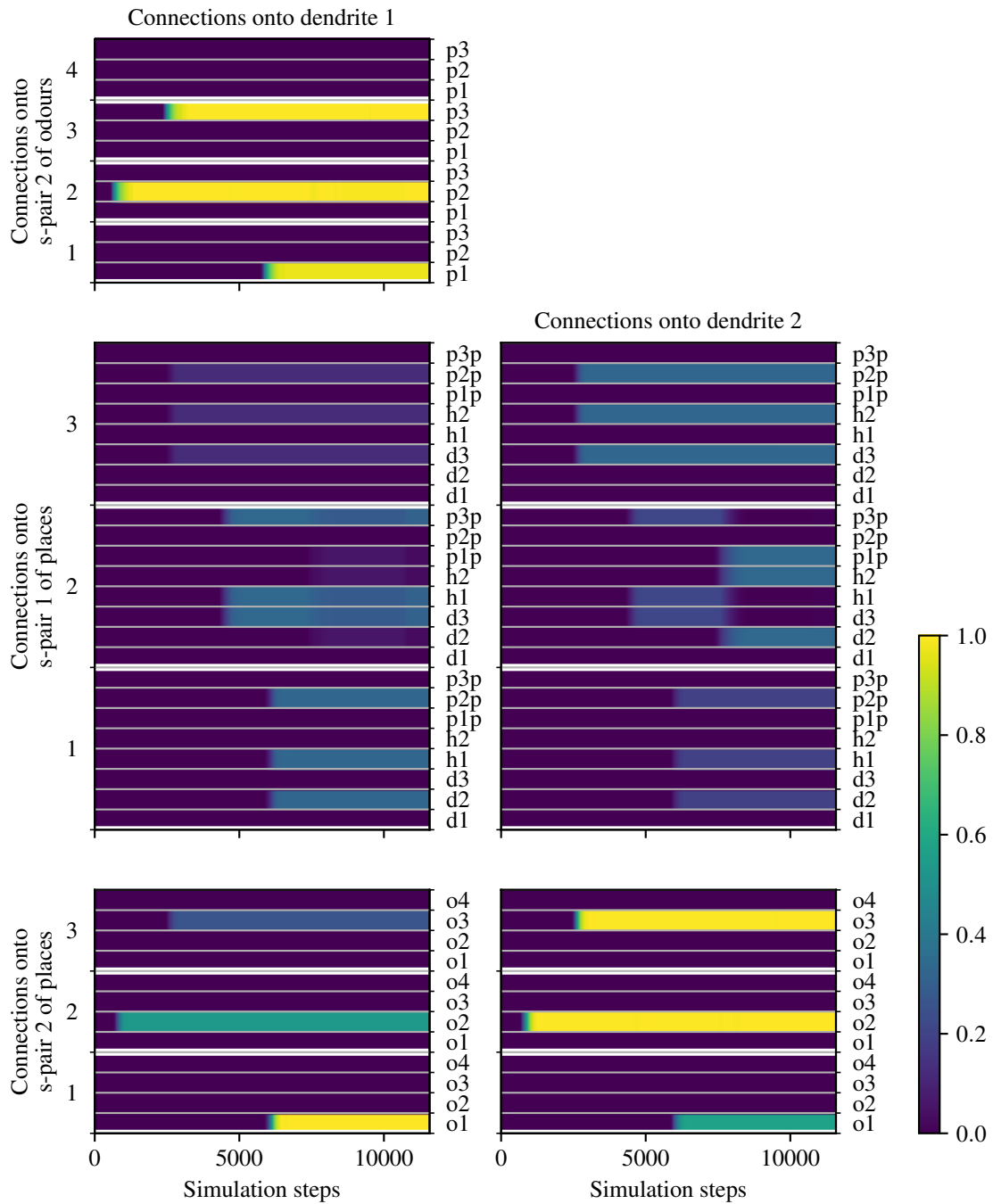


Fig. 3.19 Evolution of the weights as the network learns the environment. See text for an explanation of the plots.

