The dynamics of learning behaviour:
A situated, embodied, and dynamical systems approach

Eduardo Izquierdo

Submitted for the degree of Doctor of Philosophy

University of Sussex, UK

Submitted: June, 2008; Revised: November, 2008.
Preface

All of the experiments, analysis and interpretation described in the thesis are my own and have been carried out during my doctoral studies. The majority of the results of these investigations have already been published before the submission of the thesis in peer-reviewed conference proceedings and an international journal.


Mathematica tools developed by Randall Beer were used for parts of the dynamical analysis. The software is called Dynamica and is available online from his website. Random number generators from Numerical Recipes (Press, Teukolsky, Vetterling, & Flannery, 2007) were used.
Acknowledgements

One of the rewards of writing this thesis is the opportunity it provides to thank the many friends and colleagues who have been so generous with the help they have given me. Many thanks to all those who have provided support throughout this journey.

To my supervisor, Inman Harvey, for his skepticism and encouragement to simplify. To Ezequiel Di Paolo, who also played the role of supervisor, for his enthusiasm and willingness to explore. To Randy Beer, who also offered much needed advise, for his rigour and attention to detail. I’m particularly grateful for their collective guidance. Each has a very different but complementary perspective on how to do science.

To my friends and colleagues in the CCNR for providing such a rich scientific environment. Particular thanks to Fernando Almeida e Costa, Matthew Bardeen, Xabier Barandiaran, Edgar Bermudez, Thomas Buhrmann, Peter Fine, Tom Froese, Hiroyuki Iizuka, Simon McGregor, and Marieke Rohde. Also to the previous CCNR generations from whom I learnt how to get started. Particular thanks to Andy Balaam, Hanneke De Jaegher, Chrisantha Fernando, Paul Graham, Alex Penn, Ian Macinnes, Lincoln Smith, Rob Vickerstaff, and Rachel Wood. Thanks to Christian Caterham for technical support.

To the people from Randy’s lab with whom I interacted with most: Eldan Goldenberg, Chad Seys, Brian Daniels, Mike Brady, and Paul Williams. Also to Hillel Chiel, Peter Todd, Luis Rocha, John Kruschke, William Timberlake, and Linda Smith for the opportunity to sit down and chat about my work while at Indiana University.

To friends for putting up with my ramblings. To all those I shared discussions with at scientific meetings. Particular thanks to Chris Buckley, Elio Tuci, and Takashi Ikegami. To the colleagues that provided feedback on parts of the thesis. Also to the anonymous reviewers that provided feedback on earlier versions of the chapters that were sent to conferences and journals. And to the thesis examiners, Seth Bullock and Andy Philippides, for their insightful contributions.

I’m grateful for having had the opportunity to carry out my doctoral studies in the UK in the first place. Special thanks to Inman for his help with funding via the Overseas Research Students Awards Scheme and a Graduate Teaching Assistantship. Also to Phil Husbands. Thanks to Chrisantha and Jon Rowe for the Research Fellow position.

A very special thanks to my family for their encouragement, guidance, love, and support. Particularly to my wife, Lilia. None of this would have been possible without her.
The dynamics of learning behaviour:  
A situated, embodied, and dynamical systems approach

Eduardo Izquierdo

Summary
The aim of this thesis is to better understand how learning behaviour can be produced from a situated, embodied, and dynamical agent. To this end, we employ evolutionary techniques to synthesize dynamical system neural controllers on tasks that require learning behaviour. We vary the experimental conditions on several dimensions. First, the stimulus to be remembered is in some tasks discrete and in other tasks continuous. Second, the level of embodiment and situatedness of the model agent varies from none, to minimal, to fully embedded. The scope of the tasks is also varied. We study Hebbian learning, associative learning, object discrimination, coping with visual inversion, imprinting, and coping with changes to body morphology.

No learning algorithm is provided to the internal dynamics of the agent. Evolution has to ‘come up’ with the mechanisms that can produce the learning behaviour on its own, starting from continuous-time recurrent neural-like components as its building block. We succeed to artificially evolve networks without synaptic plasticity on all of the tasks that we set out to study. For each of the tasks, we go into some depth trying to understand how the learning behaviour is produced by the most successful networks using dynamical systems theory.

The work in this thesis demonstrates the ability of small continuous-time recurrent neural networks to perform learning behaviour under a series of different conditions. All of the work on evolving agents that learn without synaptic plasticity has focused on tasks where the agent is required to act differently in a discrete number of distinct environments, in practice two. The result was agents that swapped between two modes of interaction. We extend the approach to having to act differently in a continuum of distinct environments. Also, all of the work has focused on the role of the internal dynamics of the agent in learning behaviour. By analysing networks evolved in abstract tasks as well as more ecological versions of those same tasks, we show how plasticity can switch from being generated purely as a result of the internal dynamics to arising from the full brain-body-environment interaction.

Submitted for the degree of Doctor of Philosophy
University of Sussex, UK
Submitted: June, 2008; Revised: November, 2008.
## Contents

Summary

Table of Contents

1 Introduction

1.1 Overview

1.2 Thesis organisation

1.3 Summary of contributions

2 A situated, embodied, and dynamical systems approach

2.1 From intelligence as computation to the dynamics of adaptive behaviour

2.2 Situatedness

2.3 Embodiment

2.4 Dynamical systems

2.4.1 Basic terminology

2.4.2 Challenges

3 Evolving dynamical systems models

3.1 Modelling internal dynamics

3.1.1 On the levels of modelling

3.1.2 Feedforward artificial neural networks

3.1.3 State in neural networks

3.1.4 Continuous-time recurrent ‘neural’ networks

3.1.5 Other interpretations of the CTRNN model

3.2 Evolutionary technique

3.2.1 Evolution, not design

3.2.2 Beginning to learn how to understand evolved systems

3.2.3 Background to evolutionary algorithms

3.2.4 Microbial GA

4 Learning without synaptic plasticity

4.1 Learning behaviour

4.1.1 Learning paradigms

4.2 Evolution and learning

4.2.1 Hinton and Nowlan

4.2.2 Peter M. Todd and Geoffrey T. Miller

4.3 Evolving dynamical system controllers that learn

4.3.1 Brian Yamauchi and Randall Beer
4.3.2 Elio Tuci, Matt Quinn, and Inman Harvey .................................. 46
4.3.3 Phattanard Phattanasri, Hillel Chiel, and Randall Beer ............... 48
4.4 Other related work ................................................................. 50
  4.4.1 Comparing the evolutionary performance of networks with and without synaptic plasticity for learning behaviour ............................. 50
  4.4.2 Unexpected role of synaptic plasticity in learning behaviour ........... 51
  4.4.3 Learning behaviour as changes in other network parameters besides synaptic strength .............................................................. 52
  4.4.4 Implementing known learning algorithms in fixed-weight networks ... 53
  4.4.5 Continuous attractor neural networks models ........................... 54
  4.4.6 Describing the activity of recurrent neural networks as state machines . 55

5 Hebbian learning using fixed-weight networks ................................ 57
  5.1 Introduction ............................................................................. 57
  5.2 Methods .................................................................................. 58
    5.2.1 Hebbian learning task .......................................................... 58
    5.2.2 Dynamical neural network .................................................. 59
    5.2.3 Evolutionary algorithm ....................................................... 60
    5.2.4 Fitness evaluation ............................................................... 60
    5.2.5 Incremental evolution .......................................................... 61
  5.3 Results .................................................................................... 61
    5.3.1 Role of synaptic plasticity in nonlinear controllers ................... 61
    5.3.2 Evolutionary success using small fixed-weight circuits .............. 62
    5.3.3 Best evolved 4-node circuit ............................................... 63
    5.3.4 Performance of best evolved circuit ..................................... 65
    5.3.5 Analysis of the multiple time-scale dynamics ......................... 67
    5.3.6 Dynamical analysis of best evolved circuit ............................ 69
    5.3.7 Evolved time parameters in successful circuits ....................... 70
  5.4 Discussion .............................................................................. 70

6 Associative learning on a continuum in disembodied neural networks ... 74
  6.1 Introduction ............................................................................. 74
  6.2 Methods .................................................................................. 75
    6.2.1 Temperature preference task .............................................. 75
    6.2.2 Agent and structure of a trial ............................................. 76
    6.2.3 Dynamical neural network .................................................. 78
    6.2.4 Evolutionary algorithm ....................................................... 79
    6.2.5 Fitness evaluation ............................................................... 79
    6.2.6 Incremental evolution .......................................................... 80
  6.3 Learning Discrete Stimuli .......................................................... 81
  6.4 Learning Continuous Stimuli ....................................................... 86
    6.4.1 Evolutionary Performance .................................................. 86
    6.4.2 Learning and Memory Performance ...................................... 86
6.5 The Dynamics of Continuous Learning ......................................................... 91
6.6 Discussion ................................................................................................... 97

7 History of interactions in reactive but situated agents ............................... 100
  7.1 Introduction ............................................................................................... 100
  7.2 Background .............................................................................................. 101
  7.3 Methods .................................................................................................... 102
  7.4 Orientation experiments with visual inversion ............................................. 103
  7.5 Categorical perception experiments .......................................................... 106
  7.6 Discussion ................................................................................................. 109

8 Discrete associative learning in embodied and situated agents ................. 112
  8.1 Introduction ............................................................................................... 112
  8.2 Related Work ............................................................................................ 113
  8.3 Methods .................................................................................................... 113
  8.4 Results ...................................................................................................... 116
    8.4.1 Evolutionary performance ................................................................. 116
    8.4.2 Behavioural analysis ........................................................................... 118
    8.4.3 Dynamics of the coupled agent-environment system ....................... 121
    8.4.4 Predictions from the dynamics ............................................................ 126
  8.5 Discussion ................................................................................................. 127

9 Learning from a continuum in embodied and situated agents ..................... 128
  9.1 Introduction ............................................................................................... 128
  9.2 Related work ............................................................................................. 130
  9.3 Methods .................................................................................................... 131
    9.3.1 Agent’s body and environment ......................................................... 131
    9.3.2 Agent’s controller ............................................................................... 132
    9.3.3 Task .................................................................................................... 133
    9.3.4 Artificial evolution ............................................................................. 133
    9.3.5 Fitness evaluation ............................................................................... 133
  9.4 Results ...................................................................................................... 134
    9.4.1 Evolutionary performance ................................................................. 134
    9.4.2 Learning performance ....................................................................... 134
    9.4.3 Behaviour .......................................................................................... 136
    9.4.4 The role of the agent’s situatedness versus the agent’s internal dynamics in remembering .............................................................. 138
  9.5 Discussion ................................................................................................. 142

10 Learning with your body ........................................................................... 144
  10.1 Introduction .............................................................................................. 144
  10.2 Related work ............................................................................................ 145
  10.3 Methods .................................................................................................... 146
    10.3.1 Walking task ...................................................................................... 146
## Contents

10.3.2 Chemotaxis task .................................................. 146
10.3.3 Neural model ...................................................... 147
10.3.4 Evolutionary algorithm .......................................... 148

10.4 Results .................................................................. 148
10.4.1 Evolutionary performance ...................................... 148
10.4.2 Performance and behaviours ................................. 150
10.4.3 Dynamics of the decoupled circuit ....................... 151
10.4.4 Brain-body-environment coupled dynamics ........... 154
10.4.5 Behaviour coordination: driven circuit .................. 154

10.5 Discussion .............................................................. 156

11 Concluding remarks .................................................. 159
11.1 Contributions ......................................................... 160
11.2 Discussion: Is it really learning? ............................... 162
11.3 Future work .......................................................... 163
11.4 Conclusions .......................................................... 166

Bibliography .................................................................. 167
# List of Figures

1.1 Perspectives on learning ......................................................... 2

3.1 Feedforward artificial neural network with multiple layers ............... 23
3.2 Continuous-time recurrent neural networks ................................... 25
3.3 Microbial genetic algorithm ...................................................... 31
3.4 Gaussian mutation of real-valued genotypes .................................. 33

4.1 Dimensions of work on evolving dynamical system controllers that ‘learn’ 44
4.2 One-dimensional navigation learning task by Yamauchi and Beer .......... 45
4.3 Associative learning arena by Tuci, Quinn, and Harvey ........................ 47
4.4 Disembodied associative learning task by Phattanasri, Chiel and Beer ...... 49
4.5 T-maze learning task by Chrisantha Fernando .................................. 52
4.6 A hand-designed neural network with fixed weights that implements a three-layer feed forward network with an in-built backpropagation algorithm ........... 54

5.1 The structure of an individual Hebbian learning trial ........................ 59
5.2 Effect of weight changes on the correlation between node activity .......... 62
5.3 Fitness of the population versus generations for the best evolved 4-node population .................................................. 63
5.4 A graphical depiction of the parameters of the best evolved network ...... 64
5.5 Example sequence trial ............................................................ 65
5.6 Performance of best circuit ......................................................... 66
5.7 Relearning ability .................................................................. 66
5.8 Robustness to time-delays .......................................................... 67
5.9 Interaction of multi-timescale dynamics during correlated and uncorrelated training .................................................. 68
5.10 3D projection of the circuit’s state space ....................................... 71
5.11 Evolved time-constants for the successful 4-node circuits ................. 72

6.1 Example three-node fully-connected network, including self-connections, for the associative learning of continuous stimuli task ............................. 77
6.2 Structure of an individual trial ..................................................... 78
6.3 Plot of best fitness vs. generation for the best-evolved 3-node circuit for the discrete learning task .................................................. 81
6.4 Activity of the best 3-node circuit on a typical trial sequence .............. 83
6.5 Strobed circuit dynamics in the best 3-node circuit at selected times during a trial .................................................. 84
6.6 Extracted finite state machine from best 3-node circuit ....................... 85
6.7 Generalization performance for the best-evolved 3-node circuit on the first testing phase after a pairing .................................................. 86
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.8</td>
<td>Plot of best fitness vs. generation for the best-evolved 5-node circuit for the continuous learning task</td>
<td>87</td>
</tr>
<tr>
<td>6.9</td>
<td>Activity of the best 5-node circuit on a typical trial sequence</td>
<td>88</td>
</tr>
<tr>
<td>6.10</td>
<td>Remembering performance of the best 5-node circuit</td>
<td>90</td>
</tr>
<tr>
<td>6.11</td>
<td>Learning performance of the best 5-node circuit</td>
<td>90</td>
</tr>
<tr>
<td>6.12</td>
<td>Robustness of the best 5-node circuit to changes in the time delay before the first, second, and third testing phase</td>
<td>91</td>
</tr>
<tr>
<td>6.13</td>
<td>Internal dynamics of the best-evolved 5-node circuit</td>
<td>92</td>
</tr>
<tr>
<td>6.14</td>
<td>Transitions between each of the states superimposed over the strobed states</td>
<td>94</td>
</tr>
<tr>
<td>6.15</td>
<td>A ‘continuous’ state machine embedded in the best 5-node circuit</td>
<td>95</td>
</tr>
<tr>
<td>6.16</td>
<td>Example sequence trial of a continuous state machine as a manifold of finite state machines</td>
<td>96</td>
</tr>
<tr>
<td>6.17</td>
<td>Pearson’s correlation coefficient between the activation state of each node in the circuit and the paired temperature</td>
<td>97</td>
</tr>
<tr>
<td>7.1</td>
<td>Minimally cognitive behaviour task setup</td>
<td>103</td>
</tr>
<tr>
<td>7.2</td>
<td>Trajectories and steady-state horizontal velocity fields for normal and inverted vision for the best-evolved agents</td>
<td>105</td>
</tr>
<tr>
<td>7.3</td>
<td>Performance for circle catching and bar avoidance</td>
<td>107</td>
</tr>
<tr>
<td>7.4</td>
<td>Demonstration of labelling and discrimination over a class of parameterised hybrid object that continuously interpolates between the bar and the circle</td>
<td>108</td>
</tr>
<tr>
<td>7.5</td>
<td>Trajectories and steady-state horizontal velocity fields for the best-evolved agent for circles and bars</td>
<td>109</td>
</tr>
<tr>
<td>7.6</td>
<td>Performance as a consequence from switching the object’s identity from a circle into a bar (and vice versa) at different times on the final decision to catch or avoid in the best-evolved agent</td>
<td>110</td>
</tr>
<tr>
<td>8.1</td>
<td>Agent’s architecture and the structure of an individual trial for the situated associative learning task</td>
<td>114</td>
</tr>
<tr>
<td>8.2</td>
<td>Evolutionary performance</td>
<td>116</td>
</tr>
<tr>
<td>8.3</td>
<td>Fitness vs. generation for the best-evolved 4-node population</td>
<td>117</td>
</tr>
<tr>
<td>8.4</td>
<td>Parameters of the best-evolved 4-node circuit</td>
<td>118</td>
</tr>
<tr>
<td>8.5</td>
<td>Memory performance of one environment and re-learning ability</td>
<td>119</td>
</tr>
<tr>
<td>8.6</td>
<td>Activity of the best 4-node circuit on a typical trial sequence</td>
<td>120</td>
</tr>
<tr>
<td>8.7</td>
<td>Experience-dependent behaviour as a function of initial temperature and heading</td>
<td>121</td>
</tr>
<tr>
<td>8.8</td>
<td>Bifurcation diagrams for the best-evolved circuit in the absence of food</td>
<td>122</td>
</tr>
<tr>
<td>8.9</td>
<td>Bifurcation diagrams for the best-evolved circuit in the presence of food</td>
<td>123</td>
</tr>
<tr>
<td>8.10</td>
<td>Trajectories of neural activity for neuron y4 superimposed over its bifurcation diagram</td>
<td>124</td>
</tr>
<tr>
<td>8.11</td>
<td>3D perspective of the equilibrium points depending on temperature and positive reinforcement</td>
<td>125</td>
</tr>
<tr>
<td>8.12</td>
<td>Interactive state machine extracted from the agent-environment interaction</td>
<td>126</td>
</tr>
</tbody>
</table>
9.1 Environment and agent experimental set-up for the minimally cognitive imprinting task ........................................... 132
9.2 Evolutionary performance ........................................... 134
9.3 Performance of best-evolved agent with 3-6 internodes ................. 135
9.4 Behaviour of the best-evolved agent for different sized circuits ........... 137
9.5 Role of situatedness versus internal dynamics in remembering performance for different sized circuits ........................................... 140

10.1 Experimental set-up for the walking and chemotaxis task ................. 147
10.2 Evolutionary performance ........................................... 149
10.3 Example trial for best-evolved circuit ........................................... 150
10.4 Bifurcation diagrams ........................................... 152
10.5 Example phase portraits ........................................... 153
10.6 Brain-body-environment coupled dynamics for the two different tasks: walking and chemotaxis ........................................... 155
10.7 System driven by sinusoidal waves of different frequencies ................. 157
Chapter 1

Introduction

This chapter provides an overview of the motivation and central argument advanced in this thesis. It also includes some of the details of how the thesis is organised. The last section comprises a summary of its original contributions.

1.1 Overview

The overarching motivation of this work is to further understand the dynamics underlying learning behaviour. To this end, we employ evolutionary techniques to synthesize dynamical system neural-like networks for tasks that require learning behaviour. Our interest is in how embodied, situated, and dynamic agents can generate adaptive behaviour, and how this can be explained using dynamical systems theory.

Learning is a behaviour. It is one of the most fundamental aspects of adaptive behaviour for living organisms. Although there is no one agreed definition, it generally refers to changes in the behaviour of the organism, such that its performance on some task improves with experience. We discuss what learning is in more depth in Chapter 4.

Living organisms show a variety of behaviours that are modulated by environmental conditions and experience. A major goal in biology is to elucidate the neural basis of experience-dependent adaptive behaviour. The majority of studies of the mechanisms underlying learning and memory have focused on the activity dependence of synaptic efficacy. In fact, synaptic plasticity is conventionally thought to be both necessary and sufficient to account for learning and memory. This is reflected in most models of learning. Ultimately, this has helped cement a perspective on learning where the behaviour-producing and the learning mechanisms are neatly separated into neural activity and its synaptic modulation, respectively (see Figure 1.1A).

This traditional perspective has become less useful in the face of more recent theoretical and experimental work in neuroscience suggesting that the separation of timescales between synaptic and neural activity is less straightforward. First, it is now well accepted that neuronal activity itself modifies not only synaptic efficacy but also the intrinsic membrane properties of neurons, and changes in these properties also serve to modify the circuit’s dynamics (Marder, Abbott, Turrigiano, Liu, & Golowasch, 1996). This includes long-term potentiation of intrinsic excitabil-
Figure 1.1: Perspectives on learning. [A] Traditional view with two neatly separate mechanisms: one that produces the behaviour (e.g., the neural activity) and another that sits on top of the first modifying it (e.g., synaptic plasticity). [B] A more integrated view of learning behaviour, where the internal activity of an agent is viewed as a multiple timescale dynamical system, with no a priori separation between the learning producing mechanisms and the behaviour producing mechanisms. [C] A view of learning behaviour that takes into consideration not only the processes that are going on ‘inside the brain’ of the agent, but also the agent’s body and its interaction with the environment seriously. Figure adapted and extended from Phattanasri et al. (2007).

In modelling work, learning behaviour and plasticity has also been traditionally associated with the modification of a neural network’s parameters, especially involving changes in the synaptic connections or weights of the neural network during the lifetime of the individual (Churchland & Sejnowski, 1992). These assumptions have been carried into the studies of learning in experiments using artificial evolution to synthesize neural networks that learn (e.g., Nolfi & Floreano, 1999; Floreano & Urzelai, 2000, 2001). However, one of the main strengths of such an approach is that it allows the experimenter to reduce the amount of built-in preconceptions in the model; allowing artificial evolution to ‘figure it out’ on its own instead. The concept of plasticity is basically no more or less than change on different timescales. Whereas in conventional feedforward neural networks it is traditionally fast changes of node-activations, and slow changes of synaptic-plasticity, there is no need for the differential timescales to be split up in that particular way. In the work presented in this thesis the plasticity is non-synaptic. In fact, in dynamical recurrent neural networks, activity occurring over multiple timescales is practically inevitable (see Figure 1.1B).

Furthermore, not only has most work on learning behaviour focused on the role of synaptic plasticity, but also the little work that has not, has still focused largely on processes going on ‘inside the brain’; ignoring the role of the body and environment entirely. Behaviour, however, is the
product of the interaction between the internal dynamics of an agent, its body, and its environment. Thus, studies of learning have only considered one third of the full story. Our work explores the role of the agent’s history of interaction with the objects in the environment in learning and the role of the feedback from the dynamics of the body in learning, in addition to the role of the internal dynamics in learning (see Figure 1.1C).

By ‘internal dynamics’ we would like to refer to any activity occurring within an agent. In biological organisms, although the nervous system is often considered to be the sole generator of internal activity relevant to adaptive behaviour, there are several other components that could similarly be relevant. We are interested in the broader set of possible mechanisms responsible for the generation and modification of behaviour. Thus, we attempt to introduce as few assumptions about what those mechanisms need to be. There are, nevertheless, three main assumptions that we hold about the networks that generate the internal dynamics of an agent: (a) they are composed of a heterogeneous set of dynamic components that interact non-linearly with each other, (b) they evolved, they were not designed, and (c) they didn’t evolve in a vacuum, they co-evolved simultaneously along with their bodies and their environments. We believe these are fundamental properties of the internal dynamics of any living organism. Importantly, we do not assume any one mechanism (in the form of an in-built parameter-changing algorithm) a priori. All we ask of the agent is to perform a learning task, whichever way it can.

The field where evolving and analysing dynamical neural networks in situated and embodied agents is often referred to as Evolutionary Robotics (ER). The main strength of the field has been in the production of existence proofs. The ER approach is in an ideal position to do this because it requires fewer assumptions than most other synthetic approaches. In this thesis, however, we offer more than just existence proofs. We also attempt to understand the mechanisms that evolve to generate the behaviour of interest. Even if the precise mechanisms discovered are not those found in any living organism, by attempting to understand the mechanisms that generate situated, embodied, and dynamical behaviour, we will be developing the conceptual and practical tools with which biologists can begin to understand the neural basis of embodied behaviour in real organisms. We develop this idea further in Chapter 3.

1.2 Thesis organisation

This work is organized as follows. The first two chapters provide the conceptual and methodological framework in which our investigations take place. Chapter 2 reviews the principles of the approach taken to tackle this work: a situated, embodied, and dynamical systems perspective. Chapter 3 discusses the methodological issues: the model for the internal dynamics of the agents and the evolutionary technique employed to synthesize them. Chapter 4 provides the background literature and a review of related work.

Chapters 5-10 provide the experimental basis for this work. The experimental results have already been published, mostly in conference proceedings (except for one journal article), in versions more or less similar to the form they appear in here. The links between chapters and publications will be made below. Except where noted, all of the work has been produced in collaboration with my supervisor, Inman Harvey. The experimental chapters are as self-contained as possible on purpose. However, it is only the story they thread collectively that advances the full argument
proposed in the previous section.

Chapter 5 shows that applying the Hebb rule on the weights of a nonlinear dynamical system controller does not necessarily lead to strengthening the correlation of firing. It then demonstrates that a dynamical system controller with fixed parameters can exhibit Hebbian learning behaviour. An explanation in terms of the structure and dynamics of the best-evolved system is given. Finally, the time-scales of all successfully evolved agents are analysed and generalisations about what is required for Hebbian learning to occur are given. The importance of this chapter is that this demonstration, because of its simplicity, lays the foundation for much of the rest of the work on ‘learning without synaptic plasticity’. This chapter is based on work in Izquierdo and Harvey (2007b).

Chapter 6 proposes an associative learning task inspired on thermal-preference behaviour observed in the nematode worm. The task is modelled at an abstract level: non-embodied and minimally situated. The dynamical system controller is required to learn the temperature associated with food and then remember it for further testing. Also, the task requires the system to re-learn new preferred temperatures when re-associated with food. The chapter is divided into two sections. First is the case when the stimuli to be remembered are discrete (n=2). The dynamics of the best-evolved agent are analysed and shown to instantiate a finite state machine. Second is the case when the stimuli are on a continuum. The dynamics of the best-evolved agent for this task, on the other hand, instantiate what we define as a continuous state machine (defined in Section 6.5). The chapter also provides a discussion of the differences. This chapter is based on work in collaboration with Randall Beer (Izquierdo, Harvey, & Beer, 2008).

The two experimental chapters so far (5 and 6) will have dealt with disembodied and minimally situated agents only. Chapter 7 shows the role of an agent’s situatedness for behaviour. The idea of the chapter is to serve as a transition into the rest of the thesis, which places more emphasis to the role of the agent’s history of interactions with its environment in learning and memory. Although the aim of the chapter is to provide a conceptual transition between abstract and situated behaviour, we will provide two concrete examples that illustrate the relevance of this step: an object discrimination and a visual inversion task. We show how a reactive controller can solve these tasks by exploiting its interactions with the environment. This chapter is based on work in collaboration with Ezequiel Di Paolo (Izquierdo & Di Paolo, 2005).

In Chapter 8 we study an extension to the discrete associative learning task in Chapter 6. For the abstract case presented there, the internal dynamics instantiated a finite state machine. This chapter explores the differences in required internal dynamics when the agent has the potential to be active in choosing its own stimuli by being fully embodied and situated. The result is learning behaviour that results from internal dynamics that do not instantiate any kind of computational automata. This chapter is based on work in Izquierdo and Harvey (2007a).

Chapter 9 provides the first set of experiments in learning on a continuum in an embodied model. The task is loosely abstracted from imprinting in birds. The chapter shows how memory behaviour can arise from the agent’s situatedness, provided the environment has sufficient invariants the agent can exploit. The results in this chapter suggest that plasticity can be a feature not only of the controller but of the agent-environment interaction itself. This chapter is based on work in Izquierdo and Harvey (2005, 2006).
In previous chapters, we will have studied the role of the agent’s internal dynamics and of its situatedness in learning. However, if we take seriously the idea that behaviour is the result of the interaction between the agent’s brain, its body, and its environment, then we are still missing one third of the story. In our last set of experiments (Chapter 10), we explore the role of the body in learning. Living organisms perform a broad range of different behaviours during their lifetime. It is important that these be coordinated such as to perform the appropriate one at the right time. This chapter extends previous work on evolving dynamical recurrent neural networks by synthesizing a single circuit that performs two qualitatively different behaviours: orientation to sensory stimuli and legged locomotion. We demonstrate that small fully interconnected networks can solve these two tasks without providing a priori structural modules, explicit neural learning mechanisms, or an external signal for when to switch between them. Dynamical systems analysis of the best-adapted circuit explains the agent’s ability to switch between the two behaviours from the interactions of the circuit’s neural dynamics, its body and environment. This chapter emphasizes the role of the feedback generated through the body in shaping behaviour and suggests it as an additional source of non-synaptic plasticity. This chapter is based on work in collaboration with Thomas Buhrmann (Izquierdo & Buhrmann, 2008).

In our last chapter we summarise our results and discuss some of the implications of our analysis of evolved dynamical system agents for understanding learning behaviour in living organisms.

1.3 Summary of contributions

This thesis makes a number of original contributions, which can be summarised as follows:

- We demonstrate that increasing the strength of the connection linking two model neurons does not always increase their firing correlation. We show that neural networks without synaptic plasticity can be evolved to perform Hebbian learning behaviour. Our experiments suggest that the smallest networks must have at least four components, one of which must be an order of magnitude slower than the rest. We provide a geometric explanation for the Hebbian behaviour, as switching between two basins of attraction whose shape determines the correlation.

- We demonstrate that neural networks without synaptic plasticity are not restricted to learning behaviour in tasks that require swapping between one of two modes of interaction. We show that they are also capable of learning, remembering, and recalling continuous stimuli. In other words, neural networks without synaptic plasticity can be evolved to modulate their own dynamics on a continuum. We illustrate this with agents constrained to use only their internal dynamics to remember. Our experiments suggest that successfully evolved networks must integrate components acting on three different timescales. We also show that these networks can remain sufficiently plastic to relearn new stimuli while forgetting previously learnt stimuli during their lifetime.