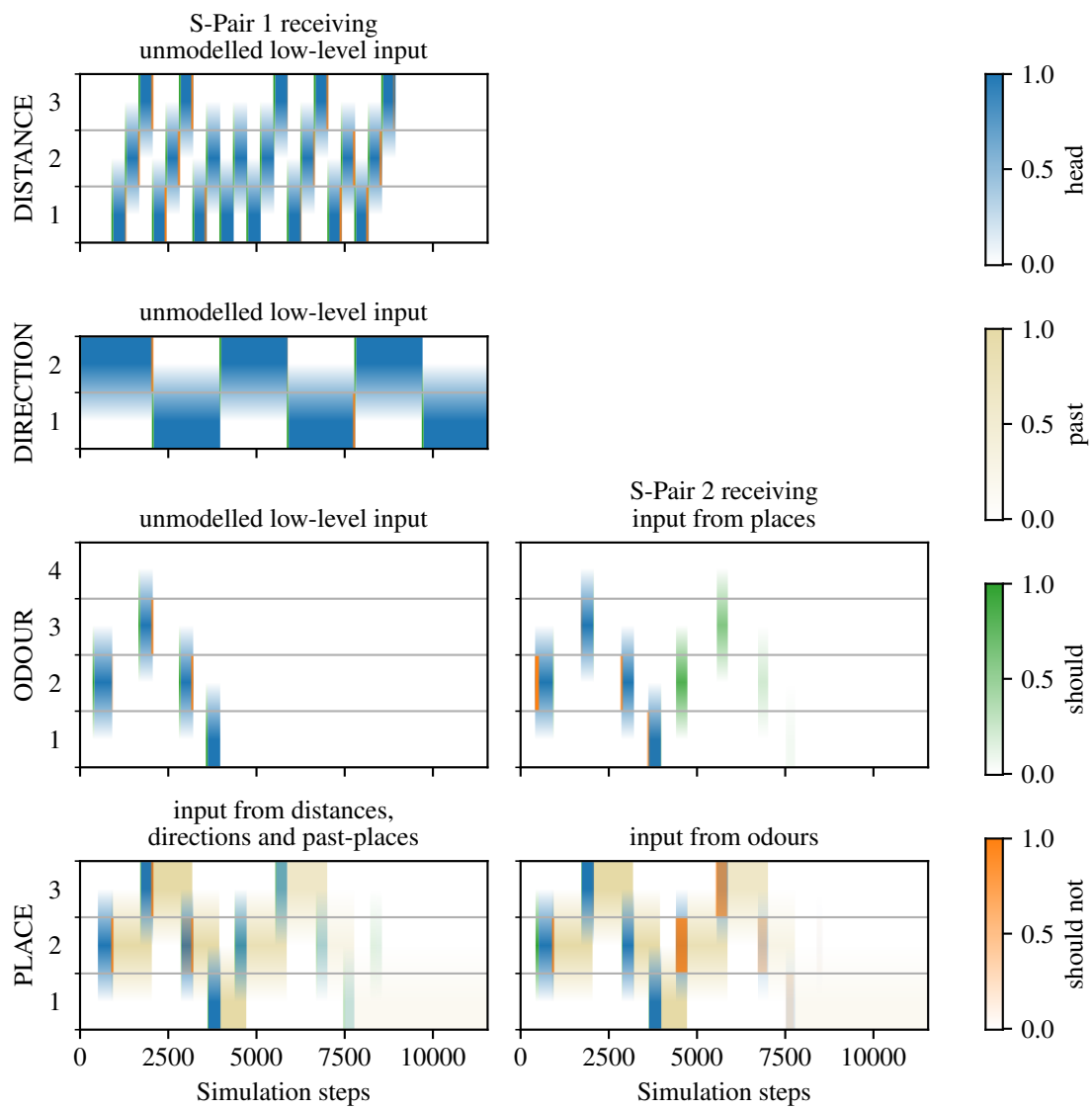


Fig. 3.20 The network gets lost when the odor landmarks are removed.

In the simulation showed in figures 3.21 and 3.22, odor 3 is substituted by odor 4. At first, when in place 3, the network will indicate that odor 3 is missing, activating its "should" neuron, and will activate the "should-not" neuron for odor 4. However, it quickly learns the new arrangement (the learning rate can of course be made slower).

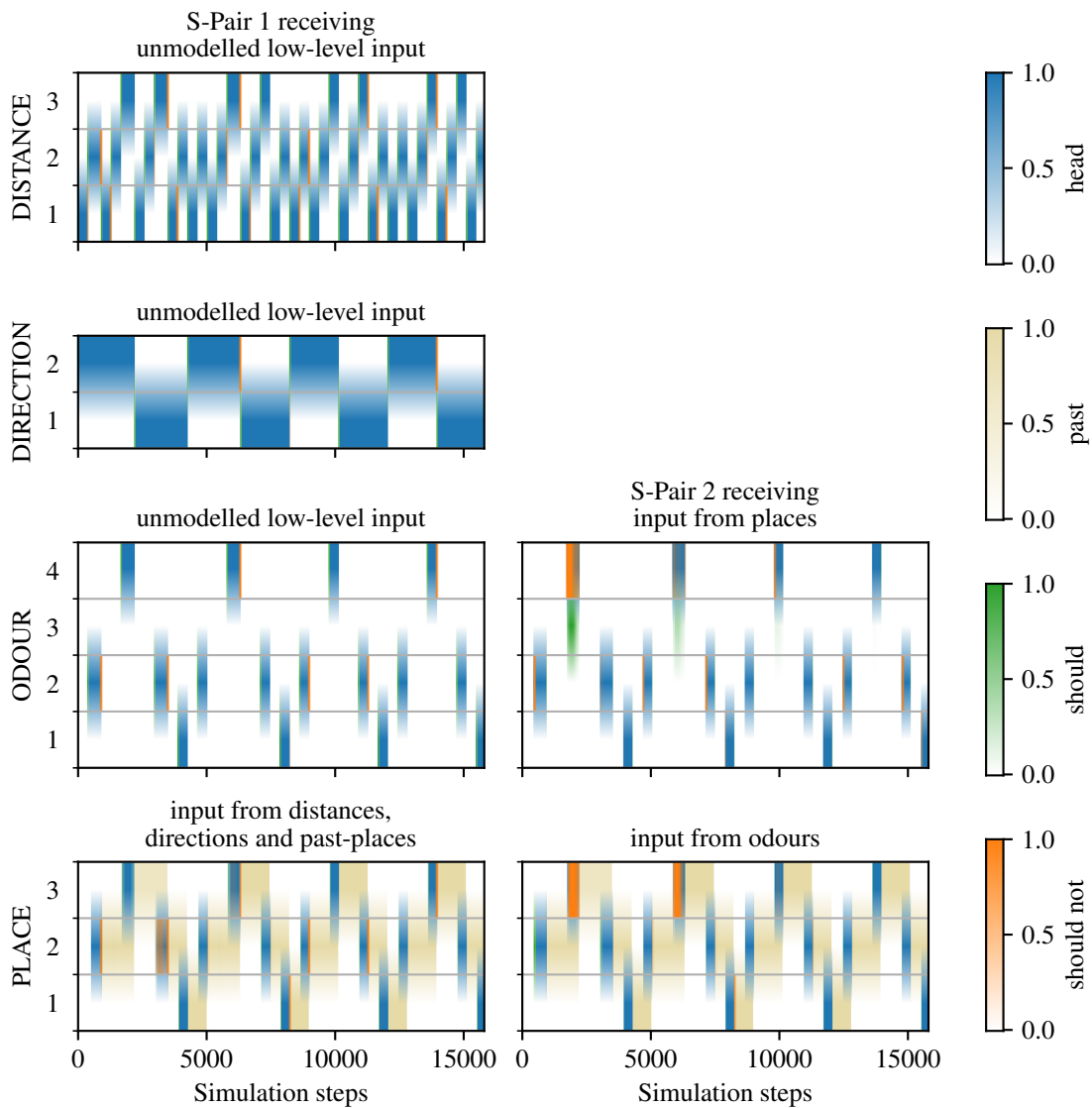


Fig. 3.21 Odor 3 is substituted for odor 4, generating mismatch signals until the network adapts to the change.

The last simulation (figures 3.23 and 3.24) shows the network learning how to go from one place to the next. So far, it had learned only the connections onto the s-pair of places receiving metric information, that is, it could recognize when it had arrived at places, but it could not use that information for traveling to them. For that, the network needs to learn

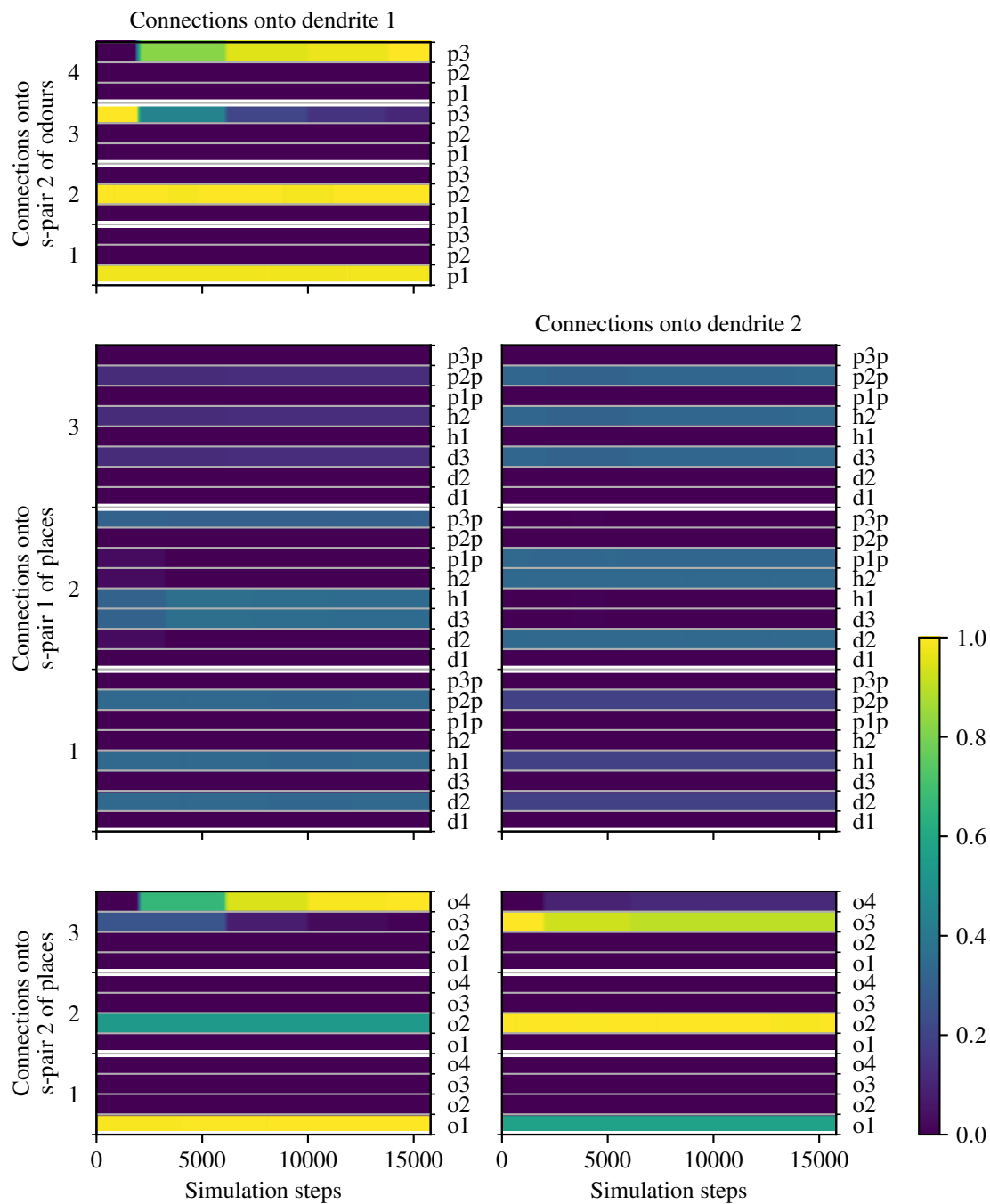


Fig. 3.22 Evolution of the weights as odor 3 is substituted by odor 4.

connections onto a motivational s-pair of distances and directions. These connections will originate from the last place active (past places) and a motivational s-pair for places indicating the place where the agent wants to go. This situation is analogous to first one showed in the section "Learning to Act". Fig. 3.25 shows the activations for the next pass after those shown in Fig. 3.23, since the scale there is too small. Each distance and direction occurs twice, for example, the direction h2 is required for going from place 1 to place 2 ($p1p + p2s$) and for going from place 2 to place 3 ($p2p + p3s$). Thus, each combination will be learned in one dendrite.

After learning (Fig. 3.25), the desire to go to a place is reflected in the activation of the motivational s-pair of distances and directions. If the current distance is not the one required for arriving at the desired place, its "should-not" neuron will be activated, whereas the "should" neuron of the required one will be activated. The same applies for the directions. These mismatch signals could then be used by the motor system to drive locomotion.

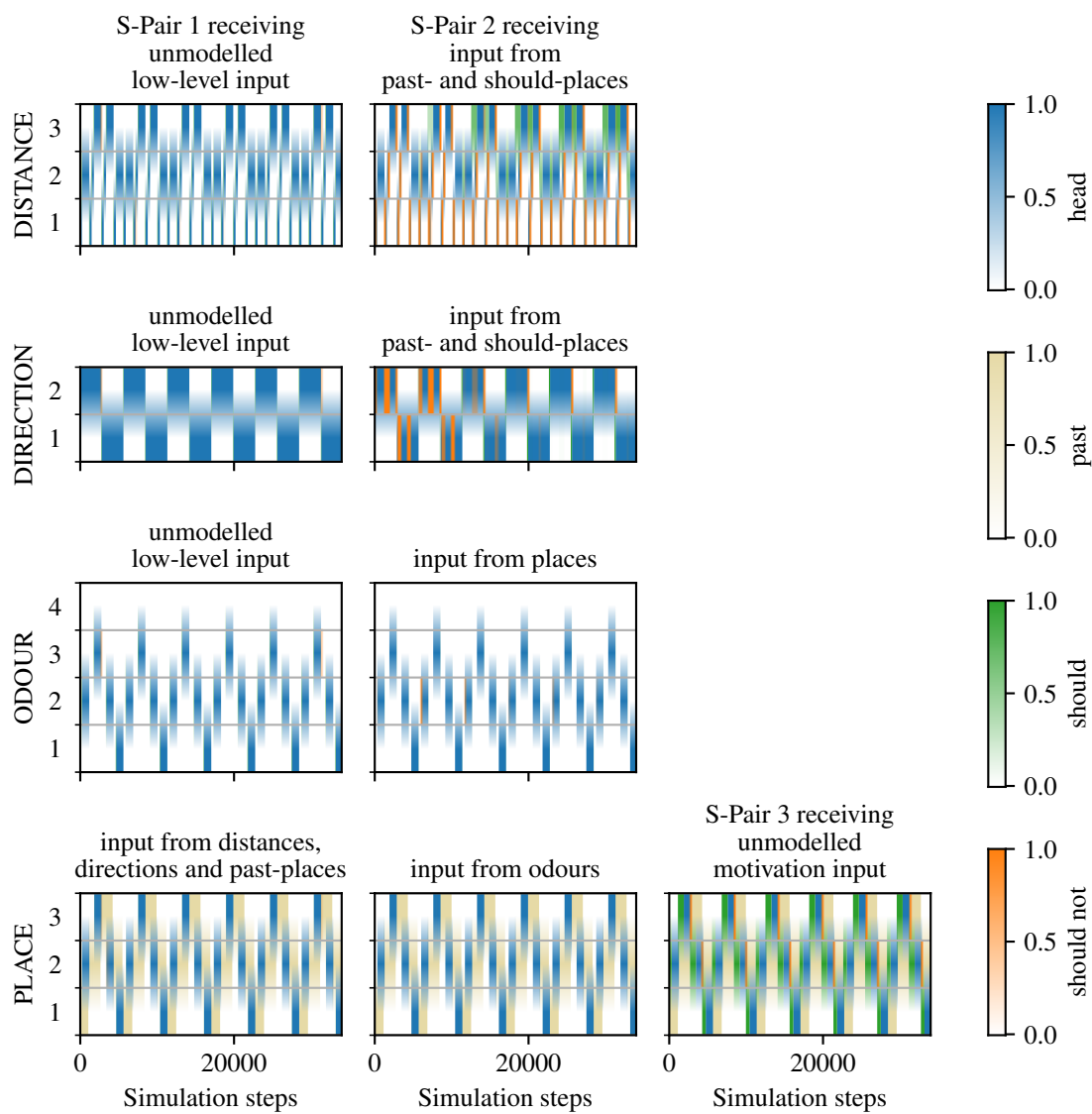


Fig. 3.23 The network learns how to navigate to places. Fig. 3.25 shows the next pass through the environment at a better scale.