- We show that it is not possible to deduce the cognitive limitations of a situated agent from the limitations of its internal dynamics. We provide two examples of reactive controllers (i.e., without internal state) performing non-reactive behaviours (i.e., demanding some state) by exploiting their history of interactions with the objects in the environment. Although there is no such thing as a living organism with purely reactive internal dynamics, the extremity of the conditions in the examples suggest that, for situated and embodied agents, we must also consider a broader range of available sources of state than just their internal dynamics.
• We demonstrate that the agent’s history of interaction with the environment also has a role in learning behaviour. We illustrate that agents, if situated, can use their interaction with the environment as a source of memory to remember continuous stimuli in a learning task. We also show that this is often the case, suggesting that agents will exploit it when and if available. Furthermore, we show that several of the successful agents use, in fact, a combination of both internal dynamics and their history of interaction to remember. This provides a concrete example of how plasticity is a feature not of the controller but of the complete agent-environment system.

• Finally, we demonstrate that the body also has an important role to play in learning. We illustrate that neural networks, when embodied, can switch between performing two qualitatively different tasks, without synaptic plasticity, without behavioural modules to switch from, and without an external signal for when to switch. We show how the difference in the form of the feedback that is generated from the movements of the different bodies generates the appropriate two behaviours and the switching, when in interaction with an attractor landscape of the internal dynamics that does not change.
Chapter 2

A situated, embodied, and dynamical systems approach

In this chapter we introduce the conceptual framework that underlies this thesis: a situated, embodied, and dynamical systems approach. The first section places the framework in the context of the (still dominant) computational view of cognition. We then review what each of the terms in the title of the approach stand for, putting them into the context of the thesis.

2.1 From intelligence as computation to the dynamics of adaptive behaviour

In the study of cognition, the conceptual framework that we adopt will inevitably affect our understanding of it. This, of course, applies to science in general. Kuhn discusses how scientists will pay attention only to “problems defined by the conceptual and instrumental techniques already at hand” (Kuhn, 1979, p. 262). Beer describes how the conceptual framework influences our study of cognition in a recent article:

They provide a set of filters through which we view the world, influencing our choice of phenomena to study, the language in which we describe these phenomena, the questions we ask about them, and our interpretations of the answers we receive. (Beer, 2008, p. 2)

For the majority of its history, cognition has been thought of in the context of a computational framework. According to Michael Wheeler this framework can be traced back as far as Plato, where the notion of an “idea” can be considered a precursor to that of a “representation” (Wheeler, 2002). The main assumption behind the computational framework is that mind is best conceived as a computer, in the sense defined in computer science: as something functionally equivalent to a universal Turing machine. Harvey sketches the notion of a computation in the context of a calculator, a chess-playing program, and a robot:

- Take some input data, such as a set of numbers, or a chess position and the history that led up to it, or the sensory inputs to a robot or animal.
- Carry out a specific algorithm on the input data, until the algorithm halts (subject to the Halting Problem).
- Present the output data that resulted (e.g., the long division, the next chess move, or the agents next motor movement). (Harvey, 1997a, p. 686)
Beer summarises what a cognitive agent must be capable of, under the computational framework:

The idea that systems are intelligent to the extent that they can encode knowledge in symbolic representations which are then algorithmically manipulated so as to produce solutions to the problems that these systems encounter. (Beer, 2008, p. 2)

Classical artificial intelligence (AI) focused almost exclusively on this framework, which is why it is also referred to as symbolic AI. Also, interest was focused on “high level” human intelligence (e.g., rational thought, playing chess, abstract reasoning). For an introduction to the tools and language developed for this framework see Russell and Norvig (2002). While symbolic AI has been useful in the development of computer science applications and technologies, it has been less so with regards to its original motivation: understanding how adaptive behaviour can be produced in living organisms.

Concerns with the computational approach to cognition arose in the late 1970’s, with the work of Dreyfus (1972) and Haugeland (1978) criticising the direction that Artificial Intelligence was taking towards a detached-from-the-real-world and information-processing-oriented view of cognition. Their work was inspired on that of philosophers such as Heidegger, Merleau-Ponty, and Wittgenstein. Also crucial was the work of Maturana and Varela (1980) and Winograd and Flores (1986). They discussed the nature of biological organisms as mechanistic structure-determined systems, yet pointed to the fundamental differences between cognition and computation. For them, cognition was about being in the world and not about rational problem solving. In essence, these people challenged the dominant view of mind, helping us understand how cognition need not be based on the systematic manipulation of representations.

However, it wasn’t until some time later, during attempts to build “intelligent robots”, that their message began to sink in. Rodney Brooks made obvious the inappropriateness of symbolic AI when he attempted to develop robots for the real world – even for the simplest of behaviours. This led him to suggest a radical departure from the classical approach (Brooks, 1990, 1991a, 1991b). Several researchers since have echoed the failure of symbolic AI. Nowadays, symbolic AI is referred to as “Good Old Fashioned AI” (GOFAI) derogatorily, a term coined in Haugeland (1998). Andy Clark summarises GOFAI’s shortcoming in the first paragraph of Being there: Putting brain, body, and world together again:

Why are even the best of our “intelligent” artifacts still so unspeakably, terminally dumb? One possibility is that we simply misconstrued the nature of intelligence itself. We imagined mind as a kind of logical reasoning device coupled with a store of explicit data – a kind of combination logic machine and filing cabinet. (Clark, 1997, p. 1)

A major shift in perspective in the approach to understand the mechanisms that produce adaptive behaviour has taken place over the last two decades. Following Beer, we will refer to this

---

1We purposefully ignore Connectionism as a major shift because, although the central metaphor changed from a physical symbol system to a feedforward neural network, and the notion of a representation changed from symbolic to distributed, the computational understanding of cognition (as described above) remained unchallenged. In fact, all of the shifts described ahead will apply similarly as departures from Connectionism. See Rumelhart, Hinton, and McClelland (1986) for more details about this paradigm.
departure as the situated, embodied, and dynamical approach (SED). We mention the six changes that we believe have been most crucial. For other treatments of the transition see Brooks (1990), Varela, Thompson, and Rosch (1991), Beer (1995a), van Gelder (1995), Pfeifer (1996), Harvey (2000), Beer (2000).

1. There was a shift from studying isolated faculties (e.g., planning, face recognition, knowledge representation) to studying ‘complete’ systems: where “everything needed for behaving in the real world has to be there” (Pfeifer, 1996, p. 3).

2. There was a shift from focusing on “high level intelligence” towards a bottom-up approach: attempting to understand systems with ‘simple’ behaviours first, and working our way up gradually. Human level intelligence was now the goal for the distant future; not the starting point. As Brooks put it:

   [W]e will never understand how to decompose human level intelligence until we’ve had a lot of practice with simpler level intelligences. (Brooks, 1991b, p. 139).

3. There was a shift from studying abstract computational tasks performed in a ‘nervous system in a vacuum’ to behaviours performed by agents with a body and in a world (e.g., walking as opposed to calculating the centre of mass). We will discuss this aspect further in Sections 2.2 and 2.3.

4. There was also a shift from studying “universal problem solvers” to studying adaptive behaviours in the context of an agent’s evolutionary history and ecological niche. What may be adaptive behaviour for some organism and environment is likely to be unfit for a different scenario. The idea that “nothing in biology makes sense except in the light of evolution” (Dobzhansky, 1973) certainly applies to understanding an agent’s adaptive behaviour. We will discuss this aspect further in Section 3.2.

5. There was a shift from constraining the systems that produce behaviour to be of a computational nature towards allowing for a broader set of possibilities: dynamical systems.

   [A] situated agent receives a continuous stream of sensations from its environment and produces a continuous stream of actions which continuously depend both on its sensations as well as its own changing internal state. This suggests that the proper way to model an agent is as a dynamical system. (Beer, 1997, p. 260).

Furthermore, because dynamic models can include discrete-time events as a special case, computational systems are but a small subset of the range of possible systems that dynamical system models can instantiate. This shift is basically about leaving open the question: What type of dynamical system is a cognitive agent? Instead of assuming it to be a subset of computational ones. We will discuss this aspect further in Section 3.1.

6. There was a shift from using the language of computation (e.g., symbols, representations, processing) to understand adaptive behaviour towards using the language of dynamical systems theory (DST). This follows directly from the previous point and is what Timothy van Gelder referred to as the “knowledge hypothesis” (van Gelder, 1998). This last shift is undoubtedly the most controversial of all. We will discuss this aspect further in Section 2.4.

---

2Even though, as Beer points out, “it is not a genuine scientific hypothesis, at least not in the traditional sense of making an empirically falsifiable claim” (Beer, 1998, p. 630).
Remarkably, some of the shifts in perspective had already been taken seriously at some stage by a good proportion of the early Cyberneticians. For example, W. Grey Walter had taken seriously the study of complete embodied and embedded systems in the real world with his pair of phototactic “tortoises” (Walter, 1950). William Ashby was one of the first researchers to attempt to understand the mechanisms underlying adaptive behaviour using a dynamical systems perspective (Ashby, 1952). Their work, however, was largely ignored during the computational AI years. For more details about the role of Cybernetics in the history of the understanding of cognition see Boden (2006), Husbands, Holland, and Wheeler (2008).

It is likely that the majority of researchers of cognition currently acknowledge that at the “most basic levels” behaviour is the result of situated, embodied, and dynamic coupling between agent and environment. But still a large proportion of those researchers would argue that, as agents evolved to manage more complex scenarios, they developed the capacity to perform “computations in the brain” and other disembodied abilities. With the result that, for some organisms with some sufficient level of intelligence, embodied, situated and dynamic cognition is merely a way to reduce the “computational load”.

We will call such a stance the “on the fence” view, to refer to those who think that up to a certain level a situated, embodied and dynamic perspective is helpful in understanding the generation of behaviour, but past a certain threshold the computational framework is most useful. The difference between most cognitive scientists is where they draw their threshold and what they assume “internal representations in the brain” to be. For examples of this view see Clark (1997), Wheeler (2002). Their underlying motivation for being “on the fence” is that:

Minds may be essentially embodied and embedded and still depend crucially on brains which compute and represent. (Clark, 1997, p. 143).

Dreyfus criticises the “on the fence” view in his more recent work reviewing Michael Wheeler’s Reconstructing the cognitive world: The next step (Wheeler, 2002):

Heidegger’s and Merleau-Ponty’s understanding of embedded-embodied coping, therefore, is not that the mind is sometimes extended into the world but rather that, in our most basic way of being – i.e. as skillful copers – we are not minds at all but one with the world. (Dreyfus, 2007, pp. 254-255)

We agree. Furthermore, we believe it will be valuable to understand cognition as embodied, embedded, and dynamic, at all levels – without exceptions and without thresholds. Thus, whatever cognitivists might want to mean by “internal representations and computations in the brain” (Harvey, 2008), as long as they are being performed by an embodied, situated, and dynamical agent, then we believe an explanation in the mathematical language of dynamical systems theory will provide a necessary foundation. This is not a very different argument to what van Gelder offers to those “on the fence”:

Cognition can, in sophisticated cases, involve representation and sequential processing; but such phenomena are best understood as emerging from a dynamical substrate, rather than as constituting the basic level of cognitive performance (van Gelder, 1997, p. 439)
Thelen, Schöner, Scheier, and Smith (2001) offer a similar argument in their study of infant development using the “A not B” task:

If we can understand this particular infant task and its myriad contextual variations in terms of coupled dynamic processes, then the same kind of analysis can be applied to any task at any age. If we can show that “knowing” cannot be separated from perceiving, acting, and remembering, then these processes are always linked. There is no time and no task when such dynamics cease and some other mode of processing kicks in. Body and world remain ceaselessly melded together. (Thelen et al., 2001, p. 2)

Above all, the debate on whether computational or dynamical understandings to how adaptive behaviour is produced in interactions with their body and environment can only be resolved on a case by case basis. As Beer puts it: “There is nothing like studying birds or trying to build an airplane to cut to the heart of the debate about what can and cannot fly” (Beer, 1998, p. 630). There has been a considerable number of advances studying learning in real organisms, but we believe there are a number of limitations of working in vitro (discussed further in Chapters 3 and 4). In this thesis we put forward six experiments using computer simulations that attempt to cut to the heart of the debate about what can and cannot learn.

2.2 Situatedness

In order to understand how the situated, embodied, and dynamical systems approach departs from the classical computational paradigm, we will review each of the main components that compose it. We discuss what being “situated” refers to and how it relates to learning. We also discuss a set of related notions. But first, we review how it has been used in some of the seminal works:

[In traditional AI] the agents are not situated at all. Rather, they are given a problem, and they solve it. Then, they are given another problem and they solve it. They are not participating in a world as would agents in the usual sense. […] A situated agent must respond in a timely fashion to its inputs. Modeling the world completely under these conditions can be computationally challenging. But a world in which it is situated also provides some continuity to the agent. That continuity can be relied upon, so that the agent can use its perception of the world instead of an objective world model. […] The key idea from situatedness is: The world is its own best model. (Brooks, 1991a, p. 15)

The robots are situated in the world – they do not deal with abstract descriptions, but with the here and now of the world directly influencing the behavior of the system. (Brooks, 1991c, p. 1227)

The agents must be situated, i.e. the whole interaction with the environment must be controlled by the agent itself, i.e. the world must always be seen from the perspective of the agent. Moreover, the agent has to be able to bring in its own experience in dealing with the current situation. (Pfeifer, 1996, p. 5)

An agent is situated if it acquires information about its environment only through its sensors in interaction with the environment. A situated agent interacts with the world on its own, without an intervening human. (Pfeifer & Scheier, 1999, p. 90)
Situatedness. An agent’s immediate environment plays a central role in its behavior. This environment is not only a rich source of constraints and opportunities for the agent, but also a context that gives meaning to the agent’s actions. (Beer, 2008, p. 3)

All of the above definitions agree to a large extent on the same notion of situatedness as the embedding of an agent in an environment, with the ability to continuously act on it and therefore play a role in the sensory stimuli that it receives.

It is crucial to note that the acting part is equally as important as the sensing part of an agent’s situatedness. Several experiments on the distortion of the visual field in humans have demonstrated that re-adaptation is possible only on the cases where the subject engages actively in the task, with movement-generated sensory feedback (Taylor, 1962; Kohler, 1964). This is also the case for making sense of ‘new’ sensory stimulation, as shown in experiments with ‘sensory-substitution’ devices (Bach-y-Rita & Kercel, 2003). Another set of experiments that have demonstrated the importance of activity in situatedness are those of Held and Hein, where kittens develop while receiving visual stimulus generated by a other kittens. These experiments have shown that self-produced movement resulting in visual feedback is necessary for the development of visually guided behaviour (Held & Hein, 1963; Held, 1965).

One the earliest examples of the relevance of an agent’s situatedness came from Braitenberg’s reactive vehicles, which while very simple in design, were capable of (sufficiently unexpected) rich behaviour (Braitenberg, 1986). But while situatedness places an emphasis on the environment’s constant influence on the agent, it does not require that the agent’s behaviour be only determined by its present situation (i.e., a reactive system). The notion of situatedness implies that the actions of an agent are continuously co-determined by the agent’s environment as well as its own internal state. Thus, it is useful to clarify that situatedness does not imply behaviourism nor reactive agents; internal dynamics of any complexity can also be occurring in a situated agent.

Being in an environment that the agent can modify, means that an agent has more than just its own internal state as a resource to accomplish adaptive behaviour; it also has the full environment as an external state. Thus, the idea of the “extended mind” developed in Clark and Chalmers (1998) follows directly from an agent’s situatedness. One useful way to think about it is in the context of your daily tasks: writing sticky notes to remember a certain job that needs doing and placing them in an appropriate place where you will see them when you need to remember it.

The interactionist role that the environment and internal dynamics of an agent must play for the generation of adaptive behaviour is not very unlike the role that genes and environmental factors play in the development of the living organism. For a critical treatment of the nature-nurture debate, in a language that will resonate with the debate underlying the understanding of adaptive behaviour, we recommend reading Oyama (2000).

When the environment includes other agents, the relevance of situated activity becomes even more pressing because there are more interesting ways in which the agent can modify its environment. Social insects, for example, are known to collectively accomplish tasks which no one single creature could have built on its own (e.g., wall building and trail formation by ants, termite mound building, nectar source selection and colony thermoregulation in honey bees) (Camazine, Deneubourg, Franks, Sneyd, Theraulaz, & Bonabeau, 2001). They accomplish this by modifying their environment, in a process referred to as stigmergy (Grassé, 1959). Similarly, the use of
language allows humans to coordinate activities, accomplishing things collectively that no individual could do on its own. Hutchins (1995) provides a beautiful example of navigating a vessel in the context of the situatedness of the crew, where the instruments in the cockpit become part of this “collective cognizer”. There are also a number of studies in social psychology that have investigated how people remember things in close relationships or groups (e.g., Wegner & Wegner, 1995). They have shown that each person tends to specialise on remembering only a small subset of all things, as well as remembering who is likely to know about what (Wegner & Wegner, 1995). They call this transactive memory. The interesting point is that this memory only exists in the collective mind of the group, not in any one individual alone (Wegner & Wegner, 1995).

The level of situatedness in the models presented in this thesis varies. The model presented in the first of the experimental chapters (Chapter 5) is the least situated of all. The neural network is not in an environment as such: its activity has no repercussion on its environment and what it senses. The model presented in the second experimental chapter (Chapter 6) is situated, but only minimally so. It is situated because a part of the sensory input that it receives (i.e., the reinforcement) depends on the activity of the network. It is minimally so, because most of the sensory input that it receives (and the timing in which it receives it) is determined by the experimenter, and is not modifiable by the activity of the network. The rest of the experimental chapters (from 7 to 10) are fully situated. These are networks that move around in an environment, thus all of their sensations are co-determined by their activity.

2.3 Embodiment

The notion of embodiment is closely related to that of situatedness, but places an emphasis on the role of the agent’s body in cognition instead. Let us first see how embodiment is defined in previous works:

The robots have bodies and experience the world directly – their actions are part of a dynamic with the world and have immediate feedback on their own sensations. (Brooks, 1991c, p. 1227)

The agents must be embodied, i.e. they must be realized as a physical system capable of acting in the real world. Although simulation studies can be extremely helpful in designing agents, building them physically typically leads to surprising new insights. [...] Physical realization often facilitates solutions which might seem hard if considered only in an information processing context. An agent existing only in simulation would not be complete. (Pfeifer, 1996, p. 5)

Embodiment implies that the agent is continuously subjected to physical forces, energy dissipation, to damage, in general to any influence in the environment. On the one hand, this complicates matters considerably. On the other hand, this often leads to substantial simplifications, because advantage can be taken of the physics involved. (Pfeifer & Scheier, 1999, p. 91)

To say that cognition is embodied means that it arises from bodily interactions with the world. From this point of view, cognition depends on the kinds of experiences that come from having a body with particular perceptual and motor capabilities that are inseparably linked and that together form the matrix within which reasoning, memory, emotion, language and all other aspects of mental life are meshed. (Thelen et al., 2001, p. 1)
Physical embodiment. The uniquely physical aspects of an agent’s body are crucial to its behavior, including its material properties, the capabilities for action provided by the layout and characteristics of its degrees of freedom and actuators, the unique perspective provided by the particular layout and characteristics of its sensors, and the modes of sensorimotor interaction that the sensors and actuators collectively support. (Beer, 2008, p. 8)

All of these treatments of embodiment refer to a physical grounding of the agent in the real world. The original motivation for this is that it forces the designer to deal with all of the issues that living organisms have to deal with. This is in contrast to simulation models, where issues that may not seem initially relevant are likely to not be modeled at all. In other words, cheating becomes harder (although not impossible, as the real environment can be simplified to a large extent) in the real world.

As a result of this movement, researchers realised that the material properties and morphology of the body would often play a key role in the generation of behaviour. Let us consider object grasping with a hand as an example. The classical approach would involve a metallic or plastic hand (little importance would actually be given to the material), a sophisticated visual system that allows the system to recognise the shape of the object, and an even more sophisticated algorithm to compute the trajectory of the hand and the angles of each of its components. A different approach was taken recently in Yokoi, Hernandez Arieta, Katoh, Wenwei, Watanabe, and Maruishi (2004), building an artificial hand from elastic, flexible and deformable materials (e.g., elastic tendons, deformable finger tips). By exploiting the morphology and the materials, the hand can easily adapt to the object it is grasping without the need to ‘know’ beforehand its shape. The result is a simplified mechanism required for the generation of the behaviour. For more examples of “how the body shapes the way we think” see Pfeifer and Bongard (2006).

But, as is the case with situatedness, the relevance of embodiment is sometimes even much more unexpected when combined with evolutionary techniques. The idea is that, given the opportunity, artificial evolution will most likely “put to use” many of the nonlinearities offered by the agent’s continuous interaction with the environment through a dynamic morphology – all of which would be often ignored (or altogether avoided) by a human designer. We will touch on this point again in the next chapter when we discuss evolutionary techniques (Section 3.2).

In Ziemke (2001), a distinction between four different notions of embodiment is made: (1) “structural coupling” with the environment (borrowing Maturana and Varela (1980)’s term), (2) physical embodiment, (3) ‘organismoid’ embodiment, and (4) organismic embodiment. The first is analogous to what we refer to by situatedness. The second refers the embodiment in, for example, a physical robot. The third notion, organismoid embodiment, refers to implementing bodies that have a correspondence with the bodies of existing living organisms. This is similar to a notion that Beer also distinguishes. He refers to it as ‘biological embodiment’ (Beer, 2008). The fourth notion brings into account that the body must also be alive. Ziemke takes inspiration again from Maturana & Varela’s distinction between autopoietic and allopoietic systems (Maturana & Varela, 1980, 1992).

In the work presented in this thesis that deals with embodiment, we will only be simulating some aspects of physical embodiment. Conducting experiments with computer simulations rather than with real robots has the advantage of reducing evolutionary run times, removing costs of
building and maintenance, and removing safety risks. Most importantly, simulations allow us to tune the level of embodiment in our experiments from completely disembodied neural circuits to models where the role of the body plays a more significant role.

As suggested above, the level of embodiment in the models presented in this thesis varies. The models presented in the first two of the experimental chapters (Chapters 5 and 6) are not embodied. We can think of them as neural networks in a vacuum. All of the models that follow (Chapters 7, 8, 9, and 10) are embodied. These neural networks have effector components that cause the body to move around in the environment. However, it is only in the very last experimental chapter (Chapter 10) that the embodiment of the model is the subject of study itself.

As a note of caution with regards to terms, “embodied cognition” is often used as an umbrella term to encapsulate a range of different ideas (often related, but not necessarily so) to the notion of embodiment portrayed here. Margaret Wilson, for example, discusses six different views within “embodied cognition”: (1) cognition is situated; (2) cognition is time-pressured; (3) we off-load cognitive work onto the environment; (4) the environment is part of the cognitive system; (5) cognition is for action; (6) offline cognition is body based (Wilson, 2002). While the term is short and catchy, the reason for the caution is that many of those views can, at least theoretically, be considered in a system that is not embodied. In fact, only the last view is directly related to the role of the body in cognition.

2.4 Dynamical systems: a language and tool to investigate adaptive behaviour

A major difference between the view of cognition as computation and cognition as a broader type of system (i.e., dynamical system) is the role of time and timing. Wheeler explains the essence of temporality (or lack of it) in the computational framework in the following passage:

> Whilst the computational architectures proposed within computational cognitive science require that inner events happen in the right order, and (in theory) fast enough to get a job done, there are, in general, no constraints on how long each operation within the overall cognitive process takes, or on how long the gaps between the individual operations are. Moreover, the transition events that characterize those inner operations are not related in any systematic way to the real-time dynamics of either neural biochemical processes, non-neural bodily events, or environmental phenomena. (Wheeler, 2002, p. 345)

All behaviours require sensations and actions. Slowing down or speeding up the movements or sensations, or any other deformation to the timings of these components, would be likely to result in the breakdown of behaviour. Thus, getting the timing right is critical to the success of any living organism; unlike the success of a computation. There is no reason to think that this is any less the case for “higher levels” of cognitive activity, as all cognitive activity requires behaviour. In fact, the role of temporality has been central to the work of several researchers of cognition at such level (Bergson, 1950; Heidegger, 1962; Merleau-Ponty, 1962; Husserl, 1964; Gibson, 1979). This is in stark contrast to computation, where all that matters is the order of execution of a sequence of events. Beer contrasts the role of temporality in computational systems and living organisms:

> Far from being inessential details, issues of rate and timing fundamentally matter to an embodied agent. Unlike computing the $n$th digit of $\pi$, which is always the same no
matter when the computation delivers it, for an embodied agent time can make all the difference between an adaptive behavior and a suicidal one. (Beer, 1997, p. 258)

As argued at the beginning of this chapter, the tools and language that we use to study cognition will influence our understanding of it. Beer has been one of the strongest proponents of applying the language of dynamical systems to study adaptive behaviour, and his argument has been:

If timing, and changes over time, are the fundamental features of adaptive behavior that we wish to understand, then we need theoretical concepts and formalism that can do justice to this dynamical nature. Fortunately, there is a well-developed body of mathematics, dynamical systems theory, whose principal concern is precisely the description and analysis of systems whose behavior unfolds over time. (Beer, 1997, p. 259)

If we agree that cognition requires behaviour, and that behaviour is the result of the temporal interaction of brain, body and environment, then not adopting the tools that have been developed to understand such systems would imply either: (a) that we wish to ignore changes over time and timing in adaptive behaviour, which is basically returning to the computational framework; or (b) that we wish to develop a new body of science to deal with systems that change over time (or to extend an existing one so that it deals with time and timing). This thesis will not develop either of those two options any further. We will rely instead on the use of dynamical systems theory as a body of mathematics that is well suited to understanding systems that unfold over time. Its application to understanding situated, embodied, and dynamical agents, however, is still very much in development. We discuss some of the challenges that we are faced with further ahead.

One obvious advantage that might be worth pointing out is that the language of dynamical systems provides a unified framework with which to talk about events occurring in the brain, body and environment at different levels (e.g., body movements, mental phenomena, brain dynamics, developmental processes, or environmental events). Thelen et al. make this point with their work:

We cast the mental events involved in perception, planning, deciding, and remembering in the analogic language of dynamics. [...] Finding a common language for behavior, body, and brain is a first step for banishing the specter of dualism once and for all. (Thelen et al., 2001, p. 2)

Dynamical systems theory has been used recently at many different levels to study living organisms. It has been used to study development (Thelen & Smith, 2006), biological circuits (Brezina, Orekhova, & Weiss, 2003; Ermentrout, 2003), neuroscience (Izhikevich, 2007), the role of chaos in the brain (Skarda & Freeman, 1990; Tsuda, 2001). But dynamical system metaphors had been used previously (Ashby, 1952; Thomas & D’Ari, 1990). Conrad Waddington, for example, described embryonic change in the language of attractors, bifurcations, open systems, stability, catastrophes, and chaos (Waddington, 1977).

2.4.1 Basic terminology
In this section we provide an intuitive picture of the basic terminology used in dynamical systems theory. It will be the minimum background required to understand the analysis in the chapters to
follow. It will be, nevertheless, a necessarily incomplete treatment. For comprehensive introduc-

A dynamical system refers to a system that changes in time. A dynamical system is char-
acterized by a set of state variables and a dynamical law that governs how the values of those
variables change with time. Mathematically, they can be described by differential equations when
time is continuous or by difference equations when time is discrete. As our interest is in adap-
tive behaviour occurring in real time, we will confine our attention to continuous-time dynamical
systems.

A dynamical system can be linear or nonlinear depending on how the dynamical law acts
on the state variables. If all of the state variables in the system changes linearly with respect to
the rest (i.e., rates of change which include first power interactions only) the system is said to be
linear; otherwise the system is nonlinear. The techniques to study linear systems are different to
that of nonlinear ones. Linear systems can be broken down into parts, each part solved separately,
and finally recombined to get the answer. The idea is that the net response generated by two or
more independent stimuli is the sum of the responses that would have been generated by each
stimulus individually. Linear systems, however, are not very common in the real world; most of
the interactions in the everyday life of living organisms are actually nonlinear. The superposition
principle (i.e., dividing the problem, solving the parts, and then combing the results to get an
answer for the full system) fails dramatically. Steven Strogatz gives an amusing everyday life
example: “if you listen to your two favorite songs at the same time, you won’t get double the
pleasure!” (Strogatz, 1994).

Nonlinear dynamical systems are very difficult (and often impossible) to solve analytically.
However, geometrical reasoning can help us understand the system without actually solving it.
Thus, pictures can often be more helpful than formulae for analyzing embodied, situated and
dynamical agents. The dynamical system analysis in this thesis is fully geometrical.

For any system under study, the choice of which components vary over time and which don’t
is arbitrary. Components that vary are referred to as the variables of the system; the rest are called
parameters. How a variable changes over time will depend on a subset of all state variables in the
system and some set of parameters, as given by the dynamical law.

The set of all possible values of the state variables constitutes the system’s state space. The
dynamical law defines a vector field on the state space. Starting from some initial state, the se-
quence of states generated by the action of the dynamical law is called a trajectory of the system.
The set of all such trajectories through every point in the state space is called the flow. In the
qualitative theory of dynamical systems one is usually interested in the geometrical or topological
structure of the entire flow (Hale & Kocak, 1991). A picture which shows all of the qualitatively
different trajectories of the system is called a phase-portrait (Strogatz, 1994).

The long-term behaviour of a dynamical system is usually of interest as well. Regardless of the
initial state, a dynamical system will always be found near an attractor after transients have passed.
The state of some systems will converge to limit sets. A limit set is a set of points that is invariant
with respect to the dynamical law. An attractor is a stable limit set. It has the property that all
trajectories passing through all nearby states converge to it. The set of initial states that converge
to a given attractor is termed its basin of attraction. The parts of the trajectories that do not lie on
the attractor itself are called transients. Repellors are limit sets that are unstable: any perturbation to the state of the system causes it to leave that limit set. Due to their instability, repellors can only be observed by starting a dynamical system on a repellor and then never perturbing it. Real-world systems are noisy and thus they could never be found on a repellor.

The dynamical behaviour of one-dimensional dynamical systems (systems with one state variable) is very limited. Trajectories are forced to move monotonically or remain constant (Strogatz, 1994). The limit sets are confined to fixed points: attractors (also called stable fixed points) and repellors (also called unstable fixed points).