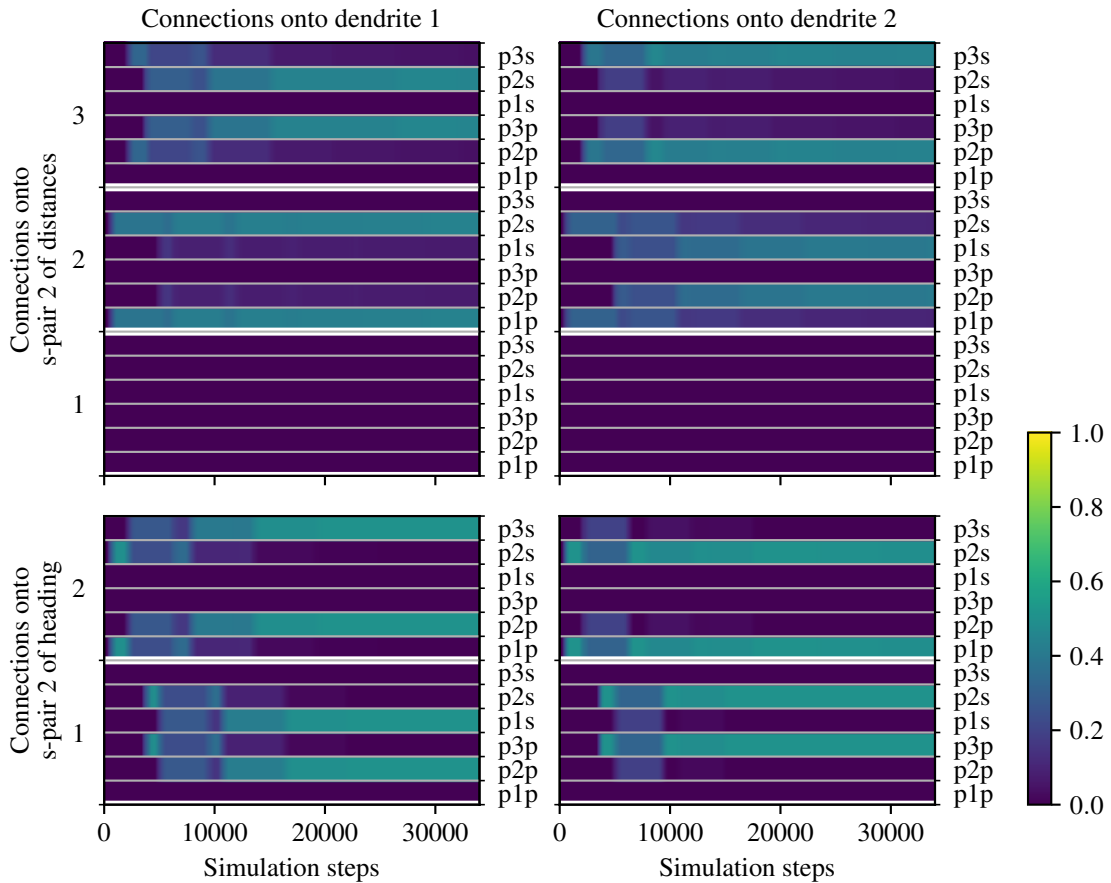


Fig. 3.24 Evolution of the weights onto the motivational s-pairs (s-pair 2) of distances and directions as the network learns to go from one place to the next. There are two combinations of past and desired places that need to be associated to each distance and direction. Each combination clusters onto a different dendrite.

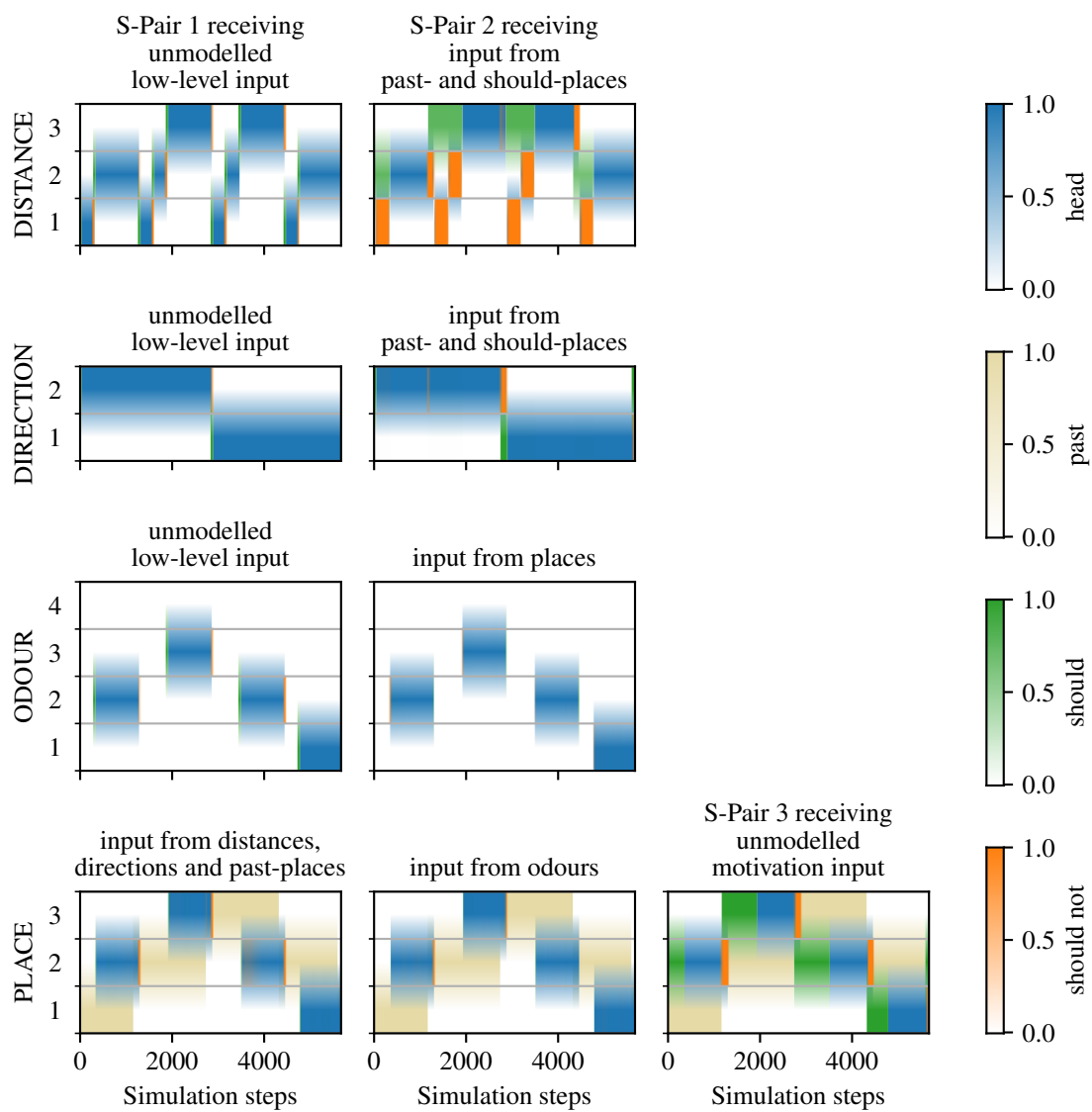


Fig. 3.25 After learning, the mismatch signals in s-pair 2 of distances and directions can be used for controlling locomotion.

Chapter 4

Discussion

4.1 Summary and Conclusions

In this thesis, I have presented a mismatch detection neural circuit and applied it to a simple navigation task. I have argued that mismatches can be of two fundamental types: mismatches by excess, when there are elements in the world that are not supported by the model; and mismatch by deficit, when there are missing elements in the world with respect to the model. The mismatch detection circuit proposed can account for both types of mismatch whereas other models typically focus on mismatch by excess.

Mismatch detection is pervasive in cognition. We engage in it constantly as we perceive the world. Doing so provides two important advantages. First, it signals the presence of elements or situations that are unexpected and thus, the agent has not prepared for, often demanding changes in behavior. This is particularly relevant in the context of a system with limited cognitive capacities that can not process all stimuli to the same extent, and thus needs to select which stimuli to attend to. It may seem paradoxical that a system with limited resources is, nevertheless, employing so many of them for mismatch detection. However, the process of detecting mismatches can be performed in a simple and massively parallel fashion, whereas other higher-level cognitive functions seem to be much more limited, for instance by the bottleneck of working memory that can only hold the information related to 3 to 5 objects (Cowan, 2010).

Mismatches can arise because the activation pattern in the model does not reflect the elements present in the environment, in which case the mismatch is corrected simply by modifying the activation pattern. Alternatively, mismatches can result from discrepancies between the world and the structure of the model, thus requiring that the structure be changed by learning. In this latter case, the detection of mismatches provides a second advantage by explicitly calculating an error signal that can be used for learning, for example, by modifying

connections in proportion to the error. Furthermore, the presence of mismatch signals can be used to determine when a new pattern is being experienced and neurons need to be recruited to represent it, as in the model by Carpenter and Grossberg (1988).

Finally, mismatches are also relevant when it comes to comparing the world to your desires. In this case, they directly indicate situations in which actions need to be taken.

The mismatch detection circuit here proposed is based on contrasting the presence of some feature with a priori predictions or a posteriori justifications. This is done by making the two elements to be compared converge on the same neuron with weights of opposite sign. Mismatch by excess results from the feature being excitatory and the prediction inhibitory, whereas mismatch by deficit results from the prediction being excitatory and the feature inhibitory. The balance between excitation and inhibition is thus fundamental for the operation of the circuit and typically excitation will be followed by inhibition. This is in agreement with observations about the behavior of excitation and inhibition in the cortex, reviewed in Isaacson and Scanziani (2011). It has been observed that in auditory, somatosensory and visual cortex, synaptic excitation is followed after a few milliseconds by a surge in inhibition. This finding is often interpreted in the context of gain control or sharpening of tuning curves, but mismatch detection may offer an alternative or complementary explanation.

The proposed mismatch detection circuit is similar to the circuits employed for predictive coding (Rao and Ballard (1999), Bogacz (2017)). However, unlike in those circuits, both types of mismatch are represented explicitly by the positive activation level of some neuron. This seems to be in agreement with findings that report increased activity in the cortex by both types of mismatch and not just mismatch by excess. A further difference between our model and predictive coding circuits stems from the fact that in the latter, only error neurons output their values to other areas, meaning that explained stimuli virtually disappear for the system. In contrast, in the mismatch detection circuit, connections between different circuits stem primarily from their "head" neurons that signal the presence of the feature. We believe that this fits better with introspective experience and is more advantageous for the production of flexible behavior.

The proposed circuit makes use of dendritic nonlinearities, similar to the ones that have been found in biological neurons. This increases the discriminatory capacity of neurons, essentially converting them into two-layer networks.

Learning is governed by a Hebbian rule, where weight update is proportional to the activations of the pre- and post-synaptic neurons. However, since it is proportional, specifically, to the negative value of the post-synaptic neuron, weights self-normalize without the need for any additional homeostatic mechanism. The learning rule developed ensures that different input combinations cluster automatically onto different dendrites of the neuron. It also makes

sure that connections from equally reliable inputs converge to the same weight. Finally, it can also be used for learning the association between desires and the actions that need to be taken in order to fulfill them. These properties rely on taking advantage of several sources of noise, and on the division of learning into a phase of weight "tagging" followed by one of weight update distribution. This was developed out of necessity, but bears a high degree of similarity to the process of synaptic tagging and capture that is believed to take place in biological neurons, and thus can offer insights into why this phenomenon occurs.

Clustered plasticity is particularly amenable to receiving insights from trying to engineer a system that performs this function, for it is conceptually a relatively simple phenomenon. It is clear that two elements are required: a mechanism that makes weights grow preferentially where there is already the largest amount of weight, and a mechanism that ensures that clusters do not overlap, which would render them useless. A biological candidate to the second mechanism has been recently reported by Oh et al. (2015), who show that stimulation at several clustered spines led to shrinkage and weakening of nearby unstimulated synapses. This shrinkage does not seem to depend on competition for resources between synapses since it can be dissociated from the growth of the stimulated synapses, thus suggesting that it depends on some local signal that is produced when a threshold of activation is surpassed. I modeled this phenomenon and, as expected, in conjunction with the first mechanism, it ensured that different input combinations clustered onto different dendrites. However, nothing prevented a new cluster from "overwriting" an older one. If memories are to be protected, some mechanism has to ensure that clusters grow onto unoccupied dendrites or dendrites that encode patterns that are not very useful for the system. To achieve this, a competition for resources or space between synapses, based on some measure of fitness, seems to be necessary and is the solution that I have implemented.

The mismatch detection circuit has been applied to a navigation task. This practical application demonstrates the capabilities of the circuit, but has also been very useful for stimulating its development. For example, multiple dendrites were introduced because of the need to separate the different ways in which a place can be reached, and the use of mismatch neurons for generating behavior was motivated by the need of having some way of telling how to go from one place to the next.