In higher-dimensional systems, the movement of trajectories can be richer. Thus, a wider range of dynamical behaviour becomes possible. Fixed points can still exist, but even these can become more interesting, depending on how the trajectories approach or repel from the equilibrium point. For example, the system could approach a stable point with a spiral or with a straight line. Also, other types of stability exist. Saddle-nodes, for example, are like fixed points in some dimension and like repellors in another. Limit cycles or orbits also become possible in multidimensional systems, where the state of the system ‘cycles’ through a set of states. There are many different kinds of limit cycles, they can be, for example, periodic or chaotic, stable, unstable, or sometimes even half-stable (Strogatz, 1994). Limit cycles are important, among other reasons, because they represent systems that oscillate even in the absence of external input.

Another fundamental idea from dynamical systems theory: parameter space. A point in parameter space reflects a specific combination of all parameter values and thus to a fixed set of basins of attraction and attractors in the corresponding state space of the dynamical system or phase portrait. Most dynamical systems are structurally stable, that is, for most parameter settings; small changes in the parameter values will produce small changes in the flow. Limit sets and basins of attraction may deform and move around a bit, but the new flow will be qualitatively similar to the old one. However, at certain parameter values, dynamical systems can become structurally unstable, so that even infinitesimal changes in parameter values can cause drastic changes in the flow, producing phase portraits that are qualitatively different from the original. For example, a basin may contract to nothing, or a new basin might appear. These qualitative changes in the types of limit sets are called bifurcations (Hale & Kocak, 1991).

Bifurcation diagrams refer to the phenomenon of a system exhibiting qualitatively different dynamical behaviour as a parameter is varied. As is pointed out in Wiggins (2003), the phrase “as a parameter is varied” deserves careful consideration, as parameters are by definition constant. The idea is that dynamical systems having parameters that change in time and that pass through bifurcation values often exhibit behaviour that is very different from the analogous situation where the parameters are constant, even if the parameter varies very slowly (e.g., Wiggins, 2003, Ch. 22). What does it mean for a system to exhibit qualitatively different behaviour? The phrase is a bit vague. And while it can be made more precise for small dimensional systems (i.e., similar topological structure), the term becomes more ambiguous for higher dimensional phase spaces (Wiggins, 2003). The values of parameters at which bifurcations occur therefore divide parameter space into disjoint volumes.
2.4.2 Challenges in understanding situated, embodied and dynamical agents

Understanding how internal state shapes behaviour is the major issue in the dynamics of cognition. The internal dynamics that operate within living organisms are composed of a complex and heterogeneous network of components interacting nonlinearly. The activity of each of the components changes over time as a function of some of the other components in the network. What’s more important, these networks are embedded in an environment: changes in the activity of the network are likely to change the environment for the agent, this changes their sensations, which in turn influences their internal activity, and so on. We present in this section what we think are the three key challenges in understanding situated, embodied, and dynamical agents.

Nonautonomous systems

The organism affects the environment, and the environment affects the organism. There is continuous feedback between agent and environment. Autonomous dynamical systems do not include any explicit time dependence. Time-dependent systems are called nonautonomous. The type of system that is embedded in an environment, as we have just described it, is a nonautonomous system. More generally, whenever the experimenter has to decompose a system into smaller subsets, each of the resulting systems will be nonautonomous. In the case of an agent-environment system, if we are interested in the dynamics of the agent, then any agent that has sensors that provide time-varying signals from its environment is nonautonomous.

Almost all of the mathematics in dynamical systems theory has been geared towards the study of autonomous systems. There are, nevertheless, some tricks that can be used to transform a nonautonomous system into an autonomous one. We can, for example, reconsider the system by including time as an extra dimension. Any n-dimensional time-dependent system is a special case of an (n + 1)-dimensional one (Strogatz, 1994). In chapters 8 and 10 we go into some depth in analysing the dynamics of agents that are closely embedded in their environment.

Multitimescale systems

Similarly, the majority of the mathematics in dynamical systems theory deals with systems that operate on the same or similar range of timescales. When a system has activity occurring over different timescales (fast and slow components), then the slow subset of components can be thought of as parameters to the fast subset. In basically all of the systems that we study in this thesis (except for the networks in chapter 7), the components of the network operate on different timescales. This is not entirely surprising as learning behaviour is closely related to systems operating on different timescales, although it is not always necessary.

The idea of variables acting as parameters to a subpart of the system is not entirely novel. W. Ross Ashby had proposed a step-mechanism acting as a parameter to the system as the key mechanism that could explain adaptivity in living organisms, which he exemplified in his ultrastable machine: the Homeostat (Ashby, 1952). For example, he defines an ultrastable system as:

Two systems of continuous variables (that we called ‘environment’ and ‘reacting part’) interact, so that a primary feedback (through complex sensory and motor channels) exists between them. Another feedback, working intermittently and at a much slower order of speed, goes from the environment to certain continuous variables which in their turn affect some step-mechanisms change value when and only when these variables pass outside given limits. The step-mechanisms affect the reacting
part; by acting as parameters to it they determine how it shall react to the environment. (Ashby, 1952, p. 98)

**Transients**

Eric Beinhocker tells a joke about two economists walking down the street: one young and the other older. The young economist looks down and says that he sees a 20 dollar bill. The more experienced economist replies: nonsense! If there had been a 20 dollar bill somebody would have already picked it up (Beinhocker, 2006, p.53). He uses it to point out how traditional models in economy treat the market as in a state of equilibrium - mainly to make the maths easier to solve. When transients are considered, however, we have an altogether different system. One in which at some point there are indeed 20 dollar bills on the floor. Asymptotically this is sufficiently unlikely as to be discarded entirely, which is why the older economist doesn’t believe the younger one. Similar to models in economy, models of cognition usually ignore transients as well. And also similar to financial markets, all of the interesting stuff in cognition actually happens in the transients; not in the state of equilibrium.

When studying a system with multiple timescales it is inevitable that the transients will play an important role in the generation of behaviour, and not only the system’s attractor landscape. Also when dealing with time-dependent systems, as the dynamical behaviour of the circuit is constantly changing, the actual circuit often never arrives at its asymptotic behaviour. In both of these scenarios, the system is usually not in equilibrium, but in a transient towards it.

All of the networks analysed in this thesis make heavy use of transients. In one case purely because of its multiple timescales (as in Chapter 5), in another case purely from its situatedness (as in Chapter 7), but more often due to a mixture of both. This is the case even though most of our experiments have been inspired from animal learning theory, which often are deliberately arranged to avoid continuous feedback and transient effects.
In this chapter we introduce and discuss in some detail the motivation for our use of: (a) continuous-time recurrent ‘neural’ networks as models of the internal dynamics and (b) evolutionary algorithms to synthesize systems that exhibit the adaptive behaviour of interest.

3.1 Modelling internal dynamics

3.1.1 On the levels of modelling

One must be careful to model the system at an appropriate level of abstraction. A frequent mistake is to assume that a more detailed model is necessarily superior. Too often criticisms are voiced that a certain model lacks some set of features and are thus irrelevant for understanding the nervous system. Taken to the extreme, only a fully detailed replica of the nervous system will be satisfactory. This would be, of course, useless as a model because it will be as difficult to understand as what it is attempting to model (Koch & Segev, 1989). Beer explains a similar idea by drawing from an analogy to Galileo’s frictionless planes:

Simpler idealized models are an indispensable tool in the early theoretical development of the field. [...] It was only with Galileo’s consideration of such highly idealized physical situations as frictionless planes that theoretical physics in the modern sense of the word became possible for the first time. [...] Thus, we need the autonomous agent equivalent of frictionless planes. We need to identify the simplest model agents that can truly be said to be behaving adaptively, perceiving their environment, making choices, learning, etc., even if these models do not realistically represent any particular animal or robot. (Beer, 1997, p. 269)

In Di Paolo, Noble, and Bullock (2000), a related distinction between “realistic simulacra” and “opaque thought experiments” is made. They summarise their view saying that “although simulations can never substitute for empirical data collection, they are valuable tools for re-organising and probing the internal consistency of a theoretical position” (Di Paolo et al., 2000, p. 497). They suggest Hinton and Nowlan’s model illustrating the guiding effects of learning for evolution (discussed further in Section 4.2.1) as an example that helped change the attitudes of theoretical biologists towards an already known phenomena. They also argue these are opaque thought experiments because the simulation models are often too complex to be easily understood.
Given that our simulation models will be purposefully idealised, what conclusions can be drawn from them? We believe there are two complementary uses of simulation models. First, simulation models have the ability to question existing theoretical frameworks by providing existence proofs. Harvey and others at Sussex have discussed the generation of existence proofs using Evolutionary Robotics (ER) in some detail:

We shall have demonstrated cognitive capacity or phenomenon X under clearly specified conditions and constraints, hence showing that these provide sufficient conditions for X. Since the underlying philosophy of ER is typically to find minimal conditions for phenomenon X, often the target of the experiment may be some alternative theory to the effect that phenomenon X necessarily requires condition Y. An ER experiment may enable us to refute this by demonstrating an existence proof for X without Y. (Harvey, Di Paolo, Wood, Quinn, & Tuci, 2005, p. 84)

We believe a second use of simulation models is to provide training examples where we can begin to develop the mindset for understanding situated, embodied, and dynamical agents, regardless of how complex they are. Although simulation models that use evolutionary techniques and dynamical system agents are in an ideal position to provide existence proofs because they require fewer assumptions than most other synthetic approaches, attempts to shed light on how the evolved dynamical system agent generates the behaviour of interest is also invaluable. Even if the precise mechanisms discovered are either: (a) not those found in certain or any living organism, or (b) cannot be easily linked to currently known biological processes, by beginning to learn how to analyse the often complex black box that generates situated, embodied, and dynamical behaviour, simulation models will inevitably sharpen our intuitions, mindset, and tools of analysis. These are likely to help experimental biologists tackle the complexities of real organisms behaving in real environments.

Therefore, the simulation models developed in this thesis hope to act as both: existence proofs and training examples. To this end, our models will be sufficiently complex to produce the adaptive behaviour of interest, but also minimal. Making them as simple as possible will help us test the boundaries of our current understanding of learning behaviour.

3.1.2 Feedforward artificial neural networks

Artificial neural networks (ANNs) are mathematical models of a set of interconnected units or nodes whose function is loosely based on real neurons. These were first developed by McCulloch and Pitts in a seminal paper written during the early Cybernetic days (McCulloch & Pitts, 1943). They were later generalized as the perceptron in Rosenblatt (1958). After failure to train multiple layer networks research in ANNs virtually stopped for over two decades, giving way to the dominance of GOFAI research (Russell & Norvig, 2002, for an overview). It was the development of the backpropagation algorithm by Paul Werbos with further refinements from David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams that led to the renaissance of neural network research in the form of connectionism(1) (Werbos, 1974; Rumelhart et al., 1986; Rumelhart, Hinton, & Williams, 1988).

---

(1) For the fascinating history behind neural network research in the context of cognitive science see Chapter 4 (mainly Sections 4.iii and 4.iv) and Chapter 12 from Boden’s in-depth work (Boden, 2006)
3.1. Modelling internal dynamics

The conventional ANN is feedforward (see Figure 3.1). The flow of activity is only in one direction: from the input nodes, through the hidden nodes (if any), to the output nodes. There are many other types of networks that have derived from the classical model (e.g., radial basis function networks, echo state networks, spiking neural networks). For an introduction to conventional artificial neural networks, the perceptron, backpropagation and some of their applications see Haykin (1998).

For the purpose of this thesis, all that is relevant about feedforward artificial neural networks is that they are (conventionally) reactive systems. When input is presented to the first layer, this is propagated forward to the output layer. As soon as the input disappears, all of the components of the network (including the output neurons) will return to their default state. Thus, one way to introduce ‘learning’ behaviour into such a reactive network was to modify some of its parameters (e.g., strengths of the connections between neurons) at a different (much slower) timescale. Changes to parameters modify the ‘default’ activity of the network. This, consequently, led to a notion of memory as a “distributed internal representation in the brain” in the form of a pattern of activations across the nodes (Hinton, McClelland, & Rumelhart, 1986). Given that changes to the weights of the connections would produce different patterns of activations, memory was understood to be ‘stored in the weights’. While this was true for conventional feedforward ANNs because of their reactive constraints, this view has been erroneously applied to networks with state. As we will see in the next sections, this is not the only way to modify the system’s behaviour in situated, embodied, and dynamical networks.

3.1.3 State in neural networks

Living organisms are dynamical systems with internal state; they are not reactive systems. Taken to the extreme, all physical objects will at least have inertia from their mass. Thus, the present behaviour of any real system will depend (to a larger or smaller extent) on its history. The internal dynamics of living organisms are, however, far richer than mere inertia. Their dynamics are instantiated from the complex network of components that comprise them (e.g., gene regulatory

Although purely reactive agents do not exist in the real world, they are a useful abstraction to compare state-full agents with. While a reactive agent is at the ‘mercy’ of its environment, an agent with internal state can: initiate behaviour independently of its immediate sensations, respond differently to identical sensory stimuli at different times, organize its behaviour in anticipation of future events, and modify its future behaviour based on its history of interaction (Beer, 2006).

Unlike conventional feedforward artificial neural networks, the internal dynamics of living organisms have three important properties: they have recurrent connections (the components inside cells and brains are messy and they interact with many other components, hardly ever in a simple feedforward fashion), they show activity over multiple timescales (with certain components acting in the timescale of milliseconds, while others act over minutes or hours), and they are always embodied and situated.

We will refer to such systems as embedded dynamical networks for short. Where can state arise from these systems? We can categorise the sources of state as coming from either: (a) their internal activity over multiple timescales, or (b) their embodiment and situatedness. Examining the first source in more detail, we can see there are several ways in which a dynamical network can have activity over multiple timescales. First, the network can have components whose intrinsic timescale is different, with some parts acting slower than others. Second, even if the intrinsic timescale of all components in the network is the same, activity over multiple timescales can arise from the recurrency alone. We can examine further the different ways in which a network with components with intrinsic different timescales can have state. The first is the addition of ‘learning rules’ that allow for the parameters of the network (e.g., strength of the weights, threshold of the neuron) to vary slower than the activity of the neurons in the network. There are many of these rules (e.g., Widrow & Hoff, 1960; Brindley, 1967; Stent, 1973; Zador, Koch, & Brown, 1990; Brown, Kairiss, & Keenan, 1990; Baxter & Byrne, 1993; Rao & Sejnowski, 2001). The second is to allow for the activity of the neurons themselves to have their own timescale of activity.

Out of all the possible ways in which a system can have state, and thus learn, only the option of adding ‘learning rules’ has been considered seriously in the literature. In this thesis we consider all of the other options (and not the traditional one) for the first time.

### 3.1.4 Continuous-time recurrent ‘neural’ networks

We use continuous-time recurrent neural networks (CTRNNs) as a model of the agent’s internal dynamics. CTRNNs are a special case of the general class of additive neural network models (Grossberg, 1988). CTRNNs are essentially continuous Hopfield neurons with the difference that self-connections are allowed and no constraints on the symmetry of the connections are made (Hopfield, 1984). For an introduction to Hopfield networks see Michel, Farrell, and Porod (1989) or Hertz, Palmer, and Krogh (1991). Each component in the network is governed by the following state equation (Beer, 1995b):

\[
\tau_i \dot{y}_i = -y_i + \sum_{j=1}^{N} w_{ji} \sigma (y_j + \theta_j) + I
\]  

(3.1)

The overdot denotes differentiation with respect to time, \(t\). The standard interpretation of this model is as follows: \(y_i\) represents the mean membrane potential of the \(i^{th}\) neuron, \(\tau_i\) represents...
3.1. Modelling internal dynamics

its membrane time constant, $\theta_i$ represents its threshold or bias, $I_i$ represents an external input, $\sigma(x) = 1/(1 + e^{-x})$ is the standard logistic activation function (see Figure 3.2B) and represents the neuron’s mean firing rate, the weight $w_{ij}$ for $i \neq j$ represents a synaptic connection from neuron $i$ to neuron $j$, and the self-interaction $w_{ii}$ represents a simple active conductance.

There are a number of reasons for employing CTRNNs to model the internal dynamics of our agents:

1. They are arguably one of the simplest nonlinear continuous-time dynamical neuron models.

2. The timescale of activity of each component is allowed to vary within any range. Also, interactions between any two components are allowed, making recurrent networks possible. We already discussed the relevance of this in the previous section.

3. They are universal approximators of smooth dynamics (Funahashi & Nakamura, 1993; Kimura & Nakano, 1998). This is an important point. What it means is that a CTRNN, given enough components, can approximate any particular dynamic with arbitrary precision. They can, for example, resemble the activity of a spiking neural network, or a set of chemicals interacting in a soup.

4. There are several ways to interpret them. They can be linked to biological processes (e.g., mean firing-rate model, nonspiking neuron) or interpreted as more general dynamical system models. We discuss this further in the next section.

5. They have been successfully evolved on a range of tasks requiring adaptive behaviour (e.g., Beer, 1996; Harvey et al., 2005). We discuss some of the more relevant examples dealing with learning behaviour in the next chapter.

6. Their dynamics and parameter space have been studied in some depth (Beer, 1995b, 2007). We discuss this briefly next.

The dynamics of the CTRNN model for small networks was first studied in some depth in Beer (1995b), providing a complete description of the possible dynamical behaviour and bifurcations

---

2This is only true if we choose to interpret the activity of spiking neurons as a continuous signal over time. It is important to note that this is not always how they are treated. In fact, because what is known about spiking neurons comes from extracellular recordings that detect when a neuron is firing, they have been studied in a digital on or off fashion (Marder, 2006). Nevertheless, recent studies by Alle and Geiger (2006) and Shu, Hasenstaub, Duque, Yu, and McCormick (2006) have shown that vertebrate neurons may have an analogue aspect to their signalling too. In invertebrate neurons this had been taken onboard for some time (Meyrand, 1992).
for 1- and 2-node circuits. A single node with a self-connection is the basic building block for any network (see Figure 3.2A). The presence of the simple but nonlinear self-interaction endows the neuron with two distinct behaviours: a single stable equilibrium point or a pair of stable equilibrium points separated by an unstable equilibrium point (Beer, 1995b). The strength of the self-connection and the external input to the node determine which of the two behaviours the node exhibits.

Any number of single nodes can be made to interact with each other using inhibitory or excitatory connections (see Figure 3.2C). The number of topologically different phase-portraits grows quickly as the number of building blocks is increased. Just two fully interconnected nodes can have 16 qualitatively different behaviours. A catalogue of most of the phase-portraits and how changes to the different parameters affect the behaviour of the network can be found in Beer (1995b). Although catalogues of phase-portraits available for larger networks doesn’t exist yet, a characterization of the global structure of the parameter space of these model neurons for any number of components has been investigated in Beer (2007).

One detail about the parameter space will be mentioned, since it will be used in some of our experiments. A circuit is said to be centre-crossing when the nullclines\(^3\) (the curve along which \(\dot{y}_i = 0\)) of each neuron intersect at their exact centres of symmetry (Beer, 1995b; Mathayomchan & Beer, 2002). What this means is that the neuron’s activation function is centred over the full range of input that it receives. Thus, a centre-crossing neuron is more likely to be sensitive to its input and less likely to always be saturated either on or off. Accordingly, a centre-crossing network is one where all of its neurons are centre-crossing. Perhaps more importantly, the area in parameter space where centre-crossing networks are is at the heart of where the ‘richest’ dynamics are to be found(Beer, 1995b). This will become more relevant when we discuss ‘seeding populations’ in evolutionary algorithms further ahead.

### 3.1.5 Other interpretations of the CTRNN model

As hinted already, the standard interpretation of the model given in the previous section is not the only possible one. The variables and parameters of the model can be meant to instantiate different things depending on the aim of the experiment. Also, there is no one interpretation that is ‘more correct’ than any other. There are two broad categories of interpretations. The first seeks to maintain a close connection to particular biological processes. The second takes a more relaxed view, treating the model as a convenient basis of dynamics.

Within those who seek to connect the model to specific details of living organisms, there are still a number of different interpretations. We provide three of the most relevant ones here:

1. The model can represent the mean firing rate of a neuron or group of neurons (as discussed previously).
2. The model can represent nonspiking neurons. The nervous system of some organisms may function mainly as a network of graded, analog, nonlinear elements. An absence of all-or-none action potentials in certain nerve cells has been observed in, for example, the nematode worm *C. elegans* (Davis & Stretton, 1989; Goodman, Hall, Avery, & Lockery, 1998). Also,\(^3\)

---

\(^3\)Nullclines is the term that applies for a two-dimensional system. For three-dimensional systems, the points where the differential equations are zero denote a surface, and thus the correct term is nullsurface. For higher dimensions, hyper-nullsurface is probably the best term to use.
although all of the focus in firing neurons is in the all-or-nothing spike, the role of the analogue aspects of the signalling of spiking neurons has begun to be recognised as similarly important (Marder, 2006).

3. The model can represent processes at the subcellular level (e.g., a signalling pathway within a neuron). This representation is appropriate because subcellular signalling pathways are usually modelled in terms of the concentration of intracellular messengers in various cellular compartments. The differential equations representing their concentration have a similar form as the equations used to represent neuronal activations (Smolen, Baxter, & Byrne, 2000, 2001; de Jong, 2002; Sontag, Kiyatkin, & Kholodenko, 2004).

Taking the view of CTRNNs as a convenient basis of dynamics is not incompatible with any of the biological interpretations. In fact, such a view recognises that components from any or all of the three interpretations mentioned above could be taking place in any one network at any time. This is the view that is held throughout this thesis, which is why we have added scare quotes to the term neural in continuous-time recurrent ‘neural’ networks.4

To analyse networks that, when embodied and situated, produce learning behaviour we first need to synthesize them (i.e., find appropriate parameters for the differential equations). There are several ways of training algorithms that have been employed to find neural networks with certain desired properties. Two of the most common ones are: recurrent extensions of traditional feed-forward learning algorithms (Pearlmutter, 1990) and evolutionary algorithms (Beer & Gallagher, 1992; Cliff, Harvey, & Husbands, 1993). The work in this thesis will focus on the latter method. The motivation and details of how we will do this are given next.

3.2 Evolutionary technique: the simplest shortcut to generate systems without design

3.2.1 Evolution, not design

The systems that we can design by hand are not necessarily similar to systems that have evolved. As engineers, when we are confronted with designing a system to solve a complex task, we decompose the task into smaller jobs. A group of experts is typically assigned to work on the components that solve each of the subtasks. We place particular emphasis on reducing the complexities of the interactions between the different components. This makes the system much more manageable.

Hand-designed systems tend to have the following characteristics: they are composed of modular components with a one to one functional mapping to parts of the task depending on how it was initially divided, they have the simplest possible interfaces between the components such that the interaction is minimized, and the different components tend to update synchronously. Engineers tend to select good designs based on how feasible they are to be developed and managed; not just on how well they solve the problem.

Less is known about the properties of systems that have evolved. While they may share, on occasions, certain properties of human-designed systems, this is not necessarily always going to be the case. One crucial difference between engineered and evolved systems is that the internal mechanisms of the latter have only been selected for their overall behaviour when embodied and

---

4Inman Harvey has suggested (in private communications) that “CTRNN” should stand for continuous-time recurrent nonlinear networks instead. It would be fitting because it refers to the feature that is common to all interpretations.
Chapter 3. Evolving dynamical systems models

situated. In other words, selection works at the level of the organism’s behaviour; not at the level of the design of the organism’s internal mechanisms. Most certainly, making systems that are easily understandable is not one of the selective pressures evolved systems are exposed to. Furthermore, it is possible that the majority of properties of evolved and designed systems be radically different from each other.

3.2.2 Beginning to learn how to understand evolved systems

Even if we imagined that ‘complete’ databases containing every single detail about the properties of each of the genes, and cells and their synapses and the rest of the cell signalling behaviour existed, we still wouldn’t be able to put the full system together and understand how all of the components interact with the body and the environment to produce adaptive behaviour. A major theoretical challenge that biology\(^5\) is faced with is that we simply do not know how to understand systems that have evolved.

We need to learn how to begin to understand complex networks of dynamic and nonlinearly interacting components. This will require the development of a set of tools and language. Most importantly, it requires model organisms where investigations can be carried out. We sketch two approaches that could, in principle, be taken.

First, we could develop these tools on real living organisms. Ideally, we would need to have an organism from which we can record the activity of all relevant internal components (e.g., calcium-imaging) (cf. variables of the system) in the least disruptive manner, while the organism is performing the adaptive behaviour it was evolved for in its ecological context (or at least an environment that has the relevant features). At present, such an approach has serious limitations. The tools to record the activity from the variables of the system are still at an early stage of development. So they tend to be imprecise. They also generally damage the system under study (e.g., prolonged exposure to light of certain wavelengths can severely damage nerve cells). Also, most of the recordings have to be made while the animal is immobilised\(^6\). Another issue is that the ecological context of their behaviours cannot be taken into consideration because it is often not well understood\(^7\).

Second, if a ‘complete’ database with all the relevant details of a biological organism did exist (or once it does exist), then we could build realistic simulations of the organism of interest. This would enable us to test hypothesis and predictions about the organism, by probing the simulation. In practice, it is fair to say that we are not there yet (even for very ‘simple’ organisms\(^8\)). And given that the simulation will be almost as complex as the real organism, it will be harder to build up intuitions and tools from the ground up (as opposed to starting with simpler idealised models).

As a shortcut to produce evolved systems

Work is on its way for a combination of the above approaches to be feasible. However, we should not need to wait until they are ready in order to begin understanding situated, embodied, dynamic,

\(^5\)This also applies to other sciences that deal with systems that evolve, such as economics.

\(^6\)Exciting new developments are allowing for recordings of calcium imaging of one or a few nerve cells on the freely moving worm (Faumont & Lockery, 2006; Clark, Gabel, Gabel, & Samuel, 2007).

\(^7\)This is the case in, for example, \textit{C. elegans} (Caswell-Chen, Chen, Lewis, Douhan, Nadler, & Carey, 2005).

\(^8\)A good example is, again, the nematode worm \textit{C. elegans}. Although a lot is known about the neurons and anatomical connectivity. Many of the connections are still uncertain. More importantly, the properties of individual neurons and nonsynaptic forms of interneuron communication (e.g., NO, CO, H\textsubscript{2}O\textsubscript{2}) are not entirely known.
and evolved systems. We can, in effect, ‘practice’ with artificial versions of non-designed systems that nevertheless perform adaptive behaviours. However, as we said before, we do not yet know enough about living organisms to be able to design evolved-like mechanisms. We can use evolutionary algorithms as a shortcut for coming up with non-designed systems (i.e., designing systems that do not necessarily have the properties of engineered systems). This is our main motivation for using evolutionary algorithms.

**Minimising built-in preconceptions of how an agent must solve a cognitive task**

Even if we did know many things about how to design evolved-like systems, when we design it is inevitable that we build-in our assumptions about how the task must be solved. We are interested, however, in the broader set of possible mechanisms that evolution can come up with. We are particularly interested in ‘solutions/designs’ that are different from what was expected. Counterintuitive solutions provide us with new ways of thinking about the task. This is our second motivation for using evolutionary algorithms. To be clear, we are not claiming that dynamical system networks and evolutionary algorithms impose no bias on the agents that we will be exploring. The point that we wish to make is that the bias is far less direct than in systems that have been hand-designed (cf. Bullock, 2006).

**Exploiting the agent’s embodiment, and situatedness**

As we discussed previously (Section 2.3), the role that situatedness and embodiment can play in the generation of behaviours is sometimes unexpected. This is particularly the case when combined with evolutionary techniques, because the evolutionary algorithm can exploit interactions that are not obvious to a person attempting to design a solution for the same task. This is often the case in evolutionary robotics. Adrian Thompson’s evolutionary hardware experiments has provided some of the better known examples (Thompson, 1995). He describes how evolution could (and unless constrained would) put to use the properties of the hardware that the designer could not know about (e.g., internal time-delays, silicon defects, the chip’s temperature, or its power supply). By relaxing the constraints on the spatial (i.e., modularisation of the design into parts with simple, well defined interactions between them) and temporal (i.e., the use of a clock to prevent the natural dynamics of the components from affecting overall behaviour) organisation of the system that would be necessary for a human designer, the resulting systems showed unprecedented power and efficiency.

A designer carefully avoids “glitches,” “cross-talk,” “transients” and “meta-stability,” but all of these things could be put to use by artificial evolution. (Thompson, 1995, p. 644).

The networks that we analyse in this thesis also show a major tendency to use their embodiment and their history of interactions with the environment whenever the body or environment provide sufficient regularities for them to do so. This is an important factor to consider for living organisms too, as the real environment also often offers regularities.

In summary, evolutionary algorithms are one way to generate agents: (a) that are selected on the basis of successfully performing some behaviour, (b) that do not necessarily have the properties of hand-designed systems, (c) where the preconceptions of how the task must be solved are
reduced, (d) that are likely to exploit their embodiment and situatedness, and (e) that are easily tractable (i.e., full access to all of the variables while interacting with their environments).

We have mentioned the motivations for using evolutionary algorithms in this thesis. To avoid confusion, it will be useful to clarify the interests that are not pursued herein. First, we do not use evolutionary algorithms to test their performance against other optimization techniques or between different flavours of evolutionary algorithms. Second, we do not investigate the evolutionary dynamics of populations. Third, we do not investigate the set of different environmental conditions under which learning arises and does not. Finally, we are not interested in evolutionary algorithms as a way of studying real biological evolution.

### 3.2.3 Background to evolutionary algorithms

Algorithms inspired by natural evolution were developed during the 1950s and 1960s as an optimization tool for engineering problems. Evolutionary strategies were originally introduced by Rechenberg in as a special form of evolutionary algorithms (Rechenberg, 1973; Schwefel, 1981). The main difference is that the genotype is encoded using vectors of real numbers as opposed to discrete units. Mutation and selection are the primary operators. Genetic algorithms were invented in Holland (1975) with the purpose to formally study the phenomenon of adaptation as it occurs in nature; not an optimization technique. The field has grown in the last 30 years. From early on the field subdivided into three branches are: genetic algorithms, genetic programming, and evolutionary strategies. A plethora of evolutionary algorithms exist today. For an introduction to genetic algorithms see Mitchell (1998) and for an introduction to evolutionary algorithms see Bäck (1996).

Evolutionary algorithms are merely inspired by biological evolution. While most of the details and complexities are abstracted away, there are three processes that must remain: heredity (i.e., offspring should be to a large extent similar to their parents), variation (i.e., offspring must in general not be exactly identical to their parents), and selection (i.e., ‘fitter’ ones are more likely to survive long enough to have offspring).