In the introductory literature review, it was argued that the animal brain does not encode a globally coherent Cartesian representation of space. This is beneficial, since maintaining such a coherent representation based on noisy sensory and odometric input is computationally challenging, as demonstrated by artificial SLAM systems. It was also argued that recognition of places and objects are likely to be performed in a similar way, and that the challenges faced in recognition of objects invariant to translation, scale and rotation further supports

a representation of space based on relative positions between features. Thus, in our model, space is represented as a series of places, identified by some landmark and linked by their relative distances and directions. Specifically, directions are specified with respect to some local reference frame. This provides more flexibility than purely relative directions and is in agreement with the well established finding of head direction cells in the rodent brain.

The model has learned a simple virtual environment composed of a "pixelated" 1D space where landmarks are odors. The choice of odors for landmarks is motivated by the fact that odors are local. If we had employed a type of stimulus that can be perceived at a certain distance from the current place, such as visual stimuli, the system would become more complex. The same kind of representation of space would be possible, but additional mechanisms would be necessary for combining distances and directions between landmarks estimated from locomotion and vision.

The system can detect when expected odors are missing, or when unexpected odors appear. This is similar to the findings reported by Fyhn et al. (2002), who measured the response of hippocampal cells to first time dislocation of target objects and found increased firing rates in some cells when objects were moved away from their learned positions.

Finally, it is worth noticing that a complete biologically inspired model of navigation probably falls close to a model of general intelligence, since navigation is a very complex task that involves a hard pattern recognition problem, the representation of past and future events, and the planning and execution of behaviors.

4.2 Outlook

The investigation carried out in this thesis has opened multiple research lines that would be interesting to pursue in the future.

First, with respect to the mismatch detection circuit, it would be valuable to further assess its biological plausibility, and correct the features that are found to be less acceptable. The rules for clustered plasticity that have been developed depend on time-varying noise for initiating the clustering. Perhaps it would be more realistic and efficient to introduce fixed noise in the "ease" with which each input can cluster onto each dendrite, which would depend in the brain on the physical proximity of the elements involved.

The learning rule dictates how connections are established between activated microcircuits, but does not say how these activations come about in the first place when nothing is known. Throughout most simulations, I manually activated the microcircuits, and only at the end a very rough system was introduced for automatically recruiting place neurons in the

navigation system. If the mismatch detection circuit is to be of any value, a self-organizing mechanism that recruits neurons to represent new patterns needs to be added.

Further refinements would involve a better measure for weight fitness, that can saturate and decrease (the current one only increases), or a mechanism for dynamically changing the learning rate based on the amount of mismatch.

Another problem that currently limits the capabilities of the circuit, as well as that of most neural systems proposed so far, is the lack of a mechanism capable of handling combinatorial structures in a flexible way (Fodor and Pylyshyn, 1988). This is very related to the binding problem (Treisman, 2002). Compare the sentences "John ate a pig" and "A pig ate John". They are composed of the same elements but the second one is much more concerning. In the context of the mismatch detection circuit, the second sentence should elicit a higher mismatch signal. However, it is not clear how a neural system can represent these two situations without resorting to specialized neurons that would bind each compound (e.g. John-subject, John-object, etc.), which would however be a very inefficient and unpractical solution. Different models have been proposed for how this binding could be performed dynamically by reuniting the bound elements under a common dimension, such as their associated position in space, the time at which neurons encoding the pattern fire (Singer (2007); as in the system developed by Shastri (1999)), or even signature activation patterns (Lange and Dyer, 1989).

Possibly related to this, the mismatch detection circuit is lacking a good representation of time where, for example, different events could be represented as having happened at different points in the past.

With respect to the navigation system, much can be done. First, distances and directions should be represented in a more continuous fashion, and one that would easily allow distances and directions that are close to each other to be recognized as such. A solution for this could be to represent distances and directions using populations of neurons such that distances and directions that are close to each other partly overlap in their representations. This would allow for them to be compared based on their degree of overlap. Using this method, the system could, for example, decide whether the memorized place for a landmark has been reached (with some path integration error) or the landmark is new and a new place neuron circuit needs to be recruited. Note that the system learns that it has arrived at a place if the triple (past-place, distance, direction) is satisfied. If distances and directions are represented by populations, and given that the learning rule homogenizes the weights across all inputs, the places should also be represented by populations of the same size if they are to be weighted in by the same amount. This would also make the system very robust to noise, and it would explain why it seems to be relatively easy to find place cells in the brain. Once continuous

distances and directions can be represented, extending the model to work in 2D would be trivial.

Another issue that needs to be resolved is how to handle the repetition of landmarks in the environment. Ideally, if the agent is disoriented and sees a landmark that occurs in several known positions, it should entertain simultaneously the hypothesis that it is at all those positions, and use future information to disambiguate its real location. Additionally, the symmetry in the weights onto "should" and "should-not" neurons would need to be broken, since a repeated landmark should not elicit a strong "should" signal for any of the places where the landmark appears, but should inhibit their "should-not" neurons.

Furthermore, in the model, heading direction is given as an unmodeled simulated input, but a neural mechanism should be developed that could account for the production of this signal based on a combination of odometric and sensory information.

Finally, the model could be extended to operate hierarchically at different spatial scales, as well as to combine distances and directions obtained from locomotion and vision such that the same model could account for the recognition of objects and places.

I look forward to keep working on these and related issues.

References

- Anderson, M. I. and Jeffery, K. J. (2003). Heterogeneous modulation of place cell firing by changes in context. *The Journal of Neuroscience*, 23(26):8827–8835.
- Aronov, D., Nevers, R., and Tank, D. W. (2017). Mapping of a non-spatial dimension by the hippocampal–entorhinal circuit. *Nature*, 543:719–722.
- Bastos, A. M., Usrey, W. M., Adams, R. A., Mangun, G. R., Fries, P., and Friston, K. J. (2012). Canonical microcircuits for predictive coding. *Neuron*, 76(4):695–711.
- Bogacz, R. (2017). A tutorial on the free-energy framework for modelling perception and learning. *Journal of Mathematical Psychology*, 76:198–211.
- Carpenter, G. A. and Grossberg, S. (1988). The ART of adaptive pattern recognition by a self-organizing neural network. *Computer*, 21(3):77–88.
- Chen, J., Olsen, R. K., Preston, A. R., Glover, G. H., and Wagner, A. D. (2011). Associative retrieval processes in the human medial temporal lobe: Hippocampal retrieval success and CA1 mismatch detection. *Learning and Memory*, 18(8):523–528.
- Collett, M., Collett, T. S., Bisch, S., and Wehner, R. (1998). Global and local vectors in desert ant navigation. *Nature*, 394:269–272.
- Conradt, J. A. (2008). *A Distributed Cognitive Map for Spatial Navigation Based on Graphically Organized Place Agents*. PhD thesis, Swiss Federal School of Technology (ETH) Zürich.
- Cowan, N. (2010). The magical mystery four: How is working memory capacity limited, and why? *Current Directions in Psychological Science*, 19(1):51–57.
- Cuperlier, N., Quoy, M., and Gaussier, P. (2007). Neurobiologically inspired mobile robot navigation and planning. *Frontiers in Neurorobotics*, 1(3):209–226.
- Douglas, R. J., Martin, K. A., and Whitteridge, D. (1989). A canonical microcircuit for neocortex. *Neural Computation*, 1:480–488.
- Duncan, K., Ketz, N., Inati, S. J., and Davachi, L. (2012). Evidence for area CA1 as a match/mismatch detector: A high-resolution fMRI study of the human hippocampus. *Hippocampus*, 22:389–398.
- Egorov, A. V., Hamam, B. N., Fransén, E., Hasselmo, M. E., and Alonso, A. A. (2002). Graded persistent activity in entorhinal cortex neurons. *Nature*, 420:173–178.