### 3.2.4 Microbial GA

Much of the discussion in the evolutionary computation literature revolves around which method is most efficient (e.g., speeds up evolution). This will not be of concern in this thesis. The evolutionary algorithm, genetic operators and their parameters were chosen as the simplest options that generated sufficiently interesting agents. For this reason we use a version of the Microbial genetic algorithm first developed in Harvey (1995).

The algorithm is extremely simple (see Figure 3.3). Starting from a population of individuals, two of them are chosen at random to compete in a tournament. The fitness of each of the two organisms is evaluated. The fitter individual is the winner of the tournament. The less fit individual is replaced by a variation of the winner. This is equivalent to horizontal transmission of genetic material from the winner to the loser of the tournament. Notice we do not employ recombination. These steps are repeated until successful individuals are found.

Tournament selection operates by taking two members of the population chosen at random, and choosing the best of the two to contribute genetic material to a new individual. This strategy for
3.2. Evolutionary technique

selecting individuals simplifies matters greatly in relation to evolutionary algorithms that require
the ranking of all individuals in the population before selection can occur. There are a variety of
ways to choose which old individual should sacrifice its places for the new, we simply replace the
loser of the tournament.

The three aspects worth mentioning with respect to how the Microbial algorithm is different
from the traditional genetic algorithms are: (a) the algorithm is a steady state, as opposed to
generational method, (b) rank-based as opposed to fitness-proportionate method, (c) the least fit
individual of the two randomly chosen for a tournament is chosen to die and be replaced by a
mutant of the winner, (d) the most fit individual remains unchanged.

Since the fitness function varies depending on the task, each will be discussed in the methods
section of the simulation chapters.

Genotype-encoding and genotype-phenotype mapping

In living organisms, genotypes are composed of a series of discrete units drawn from a finite al-
phabet (i.e., Thymine, Guanine, Cytosine, and Adenine). We will encode the parameters of our
dynamical system model using real values instead. Also, living organisms develop. Real geno-
types do not encode parameters of the final phenotype; they unfold growth patterns. Grammatical
encodings and developmental rules have been used for neural networks (e.g., Kitano, 1990). While
this is an important area of research, for simplicity we encode the parameters of the dynamical sys-
tem controller directly.

The parameters of the dynamical network are the biases ($\theta$), time-constants ($\tau$), and strengths
of the weights connecting the nodes ($w$). Thus, for a network with $N$ nodes, there are usually
$N^2 + 2N$ parameters in total. Sometimes there are more parameters because of connections from
extra sensory components.

There are two approaches to directly encoding the real-valued parameters of the circuit. The
first is to discretise the space by representing real numbers as binary strings. The major known
problem with this approach is that of ‘Hamming cliffs’ (i.e., points which are neighbours accord-
ing to the topology of the continuous space, but are not neighbours when considered as binary strings) (Whitley, 1999). ‘Gray code’ partially overcomes this issue, but it introduces a number of related problems (see Bullock, 1999, for a discussion). The second approach has been to evolve the real-valued parameters directly, which requires in turn the development of special mutation operators.

We encode all values in a genotype as a vector of real numbers over the range $[0, 1]$. Genes are mapped to network parameters linearly usually between $[-10, 10]$ for biases and inter-node and sensory weights. We use a different mapping for time-constants. In the case of time-constants, it is ideal to have greater precision for faster neurons, where issues of timing may be more crucial, and less precision for much slower ones. The difference between two neurons that are both slow, say with time-constants 70 and 71, is much less likely to be significant than the difference between two very fast neurons, with for example time-constants of 1 and 2. For this reason, time-constants are usually mapped exponentially between $[e^0, e^5]$.

**Gaussian mutation**

As we said previously, genetic algorithms were originally developed to deal with genotypes comprised of discrete units. Mutations usually correspond to flipping one unit to a randomly chosen base. Evolution strategies are the name adopted for evolutionary algorithms using real-valued genotypes. The most direct translation from the genetic algorithm approach would be to change any mutated locus to a randomly chosen real number within the appropriate range. But this is rarely used because it is very disruptive. The traditional mutation operator, instead, relies on perturbing (i.e., adding a small random number) to all loci in the genotype. Several distributions from which to generate the random number have been used. The most common is a Gaussian distribution, but Uniform, Cauchy and Harmonic distributions are also employed (e.g., Yao & Liu, 1997; Rowe & Hidovic, 2004).

We implement mutation as a random displacement on every gene drawn uniformly from a Gaussian distribution with mean 0 and variance 0.01 (see Figure 3.4). In addition to this, each gene is forced to stay within the range $[0, 1]$. When a mutation takes a gene out of this range it is reflected back (e.g., if a mutation leaves a gene with a value of 1.02, it is immediately converted to 0.98).

It is possible that our choice of mutation operator biases the direction of evolutionary change, as suggested in Bullock (1999). However, whatever such a bias might be, we do not think it will affect the principal aim of this thesis (i.e., to study non-synaptic forms of plasticity). We are certainly not aware of any obvious reasons why it should. The same applies to the rest of the choices described in this chapter. We proceed under the assumption that these choices do not impair our results, analysis, or conclusions.

**Adaptive mutation rates**

In evolutionary strategies, sophisticated methods for modifying the amount of mutation in different dimensions as evolution proceeds has been developed (Rechenberg, 1973). Similar ideas have been applied for evolving neural network controllers (e.g., Slocum, Downey, & Beer, 2000). In some of our experiments we also use a very simple adaptive mutation rate. The point of which is to reduce the size of the perturbation of the mutation as the populations gets better at doing the task, starting with ‘big steps’ in search space and fine tuning as populations get closer to their target.
3.2. Evolutionary technique

![Figure 3.4: Gaussian mutation of real-valued genotypes. [A] Probability density function of a normal distribution with mean 0 and standard deviation 0.01. [B] 3-dimensional example of $10^3$ mutations using the Gaussian mutation method described in the main text.](image)

The details of the adaptive rule will be provided in the methods section of the chapters where it is used.

*Trivial geography*

All biological populations are obviously distributed in space. This means that some organisms are close neighbours while others live at great distances. The effect of spatial distributions in the evolutionary dynamics has been discussed for a long time in evolutionary biology (Mayr, 1942; Wright, 1945). Traditionally genetic algorithms are nonspatial: they operate on populations that do not share a physical space. In other words, any two individuals can compete in a tournament with equal probability. If there is recombination, then any two organisms can also recombine to produce offspring with equal probabilities.

In Spector and Klein (2005) a ‘trivial geography’ is introduced in the context of genetic programming. The population is viewed as having a one-dimensional spatial structure. The first and last individuals are also neighbours, so the space effectively ‘wraps around’. The idea is to apply some constraints on which individuals can compete to produce new offspring based on the spatial dimension. In practice, after the first individual is chosen at random to participate in a tournament, the second individual is chosen from its ‘neighbourhood’. The neighbourhood is defined as some distance away from the original individual, in our case 10.

The intuitive reason for adding a ‘trivial geography’ into our evolutionary algorithm was that embedding the population in a spatial dimension would help to maintain its genotypic diversity. Most importantly: (a) it didn’t make the algorithm any more complicated (i.e., a minor change in only one line of code), and (b) it did not have a noticeable effect on the computational cost of evolutionary runs.\(^9\)

---

\(^9\)This is for two reasons. First, because the additional cost to the evolutionary algorithm on its own is insignificant (i.e., there is no cost difference between choosing a random number between, for example, \([0,100]\) and \([0,10]\)). Second, because most of the computational time is actually spent simulating the brain-body-environment interaction and only a fraction (around 1% for the tasks in this thesis) is dedicated to the evolutionary algorithm itself.
Chapter 3. Evolving dynamical systems models

Incremental evolution or shaping protocols

The idea of incremental evolution is inspired by co-evolutionary scenarios, where the task evolves simultaneously with the organism. For example, in a predator and prey situation, early during evolution predators would presumably not have to be too clever to catch prey. Similarly the prey’s avoidance techniques would not be highly sophisticated to escape either. As both species evolve, their repertoires could become broader – each having had to deal with new and different strategies for catching/avoiding over evolutionary time. Unlike co-evolutionary scenarios, during incremental evolution there is no chance for “red queen” effects to take place (Van Valen, 1973). The paradigm that we employ is straightforward; populations start with deliberately simple versions of the task of interest and gradually move on to more complex versions as the populations succeed.

In Harvey, Husbands, and Cliff (1994) incremental evolution was used to train physical robots to move towards a large white target first. Successful individuals were further evolved on a smaller target. They compared one evolved agent in this incremental fashion with another one evolved directly on the small target and reported better generalisation performance for the former. They acknowledge, however, that the result is merely suggestive and not statistically significant. In Beer and Gallagher (1992) a mixed approach is used, where a controller is trained to do single-legged walking. This network is then fixed and replicated six times. Sets of extra connections between them are evolved to control a six-legged walker. Incremental evolution approaches have also been used in Nolfi, Floreano, Miglino, and Mondada (1994), Floreano and Mondada (1998) and some work comparing their differences in Winkeler and Manjunath (1998).

The idea is that populations that had been previously evolved for task $X$, can be used as seeds in a new task, $Y$. This, of course, presupposes that being able to do $X$ is advantageous for doing $Y$. This approach must be used extremely carefully, as it is all too easy to make the problem harder for the controller. Particularly if tasks $X$ and $Y$ are not trivially related. We believe there are certain situations when the task lends itself to employing a shaping protocol, but for some other tasks this may not be beneficial. In some of the experiments in this thesis we use a shaping protocol. In all of the cases that it is used, $X$ is a simple subset of $Y$.

Seeding populations using centre-crossing networks

Usually, all parameters of the networks that comprise the population are set to random values drawn from a uniform distribution at the beginning of an evolutionary run. The reason for this is relatively obvious. We do not know a priori where in the parameter space of the controller model the adequate dynamics exist, such that when embodied and situated the agent will perform the desired adaptive behaviour. In fact, the mapping between the genotype (i.e., CTRNN parameter space) and the phenotype (behaviour produced by the agent in interaction with the environment) is a highly indirect and nontrivial mapping.

Seeding refers to starting the population with specific network parameters. If the difficulties involved in hand-designing a nonlinear network of interacting elements are overcome (or are experimentally known) and the hypothesis to be tested requires it, or if evolution repeatedly fails to find useful parameters, and a hand-designed has been devised. Then it may be useful to seed the evolutionary algorithm with hand-designed solutions (e.g., Drennan & Beer, 2006; Vickerstaff, 2006). Similarly, populations that had been previously evolved for task $X$, can be used as seeds in a new task, $Y$ (as discussed in the previous section).
3.2. Evolutionary technique

Thanks to the analysis of CTRNN parameter space in Beer (1995b, 2007), Buckley (2007) we do know a number of useful relations between the network’s overall dynamics and its parameters (this was briefly summarised in the CTRNN section above). Such understanding opens up the possibility to focus an evolutionary search into regions of parameter space that are most likely to yield a particular dynamics from the network.

We know, for example, that the majority of the phase-portraits of randomly generated CTRNNs will contain a single stable equilibrium point (Beer, 1995b). Furthermore, as the network gets larger the probability of encountering such phase-portraits grows (keeping the ranges where the parameters are drawn from equal). There are, however, regions of parameter space where the probability of finding ‘richer’ dynamics exist. Using this reasoning, Beer suggested that centre-crossing networks might be more evolvable in Beer (1995b). And later, together with one of his students, they tested this hypothesis in a task requiring oscillatory dynamics. They found that seeding evolutionary searches with random centre-crossing networks led to quicker evolution and better solutions (Mathayomchan & Beer, 2002).

In this thesis, however, evolutionary runs were not seeded with random centre-crossing populations. In a set of preliminary experiments, we failed to see an improvement. In the task to evolve a network for walking (oscillatory dynamics), seeding helps because it allows the easier location of the oscillatory dynamics. All of the tasks explored in this thesis used a shaping protocol (as previously described). This means that the population is first oriented to a particular region in parameter space, and then when the fitness function changes, the population is led to a second region (potentially altogether different). After the initial search, the population is likely to be genetically converged. Our intuition is that any seeding will be quickly lost by the time the evolutionary task advances further.
In this chapter we review the literature most relevant to our thesis. First, we provide a brief background into the study of learning behaviour from animal learning theory and discuss its relationship to the situated, embodied, and dynamical systems perspective. Second, we discuss the differences between the two adaptive processes that are considered in our experiments: evolutionary learning and lifetime learning. We survey two of the most important computational models that investigate the relationship between these two processes. Third, we review in some depth the work on evolving dynamical neural networks without synaptic plasticity for tasks that require learning behaviour. In the final section, we discuss a number of other ideas that are less directly related to our work, but still relevant.

4.1 Learning behaviour

One of the first to study learning and memory experimentally in humans was Hermann Ebbinghaus in 1885. Not much later, Ivan Pavlov and Edward Thorndike started experiments with other animals. Together with a number of other ‘behaviourists’, such as John Watson, Edwin Guthrie, Clark Hull, and Burrhus F. Skinner, they proposed the initial working definitions of learning. These were of the form: “Learning is a modification of behaviour based on experience”, for example:

The process which manifests itself by adaptive changes in individual behaviour as a result of experience. (Thorpe, 1956)

Learning refers to the change in a subject’s behaviour or behaviour potential to a given situation brought about by the subject’s repeated experiences in that situation, provided that the behaviour change cannot be explained on the basis of the subject’s native response tendencies, maturation, or temporary states (such as fatigue, drunkenness, drives, and so on). (Bower & Hilgard, 1981)

As a reaction from introspection practices, behaviourists intended to explain human and animal behaviour in terms of external physical stimuli and organism responses. In fact, they argued learning (as any other behaviour) would be best explained without making reference to mental events or to internal psychological processes. They were less interested in the internal mechanisms that generated such behaviour. The main objection to their work was that the organism was
treated like a black box. As a consequence of this, and after some developments in neurobiology, later definitions of learning attempted to incorporate references to the internal mechanisms, for example:

Learning ... is ... adaptive modification of behaviour ... an improvement of the physiological “machinery” whose function is behaviour. (Lorenz, 1981)

But attempts to relate behavioural and mechanistic definitions of learning ran into two main problems. First, there was a lack of a common language between processes going on inside the agent and processes going on in the interaction between agent and environment. Neurobiologists, for example, deal with molecules, excitable membranes, neurons, and synapses; while behaviourists deal with events such as “finding something pleasant”. This is how Skinner, for example, discussed the gap:

We are all familiar with the changes that are supposed to take place in the nervous system when an organism learns. Synaptic connections are made or broken, electrical fields are disrupted or reorganized, concentrations of ions are built up or allowed to diffuse away, and so on. [...] In a science of behaviour, where we are concerned with whether or not an organism secretes saliva when a bell rings, or jumps toward a grey triangle, or says bik when a cards reads tuz, or loves someone who resembles his mother, all statements about the nervous system [...] are not expressed in the same terms and could not be confirmed with the same methods of observation as the facts for which they are said to account. (Skinner, 1950, p. 193)

Second, those studying the internal mechanisms had, for the most part, not been able to study them while in interaction with their bodies and in their environments (i.e., during actual behaviour). Due to the techniques available, they had been largely constrained in what they could study. They heavily relied on studying physically constrained and far from intact animals during carefully prepared presentations of stimuli.

Instead of directly addressing either of these two issues, researchers managed to circumvent them by making the definitions of learning purely mechanistic. In Young (1987), for example, learning is defined as “The capacity to change nervous pathways”. Mechanistic definitions of learning made explicit that certain structural changes were necessary and sufficient for learning, some even which particular changes were necessary, for example:

Activity-dependent synaptic plasticity is induced at appropriate synapses during memory formation, and is both necessary and sufficient for the information storage underlying the type of memory indicated by the brain area in which that plasticity is observed. (Martin, Grimwood, & Morris, 2000)

Harvey summarises the contrast between a behavioural and a mechanistic description of learning by providing a set of different conditions which are often demanded of an agent that is learning:

1. The system changes its behaviour over time.
2. This change ‘improves’ its behaviour in some sense.
3. The change in behaviour was at the cost of some work or effort by the system.
4. Some particular kind of internal change of state must have taken place in the system. (Harvey, 1995, pp. 57-58)
He agrees that the first two conditions (and possibly also the third) would be required of a learning agent under the behavioural description, while the fourth would be required only for the mechanistic description.

In an attempt to marry the behavioural and mechanistic notions of learning, the computational paradigm muddled itself. By redefining the problem in computational terms, on how information flowed from sensory receptors to memory then processed by the brain and resulting in action, a link between the processes going on inside and out had been found. But it came at some cost. In order to achieve the link, they had to assume that “each perceptual or motor act has an internal representation in the brain” (Kandel, Schwartz, & Jessell, 2000, p. 383). In other words, learning would now require internal representations.

Rather than recognizing the shortcomings that stemmed from the lack of sufficiently sophisticated tools to investigate and record from the internal mechanisms of behaving animals; they adapted their language of analysis to suit the neurobiological technical limitations. As a consequence, animal learning experiments have become not more, but less ecological: the stimuli and responses were discretised as much as possible to facilitate computational explanations. This made support for the existence of ‘internal representations in the brain’ easier to obtain from neurophysiological evidence that cells in the brain possess firing characteristics that are consistent with something in the environment. In turn, even more techniques were put in place to study the correlation between activity in the brain and behaviour. This is currently the dominant doctrine, as shown by the present definitions of learning:

In humans, the most important mechanisms by which the environment alters behaviour are learning and memory. Learning is the process by which we acquire knowledge about the world, while memory is the process by which the knowledge is encoded, stored, and later retrieved. (Kandel et al., 2000, p. 1227)

Memories are experience-dependent internal representations, in other words, acquired models of the world, encoded in the spatiotemporal activity of brain circuits. (Dudai, 2002, p. 211)

Dynamical systems theory is a language that can be used to talk about internal as well as external processes, at several different levels of abstraction. Yet unlike the computational language, it does not presuppose the existence of ‘internal representations in the brain’. In fact, it is probably the language that makes the fewest assumptions about the subjects of study (basically that they unfold over time). While it is true that the techniques to understand learning behaviour in freely moving animals are still underdeveloped, we believe that this should not hinder the development of the mathematics and tools of analysis to understand learning behaviour in situated, embodied, and dynamical agents. On the contrary, we should begin training on examples (and thus, artificial examples) because once the neurobiological techniques to record on multiple cells on freely moving animals in ecological contexts are available, the tools of analysis that stem from our research will become indispensable. Furthermore, practice understanding artificial systems using dynamical systems theory may suggest directions in which the neurobiological techniques for recording animals should develop.

Beer puts forth the behavioural definition of learning in an evolutionary context, relating it to the internal mechanisms of the agent:
4.1. Learning behaviour

Suppose that an agent has internal dynamics on a range of timescales and let us focus on the dynamics at timescales that are long relative to the timescales of the agent’s actions. This long timescale dynamics can be deleterious, neutral, or advantageous. If, however, the agent is subjected to a selection process in which the long timescale dynamics is under evolutionary control, then any deleterious long timescale dynamics will be selected away. [...] In this sense, the improvement of behaviour over time is an inevitable property of agents with dynamics on a range of timescales subjected to selection in a changing environment. (Phattanasri et al., 2007, p. 392)

This is the view of learning that we agree with. Thus, in this thesis we do not assume that the dynamical system inside the agent is a priori capable of learning, nor that it will have ‘internal representations’.

4.1.1 Learning paradigms

There are two major forms of learning: nonassociative and associative. In nonassociative learning, the subject learns about the properties of a single stimulus. Two forms of nonassociative learning are sensitization and habituation, an enhanced or decreased response to a repeatedly presented stimulus, respectively. In associative learning, the subject learns about the relationship between either two stimuli (classical conditioning) or between behaviour and a stimulus (operant conditioning).

Classical conditioning was first studied by Pavlov (1927). A conditioned stimulus (CS) that produces no response is paired with an unconditioned response (US) or reinforcement, such as food or an electric shock, which normally produces a response (without learning). When a CS is paired or followed by a US, the CS begins to elicit a similar response. The intensity of this new conditioned response will decrease if the CS is repeatedly presented without the US, in a process known as extinction.

Edward Thorndike was the first to study operant conditioning (Thorndike, 1911). In a typical experiment, a starved animal is placed in a test chamber where it is rewarded for specific actions (e.g., figuring out how to escape or how to get food). Unlike classical conditioning, operant conditioning involves behaviours that occur either spontaneously or without an identifiable stimulus (e.g., pressing a lever). Thus, behaviours that result in favourable conditions (e.g., rewarded with food), tend to be repeated; whereas behaviours that result in unfavourable ones (e.g., electric shock) tend to be avoided. This was Thorndike’s law of effect.

Skinner expanded and refined on Thorndike’s conceptual and experimental framework. He introduced the idea of a ‘free operant’, so called because the animal was now permitted to respond at its own rate rather than in a series of trials determined by the experimenter. Skinner (1938) formulated a general learning theory based on reinforcement. The idea was that any reinforcer could strengthen any response in the presence of any stimulus, although this was problematic, as one of Skinner’s student (Keller Breland) later found out. Breland and Breland (1961) showed that not all behaviour was modifiable by reinforcement. The Brelands trained animals for entertainment purposes, shaping unusual behaviours using operant techniques. But they were not always successful. In one example, Breland and Breland (1961) described their failure to train pigs (and later also raccoons) to deposit coins in a ‘piggy’ bank, despite being rewarded with food. The association was not possible. Their experiments provided wonderful examples of how what could be
reinforced for any particular animal would depend on their evolutionary context (Hall & Halliday, 1998; Shettleworth, 1998).

There is a tendency to treat agents that learn as “general problem solvers” (i.e., anything can be learnt). But the associative learning mechanisms that animals have evolved are far from universal. Learning abilities were specifically adapted to the organisms ecological constraints instead (Bolhuis & Macphail, 2001; Shettleworth, 1998). That is, animals are not likely to associate any two stimuli chosen at random. For stimuli to be associated, it must have been related to some degree to their survival. Thus, evolutionary pressures will predispose different species to associate different stimuli more readily than others (Gould & Marler, 1987). In other words, the ecological history of the animal’s lineage will help determine the properties of its learning mechanisms (see Breland & Brelan, 1961; Dickinson, 1980; Gould & Marler, 1987; Hall & Halliday, 1998; Shettleworth, 1998; Bolhuis & Macphail, 2001, for more detail).

Pertinent for our work, timing is critical in both forms of associative learning. In operant conditioning, the reinforcer must closely follow the operant behaviour. Long delays will result in weak conditioning. Similarly in classical conditioning, if the pairing of the two stimuli is too far apart, the conditioning is poor (Lavond & Steinmetz, 2003).

Associative forms of learning are common among a wide range of organisms (Hennessey, 1979; Werner, Kadlec, & Macphail, 1996; Fanselow & Poulos, 2005). What environmental conditions could have shaped such a common learning behaviour? All animals must search out food that is edible and nutritious while avoiding food that is non-nutritious or poisonous. Depending on how variable the environment is during the lifetime of an organism, survival will require learning mechanisms or not. Coincidently, how variable an environment is will also be closely related to how mobile the agent is, as the more it moves the more likely it will be to encounter qualitatively different environments.

More complex associative mechanisms than classical and operant conditioning have been proposed (e.g., McFarland, 1998). These usually assume the manipulation of explicit ‘internal representations in the brain’. In fact, most models of learning make significant assumptions about the nature of internal representations.

Following from the computationalist paradigm on learning, memory is the place in the brain where information about the world is stored. This is what Andy Clark had referred to as the brain’s filing cabinet in the quote in our previous chapter. Memory is conventionally classified as either: implicit (nondeclarative) or explicit (declarative). Implicit memory is meant to encode perceptual and motor skills (e.g., playing tennis, driving a car, reading). Explicit memory is meant to encode more “abstract” things (e.g., names of persons and places, dates). This distinction is, however, mostly conceptual - given that even for the simplest of ecological behaviours separating between the two types of memory is far from clear cut. Some mechanistic distinctions, nevertheless, have been observed in sufficiently computationally constrained learning tasks.

Both operant and classical conditioning postulate the generation of a new connection between the stimulus and the response, due to experience. It is this hypothetical “connection” that is called memory. For those working with neural networks, the “connection” alludes to the synapses connecting neurons. Which is why learning has been traditionally equated with changes in synaptic strength in artificial neural networks. There has been, nonetheless, evidence that changes in synap-
tic strength do play a role in learning behaviour. Most of it has come from studies in the marine snail *Aplysia* where, for example, classical conditioning has been observed to involve presynaptic facilitation of synaptic transmission that is dependent on activity in both the presynaptic and postsynaptic cell. For an introduction to the currently known cellular mechanisms of learning see Chapter 63 of *Principles of Neural Science* (Kandel et al., 2000). One thing seems to be certain, according to almost all studies of learning synaptic plasticity is understood to be both necessary and sufficient for learning. In this thesis we question its necessity, and we suggest a number of other factors that can just as equally be involved in learning behaviour.

### 4.2 Evolution and learning

Evolution and learning are the two most important forms of biological adaptation. They operate on different timescales. Evolution allows populations to adapt to slowly changing environments, while learning allows an individual to adapt to changes in the environment occurring within its lifetime. Artificial evolution techniques have been combined with learning in neural networks to study the interaction between phylogenetic and ontogenetic adaptive changes. The main focus has been in answering the following two questions. First, can ontogenetic learning improve the adaptive performance of an evolving population? Second, under which conditions do organisms evolve to learn? We briefly review two of the most relevant computational modelling approaches that have begun to address each of those questions, in turn.

#### 4.2.1 Hinton and Nowlan

(Baldwin, 1896) was the first to argue that learning accelerates evolution. His argument was simple: sub-optimal individuals can reproduce by acquiring during life necessary features for survival. And as learning behaviour requires time, natural selection would favour individuals who wouldn’t have to learn (given that the environment was sufficiently stable). Baldwin called this the genetic assimilation of learnt traits. Scientific evidence was later found by Waddington (1942) who referred to it as a canalization effect.

Hinton and Nowlan (1987) produced a simple computational model illustrating how learning can guide evolution. In their model, the genotype is a string of 0s, 1s, and ?s. The metaphor invoked is that these represent the connectivity of the neural network that guides the individual’s behaviour. More specifically they represent: lack of a connection, presence of a connection, or a modifiable connection, respectively. The learning mechanism corresponds to randomly changing the ? connection to 0 or 1. Reproductive fitness is only possible when the network is perfectly connected. This corresponds to a flat fitness landscape with one skyscraper (e.g., a needle in a haystack). In the absence of learning, evolutionary search performs equally poor to random search. With learning, on the other hand, the fitness landscape effectively smooths - allowing individuals closer to the ‘skyscraper’ to reproduce slightly better than those further away from it. The result is a simplified evolutionary search. Of course, Hinton and Nowlan’s model had a number of limitations (e.g., the environment does not change, the landscape is very particular, learning is random and operates on the same space as evolution). Some of these have been addressed in latter work (e.g., Nolfi, Elman, & Parisi, 1994; Harvey & Stone, 1996; Harvey, 1997b; Simpson, 1996; Mayley, 1997; Nolfi & Floreano, 1999; Ancel, 2000; Mills & Watson, 2005)).
4.2.2 Peter M. Todd and Geoffrey T. Miller

Todd and Miller (1991) investigated the relations between evolution by natural selection and associative learning behaviour (also in Miller & Todd, 1990). They were particularly interested in discovering under what conditions learning would evolve. They used genetic algorithms to simulate the evolution of neural networks that control the behaviour of simple “creatures” in virtual environments.

The task that the agents are presented with is as follows. There are two types of edible things in the world: food and poison. An agent can detect their colour (red or green) and smell (sour or sweet). Food always smells sweet and poison always sour; but the colour of food and poison varies in different environments in the world. There are two types of environment. In environment $A$, food is red and poison green. In environment $B$, the colours of items change: food is now green and poison red. A successful agent must learn to eat food and avoid poison.

Agents are modelled using a feedforward neural network with three neurons in two layers. Each of the connections between the nodes in the first and the second layer can evolve to be fixed or varying. Those that vary follow Hebbian weight changing rules.

To determine which settings make it more likely for the agents to evolve the learning mechanisms, they varied systematically the agent’s smell-sense accuracy between 50% (random) and 100% (perfectly accurate). When they studied the average number of generations required to evolve agents that would learn to associate colour with smell during their lifetime, they observed a U-shaped curve, with the shortest average evolutionary time at around the 75% smell-sense accuracy.

Towards the lowest accuracy (near 50%), the smell of an item is effectively random. This means that there is not a sufficiently good statistical correlation between the smell and colour of an item, so as to develop the association. On the other hand, when the accuracy of smell is perfect (near 100%), there is no need to pay attention to the colour of the items. Indeed, “colour-blind eaters” would evolve under this condition. With no learning behaviour required, the connection between the smell sensor and the motor neuron (which control the eating response) would evolve to be fixed.

It is when the agent’s smell accuracy is not perfect, but not random either (near 75%), that agents can learn to associate the colour with the smell of the food - depending on the environment they are in. A successful “colour-learning eater” would evolve both smell and colour sensors, with connection to the motor neuron. While the smell-sensor neuron to motor-neuron connection was fixed, the connection between the colour-sensing neuron and the motor neuron evolved to use the Hebbian weight changing rule.

4.3 Evolving dynamical system controllers that learn

A situated and embodied agent must be able to react immediately to the events occurring in its environment. It is likely for this reason that reactive behaviour (or behaviour requiring a simple sequence of reactive behaviours) has been the primary focus of most autonomous agents research. Beer and his colleagues have, for example: evolved agents capable of chemotaxis, legged locomotion, object catching, shape discrimination, selective attention, perception of body-scaled affordances (Beer & Gallagher, 1992; Beer, 1996; Slocum et al., 2000). The group at Sussex have also
4.3. Evolving dynamical system controllers that learn

... evolved a range of networks capable of similar tasks, for example: using vision to orient an agent to an object in the environment, bipedal walking, arm motor control, insect navigation (Cliff et al., 1993; Harvey, Husbands, Cliff, Thompson, & Jakobi, 1997; Harvey et al., 2005). But the ability to modify behaviour from experience is similarly very important, particularly under changing environments.