- Eichenbaum, H., Dudchenko, P., Wood, E., Shapiro, M., and Tanila, H. (1999). The hippocampus, memory, and place cells: Is it spatial memory or a memory space? *Neuron*, 23:209–226.
- Engel, T. A. and Wang, X. (2011). Same or different? A neural circuit mechanism of similarity based pattern-match decision making. *Journal of Neuroscience*, 31(19):6982–6996.
- Etienne, A. S. and Jeffery, K. J. (2004). Path integration in mammals. *Hippocampus*, 14:180–192.
- Fiser, A., Mahringer, D., Oyibo, H. K., Peterson, A. V., Leinweber, M., and Keller, G. B. (2016). Experience-dependent spatial expectations in mouse visual cortex. *Nature Neuroscience*, 19:1658–1664.
- Fodor, J. A. and Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.
- Fonseca, R., Nägerl, U. V., Morris, R. G. M., and Bonhoeffer, T. (2004). Competing for memory: Hippocampal LTP under regimes of reduced protein synthesis. *Neuron*, 44:1011–1020.
- Foo, P., Warren, W. H., Duchon, A., and Tarr, M. J. (2005). Do humans integrate routes into a cognitive map? Map- versus landmark-based navigation of novel shortcuts. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 31(2):195–215.
- Friston, K. J. (2005). A theory of cortical responses. *Phil. Trans. R. Soc. B*, 360(1456):815–836.
- Fyhn, M., Molden, S., Hollup, S., Moser, M.-B., and Moser, E. I. (2002). Hippocampal neurons responding to first-time dislocation of a target object. *Neuron*, 35:555–566.
- Glimcher, P. W. (2011). Understanding dopamine and reinforcement learning: The dopamine reward prediction error hypothesis. *PNAS*, 108(S3):15647–15654.
- Gothard, K. M., Skaggs, W. E., Moore, K. M., and McNaughton, B. L. (1996). Binding of hippocampal cal neural activity to multiple reference frames in a landmark-based navigation task. *Journal of Neuroscience*, 16(2):823–835.
- Govindarajan, A., Kelleher, R. J., and Tonegawa, S. (2006). A clustered plasticity model of long-term memory engrams. *Nature Reviews Neuroscience*, 7:575–583.
- Gray, N. W., Weimer, R. M., Bureau, I., and Svoboda, K. (2006). Rapid redistribution of synaptic PSD-95 in the neocortex in vivo. *PLoS Biology*, 4(11).
- Hafting, T., Fyhn, M., Molden, S., Moser, M. B., and Moser, E. (2005). Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801–806.
- Haikonen, P. O. (2014). Yes and no: match/mismatch function in cognitive robots. *Cognitive Computation*, 6:158–163.
- Hartley, T., Lever, C., Burgess, N., and O’Keefe, J. (2014). Space in the brain: How the hippocampal formation supports spatial cognition. *Phil. Trans. R. Soc. B*, 369(1635).

- Hasselmo, M. E. and Schnell, E. (1994). Laminar selectivity of the cholinergic suppression of synaptic transmission in rat hippocampal region CA1: Computational modeling and brain slice physiology. *The Journal of Neuroscience*, 14(6):3898–3914.
- Hetherington, P. A. and Shapiro, M. L. (1997). Hippocampal place fields are altered by the removal of single visual cues in a distance-dependent manner. *Behavioral Neuroscience*, 111(1):20–34.
- Hollerman, J. R. and Schultz, W. (1998). Dopamine neurons report an error in the temporal prediction of reward during learning. *Nature Neuroscience*, 1(4):304–309.
- Isaacson, J. S. and Scanziani, M. (2011). How inhibition shapes cortical activity. *Neuron*, 72:231–243.
- Johnson, J. S., Spencer, J. P., Luck, S. J., and Schöner, G. (2009). A dynamic neural field model of visual working memory and change detection. *Psychological Science*, 20(5):568–577.
- Kanitscheider, I. and Fiete, I. (2016). Training recurrent networks to generate hypotheses about how the brain solves hard navigation problems.
- Keller, G. B., Bonhoeffer, T., and Hübener, M. (2012). Sensorimotor mismatch signals in primary visual cortex of the behaving mouse. *Neuron*, 74:809–815.
- Klauss, T., Marsh, W. E., Zetsche, C., and Schill, K. (2015). Representation of impossible worlds in the cognitive map. *Cognitive Processing*, 16:S271–S276.
- Kraus, B. J., Brandon, M. P., Robinson, R. J., Connerney, M. A., Hasselmo, M. E., and Eichenbaum, H. (2015). During running in place, grid cells integrate elapsed time and distance run. *Neuron*, 88:578–589.
- Kropff, E., Carmichael, J. E., Moser, M.-B., and Moser, E. I. (2015). Speed cells in the medial entorhinal cortex. *Nature*, 523:419–424.
- Kumaran, D. and Maguire, E. A. (2007). Match–mismatch processes underlie human hippocampal responses to associative novelty. *The Journal of Neuroscience*, 27(32):8517–8524.
- Lange, T. E. and Dyer, M. G. (1989). High-level inferencing in a connectionist network. *Connection Science*, 1(2):181–217.
- Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- Legenstein, R. and Maass, W. (2011). Branch-specific plasticity enables self-organization of nonlinear computation in single neurons. *The Journal of Neuroscience*, 31(30):10787–10802.
- Lisman, J. E. and Grace, A. A. (2005). The hippocampal-VTA loop: Controlling the entry of information into long-term memory. *Neuron*, 46:703–713.
- Major, G., Larkum, M. E., and Schiller, J. (2013). Active properties of neocortical pyramidal neuron dendrites. *Annual Review of Neuroscience*, 36:1–24.

- Makino, H. and Malinow, R. (2011). Compartmentalized versus global synaptic plasticity on dendrites controlled by experience. *Neuron*, 72:1001–1011.
- Mel, B. W. (1991). The clusteron: towards a simple abstraction for a complex neuron. *Advances in Neural Information Processing Systems*, 4.
- Milford, M. and Wyeth, G. (2009). Persistent navigation and mapping using a biologically inspired slam system. *The International Journal of Robotics Research*, 29(9):1131 – 1153.
- Miller, R. R., Barnet, R. C., and Grahame, N. J. (1995). Assessment of the Rescorla-Wagner model. *Psychological Bulletin*, 117(3):363–386.
- Moser, E. I., Kropff, E., and Moser, M.-B. (2008). Place cells, grid cells, and the brain’s spatial representation system. *Annual Review of Neuroscience*, 31:69–89.
- Noton, D. (1970). A theory of visual pattern perception. *IEEE Transactions on Systems Science and Cybernetics*, 6:349–357.
- Nääätänen, R., Paavilainen, P., Rinne, T., and Alho, K. (2007). The mismatch negativity (MMN) in basic research of central auditory processing: A review. *Clinical Neurophysiology*, 118:2544–2590.
- Oh, W. C., Parajuli, L. K., and Zito, K. (2015). Heterosynaptic structural plasticity on local dendritic segments of hippocampal ca1 neurons. *Cell Reports*, 10(2):162–169.
- O’Keefe, J. and Burgess, N. (1996). Geometric determinants of the place fields of hippocampal neurons. *Nature*, 381:425–428.
- O’Keefe, J. and Dostrovsky, J. (1971). The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, 34(1):171–175.
- O’Keefe, J. and Nadel, L. (1978). *The hippocampus as a cognitive map*. Oxford University Press.
- Pearce, J. J. and Hall, G. (1980). A model for Pavlovian learning: Variations in the effectiveness of conditioned but not of unconditioned stimuli. *Psychological Review*, 87(6):532–552.
- Polsky, A., Mel, B. W., and Schiller, J. (2004). Computational subunits in thin dendrites of pyramidal cells. *Nature Neuroscience*, 7(6):621–627.
- Poucet, B., Sargolini, F., Song, E. Y., Hangya, B., Fox, S., and Muller, R. U. (2013). Independence of landmark and self-motion-guided navigation: a different role for grid cells. *Phil. Trans. Roy. Soc. B*, 369(20130370).
- Quiroga, R. Q., Mukamel, R., Isham, E. A., Malach, R., and Fried, I. (2008). Human single-neuron responses at the threshold of conscious recognition. *PNAS*, 105(9):3599–3604.
- Rao, R. P. N. and Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2:79–87.

- Redondo, R. L. and Morris, R. G. M. (2011). Making memories last: the synaptic tagging and capture hypothesis. *Nature Reviews Neuroscience*, 12:17–30.
- Rescorla, R. A. and Wagner, A. R. (1972). A theory of Pavlovian conditioning: variations in the effectiveness of reinforcement and nonreinforcement. In Black, A. H. and F., P. W., editors, *Classical conditioning II: current research and theory*, chapter 3, pages 64–99. Appleton-Century-Crofts, New York.
- Rybak, I., Gusakova, V., Golovan, A., Podladchikova, L., and Shevtsova, N. (1998). A model of attention-guided visual perception and recognition. *Vision Research*, 38(15):2387–2400.
- Sajikumara, S., Morris, R. G. M., and Korte, M. (2014). Competition between recently potentiated synaptic inputs reveals winner-take-all phase of synaptic tagging and capture. *PNAS*, 111(33):12217–12221.
- Salisbury, D. F. (2012). Finding the missing stimulus mismatch negativity (MMN): Emitted MMN to violations of an auditory gestalt. *Psychophysiology*, 49(4):544–548.
- Samsonovich, A. and McNaughton, B. L. (1997). Path integration and cognitive mapping in a continuous attractor neural network model. *The Journal of Neuroscience*, 17(15):5900–5920.
- SanMiguel, I., Saupe, K., and Schröger, E. (2013). I know what is missing here: electrophysiological prediction error signals elicited by omissions of predicted "what" but not "when". *Frontiers in Human Neuroscience*, 7(407).
- Schultz, W. and Dickinson, A. (2000). Neural coding of prediction errors. *Annual Review of Neuroscience*, 23:473–500.
- Schöner, G., Spencer, J., and the DFT Research Group (2016). *Dynamic Thinking: A Premier on Dynamic Field Theory*. Oxford University Press.
- Shastri, L. (1999). Advances in SHRUTI — a neurally motivated model of relational knowledge representation and rapid inference using temporal synchrony. *Applied Intelligence*, 11:79–108.
- Singer, W. (2007). Binding by synchrony. *Scholarpedia*, 2(12).
- Stachniss, C., Leonard, J. J., and Thrun, S. (2016). Simultaneous localization and mapping. In Siciliano, B. and Khatib, O., editors, *Springer Handbook of Robotics*, chapter 46, pages 1154–1171. Springer.
- Sutton, R. S. and Barto, A. G. (1981). Toward a modern theory of adaptive networks: Expectation and prediction. *Psychological Review*, 88(2):137–170.
- Sutton, R. S. and Barto, A. G. (1990). Time-derivative models of Pavlovian reinforcement. In Gabriel, M. and Moore, J., editors, *Learning and Computational Neuroscience: Foundations of Adaptive Networks*, chapter 12, pages 497–537. MIT Press.
- Taube, J. S. (2017). New building blocks for navigation. *Nature Neuroscience*, 20(2):131–133.

- Taube, J. S., Muller, R. U., and Ranck, J. B. (1990). Head-direction cells recorded from the postsubiculum in freely moving rats. I. description and quantitative analysis. *The Journal of Neuroscience*, 10(2):420–435.
- Treisman, A. (2002). The binding problem. *Current Opinion in Neurobiology*, 6(2):171–178.
- Urbanczik, R. and Senn, W. (2015). Learning by the dendritic prediction of somatic firing. *Neuron*, 81:521–528.
- Wacongne, C., Changeux, J. P., and Dehaene, S. (2012). A neuronal model of predictive coding accounting for the mismatch negativity. *Journal of Neuroscience*, 32(11):3665–3678.
- Wang, R. F. and Spelke, E. S. (2000). Updating egocentric representations in human navigation. *Cognition*, 77:215–250.
- Warren, W. H., Rothman, D. B., Schnapp, B. H., and Ericson, J. D. (2017). Wormholes in virtual space: From cognitive maps to cognitive graphs. *Cognition*, 166:152–163.
- Wu, X. E. and Mel, B. W. (2009). Capacity-enhancing synaptic learning rules in a medial temporal lobe online learning model. *Neuron*, 62:31–41.
- Zetsche, C., Wolter, J., and Schill, K. (2008). Sensorimotor representation and knowledge-based reasoning for spatial exploration and localisation. *Cognitive Processing*, 9:283–297.