There has been a good amount of work evolving dynamical neural networks in tasks that do require learning behaviour (e.g., Nolfi et al., 1994; Nolfi & Parisi, 1996; Floreano & Mondada, 1996, 1998; Nolfi & Floreano, 1999; Floreano & Urzelai, 2000; Urzelai & Floreano, 2000; Floreano & Urzelai, 2001; Nolfi, 2003). However, most of it has assumed that synaptic changing mechanisms have to be provided as a building block for evolution to use (same as neurons, for example). Furthermore, even if the tasks have not required learning behaviour per se – or if there isn’t even any behaviour at all (just neural activity) – once synaptic changing rules are provided (or artificial evolution allowed to add them to the network at any point), then the system is said be able to ‘learn’. This is a simple case of treating learning purely as a mechanism, not behaviour. For this reason we will not discuss that work in any detail here.

A more integrated view of learning, as a dynamical process occurring over multiple timescales, was first made concrete in a set of ER experiments by Yamauchi and Beer (1994a, 1994b). Continuous time recurrent neural networks (CTRNNs) without synaptic plasticity were successfully evolved in tasks that require learning behaviour. The interests in this work is in how the learning behaviour is produced. We will review this ahead in Section 4.3.1.

In the diagram in Figure 4.1 we show the different dimensions on which related work on evolving dynamical system controllers that ‘learn’ has been traditionally developed (white circles). The circles shaded grey represent the dimensions in which work is being further extended in this thesis. In the section ahead we will discuss the most relevant related work to ours. The points that make each of the works presented in this thesis different from each of the research efforts presented here will be made in a related work section for the different experimental chapters.

4.3.1 Brian Yamauchi and Randall Beer

Yamauchi and Beer (1994a, 1994b) were the first to explore the idea of using dynamical recurrent neural networks without synaptic plasticity in tasks that required learning.

In their paper titled “Integrating reactive, sequential, and learning behaviour using dynamical neural networks” (Yamauchi & Beer, 1994a), they explore three different tasks, all of which fall on different levels of behavioural plasticity: landmark recognition, one-dimensional navigation, and sequence learning.

In their first task, landmark recognition, an eight-neuron dynamical neural network was evolved to identify one of two different objects. One neuron is designated to signal which of the two objects it has sensed. The sensor is such that a single reading is not sufficient to differentiate between the two objects. Thus, the task requires the integration of perception over time. Circuits were successfully tested on a physical robot. How to signal each of the objects does not change either from task to task, nor within the lifetime of the agent (i.e., perception of object A is always signalled by turning output neuron on and object B is always signalled by turning output neuron off). Thus, for our purposes, we will not categorise this task as requiring learning behaviour (they don’t do so
Chapter 4. Learning without synaptic plasticity

Tasks that require learning behavior
- Evolving dynamical system models that ‘learn’
- Use of models with plastic synapses
- Use of models without plastic synapses
- Focus on the relation between evolution and learning
- Focus on the mechanisms responsible for the learning behaviour

Tasks that don’t require learning behavior
- Task set-up is mostly ecological
- Task set-up is mostly abstract
- Stimuli to be remembered discrete
- Stimuli to be remembered on a continuum

Focus on the mechanisms responsible for the learning behaviour

Figure 4.1: Relevant dimensions of related work on evolving dynamical system controllers that ‘learn’. The black bubble in the centre represents the main theme of this thesis. White bubbles represent the areas that previous work has traditionally focused on. The bubbles shaded grey represent the dimensions that this thesis develops further.

either).

Their second task involves a one-dimensional environment with two objects: a goal and a landmark (see Figure 4.2). The relation between the goal and landmark can vary from trial to trial, so that there are two types of environment: landmark-near and landmark-far. The agent starts in the centre of the one-dimensional continuum, and the goal is positioned randomly at either the left or the right end. The landmark is positioned on the same or opposite side of the goal, depending on whether the agent is in a landmark-near or far environment, respectively. A successful agent needs to learn which of the two types of environments it is in, based on the location of the landmark, so as to navigate most efficiently towards the goal.

Yamauchi and Beer failed to evolved a single network to solve this learning task. So they resorted to evolve separate networks for subtasks, such that when combined would produce the original target behaviour. They evolved one network for landmark-far environments only, one for landmark-near ones, and one for environment classification. They used the output of the classifier network to activate or deactivate the landmark-near and far networks, depending on the environment.

In their last task, they attempt to answer whether a single network can switch between several different modes of behaviour based on experience. To answer this, they make the task more abstract, so the embodiment of the network is no longer considered. The task is, nevertheless, inspired on a rat attempting to navigate a binary maze. At each intersection, the rat can choose to go left or right. Reinforcement is offered for each correct decision. This is abstracted into a problem of learning a sequence of \( n \) binary digits, each representing a junction in the ‘maze’.
4.3. Evolving dynamical system controllers that learn

They managed to evolve a 5-node network on the one-bit sequence task, which basically has to generate an on or off signal depending on whether it receives a reinforcement or not, and the current state. The successful network adopts an output state of 0, and if reinforced, then remains at 0; otherwise it outputs 1 and remains like this permanently.

They go on to analyse this network using dynamical systems theory in Yamauchi and Beer (1994b), which we summarise here. When no sensation is present, the dynamics of the circuit are bistable (two equilibrium points), with the output neuron either ‘on’ or ‘off’. The network always starts in the ‘off’ state; receiving reinforcement allows it to stay in this state. The absence of the reinforcement signal makes the state of the agent jump to the ‘on’ region. From this attractor, none of the other inputs allow it to escape its basin. In terms of behaviour, the circuit assumes it is in one of the two environments, and changes when it doesn’t receive reinforcement the first time. This explains why the network is not permanently plastic.

In their second work, Yamauchi and Beer (1994b) also evolve and analyse one 5-node network that can switch between the two ‘environments’ during its lifetime, as opposed to only at the start of the trial (by including changes of environment during the circuit’s lifetime during evolution). The difference in the learning behaviour is between an initial critical plastic period for the former and permanent plasticity for the latter. The dynamical analysis of the network reveals the presence of a limit cycle in the absence of sensation phase of the task. Crucially, the limit cycle alternates between the on and off state of the output neuron. In one of the environments, the reinforcement helps keep the state of the network always on the off part of the cycle. In the absence of the reinforcement, the network moves to the on state, in appropriate time for its response to be evaluated. The reinforcement in this case, still leaves the state sufficiently close for it to remain in the on part of the cycle. Unlike their first circuit, absence of the reinforcement signal switches the state of the network to the opposite part of the cycle. Thus, the limit cycle (as opposed to the two equilibrium points) helps explain why the second network can maintain its plasticity throughout its lifetime.

Attempts to evolve a two-bit learner directly again failed, so the task was again divided, with...
the two successful networks for each subtask combined to produce a solution. Some success is reported on the three-bit sequence learning experiments, but only for a very limited subset of the three-bit sequences. In fact, all of the three bit sequences that were employed for the task were related by a phase shift. So, effectively, it was always the same pattern that had to be produced, with no real change in the behaviour required.

4.3.2 Elio Tuci, Matt Quinn, and Inman Harvey

Tuci (2003) also investigated associative learning in networks without synaptic plasticity. Their focus was on the evolutionary factors that facilitated the evolution of associative learning. We will review two of the most relevant experimental results described in Tuci (2003).

In the first of them, the interest was to replicate Yamauchi and Beer’s one-dimensional landmark experiment (see Section 4.3.1) (Tuci, Quinn, & Harvey, 2002a). Remember that for this task a successful agent has to learn in which of two environments (landmark-near or landmark far) it is in during its lifetime. The agent is given several trials to learn the relation between the landmark (light gradient) and the goal. The main difference being that they would only settle for an “integrated” controller (i.e., single network, as opposed to a mixture of several subnetworks evolved on parts of the task).

Although the associative learning part of the task was essentially the same, there were several differences in the experimental set-up that made the task arguably more ecological. The major differences were: (a) instead of a one-dimensional arena they used a two-dimensional arena (see Figure 4.3), (b) instead of a vehicle on a train-like track with two neurons for opposing left and right forces and dedicated goal and landmark sensors, they used a khepera-like robot simulation of movement and sensing, (c) instead of using a landmark that could only be sensed locally, they used one that could be sensed throughout the arena. There were also differences in the fitness function. The two most important were that: (a) fitness was more gradual (points were given for getting close); as opposed to an all or nothing fitness evaluation, and (b) trials did not terminate if the agent incurred in the wrong behaviour.

However, similar to the work from which they were drawing their inspiration, their fitness function rewarded equal amounts of score for each of the four possible situations an agent could find itself in (see Figure 4.2). And also similarly, they failed to evolve controllers that could learn to exploit the relation between goal and landmark. No agent did significantly better than 50% in either of the landmark-near or far environments. Analysis of the behaviour of the failed agents showed that none were “paying attention to the landmark” (i.e., neither going towards nor away from the light source).

As the fitness function was devised, an agent that headed towards the light source (or in the opposite direction) would receive an equal score to an agent that always went left (or right, or indeed uniformly random to either left or right). In a second set of experiments, such fitness symmetry was broken by allowing the trials in landmark-near environments to contribute to a larger proportion of the total fitness, 75%; whereas the trials in landmark-far environments would contribute a mere 25%. The justification was to increase the selection pressure towards the relevance of the landmark. Then, whether in any environment the agent had to go towards the landmark or away from it was more a matter of fine-tuning.
Figure 4.3: Replication of associative learning task in a more ecological setting. A two-dimensional arena with a goal and a landmark (light source). The relation between the goal and landmark can vary from trial to trial, so that there are two types of environment: landmark-near and landmark-far. Figure reproduced from Tuci (2003).

Successful circuits were obtained once the fitness symmetry was broken. And this is for us the main interest in this work (Tuci et al., 2002a). Because the focus of their work was on the conditions that facilitated learning to evolve, we know very little about the mechanisms that produce the learning behaviour in these 13-neuron dynamical network controlling the khepera-style robot. What we do know is that there was a critical learning period for the successful agents after which their plasticity disappeared. Similar to the first sequence-learner analysed in Yamauchi and Beer (1994b), one type of transition (e.g., from landmark-near to landmark-far) but not the other was possible during their lifetime. This was expected for us (although it seemed they were disappointed from that result), as no transitions had been included during evolution.

Their second work we will review more lightly. It is based on work in Tuci, Quinn, and Harvey (2002b). Their interest is in obtaining a network that learns without having to break the symmetry of the fitness function. Although the task remains the same, the neural network model, genetic algorithm, and the genotype-encoding (which now allows for variable number of neurons and connections) scheme changed. The reason for changing the network, for example, is in hope that a different dynamical system will be more likely to do light seeking or avoidance than the previous one. But similar to the rest of the changes, we do not feel there is a clear justification as to why (besides getting lucky) this would be the case (any more so than in the previous set-up). Yet the experiments did succeed, even with the symmetric fitness function. So while the aim of the study was accomplished, given the multiple simultaneous changes in the set-up, we were left knowing very little about the reason for the improvement. This caveat is also recognised by
the authors when they say that “The results of the simulation in this chapter certainly leave some open-questions which would require further investigations” (Tuci, 2003, p. 145). We agree.

In all of the work presented here, there is some emphasis on making the tasks “more complex”. For example, Tuci says: “From the description above, it should be clear that this task is more complex than that designed by Blynel and Floreano (2002)” (Tuci, 2003, p. 152). It is undeniably true that all of their tasks require more elaborate sensory-motor coordination, because the agents move freely in a 2D arena, as opposed to back and forth on a one-dimensional rail. However, for us it is unclear whether this change towards more elaborate sensory-motor coordination makes the task easier or harder for the agent. Their work seems to suggest that it makes the task harder, but our intuition is the opposite (see discussion on embodiment in Section 2.3). More importantly, however (and regardless of whether the task is made harder or easier), what is certainly the case is that changing the set-up in so many dimensions as they did makes it harder for us to make a clear comparison with the work that they are extending/criticising, and to evaluate the details that make the difference.

4.3.3 Phattanard Phattanasri, Hillel Chiel, and Randall Beer

Phattanasri, Chiel, and Beer (2007) also investigated associative learning in networks without synaptic plasticity. Their work focused more centrally on understanding the evolved mechanisms using dynamical systems theory. The associative learning task is a similar but abstract version of the landmark navigation task. A successful network has to learn to associate food with one of two possible smells. But there were also a number of differences in their set-up, possibly influenced by their motivation above, such that the task was made abstract, less ecological. In this respect, the task also takes inspiration from the \( n \)-digit sequence-learner that they had worked on previously (and on which they managed to better grasp an understanding of the dynamics of their resulting circuits).

Their experiment is based on associative learning studies by Chiel and Susswein (1993). Also inspired by the color-learning eater in Miller and Todd (1990). Basically, in a world with edible and inedible food, an agent has to eat as much of the edible and as little of the inedible food as possible. The agent can sense the food by its smell. There are only two smells possible in this world. The agent can also sense whether the food last eaten had been edible or not, via a reinforcement signal that can be positive or negative, respectively – a sort of “gut” sensor.

If edible food could reliably be distinguished from inedible food always during the lifetime of the agent, then no learning behaviour would be required. However, the smell of the edible and inedible food is sometimes “switched around”, within trials of evaluation of the same agent (and thus within the timescale of an evolutionary generation). The agent is, therefore, unable to rely only on its sense of smell to recognise which food to eat and which to avoid eating. A successful agent has to also pay attention to what its gut sensor tells it after it has eaten a food with one of the two smells, to find out which of the two smells corresponds to the edible foods.

The task follows a schedule of smells, evaluations, and reinforcements, with delays in-between (see Figure 4.4). The agent is a disembodied neural network that receives sensory stimulation according to a schedule (with some variability) and similarly the activity of its designated output (i.e., mouth) is evaluated at certain established times. Although it is not very ecological, it resembles
4.3. Evolving dynamical system controllers that learn

Figure 4.4: Disembodied associative learning task by Phattanasri, Chiel and Beer. Figure reproduced from Phattanasri et al. (2007).

conventional animal experiments as carried out in the labs.

Also similar to the rest of the work reviewed here, not including transitions between the two environments during evolution resulted in agents that wouldn’t necessarily maintain their plasticity during their lifetime. They would manage a transition from one environment to the other, but not in the other direction.

The sequence of a single task can comprise several tests and transitions. After conducting a series of preliminary experiments, they noticed that it was more difficult to evolve successful circuits on a task with large number of trials and transitions between environments directly. So they applied an incremental evolution technique (see discussion on Section 3.2.4). Populations started with the simplest version of the task and as they succeeded, the task was made more complicated.

In their work, an effort to evolve the smallest possible circuits that managed the learning task was made. In one case, after observing that a successful six-neuron circuit solved the task using only four of its neurons, experiments with three and four-node circuits were attempted successfully. The study of the evolved parameters of successful agents indicates that there is at least one neuron with a longer time constant than the rest.

For this task, as for all other tasks examined in this chapter, the agent has to be able to modify its behaviour between two different environments. In this task, it is the negative reinforcement feedback that triggers the modification of the agent’s behaviour.

They then consider the dynamics of the best evolved three-node circuit in some depth. First, using the insight from the behavioural studies, they summarise the behaviour of a successful circuit using a finite state machine. Second, they use a ‘strobing’ technique to detect where the state of the system is at different crucial points of a learning sequence. This technique refers to observing the state of the system only at particular points in time. This can, in certain occasions, simplify the analysis of a continuous-system. The name comes from the metaphor of studying the system in the dark while using a strobe light at specific points in time. Third, they use the strobed states to relate the finite state machine abstraction with the operation of the circuit. They do this by matching the states of the FSM with the inner states of the circuit resulting from the strobing
analysis, as well as the transitions from state to state. Fourth, they study the phase-portraits of the circuit in each of the different stimulus conditions. The behaviour of the successful circuit can be understood as switching between two groups of basins of attraction — one group of basins for each of the environments. They also show how the negative reinforcement caused the system to switch between these two groups of basins. Finally, they observed that the state trajectory seldom reaches an attractor before the arrival of a new external input. In other words, the operation of the agent is based on the transient states.

4.4 Other related work

4.4.1 Comparing the evolutionary performance of networks with and without synaptic plasticity for learning behaviour

Blynel and Floreano (2002) compare the evolutionary performance of dynamical recurrent neural networks with and without synaptic plasticity on a task requiring learning behaviour. The neural network without synaptic plasticity is a CTRNN. The neural network with synaptic plasticity is a discrete-time version of the CTRNN, with synaptic plasticity implemented as one of four possible Hebbian learning rules with evolvable learning rates (Floreano & Urzelai, 2000). They compared the evolutionary progress of both networks on two tasks. However, only the first would appear to require learning behaviour. The task is very similar to Yamauchi and Beer’s one-dimensional landmark-near or far navigation. Counterintuitively\(^1\), their results suggested that networks without synaptic plasticity produced better performing agents than networks with synaptic plasticity. There are two main reasons why it is hard to conclude anything strongly from their result. First, the comparison was not ideal (for example, the network with synaptic plasticity had nearly three times as many synapses as the network without synaptic plasticity). Second, in their version of the task, since the landmark is fixed to one side of the arena and the goal is the only thing that changes, it is possible for the agent to employ a reactive turn left or right strategy, as opposed to switching between approaching or avoiding the landmark; making it unnecessary to form an association between landmark and goal.

Tuci and Quinn (2003) reviewed and extended the work by Blynel and Floreano. The major difference in their approach was that the agent was free to move in a 2-dimensional arena. However, the difference between the performance of the controllers with and without synaptic plasticity was more stark: with 14 out of 20 evolutionary runs resulting in successful learning agents for the controllers without synaptic plasticity and none for controllers with Hebbian learning mechanisms. However, similar to Blynel and Floreano’s experiment, it is hard to conclude anything from the difference in evolutionary success, since it could be that the range of learning rates available weren’t appropriate, or that a larger selection of synaptic plasticity rules be available altogether for the networks with synaptic plasticity to evolve successfully. Another reason that makes the comparison less useful is the other differences between the networks, besides the synaptic plasticity. In particular, the network with synaptic plasticity is a discrete-time dynamical system, whereas the CTRNN is a continuous-time one.

Finally, in Phattanasri (2002) evolutionary experiments were also run in their associative learning task with networks with plastic synapses. One difference between their work and that in Blynel

\(^1\)At least for those assuming that changes in the strength of the synapses equates with learning behaviour.
and Floreano (2002), Tuci and Quinn (2003), was that both networks (with and without synaptic plasticity) were CTRNNs. When they constrained the learning rates for the synaptic plasticity rules to be in a range such that their dynamics was slower than that of the activity of the neurons, they failed to obtain successful circuits. It is fair to mention, that failure was possibly expected for the task at hand, since in their task learning amounts to switching behaviour when negative reinforcement is applied. So the learning process must occur relatively fast.

There have been other works reporting similar results when comparing the evolutionary performance of networks with and without synaptic plasticity mechanisms, but mainly in tasks that do not actually require learning behaviour (e.g., Stanley, Bryant, & Mikkulainen, 2003).

### 4.4.2 Unexpected role of synaptic plasticity in learning behaviour

When learning behaviour is observed in a network that also employs variable strength synapses, it is generally assumed that the changing synapses are doing the learning. Little attention has been actually paid to the seemingly crucial question: do the activities of the synapses conform with the traditional view of learning, with changes in the synapses occurring only while the network is learning and remaining stable thereafter to remembering whatever was learnt until needed? As we will see ahead, in the few examples where researchers have probed deeper, this has not been what has been observed.

Phattanasri (2002) also attempted to evolve neural networks with synaptic plasticity on the same two-environment associative learning task with success. The number of variables in the resulting successful circuits was, however, much larger - with 6 varying neurons and 30 varying synapses. The result was that all of the synapses evolved to be active at very fast timescales, not very different from the timescales of the fastest neurons in their counterpart fixed-weight networks. They also ran evolutionary experiments forcing the learning rate of the synaptic rules to be much slower but failed to obtain successful circuits. Failure was possibly expected for the task at hand, because learning requires switching behaviour when negative reinforcement is used.

Fernando (2002) explores a very similar two-environment associative learning task in a slightly more complicated T-maze arena. A CTRNN with additional rules to generate plasticity in the synapses was employed as the controller of the robot. Despite not being able to evolve an agent that solved the task completely, an analysis of the best performing agent, using concepts from animal learning theory, was attempted. His motivation was to demonstrate the limitations of such a method. He used a khepera-like simulated robot in a T-maze arena (see Figure 4.5). Food could appear either towards the left or right end of the top arm of the maze. The food could only be sensed in the near-vicinity. The task is essentially the same as Yamauchi and Beer’s one-dimensional navigation task and Tuci et al’s ecological extension: the association between two binary stimuli. The difference is in the “ecological degree”, falling somewhere between Yamauchi and Beer’s least ecological scenario and Tuci’s most ecological scenario. While the agent is “freer” to roam in the environment, the shape of the environment (T-maze) constrains the possible movements. This was, in fact, an intentional move from the designers of the first T-mazes assays for the study of rat learning (Watson, 1924).

Similar to the work in Phattanasri (2002), analysis of the most successful agent suggested unexpected roles for the Hebbian weight changes. Primarily, despite having available a broad
range of learning rates, they evolved to be as large as possible. Thus, weight changes occurred at a similar timescale to the activity of the neurons. As a result, no “memory trace” could be reliably linked to any of the synaptic strengths. Instead, a greater proportion of such a trace could be said to reside in the activity of the nodes themselves.

4.4.3 Learning behaviour as changes in other network parameters besides synaptic strength

Synaptic plasticity is usually the only parameter associated with learning behaviour. But plasticity in the membrane excitability (i.e., threshold or bias) of the neuron or in its rate of decay (i.e., time-constant) is also possible. We briefly review here three works that consider networks with plasticity in a combination of these parameters. Our work differs from all of these in that we do not implement plasticity in any of those three parameters to achieve learning behaviour successfully.

Inspired by the plasticity mechanisms suggested from studies in *Hermissenda* (Alkon, 1983), Tesauro (1998) developed a theoretical model of a neural network where the learning behaviour takes place as a consequence of the online modification of the properties of the neurons (i.e., threshold parameter), instead of the network’s synaptic strength values. His motivation is similar to ours in that he is questioning the necessity of modifiable synapses to achieve different forms of learning behaviour, but it is different in all other respects. Mainly in that we do not implement additional rules for either variable synapses or variable thresholds. In fact, the variability in the threshold that is implemented in Tesauro’s model can also be produced with a model lacking such prescribed modification rule. If we suppose that there are additional neurons connected to the neuron whose threshold should change, then slow changes in the activation of these neurons will effectively lower or increment the threshold of the neuron with respect to the rest of its incoming connections.

Inspired by homeostatic plasticity observed in neurons from the stomatogastric ganglion of decapod crustaceans (Turrigiano, 1999), Williams (2006) developed a model where changes in the synaptic strength as well as in the membrane excitability (i.e., threshold) take place. The motivation of their model is to stabilise the firing of a neuron as its input from other neurons
is varied from the modulation of the synaptic strengths connecting them. In the context of our work, however, no experiments were performed to study the role of such stabilising mechanisms on learning behaviour.

Vickerstaff (2006) aimed to take a step further by making all parameters in the neural network model variable, including synaptic strengths, thresholds, and time-constants. In practise, this meant converting only synaptic strengths and time-constants into variables, given that modification of the threshold is equivalent to modification of all the incoming connections to a neuron and is thus implicitly included. A network was hand-designed to perform a vector navigation task, where an agent has to remember not only the nest, but every previous location visited (e.g., beacon) on its way. The design included a “latching” mechanism that the agent used to remember the location of the food. The idea of the “latching” mechanism is that the time-constant of a neuron changes from being very small, and thus quickly modifiable, to effectively infinite, and thus able to retain its value indefinitely. This is, essentially, analogous to storage in a computational system. To achieve the vector navigation task, the system is said to “require” the latching feature of the new controller (Vickerstaff, 2006, p. 138). We believe, however, that this is misleading. Perhaps hand-designing a functionally equivalent mechanism to that of “latching” using a network with fixed synaptic strengths and time-constants is “hard”. It may also be nontrivial to evolve. But, in principle, neither designing by hand nor evolving one should be impossible. Most importantly, whether such “latching” mechanism is required for the vector navigation task or not is not clear from their studies. Our work is different from theirs in that our interest is to find non-obvious ways in which remembering mechanisms can be instantiated in dynamical system controllers, without the need to create additional special mechanisms for them.

4.4.4 Implementing known learning algorithms in fixed-weight networks

In Cotter and Conwell (1990), Younger, Conwell, and Cotter (1999), neural networks with fixed weights are hand-designed to incorporate one of several different learning algorithms, including a gradient descent algorithm and backpropagation. For an example of a network designed to implement the latter see Figure 4.6. The weights in their networks specify the learning algorithm. The networks have been tested on Boolean logic functions (e.g., and, or, xor). In their work, choosing the values for the fixed weights amounts to learning how to learn. This is somewhat similar to our work, except for a number of important factors. First, learning is achieved through the full network configuration, including neurons’ thresholds and time-constants, not just the synaptic weights. They dedicate a set of additional neurons in the network to implementing the traditional learning algorithm of choice, which they call the auxiliary network. Thus, our work also differs in that their networks still follow the classical clear-cut distinction between the behaviour-producing mechanisms and the learning-producing mechanisms, despite both of them being implemented using neural dynamics instead of synaptic dynamics. Second, their systems are hand-designed, while we use a search technique. Finally, we are interested in learning as a behaviour, not as a mechanism. While their network includes an in-built backpropagation algorithm, whether it performs learning behaviour when situated and embodied is not studied. Our networks are selected based on their interaction with their environment, regardless of what the internal modulatory mechanisms are.
Chapter 4. Learning without synaptic plasticity

Figure 4.6: A hand-designed neural network with fixed weights that implements a three-layer feed forward network with an in-built backpropagation algorithm. Figure reproduced from Younger et al. (1999).

4.4.5 Continuous attractor neural networks models

Continuous attractor neural networks have been proposed by Amari (1977), Seung (1996), Ermentrout (1998). This is basically a network with parameters appropriately tuned to display a “line attractor” (see Figure 4.7). The motivation for the development of their model is to provide an explanation for how a memory of a continuous feature could be stored in the brain. For example, Seung (1996) investigates how the brain can hold the eyes still between saccades and proposes that it stores a memory of its position in the synaptic strengths. It has been shown how linear attractors can be implemented in a linear system and approximated in nonlinear ones (Seung, 1998b, 1998a; Stringer, Rolls, & Trappenberg, 2004; Boucheny, Brunel, & Arleo, 2005).

The downside of such “linear attractor” dynamics is that they cannot be perturbed without loosing its ‘memory’, given that the state of any one of the fixed points along the line is structurally unstable (e.g., imagine small movements of a pebble that is sitting at the bottom of the U-shaped landscape in Figure 4.7). The main difference between our work and theirs is that, to achieve the line attractor in their neural network model, mechanisms of synaptic plasticity are required. Their model also requires that the system be linear for the line attractor to be effectively continuous. It also requires that the time-constants of some of the components in the network be infinitely slow. Also their model does not involve feedback between the agent and environment, and has not been tested on a situated and embodied behavioural learning task. All the previous mentioned issues are the main reasons why the continuous attractor neural network model is not consider any further in this thesis.
4.4.6 Describing the activity of recurrent neural networks as state machines

A finite state machine (FSM) is a model of behaviour composed of a finite number of states, transitions between those states, and actions. The idea of a state machine was first introduced by Turing (1936). They are mathematical models that serve as approximations to a physical or abstract phenomenon. In discussing FSMs, it is important to understand the distinction between the implementation of a particular model in hardware and the model itself. FSMs are abstract representations from which the properties and limitations of a system can be studied. There is a developed body of discrete mathematics (automata theory) that studies FSMs. For an introduction see Gill (1962), Hopcroft and Ullman (1979). There are two key assumptions about the nature of the system under observation that must be made, when using an FSM to describe them:

**Time-discreteness:** Every system representable by a finite state machine is assumed to be controlled by an independent synchronizing source. Thus, the behaviour of the system at any sampling time \( t_v \) is independent of the interval between \( t_v \) and \( t_{v-1} \).

**Alphabet-finitude:** Each variable can assume only a finite number of distinct values (which may or may not be numerical in nature). As such, the state of a finite-state machine at any given sampling time is that variable which, together with the input to the system, enables one to predict the output symbol at this sampling time and the variable at the next sampling time. The assumption that the number of possible internal states is finite is fundamental to the theory.

Cleeremans, Servan-Schreiber, and McClelland (1989) was one of the first to explore the relationship between recurrent neural networks and finite state machines. A ‘simple recurrent network’ (Elman, 1988) was trained to predict successive elements of a sequence. It was trained
Chapter 4. Learning without synaptic plasticity

with examples taken from a finite state grammar, making the network a ‘finite state recognizer’. When the network was constrained to a minimal number of hidden units, it was possible for the patterns of these units to correspond to the nodes in the grammar. In other words, the ‘neural’ implementation would closely approximate the FSM model.

There has been a number of other work training recurrent neural networks to behave like deterministic FSMs (e.g., Pollack, 1991; Giles, Miller, Chen, Chen, Sun, & Lee, 1992; Kolen, 1994b, 1994a; Casey, 1996; Omlin & Giles, 1996; Tino & Koteles, 1999; Rodriguez, Wiles, & Elman, 1999). The most interesting of these is the work by Tino, Horne, and Giles (1995), where dynamical systems theory is used to study the relation between the activity of the network and the FSM models that they instantiate. The main difference between these works and our own is that the majority of it has been carried out in the context of natural language processing, and with the motivation to find ‘internal representations in the brain’ of the grammars being produced. None of that work has taken into consideration situated and embodied agents involved in ecological tasks requiring learning behaviour. We address some of the difficulties involved in representing a situated, embodied, and dynamical agent’s behaviour using FSMs in Chapters 6 and 8.
Chapter 5

Hebbian learning using fixed-weight networks

This chapter shows that applying the Hebb rule on the weights of a nonlinear dynamical system controller does not necessarily lead to strengthening the correlation of firing. It then demonstrates that a dynamical system controller with fixed parameters can exhibit Hebbian learning behaviour. An explanation in terms of the structure and dynamics of the best-evolved system is given. Learning is shown to arise from the interaction between the multiple timescale dynamics. In particular, we show how the fast-time dynamics alter the slow-time dynamics, which in turn shapes the local behaviour around the equilibrium points of the fast components by acting as a parameter to them. Finally, the time-scales of all successfully evolved agents are analysed and generalisations about what is required for Hebbian learning to occur is given. This chapter lays the foundation for much of the rest of the work on ‘learning without synaptic plasticity’.

5.1 Introduction

Among the insights attributed to Hebb (1949) are ideas about how the connection between two neurons should strengthen or weaken according to the correlation of their activity. The main principle behind his reasoning is that if one neuron stimulates another repeatedly then the strength of their link should increase. Hebb’s postulate makes intuitive sense when the two components that are connected behave linearly. That is, the more one is active the more the other is as well. In fact, already half a century before him, James in his Principles of Psychology had written:

When two elementary brain-processes have been active together or in immediate succession, one of them, on reoccurring, tends to propagate its excitement into the other. (James, 1901, p. 566, italics in original)

Hebb famously wrote:

When an axon of cell A is near enough to excite cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A’s efficiency, as one of the cells firing B, is increased. (Hebb, 1949, p. 62)

Hebb’s law is most often paraphrased as “Neurons that fire together wire together”.
Following Hebb’s insight a plethora of neural network models has been proposed to account for learning behaviours. Generally, these models postulate that learning takes place by modification of the efficacy of individual synapses; or similarly the modification of other parameters in the system. In Tesauro (1998), for example, the learning mechanism suggested is the modification of the neuron’s threshold parameter, instead of the network’s synaptic strength values. This approach assumes that the learning-producing mechanisms and the rest of the internal mechanisms are separate. It also assumes that changes to the weight in the connection between two nodes will lead to changes to their correlation. Although this is the case in simple linear systems, as we will see in this chapter, this is not necessarily the case in more complex nonlinear systems.

As discussed in Section 4.4.2, when networks with synaptic-plasticity have been evolved on tasks requiring learning behaviour, the role that the synapses have played has been different from what is traditionally expected (e.g., Fernando, 2002; Phattanasri, 2002). The motivation for the work in this chapter is first to understand the effect of variations in the connection strength on the correlation between two nodes in a nonlinear system. Second, we are interested in knowing whether a fixed-weight network can generate Hebbian learning-like behaviour. We are also interested in knowing whether it is possible to synthesize such systems using simple evolutionary techniques. If it is possible, then we would like to understand the evolved internal mechanisms.

In order to tackle these questions, we artificially evolve CTRNNs to perform a Hebbian learning task. The details of the methods are described in Section 5.2. In Section 5.3, we report on the evolutionary success for different sized circuits. We also describe the performance of the best and smallest successful circuit, providing some detail on the analysis of the evolved internal mechanisms. Towards the end we discuss the implications of our results for the different views on what CTRNNs represent.

5.2 Methods

We use evolution to synthesize a continuous-time recurrent neural network that implements some version of a Hebb rule. The conventional rule is as follows: 2 nodes A and B, if the activations between these two nodes are highly correlated the weight of their connection increases; if not, it decays. However, we do not want to explicitly specify this in terms of what happens to the weights of the connections, since this carries built-in assumptions about the mechanisms and architecture required to do so. We want to define the task at a ‘higher’ behavioural level.

5.2.1 Hebbian learning task

Suppose we have a black box and we have access to two of its variables: A and B. We can perturb these using time-varying inputs. The mechanism we seek is such that: (1) If there is no long-term correlation observed between the activity of nodes A and B, then experimental perturbations to A (in the absence of any perturbation to B) have no (or little) discernible influence on B. That is, the activation of nodes A and B become uncorrelated. (2) But if there is some long-term correlation between their activities, a stage is reached where experimental perturbations to A (in absence of any perturbations to B) have a correlated influence on the activation of B.

The task has two phases (see Figure 5.1). During the first phase time-varying perturbations are applied to the input of nodes A and B. This phase lasts 100 units of time and we will refer
5.2. Methods

Figure 5.1: The structure of an individual Hebbian learning trial. The trial starts with a random delay, where no input is applied. Then, inputs are applied to nodes $A$ and $B$ (training phase). The inputs can be correlated or uncorrelated. In the example shown, the perturbations have different frequencies. Here we show the minimum and maximum frequencies used: $f_A = 1$ and $f_B = 2$. This is followed by another delay. Perturbations are then applied to node $A$ only (testing phase), while the correlation between the activities of $A$ and $B$ is evaluated.

to it as training. During the second phase a time-varying perturbation is applied only to node $A$, also for 100 units of time. This phase we will refer to as testing. Before each phase a time delay is introduced where no perturbations are applied. The length of the delay is chosen at random between $[10, 20]$ units of time. The time-varying input is a sine wave, with variable frequencies: $I_t = \sin(k ft)$, where $I_t$ is the external input at time $t$; $k$ (0.2) is a constant of proportionality; and $f$ determines the frequency of the wave, chosen uniformly at random between $[1, 2]$. The perturbations to the system during training can be of two different types: correlated or uncorrelated. For the correlated training, sine waves with similar frequencies are used, randomly chosen every time so that $|f_A - f_B| < 0.05$, where $f_i$ is the frequency given as input to node $i$. For uncorrelated training, perturbations are given with sine waves of different frequencies, also randomly chosen. It is ensured that frequencies are not too similar during uncorrelated training by constraining $|f_A - f_B| > 0.2$.

5.2.2 Dynamical neural network

For the ‘black box’, we use a continuous-time recurrent neural network (CTRNN) with the following state equation (Beer, 1995b):

$$
\tau_i \dot{y}_i = -y_i + \sum_{j=1}^{N} w_{ji} \sigma(y_j + \theta_j) + s_i I_i
$$

where $y$ is the activation of each node; $\tau$ is its time constant; $w_{ji}$ is the strength of the connection from the $j^{th}$ to the $i^{th}$ node; $\theta$ is a bias term; $\sigma(x) = 1/(1 + e^{-x})$ is the standard logistic activation function.
function; \( I \) represents the external input (e.g., from a sensor); \( N \) represents the number of nodes in the network; and \( s_i \) is a ‘sensory’ weight for the external input. Only the two nodes that receive external input (\( A \) and \( B \)) have sensory weights. In simulation, node activations are calculated forward through time by straightforward time-slicing using Euler integration with a time-step of 0.1. The network is fully connected.

### 5.2.3 Evolutionary algorithm

The parameters of each circuit (i.e., weights, biases and time-constants for each node) are evolved using a version of the Microbial genetic algorithm (Harvey, 2001). These are encoded in a genotype as a vector of real numbers over the range \([0, 1]\). Offspring of Microbial tournaments are generated as a mutation of the winner of the tournament (without recombination). The mutation is implemented as a random displacement on every gene drawn uniformly from a Gaussian distribution with mean 0 and variance 0.05. Each gene is forced to be in \([0, 1]\): when a mutation takes a gene out of this range it is reflected back. The offspring replaces the loser of the tournament. Genes are mapped to network parameters linearly between \([-10, 10]\) for biases and inter-node weights, and to \([0, 10]\) for sensory weights. Time-constants are mapped exponentially to \([e^0, e^5]\). The size of the population used is 50. We define a generation as the time it takes to generate 50 new individuals. The evolutionary algorithm uses a ‘geographical’ method to allow different subpopulations to evolve semi-independently in ‘neighbourhoods’ within the population (see Section 3.2.4 for more detail). A minimal 1D wrap-around geography with neighbourhoods of size 10 was used: such that only individuals 10, or less than 10, positions away from each other could compete in tournaments. In principle, this allows for genotypic diversity to be maintained for longer in the population at almost no extra computational cost. Finally, because the fitness is noisy (described below), agents are re-evaluated every time they participate in a tournament.

### 5.2.4 Fitness evaluation

The fitness of a circuit is obtained by: (a) maximizing the correlation coefficient on trials where inputs have similar frequencies, and (b) minimizing the absolute value of the correlation coefficient on trials where the inputs have different frequencies. We define the correlation of two time series, \( Y \) and \( X \), during phase \( p \) (\( p = 1 \) for training and 2 for testing) using the Pearson product-moment correlation coefficient:

\[
r(Y, X, p) = \frac{\sum_{i=ts}^{te}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=ts}^{te}(x_i - \bar{x})^2} \sqrt{\sum_{i=ts}^{te}(y_i - \bar{y})^2}}
\]

where \( ts \) and \( te \) denote the start and end of phase \( p \); \( x_i \) and \( y_i \) are the values in the time series \( X \) and \( Y \); \( \bar{x} \) and \( \bar{y} \) are the average activations of \( X \) and \( Y \), respectively. The correlation is calculated only if both of the standard deviations are nonzero; otherwise a correlation of zero is given.

The fitness is composed of six parts and it is calculated by evaluating the circuit on \( R \) independent trials. The fitness is, thus, given by the multiplication of the averages of each component as follows:

\[
f = \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot (1 - \bar{d}) \cdot (1 - \bar{e}_1) \cdot (1 - \bar{e}_2)
\]

The first three components measure the correlation between the input signals and the activities of the nodes: \( a = r(A, I_A, 1) \) is the correlation between the input to \( A \) and the activity of \( A \) during
training; \( b = r(B, I_B, 1) \) is the correlation between the input to \( B \) and the activity of \( B \) also during training; \( c = r(A, I_A, 2) \) is the correlation between the input to \( A \) and the activity of \( A \) during testing. We would like to maximize these. Negative values are clipped to 0.

The last three components measure the relation between the correlations of the two nodes during training, \( x = r(A, B, 1) \), and testing, \( y = r(A, B, 2) \), phases: \( d = |x - y| \) is the absolute difference between the correlation of nodes \( A \) and \( B \) during training and testing; \( e_1 = \sqrt{(1-x)^2 + (1-y)^2} \) for trials where similar frequencies are applied we would like the activities of \( A \) and \( B \) to be highly correlated; \( e_2 = \sqrt{x^2 + y^2} \) for trials where different frequencies are applied we would like the activities to be as uncorrelated as possible. These components we seek to minimize.

### 5.2.5 Incremental evolution

During the first stage of evolution, individuals in the population are evaluated once after they are initialized. This is repeated 100 times per circuit \((R = 100)\). During this stage the state of the nodes are initialized to 0 at the start of the trial. Once any individual in the population obtains a fitness greater than 0.75, the task becomes harder. During the second stage, the nodes are initialized at random in \([-10, 10]\) and the task involves evaluating the circuit on two consecutive trials without reinitializing the circuit’s state. To keep evaluation times similar \( R = 50 \), also the variance in the mutation is decreased to 0.01. The third and last stage consists of 5 subsequent trials and \( R = 20 \).

### 5.3 Results

#### 5.3.1 Role of synaptic plasticity in nonlinear controllers

Before we begin the main part of our results it will be useful to understand what is the effect of changing the strength of the connection from node \( A \) to \( B \) on the correlation of their activities in nonlinear dynamical networks. In order to gain some insight into this question, we analysed \( 10^3 \) randomly chosen 2-node CTRNNs from the same range as that used for the evolutionary experiments. For each, we evaluated the correlation between the activities of both nodes for 141 weights connecting \( A \) to \( B \) evenly spaced between -7 and 7. Each circuit is first integrated for 200 units of time from a random initial activation (between \([-10, 10]\)). This allows the system to settle in its long-term state. The correlation is measured for the following 100 units of time, as node \( A \) is perturbed by a sine wave with a randomly chosen frequency. This is repeated \( 10^3 \) times for each circuit and for each weight. The thick black line in Figure 5.2 shows the mean correlation over all generated circuits. As can be observed, when the weight is negative there is likely to be some negative correlation between the two nodes. Analogously, when the weight is positive there is a good chance of some positive correlation. Note that the correlation, on average, does not grow particularly stronger or weaker as the absolute value of the weight changes. Also, note that (on average) the correlation drops to 0 only when the weight is exactly 0.

However, the most relevant insight in this preliminary study is not to be obtained from the average over hundreds of circuits, but in particular examples of these nonlinear systems. The thinner lines show two examples out of all those randomly generated. In the example shown in the thin black line, the activities are most correlated at a particular inhibitory value for the weight, but decreases to near zero correlation when weakened until there is virtually no weight. Notably,
Chapter 5. Hebbian learning using fixed-weight networks

Figure 5.2: Effect of weight changes on the correlation between node activity. Average correlation coefficient as the weight from node A to B is varied (thick line). Average taken from $10^3$ randomly generated 2-node CTRNNs. Example effects of the weight on the correlation from two of those circuits (thin black and grey lines). The networks at the bottom reflect the changes to the synaptic strength from A to B (black connection) from strongly inhibitory to strongly excitatory.

the activity remains almost completely uncorrelated after the synapse is strengthened. Also, increasing the strength of the inhibitory connection results in slightly stronger negative correlations. In the second example (grey line) the correlation is zero when there is no weight. The slightest excitatory weight makes the activity highly correlated. However, increasing the strength results in less correlation, not more. This is until a minimum is hit and the correlation starts going up again. On the other hand, setting the weight to any inhibitory strength results in a steady strong negative correlation, regardless of the strength. These individual examples show just how complex and unintuitive the relation between increasing/decreasing the weight and the correlation between the activities of both nodes can be.

5.3.2 Evolutionary success using small fixed-weight circuits

Is it possible to evolve a non-weight-changing circuit to perform Hebbian learning behaviour? To answer this question we performed evolutionary runs using 3 and 4 node circuits. No 3-node circuits out of 20 advanced to the second stage. Much more successful were populations of circuits with extra components. After $10^4$ generations, we found $11/20$ 4-node populations that reached the last stage of the incremental protocol. The best fitness of $9/11$ of those populations was greater than 0.78 at the end of the runs. From preliminary analysis, fitnesses greater than or equal to 0.78 are regarded as successful.

Figure 5.3 shows the run that led to the best-performing circuit. During the first hundreds of generations the population's fitness remains in stasis around 0.28. Around generation 1700 the population finds a portal to higher fitness. At generation 3141 (first dashed line) the task changes to the second stage after the best performing circuit in the population reaches the 75% threshold. At this point, the fitness of the best individual (as well as the average) drops significantly. This
5.3. Results

Figure 5.3: Plot of the best (black) and average (grey) fitness of the population versus generation resulting in the best evolved 4-node circuit. Only the first 6000 generations are shown. The transitions between the stages in the incremental evolutionary protocol are marked with dashed lines. Transitions occur when the best fitness exceeds 0.75 (dotted grey line). The evolutionary run shown here is representative of a typical successful one.

means that the circuits in the current population had not generalized to more than one consecutive trial. After less than $10^3$ generations more, the circuit reaches (second dashed line) again the 75% threshold on the task involving 2 consecutive trials. At this point the task changes to 5 consecutive trials and the circuit’s performance does not drop as much, meaning that the circuit generalized well. All of the features mentioned for this evolutionary run are qualitatively similar for the rest of the successful runs. It is the best circuit evolved in this run that we analyse in detail in the rest of this chapter.

The best-evolved network achieved a fitness of 0.83 on a more thorough fitness evaluation performed at the end of its evolutionary run. This involved $10^6$ tests using randomly chosen frequencies, time delays and initial activations on 5 consecutive training and testing trials. This was performed using a time-step of integration an order of magnitude smaller than that used during evolution (0.01) to avoid time integration errors.

5.3.3 Best evolved 4-node circuit

What can we say about the best-evolved architecture? Table 5.1 provides the evolved parameters and Figure 5.4 shows a graphical depiction. There are a number of important things to note. First, components $A$, $B$ and $D$ are fast acting, while the $C$ component is significantly slower. Node $A$ has a high threshold and strongly inhibits itself. Nodes $A$ and $B$ have similarly strong connections to $D$, but one of them excitatory and the other inhibitory. $D$ has virtually no self-connection but a strong excitatory connection to the slower node in turn. The slow node has a very low threshold and highly excites itself. In general, the connections between $A$, $B$ and $C$ are very weak. Thus, most of their interactions are mediated by node $D$. These factors play an important role in the generation of the circuit’s learning behaviour as we will see ahead.
Figure 5.4: A graphical depiction of the parameters of the best evolved network. The shading scheme is similar to that used in (Beer, 2003). Nodes are shaded according to their bias, with higher threshold nodes (which require more excitation to activate) shaded darker. Excitatory connections are shaded black and inhibitory connections are shaded grey, with the width of the line proportional to the strength of the connection. The time-constant parameter for each node is represented by the size of the circle, with larger circles representing slower nodes (integrating over longer periods of time). The slow node (C) should be displayed 7 times larger than the fast nodes (A, B and D), but this is impractical for this figure.

Table 5.1: Parameters for the best evolved 4-node Hebbian learning circuit

<table>
<thead>
<tr>
<th></th>
<th>( y_a )</th>
<th>( y_b )</th>
<th>( y_c )</th>
<th>( y_d )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y_a )</td>
<td>-9.71784</td>
<td>-1.65346</td>
<td>-3.74262</td>
<td>9.76954</td>
</tr>
<tr>
<td>( y_b )</td>
<td>-3.60944</td>
<td>-5.22186</td>
<td>1.3865</td>
<td>-8.73396</td>
</tr>
<tr>
<td>( y_c )</td>
<td>-2.33072</td>
<td>2.80932</td>
<td>9.87038</td>
<td>-5.58786</td>
</tr>
<tr>
<td>( y_d )</td>
<td>-7.41598</td>
<td>7.74034</td>
<td>9.89928</td>
<td>-0.17918</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>( I_a )</th>
<th>( I_b )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_a )</td>
<td>9.0805</td>
<td></td>
</tr>
<tr>
<td>( I_b )</td>
<td></td>
<td>9.0419</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>( \theta )</th>
<th>( \theta )</th>
<th>( \tau )</th>
<th>( \tau )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \theta )</td>
<td>8.4201</td>
<td>0.00044</td>
<td>-5.93522</td>
<td>-0.1288</td>
</tr>
<tr>
<td>( \tau )</td>
<td>1.40292</td>
<td>1.23676</td>
<td>6.97931</td>
<td>1.09622</td>
</tr>
</tbody>
</table>
5.3. Results

Figure 5.5: Example sequence trial. Top two rows show the external perturbations to nodes A and B. The bottom four rows show the sigmoided activity of each of the nodes in the circuit. The areas shaded in light grey depict training phases. The areas shaded in darker grey depict testing phases. Time proceeds from left to right along the horizontal axis. The circuit is first trained using similar time-varying perturbations (1). After a delay, the circuit is evaluated while receiving a different frequency input on A (2). The correlation between the activations of node A and B is 0.98. Subsequently, the circuit is trained on two random but different frequencies (3). When evaluated (4), the correlation between the activities of A and B is -0.02. The ‘behavioural connection’ between nodes A and B has strengthened during the first part of the trial (1, 2) and then weakened during the second part of the trial (3, 4).

5.3.4 Performance of best evolved circuit

What does the circuit do? In order to visualize the circuit’s behaviour, we can record the activity of each of the components of the circuit as well as the external perturbations applied during an example trial. Figure 5.5 shows this in a trial where the input to A and B are first correlated and later when the input is uncorrelated, separated by a dashed line. We can observe how, after correlated perturbations have been applied, the activities of A and B remain highly correlated: every time node A ‘fires’, so does node B. In contrast, after uncorrelated perturbations are applied, the activity of these nodes becomes highly uncorrelated. In the figure we can also see how the slower node (C) adopts different states for each of the scenarios; similarly node D adopts different patterns of activity altogether.

How well does the circuit learn? In Figure 5.6 we visualize the relation between the correlation of nodes A and B during training (horizontal axis) versus the correlation during testing (vertical axis). The figure shows $10^3$ examples using randomly chosen frequencies, time-delays and initial activations for similar (black points) and different (grey points) time-varying perturbations. As can be observed, application of similar frequency inputs generates correlation in the activities of
Chapter 5. Hebbian learning using fixed-weight networks

Figure 5.6: Performance of best circuit. Correlation during training versus correlation during testing; for trials with different time-varying perturbations (grey) and similar perturbations (black).

Figure 5.7: Relearning ability. Each point shows the average correlation coefficient on the $i^{th}$ trial over $10^5$ different runs. Black triangles correspond to the correlation coefficients after correlated training (good performance should be near 1). Grey squares correspond to the correlation coefficients after uncorrelated training (good performance should be near 0). The grey vertical bar represents the range of trials the circuit was evolutionary trained for [1-5].

A and B during training and this leads to correlation during the test phase, even though no input is applied to B. Similarly, presentation of different frequencies generates uncorrelated activity during training, which in turn leads to uncorrelated activity during testing.

There are two other questions of interest in the context of a circuit that learns. The first is to do with the plasticity of the circuit. How many times can the link between A and B re-strengthen and re-weaken? Does the plasticity harden after some trials? Figure 5.7 shows the correlation on the $i^{th}$ trial following a correlated training phase (black triangles) or following an uncorrelated training phase (grey boxes), when all previous trials were randomly chosen to be either correlated or uncorrelated with 50% chance. As can be seen, the circuit is capable of maintaining high performance even after 100 re-learning experiences. This demonstrates generalized behavioural plasticity.

The second question of interest concerns the circuit’s memory. How sensitive is the circuit to longer time delays? We address this by testing its performance on a bigger range of time delays than it was evolved for. Figure 5.8 shows the average correlation coefficient for delays between 0 and 100 units of time. The circuit is evaluated during the first trial $10^5$ times for
5.3. Results

Figure 5.8: Robustness to time-delays. Average correlation coefficient of the activity of node A and B during the testing phase for $10^5$ repetitions with randomly chosen train and test inputs, for correlated input (black line) and uncorrelated input (grey line) using different time-delays between training and testing phases. The grey area represents the range of delays the circuit was evolutionary trained on [10-20].

correlated as well as uncorrelated trials using different initial activations and frequencies of sine waves. As can be seen, the memory for the learnt correlated input does not decay across the range measured. In contrast, as the time range increases after uncorrelated training, the activities of the two nodes cease to be uncorrelated gradually. Eventually, it becomes indistinguishable from the activity after correlated training. In other words, after extended delays, the circuit ‘forgets’ how to ‘behave’ following uncorrelated training; yet ‘remembers’ perfectly well what to do after correlated training. We will come back to this point in the dynamical analysis.

5.3.5 Analysis of the multiple time-scale dynamics

How does this nonplastic circuit produce Hebbian learning behaviour? We know dynamics are occurring at two different time-scales. The inputs change the activations of A and B almost instantaneously because both are fast acting. These change D in turn. We also know that C’s slower dynamics is influenced mostly by the activity of D. In order to understand this circuit’s mechanisms, we need to understand (a) how the fast dynamics influence the slow dynamics and (b) how the changed slower dynamics affect the fast dynamics, in turn. In order to address these questions, we visualize the activity of the circuit in its fast and slow state spaces using a set of trials with different frequencies covering the whole range. We are particularly interested in the differences between the structures of the activities when the perturbations during training are correlated and when they are not.

Figures 5.9ai and 5.9bi show the trajectories of the sigmoided output of nodes A versus B during the last 80 units of time of the training during uncorrelated perturbations (5.9ai) and during correlated perturbations (5.9bi). The shades of grey represent the state of node D for each possible state of nodes A and B. The cancellation of activities from A and B during correlated activity means that node D is maintained in a mid-level of activation (5.9bi). In contrast, during uncorrelated activity node D oscillates with the difference between nodes A and B (5.9bi).

How does this affect the slower component? Figures 5.9a1i and 5.9b1i show the activity of C throughout each of the trials, the light grey region represents the training phase and the darker grey
Figure 5.9: Interaction of multi-timescale dynamics during uncorrelated training (a) and correlated training (b). (i) Activity in output state space of the fast nodes A and B during a set of training examples with (a)(i) different-frequency perturbations to both nodes and (b)(i) same frequency perturbations. The shades of grey represent the state of node D for each possible state of nodes A and B. White represents $D=0$ and the darkest shade of grey represents $D=1$. (ii) Activity of slow dynamics (output of node C) through time for the same set of training examples. The region shaded in light grey corresponds to the training phase and the region in darker grey corresponds to the testing phase. Delays are not shaded. (iii) Activity in state space of fast nodes after training, while node A is being perturbed (testing phase).
5.3. Results

represents the testing phase. During uncorrelated input (5.9aii), the firing of node \( D \) results in the saturation of \( C \). In contrast, during correlated input (5.9bii), the absence of activity in \( D \) allows \( C \) to rest in its off state. As a result of training, \( C \) is left in two different states. More importantly, this difference in state persists during the testing phase (see darker shaded regions in 5.9aii and 5.9bii).

The long-term changes in \( C \) modulate the fast dimensional dynamics in turn. Figures 5.9aiii and 5.9biii show the activity of the fast nodes during the testing phase. Even though the perturbations to \( A \) at this point are the same, the patterns of activity are significantly different. In fact, the only difference in the system is maintained by \( C \). The different patterns correspond to the correlation coefficient in a straightforward manner: activity along the diagonal means that both nodes are firing in phase (5.9biii). This results in an average correlation coefficient of 0.99. In contrast, activity parallel to the horizontal axis corresponds node \( A \) having little or no influence on node \( B \) (5.9aiii), resulting in an average correlation coefficient of 0.01. Finally, it is important to note that the \( C \) component does not only modulate the dynamics of \( A \) and \( B \). It also plays an active role in the generation of correlated activity, as can be seen from Figure 5.9bii.

5.3.6 Dynamical analysis of best evolved circuit

We can use tools from DS theory to further understand the evolved mechanisms. We will focus on two questions: what are the equilibrium points of the evolved system? and how do the trajectories flow into these equilibrium points for each of the different scenarios?

The global dynamics of the evolved system are rather simple: with just one equilibrium point (EP). When unperturbed, this point attracts all trajectories in the phase plane. In the context of the Hebbian learning task, it is near this EP where the system arrives at after training with correlated input. Hence, determining the local behaviour (how the trajectories proceed into this point) corresponds to the system’s performance during the testing phase after correlated training. We will refer to this as the correlated EP.

What about the system after uncorrelated training? Is there not a long-term behaviour associated with this scenario? There isn’t, at least not in the strict sense. This explains why, when longer time delays are introduced, the performance of the system falls back into what it would be had been trained with correlated input. However, it is here where the system’s operation on two rather different time-scales allows it to have a ‘temporary equilibrium point’. Although for the time-scale of the slow node this point is merely a transient, for the much faster subset of the dynamics the state will arrive at this point in the relative long term. From the previous section, we know \( C \) is near one after training with uncorrelated input. Hence, its influence on the states of the fast nodes is nearly constant. We can approximate the relatively long-term behaviour of the system after training with uncorrelated signals by studying the dynamics of the modified circuit that assumes \( C \) is always one. Similar to the correlated case, the reduced circuit has only one EP. This is also a stable attractor, shifted on the \( AB \) plane. We will refer to it as the uncorrelated EP.

While there is no global difference in the dynamics between the two scenarios, we can still ask about the local behaviour around their associated EPs. How do the trajectories proceed in their vicinities? This is determined by the eigenvalues of the Jacobian matrix of the system evaluated at these points (Beer, 1995b). Both EPs are mixed: the trajectories proceed in a straight line to-
wards the fixed point in some dimensions but proceed in spirals in other. Figures 5.10a and 5.10b show the flow around the correlated and uncorrelated fixed points, respectively. It is the qualitative difference in the local behaviour around the relative-long-term behaviour of the two EPs that generates the two different correlations patterns between A and B.

Next we visualize the system’s dynamical properties in a 3D projection of its 4D activation space. Figure 5.10c shows the circuits EPs and the behaviour of the system around its EPs as it is systematically perturbed through input to node A. The perturbations are in the form of positive and negative parts of a sine wave with different frequencies. The black disk represents the correlated EP. Although the uncorrelated EP only has A, B and D coordinates, we can visualize it where the sigmoided output of C nears 1. The grey disk represents this. The trajectory in grey depicts the flow of the system from the uncorrelated to the correlated EP after longer delays. The black trajectories to the left depict the system in activation state space when perturbed away from the correlated EP. The state of the system remains at all times very near a 2-dimensional plane that cuts the A and B plane diagonally. This results in their activity being correlated. As remarked earlier, it is important to note that C plays a role in maintaining the activities of A and B on this plane. The black trajectories around the uncorrelated EP for the same pattern of perturbations show a completely different structure. Most importantly the activity in the B plane is minimized. This results in uncorrelated activity. Hence, the ‘weakening’ and ‘strengthening’ of the behavioural connection between nodes A and B corresponds to the shift between these two distinct regions of dynamics: the global EP (black disk) and the ‘temporary’ EP (grey disk).

5.3.7 Evolved time parameters in successful circuits

Most of the analysis in this chapter deals only with the internal mechanisms of the best-evolved and smallest possible circuit (sections C through F). Is there something in common in the parameters of all successfully evolved circuits? Due to the high-dimensionality of parameter space, we will not attempt to characterize successful circuits in general. Instead, we will focus only on the time-constants because of their relevance in the analyzed circuit. Figure 5.11 shows the average and standard deviation of the evolved time-constants for all successful 4-node circuits. The C node was arbitrarily chosen as the slowest of the two extra components. As can be seen similar to the analyzed circuit, all A, B and D nodes evolved to be as fast as possible. In contrast, one of the extra components (C) is, on average, much slower acting. More importantly, we observed the behaviour of two of these other successful circuits and found much similar mechanisms at play.

5.4 Discussion

In this chapter, we have (a) shown how unintuitive the relation can be, in nonlinear systems, between changing a weight and the correlation between the two nodes connected by that weight; (b) demonstrated that CTRNNs with fixed weights can be evolved to produce Hebbian-like learning; and (c) analyzed the behaviour and dynamics of the best and smallest evolved circuit. Learning is shown to arise from the multi-timescale dynamics. In particular, the evolved mechanism requires two extra components: one fast and the other slow. The fast node fires when the activities of nodes A and B are different, and the firing of that node influences the slow one which then settles into two different persistent states. The changed slow component, by acting as a parameter
Figure 5.10: 3D projection of the circuit’s state space. (a) Flow (grey trajectories) around the correlated EP (black disk). (b) Flow around the uncorrelated EP (light grey disk). (c) The light grey disk represents the system’s temporary EP associated with the state after uncorrelated training. The black disk represents the system’s EP associated with the state after correlated training. Trajectories in black depict the behaviour of the system in activation state space when systematically perturbed away from the equilibrium through input to node A. The grey trajectory depicts the movement between the two points corresponding to the strengthening and weakening of the ‘connection’ between A and B.
Chapter 5. Hebbian learning using fixed-weight networks

Figure 5.11: Evolved time-constants for the successful 4-node circuits. Averages for each of the nodes indicated by the bars and standard deviations with the lines. Dashed grey line indicates the smallest (i.e., fastest) time-constants allowed.

to the fast components, shapes their patterns of activity in turn.

In connectionist artificial neural networks, it is common for all neurons to be constrained to operate at a single time-scale. It is for this reason that longer-term changes to the behaviour of the system have to be introduced as additional parameter changing rules. These rules are generally applied to the weights of the connections. One example is backpropogation learning; another is Hebb’s rule. It is now known that neurons can act over a range of different time-scales, and so can changes in synaptic efficacy. Furthermore, the time-scales of activity of the two can overlap (see Toledo-Rodriguez et al., 2005, for a review). If we think of CTRNNs as models of interconnected neurons, then we are essentially allowing neurons to act in a range of different time-scales. Under this view, this work suggests: (a) that Hebbian forms of learning can arise without synaptic plasticity; and perhaps more importantly: (b) that it can do so via neurons (or sets of neurons) interacting over multiple time-scales. There are at least two ways in which this is feasible in nervous systems: single cell (i.e., changes to their intrinsic properties) and network (i.e., active reverberation in currently connected circuits) mechanisms. Both are likely to co-operate in generating persistent activity (Major & Tank, 2004). The circuit analyzed in this chapter resembles more closely the former.

In the broader ‘CTRNN as a dynamical system’ view, each of the nodes can evolve to instantiate any possible component, with certain intrinsic properties and with particular rules for interacting with the rest of the nodes. Some nodes could resemble, for example, neurons; others: population of neurons, a range of neurotransmitters, synapses, and so on. In fact, the components need not be constrained to at the level of inter-neuronal communication in the central nervous system. As we discussed in 3.1.5, they could also, for example, represent biochemical networks inside a single cell or in plants. In this perspective, our work suggests that Hebbian forms of learning can be present in a broader range of systems, regardless of whether they have synapses and neurons or not. The interesting question becomes: can we extract general principles about the properties of the components required for learning? More importantly still, we believe further understanding of these systems could lead to practical suggestions to the community studying learning and memory in living organisms. Suggestions to study components, processes or even organisms not generally considered, such as unicellular organisms (Hennessey, 1979) and plants (Trewavas, 2003).
This work raises several interesting directions for future research. First, the task could be further developed in several directions. Of particular interest would be to alter the task to allow a continuum of different strengths of correlation. At present, it suffices for the system to have just two modes of long-term dynamics: fully correlated or nil correlated. Also interesting would be to increase the range of delays experienced to encourage the evolution of systems that do not easily forget. Second, we need to understand better how the slower components relate to synaptic plasticity mechanisms. One question that we can ask is, can we extract a weight-changing rule (or any parameter-changing rule) from the evolved dynamics of one of these circuits? Finally, although all of the successful circuits evolved slower-acting extra components, slow components are not the only way to generate slower-time dynamics, as remarked earlier. An interesting direction of research for the future is to encourage the evolution of reverberatory dynamics by evolving larger systems and/or constraining components to a smaller range of time-constants.
In this chapter we study an associative learning task inspired on the thermal-preference behaviour observed in the nematode worm, *Caenorhabditis elegans*. We extend previous work on evolving ‘learning without synaptic plasticity’ from remembering discrete to continuous stimuli. The task is modelled at an abstract level (i.e., non-embodied). It consists in associating paired stimuli: temperature and food. It has been designed so that the temperature to be associated can be either drawn from a discrete set or range over a continuum of values. It also requires that the evolved circuits remain sufficiently plastic to learn new associations and forget previous ones during their lifetime. We address two main questions: can the learning without synaptic plasticity approach be extended to continuous tasks? And if so, how does learning without synaptic plasticity work in the evolved circuits? Analysis of the most successful circuits to learn discrete stimuli reveal finite state machine (FSM) like internal dynamics. However, when the task is modified to require learning stimuli on the full continuum range, it is not possible to extract a FSM from the internal dynamics. In this case, a continuous-state machine is extracted instead.

6.1 Introduction

All of the work on evolving agents that learn without synaptic plasticity has focused on tasks where the agent is required to behave differently in a discrete number of different environments, in practice two (e.g., Yamauchi & Beer, 1994b, 1994a; Tuci et al., 2002a; Blynel & Floreano, 2003; Phattanasri et al., 2007). As discussed in Section 4.3, the learning in these experiments, corresponds to swapping between two different modes of interaction; depending on which environment the agent finds itself in (e.g. going towards the landmark in landmark-near environments or going away from it in landmark-far ones). A different, but arguably more common, form of learning requires the ability to adapt to changes in the environment ranging over a continuum of values. Temperature preferences in *C. elegans* (Hedgecock & Russell, 1975), song learning (Funabiki & Konishi, 2003) and parental imprinting (Lorenz, 1981) in birds, and face recognition in humans (Cohen & Tong, 2001) are a few examples of such learning of continuous stimuli.

In Chapter 9, experiments on learning from a continuous range using circuits without synaptic
plasticity will be provided. In that work, however, the analysis of evolved circuits illustrated how the rich environmental regularities arising from the agent’s situatedness provided a way to ‘offload’ the plasticity to the agent-environment interaction. In this chapter we study a similar ‘associative learning on a continuum’ task, but this time in a disembodied neural network. This constraints the plasticity to the dynamics of the circuit.

The central contribution of this chapter will be to extend previous work on evolving learning without synaptic plasticity from discrete tasks to continuous tasks, where the plasticity arises entirely from the neural network controller. As such, there are two major questions it will address. First, can this approach be extended to continuous tasks? Second, if so, how does learning without synaptic plasticity work in the evolved circuits?

The particular associative learning task, neural models, and evolutionary algorithm that we employ are described in Section 6.2. Section 6.3 presents a ‘baseline’ for the ‘associative learning on a continuum’ work by reproducing the results in Phattanasri et al. (2007) with a different training paradigm and evolutionary technique, but similar task and shaping protocol, and the same neural network model. This section shows that a FSM can be extracted from the functioning of the circuit’s internal dynamics. In Section 6.4, we demonstrate that CTRNNs lacking synaptic plasticity can be successfully evolved to exhibit associative learning for a continuous version of the task. The dynamical operation of the best such circuit is then analyzed in detail in Section 6.5. Finally, Section 6.6 concludes with a discussion of the broader implications of our results and directions for future work.

6.2 Methods

We use evolutionary techniques to synthesize dynamical system circuits for an associative learning task abstracted from one of several learning behaviours studied in the nematode worm *Caenorhabditis elegans*. The behaviour is known as temperature preference\(^1\) (Hedgecock & Russell, 1975). It consists in associating paired stimuli: temperature and food. This paradigm was chosen to be the simplest possible scenario requiring associative learning, yet sufficiently sophisticated to allow for the ‘remembered stimulus’ to be either discrete or continuous.

6.2.1 Temperature preference task

As we are interested in the broader set of possible mechanisms that can give rise to such behaviour, we do not want to explicitly specify the task in terms of the internal mechanisms. This would carry built-in assumptions about the mechanisms and architecture required to do so. We would like to define the task at a ‘higher’ behavioural level, instead.

\(^1\) Although the behaviour has been observed by several others (e.g., Mori & Ohshima, 1995, 1997; Mori, 1999; Ryu & Samuel, 2002; Zariwala, Miller, Faumont, & Lockery, 2003; Mohri, Kodama, Kimura, Koike, Mizuno, & Mori, 2005; Murakami, Bessinger, Hellmann, & Murakami, 2005; Biron, Shibuya, Gabel, Wasserman, Clark, Brown, Sengupta, & A.D.T., 2006; Luo, Clark, Biron, Mahadevan, & Samuel, 2006; Clark, Biron, Sengupta, & Samuel, 2006) since it was first discovered by Hedgecock and Russell (1975), a more recent study has challenged the validity of the phenomena due to possible effects of body temperature on movement (Anderson, Albergotti, Proulx, Peden, Huey, & Phillips, 2007). As a result, temperature preference is currently a debated topic of investigation in *C. elegans*. 
C. elegans can sense temperature at the tip of their head, and (although less directly) they can also sense food (primarily bacteria). In the thermal preference behavioural paradigm, animals placed onto a thermal gradient will migrate to the temperature that they had been previously cultivated at. The behaviour has been described as one of the most complex in the C. elegans repertoire (Hobert, 2003). They will only ‘remember’ their cultivation temperature if that temperature was paired with the presence of ample food supply. In contrast, if the previous cultivation temperature was paired with starvation (an aversive stimulus), animals will avoid that temperature. They are also known to learn new preferred temperatures: if starved in their current temperature they will move until they find bacteria and remember that temperature for the future (Mohri et al., 2005). In summary, the nematode worm modifies its behaviour in relation to previous experience based on two distinct, yet paired, sensory inputs. The behaviour involves memory formation as well as acute sensory input comparison to a reference value. Although ablation studies have been performed to identify the circuit of neurons responsible for the behaviour (Mori & Ohshima, 1995), the underlying mechanisms are still poorly understood.

In our abstracted model, the task involves pairing the agent’s food with a particular temperature (chosen at random). After some time has passed, the agent is presented with a temperature stimulus for testing. This can be similar or different from the originally paired signal. A successful agent is required to open its ‘mouth’ when it is the same stimulus and close it otherwise. The agent is required to identify the pairing signal for several consecutive tests separated by random delays. Also, the agent can be paired with a different temperature signal from the original at any point during its lifetime. Successful agents are required to re-learn new associations and identify subsequent test signals based on their last pairing.

6.2.2 Agent and structure of a trial

An agent is modeled as a circuit with two sensors and one output (see Figure 6.1). The ‘thermal’ sensor ($T$) provides the local temperature in the agent’s environment. The ‘food/reward’ sensor ($F/R$) provides a binary signal corresponding to the presence or absence of food. This sensor can also act as a positive or negative reward signal, which can be loosely interpreted as coming from a gut sensor that signals the consequences of the agent’s previous action. The only action available to the agent is to open or close its ‘mouth’ via a continuous effector output ($M$). Both temperature and food signal inputs can be perceived by any of the nodes in the network via a set of connections.

A single trial is structured into phases (see Figure 6.2). There are two kinds of phases the circuit can be exposed to: ‘pairing’ and ‘testing’. During pairing, first the circuit is exposed to food and a particular temperature simultaneously. This lasts a fixed 20 units of time. Next, both the food and the temperature signals are removed (signals return to 0). The duration of the delay is random, lasting anywhere between 16 and 24 units of time. During testing, the circuit is first exposed to a particular temperature that may or may not be the same as the one applied during the pairing phase, but in the absence of the food signal. This lasts 10 units of time. The temperature signal is then removed and the circuit evaluated for another 10 units of time by observing the state of the ‘mouth’. Ideally, it should be ‘open’ if the recent and original pairing temperature are the same.

---

2The main thermosensory neuron is called AFD, another neuron called ASH is known to be involved in sensing food. It is important to note that the sensory function of neurons is often determined using behavioural experiments on worms where the neuron has been laser-ablated. See de Bono and Maricq (2005) for more detail.
Figure 6.1: Circuit architecture. Example three-node fully-connected network, including self-connections. The ‘thermal’ and ‘food/reward’ sensors are also fully connected with all nodes. The strengths of all connections are evolved, but stay fixed within the lifetime of the agent. One node is specified as the ‘mouth’ neuron.

same, and ‘closed’ if they are not. Then a delay of random duration is applied, lasting between 8 and 12 units of time. After this, the agent receives either a positive or a negative reward. This is based on the correctness of its previous action (mouth open or closed) in relation to the recently experienced temperature and its environment, as determined by the original temperature. The testing phase ends with another delay of random duration in the same range as the previous one.

The training paradigm is given by the structure of a trial. For any single trial, there must always be at least one pairing and one testing phase, as illustrated in Figure 6.2. However, trials can be comprised of more than one pairing and testing phases. The pairing phase indicates the ‘environment’ the agent is in. The particular temperature where food is found defines an environment. In different environments food can be found in different temperatures. Thus, multiple pairing phases represent changes of environment. A circuit can also be subject to several subsequent testing phases within a particular environment. This requires the circuit to remember the paired temperature for longer. Thus, multiple testing phases represent subsequent evaluations of the agent’s performance within an environment. For brevity, a length-LK trial corresponds to one with L different environments (i.e., L − 1 changes of environment) and up to K testing phases per environment. During a trial, the state of the circuit is never altered. Only at the start of a trial is the state of the circuit ‘reset’ (more detail in the next section).

The stimuli to be remembered can be drawn from a discrete set or from a range on a continuum. Consider a situation where only n types of environment exist for a particular agent. For the simplest n = 2 case, this would correspond to a ‘world’ where food could be found in a particularly ‘cold’ temperature, \( t_1 \), in some environments, and in a particularly ‘hot’ temperature, \( t_2 \), in the second environment. A successful agent would have to find out which of the two environments it finds itself in, and act accordingly. This is the discrete case. For the continuous case, the number of possible environments is, in principle, infinite. Pairing temperatures are chosen at random uniformly from the range [1, 2]. The temperatures used for testing are either equal or different from the paired temperature with 50% chance. Test temperatures, when different, are also chosen at random uniformly from the full range. A successful agent must remember the particular temperature where food is found in each particular environment. In practice, there will be some
Chapter 6. Associative learning on a continuum in disembodied neural networks

Figure 6.2: Structure of an individual trial. A trial can consist of a combination of the following two phases: pairing and testing (vertical grey line). Each of these phases is composed of distinct events (vertical dashed lines). During pairing, food and temperature signals are applied simultaneously. This is followed by a variable random delay. During testing, first a temperature signal is applied. Second, the state of the mouth is evaluated relative to the correct action (bold line). The error between the agent’s action and the desired response is shaded in dark grey. Third, there is a variable random delay. Fourth, a reward signal is applied based on whether the previous action was correct or not. Finally, there is another variable random delay before the next trial begins. Multiple pairing and testing phases can occur during a single trial. The state of the network is never reset within a trial, only between different trials.

Differences that may be too small to detect given some precision. For this reason, although pairing and testing temperatures are drawn at random, the performance on test signals that are different from the pairing temperature by less than 0.1 do not count towards fitness.

The aim is to study the difference in the evolved internal dynamics of agents evolved on stimuli drawn from a discrete set (n = 2) versus a continuum.

6.2.3 Dynamical neural network

We use continuous-time recurrent neural networks (CTRNNs) as a model of the agent’s internal dynamics. These are continuous-time nonlinear dynamical systems that can, in principle, approximate any dynamics with an arbitrary precision, given enough components (Funahashi & Nakamura, 1993). For this reason, our model does not include any form of explicit plasticity mechanisms (such as weight-changing or any other parameter-changing rules). Each component in the network is governed by the following state equation (Beer, 1995b):

\[
\tau_i \dot{y}_i = -y_i + \sum_{j=1}^{N} w_{ji} \sigma(y_j + \theta_j) + tw_i T(t) + fw_i F(t)
\]

(6.1)

where \(y\) is the activation of each node; \(\tau\) is its time constant; \(w_{ji}\) is the strength of the connection from the \(j^{th}\) to the \(i^{th}\) node; \(\theta\) is a bias term; \(\sigma(x) = 1/(1 + e^{-x})\) is the standard logistic activation
6.2. Methods

function; and $N$ represents the number of nodes in the network. All nodes can sense the external environment via a set of extra connection weights: $tw_i$ is the weight of the connection from the ‘thermal sensor’, $T(t)$, to node $i$; $fw_i$ is the weight of the connection from the ‘food sensor’, $F(t)$, to node $i$. The activation of all nodes is ‘reset’ to 0 at the beginning of each trial. Remember that each trial comprises a sequence of pairing and testing phases. In simulation, node activations are calculated forward through time by straightforward time-slicing using Euler integration with a time-step of 0.1. The network is fully connected.

6.2.4 Evolutionary algorithm

The parameters of each circuit (i.e., biases, time-constants, inter-node and sensor-node weights for each node) are evolved using a version of the Microbial genetic algorithm (Harvey, 2001). There are $N^2 + 4N$ parameters in total. These are encoded in a genotype as a vector of real numbers over the range $[0, 1]$. Offspring of Microbial tournaments are generated as a mutation of the winner of the tournament (without recombination). The mutation is implemented as a random displacement on every gene drawn uniformly from a Gaussian distribution with mean 0 and variance 0.01. Each gene is forced to be in $[0, 1]$: when a mutation takes a gene out of this range it is reflected back. The offspring replace the loser of the tournament. Genes are mapped to network parameters linearly between [-10, 10] for biases and inter-node and sensory weights. Time constants are mapped exponentially between $[e^0, e^5]$. The size of the population used is 50. We define a generation as the time it takes to generate 50 new individuals. The evolutionary algorithm uses a ‘geographical’ method to allow different subpopulations to evolve semi-independently in ‘neighbourhoods’ within the population (see Section 3.2.4 for more detail). A minimal 1D wrap-around geography with neighbourhoods of size 10 was used: such that only individuals 10, or less than 10, positions away from each other could compete in tournaments. In principle, this allows for genotypic diversity to be maintained for longer in the population at almost no extra computational cost. Finally, because the fitness is noisy (described below), agents are re-evaluated every time they participate in a tournament.

6.2.5 Fitness evaluation

A successful circuit must maximize its consumption of edible food and minimize the consumption of inedible food, regardless of the environment it is in. Each individual is evaluated over a set of $R$ length-$LK$ trial sequences. The fitness of a circuit is given by one minus the average consumption error over each testing phase:

$$F = 1 - \frac{\sum_{r=1}^{R} \sum_{l=1}^{L} \sum_{k=1}^{K} E_{rlk}}{RLK} \quad (6.2)$$

where $E_{rlk}$ stands for the error of an agent’s action on the $k^{th}$ test, as part of the $l^{th}$ environment, and on the $r^{th}$ trial repetition. The error is depicted in Figure 6.2 as the dark grey region and it is calculated by integrating the difference between the desired output and the agent’s action during the evaluation phase:

$$E_{rlk} = \int_{T_{rlk} + 10}^{T_{rlk} + 20} |A_{rlk} - M(t)| \psi(T_{rlk} + 10, t) \, dt \quad (6.3)$$
where $T_{rlk}$ is the time the $k^{th}$ trial begins as part of the $l^{th}$ environment and $r^{th}$ trial; $A_{rlk}$ is the correct motor output given the temperature applied during the $k^{th}$ testing phase and the pairing temperature corresponding to the $l^{th}$ environment type, during the $r^{th}$ trial; $M(t)$ is the agent’s actual motor output during the evaluation period; and $\psi(t_0, t) = \exp((-t - t_0 - 5)^2/5.12)/4.0034$ is a Gaussian weighting function normalized so that $E_{rlk}$ runs between 0 and 1. Gaussian weighting assigns maximum importance to the error of the agent’s action at the center of the evaluation period, with the importance smoothly falling off at earlier and later times. This weighting function was implemented following Phattanasri et al. (2007). There are two components that make the fitness evaluation of a single trial noisy: (a) the random values of the pairing and testing temperatures and (b) the random duration of the delays.

### 6.2.6 Incremental evolution

An incremental shaping protocol was employed during evolution. The strategy is straightforward. Evolution starts with the most basic form of the task. As the population succeeds, the level of difficulty of the task increases. This not only saves important evolutionary computation time, but in some situations it can also increase the chances of evolutionary success (Phattanasri, 2002).

The shaping protocol includes four stages. During the first stage, one pairing and one testing phase are applied. This is the most basic form the task can take ($L = 1, K = 1$), and it requires learning one association and remembering it for at least one test. During the second stage, agents are still exposed to only one environment (one pairing phase), but the number of testing phases is increased to three ($L = 1, K = 3$). The change from the first to the second stage serves to increase the selection pressure on the agent’s memory. During the third stage, a change of environment is introduced for the first time. A trial consists of a pairing phase, followed by three testing phases, followed by another pairing phase, and finally followed by another three testing phases ($L = 2, K = 3$). This change of stage introduces new selection pressure on the agent’s ability to re-learn and remain plastic, while still retaining selection pressure on the agent’s memory. The fourth and last stage consists of three environments, each with a variable number of testing phases ($L = 3, K = [0, 5]$). When $K = 0$, the agent is subject to two consecutive pairing phases. It is included so that the agent is trained on situations where it is asked to remember a temperature, and without being tested on it, to have to relearn a new one. Finally, given that the fitness evaluation of a single trial is noisy, agents are assessed on 100 trials per fitness evaluation.

There are some differences in the shaping protocols for the discrete and continuous scenarios. Transitions between the first three stages of the shaping protocol are triggered whenever the fitness of the best agent in the population exceeds a certain threshold (95% for the discrete case and 90% for the continuum case). The last stage is applied only after a certain number of generations and only for the most successful populations. How many generations and which populations are regarded as successful was also different for the discrete and the continuous versions of the task. This was based on two observations from preliminary experiments. First, populations evolving on the continuous task required more time to achieve appropriate performance than on the discrete case. Second, while it was possible for successful circuits to achieve near perfect fitness (99.9%) on the discrete task, the fitness of circuits evolved for the continuous version of the task never reached similar levels of performance. Thus, for the discrete version, the last stage was applied.
6.3 Learning Discrete Stimuli

Our first set of experiments examines the ability of CTRNNs to solve the discrete version of the associative learning task. The main purpose of this section is to form a ‘baseline’ for the continuous work (Section 6.4) by reproducing the results in Phattanasri et al. (2007) with a different training paradigm and evolutionary technique, but similar task and shaping protocol, and the same neural network model. Another important role of this section is to show how the dynamical analysis leads to the FSM.

Evolutionary searches with 3- and 4-node circuits were performed. Successful agents were reliably evolved using such small circuits. For the most part, successful circuits had evolved to a fitness of 99% after the first 3000 generations. After $10^4$ generations, we found 4/20 evolutionary runs using 3-node circuits and 12/20 using 4-node circuits that achieved fitness greater than 99% at the third stage of the shaping protocol. The most successful populations (those that reached fitness greater than 99% at the third stage of the shaping protocol) were further evolved on the last stage of the shaping protocol for an additional 5000 generations. Figure 6.3 shows the fitness versus generation plot for the best 3-node population. The evolutionary runs from successful populations are all relatively similar. Periods of fairly steady fitness values are punctuated by sudden jumps to regions of higher fitness. Also, the fitness tends to drop significantly following changes to the difficulty of the task from the shaping protocol (dashed lines). In the reminder of this section we describe the performance and internal dynamics of the best-evolved 3-node circuit in some detail.

Figure 6.3: Plot of best fitness vs. generation for the best evolved 3-node circuit for the discrete learning task. The x-axis is plotted with a log scale, as the first transitions occur early during the evolutionary run. Transitions between stages of the incremental evolutionary technique are marked with dashed lines and labeled accordingly. First two transitions occur when the best fitness exceeds 0.95, the last transition occurs after $10^4$ generations. The fitness drops sharply during the first two transitions, before the circuit can generalize to sequences of arbitrary length.

after $10^4$ generations only to those populations whose best fitness reached greater than 99% at the third stage of the shaping protocol. For the continuous version, the last stage was applied after 20000 generations and only to those populations whose best fitness reached greater than 91% at the third stage of the shaping protocol. Finally, the variance of the mutation was halved during this last stage, to allow fine-tuning of the evolved parameters.
The best 3-node circuit attained a fitness of 99.99%, the parameters of which are specified in Table 6.1. To verify that this circuit had truly generalized to longer sequences, we tested it on $10^6$ trials with 5 changes of environment and up to 10 tests per environment during its lifetime. Pairing and testing temperatures were chosen at random (from the two available). The time-step was made an order of magnitude smaller (0.01) to avoid possible integration errors. The circuit performed correctly on 98.36% of evaluations for this set, demonstrating that it does indeed represent a solution to the discrete temperature preference task that can generalize to longer sequences.

The behaviour of this circuit on a typical sequence of trials is shown in Figure 6.4. As there are only two possible temperatures (1 and 2) that food can be associated with in this version of the task, we will refer to them as environments A and B, respectively. During the sequence shown, the environment-type switches from A to B and then back to A again at the points indicated by the dashed vertical lines. We can observe that the activity of nodes $o_1$ and $o_3$ are nearly inverted during environment B, in relation to A. For example, the ‘mouth’ node ($M$) is open (highly activated) throughout most of the duration of environment A, closing only during the presentation of the ‘wrong’ signal. Exactly the opposite is the case throughout environment B. The same node is now mostly closed, opening only during the presentation of the wrong signal in this environment. For both environments, the presentation of the different signal serves to change the current state of the ‘mouth’, while the reward serves to replace the former state. Similarly, all other nodes are, in part, keeping track of the environment the agent finds itself in. Also important to note is that the timescale of activity of the second node ($o_2$) is relatively slower than that of nodes $o_1$ and $o_3$.

How does this circuit work? In order to visualize the overall structure of this circuit’s operation we apply a similar technique as that used in Phattanasri et al. (2007) by ‘strobing’ the state of the system at selected times during a trial. In particular, we observe the state of the system at the end of the pairing signal, the resting time, the testing signal and the reward signal. Given that the evolved system has a three dimensional state space we can visualize the entire space directly. The ‘strobes’ fall into relatively distinct clusters (Figure 6.5). Each of the clusters can be labeled according to the previous environmental interaction that the circuit had undergone. The labels A and B for the states represent one of the two temperatures the circuit has been initially paired with (i.e., the environment type). Clusters A1 and B1 denote the state of the system after a pairing. Clusters A2 and B2 represent the state of the system after a random delay. Cluster B2 is also the
6.3. Learning Discrete Stimuli

Figure 6.4: Activity of the best 3-node circuit on a typical trial sequence. From top to bottom the traces correspond to the temperature signal ($T$), the food/reward signal ($F/R$), the mouth state ($M$) and the outputs of the remaining nodes ($o_2$ and $o_3$). Small rectangles mark the time when the mouth state is evaluated and the state that the mouth should be during this time. Dashed lines mark transitions between environments.

default starting state of the system. Clusters $A3$ and $B3$ represent the state of the system after the presentation of a test temperature. The presentation can be of one of two temperatures; these are sub-labeled with a further $A$ or $B$, accordingly. For a successful circuit this means that clusters $A3A$ and $B3B$ correspond to states where the mouth is open, while clusters $A3B$ and $B3A$ correspond to closed-mouth states. Finally, clusters $A4$ and $B4$ correspond to the state of the system after a positive reward.

We can consider the dynamics of the circuit when decoupled from the environment. For each of the different combinations of input we can determine the limit sets of the circuit. Furthermore, we can compare the relation of the circuit’s asymptotic behaviour with the clusters. Although a number of these clusters are centered on the equilibrium points in the distinct phase-portraits of this system, many are not (data not shown). Also some of the clusters are not entirely contiguous in state space. This highlights the importance of the transients. Some of the non-contiguous clusters can be further subdivided according to the system’s previous state. The labels in parenthesis denote the particular cluster from which the system departed to form those subclusters.

The strobed circuit dynamics from Figure 6.5 can be interpreted as implementing a FSM with input. The strobed states correspond to the FSM states. Although not shown, transitions between strobed states correspond to input-driven transitions of the FSM. The FSM extracted from this circuit is shown in Figure 6.6. States $A1$ through $A4$ and $B1$ through $B4$ represent the states of the system described previously. There are 4 different types of transitions that can occur. First, the application of one of two possible temperatures $T = 1$ or $T = 2$, labeled ↓ and ↑, respectively. Second, the application of a positive or negative reward, labeled $+$ or $-$, respectively. Third, a pairing that involves the application of a temperature and food simultaneously, denoted as ↓+ or
Figure 6.5: Strobed circuit dynamics in the best 3-node circuit at selected times during a trial. See the main text for the meaning of the labels.

↓ depending on the temperature. Finally, it is useful to treat the lack of stimuli as a transition that the circuit is exposed to, because it directs the state of the system to a relevant state in the machine. Thus, the last transition that can occur is a delay, corresponding to the absence of stimuli and denoted by $o$. Transitions are shown as arrows with labels in Figure 6.6. The start state is shown by an arrow pointing at it from $s$. Although this is not a situation encountered during evolution, at the start of a trial, if no pairing is applied, the system will move towards state $B2$. Notice subclusters (e.g. $A4(A3A)$ and $A4(A3B)$) are grouped together. A more detailed FSM could be provided, but the extra detail does not add to our understanding of the operation of the state machine, as all of the incoming and outgoing transitions remain the same.

In our temperature preference task, an agent discovers which of the two environments it is in when there is a pairing. This means that the circuit doesn’t have to wait to get a negative reward to learn. In fact, a successful circuit will never have to experience a negative reward. This is the case for the best-evolved 3-node circuit. For this reason, we can ignore the punishment transitions and states altogether from our analysis. We can, however, artificially induce a negative reward. This makes sense only for this two-environment task, where being maladapted in one environment means inevitably that the circuit is in the only other possible environment. Application of a negative reward while in environment $A$ does switch the circuit’s behaviour to what it would be if it were in the opposite environment, however this switch does not occur the other way around. Thus, negative reward drives the state of the system to $B2$’s basin of attraction. This is linked to its role as the default initial state.

Before we move on to the continuum task, one question that we can ask is how this circuit deals with stimuli in between the discrete. We can test the performance of this system for the full range of possible combinations of pairing and testing temperatures. Figure 6.7 shows the
6.3. Learning Discrete Stimuli

Figure 6.6: Extracted FSM from best 3-node circuit. Each state corresponds to a cluster of strobe states from Figure 6.5. Transitions between states are induced by the application of some combination of food, temperature or delays.

generalization performance of the best-evolved circuit across all combinations. The vertical axis corresponds to the pairing temperature signal. The horizontal axis corresponds to the test temperature signal. The shading represents how well the circuit performs: lighter shades correspond to better performance. What the circuit is expected to do changes depending on the pairing signal.

For test temperatures equal to the pairing temperature the circuit is required to open its mouth. This corresponds to the line on the diagonal of the figure. For any test temperature that is different from the pairing temperature the circuit is required to close its mouth. This corresponds to all regions not directly on diagonal. For the discrete version of the task, the circuit is only evolved on the highest and lowest possible temperatures (denoted by circles in the figure). For this reason it is not expected to generalize to signals in-between. What we observe, instead, is an example of a binary categorization. When paired with ‘cold’ temperatures (below 1.6), the agent opens its mouth to all test temperatures below 1.3. When paired with ‘hotter’ temperatures (above 1.6), the agent opens its mouth to any test temperature above 1.3. Temperatures fall into one of two broad categories: ‘cold’ or ‘hot’; with little or no generalization to temperatures in-between.

Finally, it is important to mention that other successfully evolved 3-node circuits displayed overall similar properties to the one analyzed here. Namely: (a) equivalent FSMs could be extracted from the evolved internal dynamics; (b) strobed states would not necessarily correspond to equilibrium points of the non-autonomous system with transients playing an equally important role; (c) categorization into two behavioural groups was observed without generalization within previously unseen environments; and (4) evolved time-constants would consistently fall into at least two relatively different time-scales: fastest possible (near 1.0) and slower by an order of magnitude.
Chapter 6. Associative learning on a continuum in disembodied neural networks

6.4 Learning Continuous Stimuli

Can the same ‘learning without synaptic plasticity’ approach used in our previous section be extended to the continuous task? Our second set of experiments examine the ability of CTRNNs to solve the same associative learning task when the stimuli can be anywhere along a continuum. The main motivation for these experiments is to evaluate the similarities and differences between agents evolved for this task and the previous version.

6.4.1 Evolutionary Performance

Evolutionary searches were performed using 3- to 6-node circuits. We carried out 20 evolutionary runs with different seeds per group. After $2 \times 10^4$ generations, we found that none of the 20 evolutionary runs using 3- and 4-node circuits reached the third stage with a fitness greater than 91%, while 12/20 5-node and 13/20 6-node populations did. The most successful populations (those that reached fitness greater than 91% at the third stage of the shaping protocol) were further evolved for 5000 generations on the fourth stage of the shaping protocol, with the most successful 5-node population reaching a 96.85% best fitness. Figure 6.8 shows the best fitness versus generation plot for this population. Although success is achieved early on the first stages of the task, it takes many more generations to achieve sufficiently high scoring individuals. This was similar for most other successful evolutionary runs. Also, similar to evolution on the discrete version of the task, the fitnesses of the best individuals in the population drop after the difficulty of the task is increased. In the remainder of this section we describe the performance and internal dynamics of the best-evolved 5-node circuit in some detail. The evolved parameters of this circuit are specified in Table 6.2.

6.4.2 Learning and Memory Performance

How well can this circuit learn and remember on trials involving more changes of environment and more tests than it was evolved for? To answer this question we tested it on $10^6$ trials with 5 changes of environment and up to 10 tests per environment during its lifetime. Pairing and
6.4. Learning Continuous Stimuli

Figure 6.8: Plot of best fitness vs. generation for the best evolved 5-node circuit for the continuous learning task. Labeling conventions are the same as in Figure 6.3. The x-axis is plotted with a log scale, as the first transitions occur early during the evolutionary run. The first two transitions occur when the best fitness exceeds 0.90, and the last transition occurs after $2 \times 10^4$ generations. Similar to the discrete case, the fitness drops after the first two transitions, before the circuit can generalize to longer sequences.

Table 6.2: Best 5-node circuit for the continuum version of the temperature preference task

<table>
<thead>
<tr>
<th></th>
<th>$y_1$</th>
<th>$y_2$</th>
<th>$y_3$</th>
<th>$y_4$</th>
<th>$y_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_1$</td>
<td>9.6712</td>
<td>1.8562</td>
<td>4.3859</td>
<td>-0.6227</td>
<td>-0.7912</td>
</tr>
<tr>
<td>$y_2$</td>
<td>-5.1801</td>
<td>-2.4098</td>
<td>-3.5321</td>
<td>8.8119</td>
<td>-9.9244</td>
</tr>
<tr>
<td>$y_3$</td>
<td>-6.5417</td>
<td>9.3668</td>
<td>1.3050</td>
<td>9.7333</td>
<td>-5.9810</td>
</tr>
<tr>
<td>$y_4$</td>
<td>7.0174</td>
<td>5.7888</td>
<td>-9.9125</td>
<td>-4.5742</td>
<td>-8.5561</td>
</tr>
<tr>
<td>$y_5$</td>
<td>8.3435</td>
<td>-9.2429</td>
<td>2.8769</td>
<td>-4.9563</td>
<td>5.2326</td>
</tr>
<tr>
<td>$T$</td>
<td>-2.8401</td>
<td>5.5182</td>
<td>5.5025</td>
<td>-1.3114</td>
<td>9.3223</td>
</tr>
<tr>
<td>$F$</td>
<td>-7.6829</td>
<td>-1.5416</td>
<td>4.2153</td>
<td>3.9677</td>
<td>7.6002</td>
</tr>
<tr>
<td>$\theta$</td>
<td>-4.8178</td>
<td>-4.4765</td>
<td>-9.9440</td>
<td>-0.7085</td>
<td>-5.2138</td>
</tr>
<tr>
<td>$\tau$</td>
<td>1.7630</td>
<td>16.9439</td>
<td>1.5663</td>
<td>73.9571</td>
<td>1.0663</td>
</tr>
</tbody>
</table>
testing temperatures were chosen at random from the full range, and the time-step was an order of magnitude smaller (0.01). The circuit performs correctly on 95.77% of the trials on this set of experiments, indicating that it does indeed represent a solution to the associative learning task for stimuli that can range over a continuum of values that can generalize to longer sequences.

The behaviour of this circuit on a typical sequence of trials is shown in Figure 6.9. During this sequence, the environment type switches from A to B, and then to an in-between environment C (temp=1.5) at the points indicated by the dashed vertical lines. From Figure 6.7 we know that switching to an in-between environment (such as C) for the 3-node network discussed previously would have led to poor performance. In contrast, as depicted in Figure 6.9, the 5-node circuit does manage to remember the in-between signal correctly. One thing to note is the range of time-scales of activity displayed by the components in the circuit: $o_1$, $o_3$ and $o_5$ are fast acting with time-constants in the range between [1.06, 1.76], $o_2$ is somewhat slower than those ($\tau=16.9$) and $o_4$ is much slower acting than all others ($\tau=73.9$). Although this bears some resemblance to the different time-scales evolved for the smaller circuit, the differences between these ranges are much larger.

How well does this circuit generalize to signals in-between the border cases? To answer this we can study the learning map for the best-evolved 5-node circuit on the complete range of pairing and test temperatures. Remember from the circuit evolved for the discrete task (Figure 6.7) that two types of behaviour dominate its performance, since any signal above a certain threshold is treated as ‘cold’ and anything above is treated as ‘hot’. In Figure 6.10A we show the learning map...
for the first test after a pairing systematically covering the full-spectrum of combinations. The dominance of the white shade reflects the good generalization performance. The shades of black around the white diagonal line depict the precision of the system’s evolved memory. Any testing temperature that is different from the pairing temperature by less than some variable amount is regarded as the same. Although one way to see this is as a ‘lack of precision’, it is more interesting to treat it as a behavioural generalization: treating temperatures similar to the paired temperature as correct. This is also a direct reflection on the evolutionary conditions, avoiding too similar signals when different. The dashed diagonal lines in the figure depict the range within which there was no selection pressure (except for when the pairing and test temperatures are exactly the same).

The first test is performed directly after the circuit has experienced the to-be-remembered temperature (i.e., the last paired temperature). How does the memory decay over multiple tests? Figure 6.10B shows the remembering performance for the second test after a random delay. Although there is some degradation in the memory of the original signal, as can be seen by the shades of grey around the diagonal line, it is still mostly appropriate. Figure 6.10C shows the remembering performance on the 10th test signal. As can be seen, the performance continues to degrade slowly as more and more tests and random delays are applied consecutively after the original pairing.

How fast does memory decay over many more presentations? How does memory depend on the circuit’s experience? Why does it decay, and can it be preserved for longer? In Figure 6.10D we show the circuit’s ‘forgetting curve’: the remembering performance as a function of the number of test phases experienced. Shaded in grey we show the number of trials the circuit was evolved for. The solid line represents the remembering performance when pairing and testing temperatures are chosen at random. As is expected, the memory of the originally paired temperature decays with the number of tests. But does how fast it decays depend on what the temperatures of the tests are? The dotted line represents the remembering performance when all of the testing temperatures are random but different from the paired temperature. As can be seen, the remembering performance falls much more dramatically, in a classical exponential decay curve. This corresponds well with the literature in experimental psychology, where memory retention is known to decay exponentially as a function of time, in the absence of revision of the learnt material (Ebbinghaus, 1885). This memory decay also suggests that not re-experiencing the original temperature decreases the chances of remembering it correctly. Thus, we should expect good remembering performance if every one of the testing temperatures are the same as the one to-be-remembered. The dashed line shows the performance when all previous test signals are equal to the original pairing signal. Indeed, the continued presentation of the to-be-remembered signal strengthens the circuit’s memory of it, while long absences result in the degradation of the original memory.

Does this circuit remain sufficiently plastic to re-learn new associations between temperatures and food throughout its lifetime? Or does the plasticity decay after some time or usage? We can test the agent’s ability to learn new temperatures by changing the environment multiple times. In Figure 6.11 we observe the long-term performance for several different re-learning environments. We again observe satisfactory performance. It is as if re-learning resets the state of the circuit entirely. This shows that the agent remains fully plastic outside of the ranges that it was evolved for (i.e., only two changes of environment). In fact, the circuit shows no sign of losing its plasticity with time.
Chapter 6. Associative learning on a continuum in disembodied neural networks

Figure 6.10: Remembering performance of the best 5-node circuit. Generalization map on the first [A], second [B] and tenth [C] test signal. Labeling conventions are the same as in Figure 6.5. [D] A measure of the circuit’s performance on the $i^{th}$ test: when the signals for all of the previous tests where randomly chosen to be equal or different than the pairing signal with 50% chance (solid line), when the previous test signals are always the same as the pairing signal (dashed line), and when the previous test signals are random but always different from the pairing signal (dotted line). Each point corresponds to the average over $10^5$ random runs. The points marked with diamonds correspond to the performance maps shown for parts [A], [B] and [C], respectively. The grey shaded region corresponds to the range of conditions the circuits were evolved in.

Figure 6.11: Learning performance of the best 5-node circuit. A measure of the circuit’s performance for many changes of environments: on the first (solid line), second (dashed line) and third (dotted line) test after the $i$-th change of environment. Each point corresponds to the average over $10^5$ random runs. The grey shaded region corresponds to the range of conditions the circuits were evolved in.
Finally, we can ask about the robustness of the performance in relation to the time delays between tests. Although transients play a role in the dynamics, ideally memory should be more permanent. In other words, it is important to know how stable the memory of the evolved circuits is. Figure 6.12 shows the robustness of the circuit’s learning performance as a function of the length of the time delays. The circuit manages to be quite robust to time-delays shorter and longer than the range that it was evolved for. Although this was not selected for during evolution, it is a relevant feature of this circuit.

In summary, evolutionary runs for the continuous task were successful only with circuits of size 5 and larger\(^3\). The behaviour and performance of the most successful and smallest circuit was studied. The circuit manages to learn and generalize over the full range of signals on the continuum that it was evolved for. Also, it can remember paired temperatures for longer than it was evolved for, as long as it continues to experience that temperature during tests. The circuit can also remain plastic enough to re-learn new associations within its lifetime. Furthermore, we observed no degradation in the circuit’s plasticity over time. Finally, the circuit’s memory was relatively robust to longer time delays than those experienced by its ancestors.

### 6.5 The Dynamics of Continuous Learning

How does this circuit work? How do the evolved circuit’s mechanisms differ from those evolved for the discrete version of the task? Can a FSM be extracted to capture the workings of the dynamics of the best-evolved circuit? In order to answer these questions, we have to visualize the overall structure of this circuit’s operation, and for this we will use a similar approach to that developed for the discrete version. We ‘strobe’ the state of the system at selected times during a trial. The main difficulty that arises in this case is that the internal state of the evolved system is composed of more components. Therefore, part of the work in analyzing the internal dynamics of this evolved circuit will involve: (a) looking at several different 3 dimensional slices of this 5

---

\(^3\)We do not investigate in this chapter whether the ability to generalize is related to the size of the network. The task studied requires that successful circuits generalize to signals in-between the border cases, but it also requires: (a) that they remember the original signal on subsequent tests, and (b) that they remain sufficiently plastic to re-learn during their lifetime.
Figure 6.13: Internal dynamics of the best-evolved 5-node circuit, strobes while tested on the continuum of signals after a pairing (Q1), a rest (Q2), a test (Q3), or a positive (Q4) or negative reward (Q5). The Q3 strobed region can be further subdivided according to the relation between the pairing and testing temperature. The pairing temperature can be ‘hotter’ (i), the same (ii), or ‘colder’ (iii) than the original.

dimensional space, (b) building up our intuitions about the structure of the manifolds of activity, and (c) choosing the variables and perspectives that provide the most useful insights. In the figures to follow we look at some of these 3-dimensional slices of the space of activations of the evolved circuit. In particular, we look at slices from the two slowest nodes, y_2 and y_4, and the fastest one, y_5.

In Figure 6.13, we visualize the state of the system when strobed on the full range of stimuli. We observe that each of the states that correspond to the same activity form a ‘stretched out’ cluster. Each of the clusters represents the state of the system after a pairing, a rest, a test or a positive or negative reward (labeled Q1 through Q5, respectively). Notably, each different state remains relatively well connected and separately clustered. We can also observe that each of the clusters forms a one-dimensional ‘tube-like’ structure, except for cluster Q3 that forms something that looks more like a two-dimensional ‘wing’ structure. We will come back to this point later in the analysis.

An extended behavioural sequence such as the one shown in Figure 6.9 can then be understood as a set of trajectories between these strobe clusters (see Figure 6.14). A question of interest is: do these stretched out clusters and the trajectories between them have any further internal structure to them? We can visualize the trajectories in relation to the environment the circuit is in. In Figure 6.14, trajectories are coded in shades of grey according to their original paired temperature: lighter shades corresponding to hotter temperatures. What can be observed is that the transitions
have a relatively structured pattern. The trajectories are arranged from top to bottom according to their paired temperature: with ‘hotter’ ones at the top and ‘colder’ ones towards the bottom. Most importantly, this pattern is maintained as the state of the system flows between each of the different clusters. This corresponds to the ‘memory’ of the environment.

A crucial aspect to the learning behaviour under study is the circuit’s ability to make different decisions depending on its experience. One question of interest is, how does the decision arise internally? To answer this, it will be useful to take a closer look at the transitions between the resting state (Q2) and the testing state (Q3) (plots E, F and G in Figure 6.14). During the resting state the agent can be tested with the same temperature as it was paired with originally, or it can be tested with a different temperature. In the latter case, the temperature can be either ‘colder’ than the original or ‘hotter’. Figure 6.14 shows the state of the system as it transitions from its resting state to being tested on any of the possible signals. The state of the system moves to the middle part of the cluster Q3(ii) for signals that are similar to the original (Figure 6.14F). What this means is that the original temperature is known from the level of the Q2 cluster, where the system is operating. When the temperature is different than the original, the state of the system falls away from the middle into one of the two outer ‘wings’ of the structure. Falling to the top left ‘wing’ Q3(iii) when the testing temperature is ‘colder’ than the original pairing temperature (Figure 6.14E) and to the bottom right ‘wing’ Q3(i) when it is ‘hotter’ (Figure 6.14G).

As we have seen, the circuit discovers the environment it is in through the simultaneous pairing of food and temperature. Yet, unlike the discrete scenario (in particular for \( n = 2 \)), receiving a negative reward is not enough to modify the state of the agent such that it ‘finds out’ which temperature is the right one. Furthermore, a successful circuit could simply never receive negative reward during its lifetime. This is not, however, the case for the circuit under analysis. From Figure 6.14C, we know that some signals end up near the border between the ‘wings’ and the middle part of the Q3 cluster, which then receive a negative reward moving the state of the system to Q5. From Figure 6.10A, we know that these correspond to test signals that are very similar (but different) from the paired temperature. The negative reward, however, does not ‘correct’ these borderline cases. We examined this by artificially inducing a negative reward after the agent opens its mouth when tested on the paired temperature (for which it usually receives a positive reward). When tested again using the same temperature the agent would still open its mouth. Thus, the negative reward cannot override the original pairing memory in this circuit. Similarly, we examined whether the positive reward could trigger the circuit to relearn a new association. We artificially induced a positive reward after the agent closes its mouth when tested on a different signal to the paired temperature (for which it usually receives a negative reward). When tested again using this new temperature (which provided a positive reward despite not being the original paired temperature), the response of the circuit was still to close its mouth. Thus, the circuit only learns new associations through the simultaneous pairing of food and temperature; not through the reward signal. The most likely reason for this is that no changes of environment were experienced of the latter form during evolution\(^4\).

A key question that we would like to ask, then, is whether we can extract an FSM from its internal dynamics, such that it explains the learning behaviour? Given that the network has to

---

\(^4\)Social learning experiments in Noble, Todd, and Tuci (2001) present an example where the learner ignores a cue that would be useful because the situation in which they have utility is not a frequent part of evolution.
Figure 6.14: Transitions between each of the states superimposed over the strobed states. Each transition is shaded in grey according to the original pairing temperature, with lighter shades representing hotter temperatures and darker shades colder temperatures. [A] Transitions between pairing and resting. [B] Transitions while positive reward is applied. [C] Transitions while negative reward is applied. Notice that negative reward occurs naturally only when the test temperature is different but too similar to the pairing temperature. The circuit makes the mistake of classifying these as the same, despite the small difference. Thus, the transitions depart from $Q_3^{(ii)}$ mostly. [D] Transitions during resting after a positive or negative reward. The bottom three figures depict the transitions between the resting state $Q_2$ and the presentation of test temperatures over the whole continuous range: [E] shows the transitions when the test temperature is lower (‘colder’) than the pairing temperature; [F] when the test temperature is the same as the pairing temperature; and [G] when the test temperature is higher (‘hotter’) than the original pairing temperature. As can be seen by the predominance of lighter-shaded transitions in [E], when the original pairing is of a ‘hotter’ (lighter grey trajectories) temperature, the number of test temperatures that are classified as ‘colder’ is greater than when the original pairing is of a ‘colder’ (darker grey) temperature. The inverse is true for [G], where there is a predominance of darker-shaded transitions.
6.5. The Dynamics of Continuous Learning

Figure 6.15: A ‘continuous’ state machine embedded in the best 5-node circuit. The states are labeled according to the strobed states from the previous figures. The shade of grey for each state represents the real-valued register. Where the agent’s state ‘lands’ inside the \( Q3 \) region is determined by the pairing temperature \( (t_p) \) along the horizontal axis and the testing temperature \( (t_t) \) along the vertical axis: when the test temperature is ‘hotter’ than the original pairing temperature \( (t_t > t_p) \) the state falls into the \( Q3(i) \) region; when they are the same \( (t_t = t_p) \) it falls into the \( Q3(ii) \) region; finally, when the test temperature is ‘colder’ than the original pairing temperature \( (t_t < t_p) \) it falls into the \( Q3(iii) \) region.

To remember a continuous signal, a ‘machine’ is required that will allow for a continuum of states to represent the environment. No finite state machine can represent such internal mechanisms. A richer structure is needed: a machine that includes for each of the discrete states an inner (relatively independent) continuous state. We are calling this set of machines, continuous state machines (CSM). One way to think of these is as a continuous manifold of finite state machines. Accordingly, we can think of a FSM as a CSM with only one inner ‘level’. Figure 6.15 shows two of the FSMs, on top of each other with a transition from one to the next. The dynamics are just like an FSM but with stretched-out regions for each state. We can think of each of the states as containing a real-value register. This inner state is continuous and is instantiated as the level within the extended strobe clusters.

At the behavioural level, the CSM denotes two seemingly distinct processes operating at two different scales. While the discrete states resemble states of an FSM, the regions inside each of the discrete states resemble something more like a continuous (but bounded) tape. We can illustrate this idea using an example sequence trial in our evolved circuit. In Figure 6.16, we show the trajectory of the state of three of the nodes during an example sequence trial where the agent is first paired and tested with temperature \( X \) (1.3), then paired with and tested with a different temperature \( Y \) (1.7), and vice-versa, several times. The trajectory is placed within the context of the ‘strobed’ states (in grey). This illustrates the notion of the system’s operation at different levels within the manifold of finite state machines (\( X \) and \( Y \)), as well as the transitions between these \( (X \rightarrow Y \text{ and } Y \rightarrow X) \) corresponding to re-learning new associations. It is important to point out that this evolved circuit requires ongoing interaction with an environment in order to maintain a given operational
level within the manifold of FSMs. This can be observed best from the forgetting curve shown in Figure 6.10D. Without this ongoing interaction, the state will eventually decay to some fixed level.

Finally, we would like to know how the state machine is related to the circuit’s evolved components. The best 5-node circuits taken from the five best evolutionary runs (using different seeds) show a distinct distribution of time-constant parameters: the majority of the components are as fast-acting as is allowed but a few are much slower. Is there a functional relation between the discrete states and the fast nodes, and between the continuous internal state and the slow nodes? We can answer this for the case of the best-evolved circuit. Although the full circuit is responsible (and necessary) for the learning phenomena, we can test the correlation between the paired temperature and the state of the system, at different times during a trial and for every component in the circuit. We do this using Pearson’s product-moment correlation coefficient (Moore, 2006):

$$r(p, y) = \frac{\sum_{t=1}^{2}(p_t - \bar{p})(y_t - \bar{y})}{\sqrt{\sum_{t=1}^{2}(p_t - \bar{p})^2 \sum_{t=1}^{2}(y_t - \bar{y})^2}}$$  \hspace{1cm} (6.4)$$

where $p_i$ is the original paired temperature and $y_i$ is the state of one of the nodes at a particular stage during the trial (i.e., $Q1$ through $Q5$) in an environment $t$; $\bar{p}$ and $\bar{y}$ are the averages of $p$ and $y$, respectively. Temperatures, $t$, in the full range $[1, 2]$ (incremented in steps of 0.01), were used to record the state of each of the nodes during each of the different stages.

As can be seen from Figure 6.17, all of the nodes are highly correlated with the remembered temperature (either positively or negatively) at most of the stages of a sequence trial. However, while the correlation among the fast set of components (grey) varies within a trial, the correlation of the slow components (black) remains remarkably stable. This suggests that the role they play in the maintenance of the ‘memory trace’ is stronger than the faster subset of nodes.
In summary, to understand how the most successful and smallest circuit works we strobed the state of the system at selected times during a trial. We observed separate clusters that stretched out with a relatively structured inner dimension. The different clusters corresponded to the different events in the trial (i.e., pairing, rests, tests, rewards). However, the attractors of the circuit did not always correspond to the clusters. The inner dimension, on the other hand, corresponded to the to-be-remembered signal. As the trial proceeds, the state of the system transitions from cluster to cluster, while maintaining the structure of the inner dimension. The decision process was shown to involve a more complex two-dimensional-like cluster, where a relational categorization process was observed: with ‘hotter’, ‘similar’, and ‘colder’ test temperatures neatly separated. We described the evolved learning mechanism in terms of a continuous manifold of FSMs. Finally, although all of the components in the network are involved in the generation of the learning behaviour, we observed a stronger maintenance of the correlation between the to-be-remembered signal in the slower acting components of the network compared to the faster ones.

### 6.6 Discussion

In this work we have extended previous work on evolving learning without synaptic plasticity from discrete (in practice 2-choice) tasks to continuous tasks. We address two main questions.

First, can this approach be extended to continuous tasks? We show that continuous-time recurrent neural networks without synaptic plasticity are successfully evolved on an associative learning task abstracted from a temperature preference behaviour observed in *Caenorhabditis elegans*. The behavioural task studied in this chapter is, of course, not exclusive to *C. elegans*. Broadly, it involves learning an environmental feature that can range over a continuum of values and remembering it for later as a preference. It also involves the ability to change this preference when appropriate. This is a rather common ability amongst living organisms, including humans.

Second, how does learning without synaptic plasticity work in the evolved circuits? In this work we have shown how the evolved internal dynamics differ in an associative learning task when the stimuli to-be-associated is on a continuum as opposed to a discrete set. The analysis of
evolved agents for associative learning, where the stimuli to-be-remembered are discrete signals, display FSM-like internal mechanisms. This agrees with recent results presented in Phattanasri et al. (2007). A different and richer type of state machine is found when analyzing agents evolved to remember and discriminate between signals from a continuum. Because of the ability of the evolved circuit to use a continuous state inside a set of finite states, we have come to consider it as a different class of automata that we call a continuous state machine.

It has been known for some time that artificial neural networks have the capacity to act as FSMs (McCulloch & Pitts, 1943; Minsky, 1967). In particular, the relation between recurrent neural networks and automata has been treated by several authors (e.g., Cleeremans et al., 1989; Servan-Schreiber, Cleeremans, & McClelland, 1991; Pollack, 1991; Giles et al., 1992; Casey, 1996). None of this work has discussed the notion of a manifold of finite state machines or a continuous state machine, nor have they been observed to arise in neural networks. A relation between FSMs and the state space representation of continuous control theory has been indicated in Elgerd (1967). A related notion has been developed in the context of grammar recognition using recurrent networks in Servan-Schreiber et al. (1991) called graded state machines. The notion of continuous state and graded state machines is different in two important ways. First, the infinite and graded states of a CSM are clustered around discrete and separate finite states. Second, there is a relevant relationship between the continuous dimension across the separate clusters of finite states. Only an intuitive notion of continuous state machines has been provided in this work. Developing a formal account in the context of automata theory may be of interest in the future.

What is the role of transients over multiple timescales? First of all, it is important to note that the strobed points are not, in general, attractors of the evolved circuit. Rather, the system is always being pulled from one attractor to the next by the changing sensory input. Thus the evolutionary algorithm has shaped the transient dynamics of the circuits to solve the task at hand, not its attractor structure. It is also important to note that the best 5-node circuits taken from the 5 best evolutionary runs (using different seeds) all consistently showed at least two different timescales in their evolved internal components. Although the majority of neural components evolved to be as fast-acting as possible (with time-constants near 1.0), for each circuit at least one (but in some cases two) of the neural components evolved to be much slower-acting (by at least an order of magnitude). This points to the importance of developing the tools and language to understand dynamical systems with components interacting over multiple timescales.

But can circuits with fixed weights really learn? Whenever a system has dynamics on a range of timescales (regardless of where this dynamics is instantiated), then it is capable of demonstrating ‘learning-like’ behaviour. The nodes in CTRNNs can change relatively slowly, compared to other variables; much as synaptic weights do in a conventional neural network under learning rules. These nodes just do not happen to be labeled as ‘weights’. It would be possible to take the nodes with slower time-parameters and arbitrarily label them ‘synaptic-weight-equivalent nodes’. However, ‘nodes acting as synaptic weights’ is only one possibility. A CTRNN can exhibit dynamics on a range of timescales even if the time constants of all the nodes are fixed at 1, due to the

---

5Classical examples of multiple timescale systems (e.g. weakly-coupled and relaxation oscillations) are covered in most introductory dynamical systems textbooks (e.g., Strogatz, 1994). For an example of multiple timescale techniques to understand bursting in neurons see Izhikevich (2000). For applied dynamical systems describing the analysis of 3-timescale dynamics see Krupa, Popović, and Kopell (2008).
interactions between the nodes. We will revisit this issue again in Chapter 11. Notwithstanding, for the associative learning task studied here, artificial evolution exploited predominately the ability to use components with inherently different timescales of activity. Thus, while memory could have arisen from, for example, reverberatory dynamics (Lau & Bi, 2005); it was the dynamics of the slower-acting components that instantiated the ‘memory trace’, as could be observed from the maintenance of a high correlation between their levels of activity and the to-be-remembered signal.

The circuits evolved for the continuous version of this task required constant interaction with their environment to maintain their temperature memory. This highlights the situated nature of the learning task. Would it be ‘better’ if the circuits could maintain their memory indefinitely and in the absence of environmental interaction? Traditionally in robotics, ‘memory mechanisms’ are designed to remember everything indefinitely, regardless of how old or new, main or secondary the information is. This is not necessarily the case for living organisms. What accounts for appropriate memorization behaviour when an agent needs to interact with its environment is likely to be very different than the memory required by information processors. Issues of context and time-sensitivity become relevant. For living organisms, remembering recent experiences is usually more important than older ones. For example, you would like to remember where you parked your car this morning, not necessarily all of the locations on previous days! Similarly, remembering highly recalled memories is also more important than recalling less frequently needed ones. For example, you would like to remember the names of people you interact with on a daily basis at work better than those of whom you haven’t seen in a really long time. In fact, several studies have found consistency in forgetting curves across tasks, measurement metrics and even species (Wixted & Ebbesen, 1991). These studies suggest that memory declines as a power function of time. This is the first example known to the authors where similar forgetting curves are observed in artificially evolved circuits for learning behaviour. It is important to note that this is not a limitation of the evolved circuit, but a consequence of its situated nature.

Continuous-time recurrent neural networks without synaptic plasticity have now been demonstrated to be capable of associative learning on both discrete and continuous stimulus spaces. How much further can this approach be taken? The most obvious next step could be to study second and higher-order conditioning, where the initially associated stimulus can consequently be used to learn about some new stimulus. Another useful next step would be to study the blocking effect, a phenomenon observed whereby conditioning to a stimulus is blocked if the stimulus has been reinforced in compound with a previously conditioned stimulus. Both phenomena are discussed in most textbooks on learning. Finally, it is important to note that one of the major differences between our task and the behaviour performed by the nematodes is the agent’s embodiment. In the case of the worm, it influences the sensory stimuli that it receives next by moving up or down the thermal gradient. In our task, the situation is more akin to traditional psychology experiments, where the experimenter immobilizes the subject (e.g. glues the worm to a petri dish) while applying different stimuli to it and studying its responses in a highly structured manner. One important direction of future work will be to analyse evolved circuits using more ecological learning scenarios.
Chapter 7

History of interactions in reactive but situated agents: A visually-guided orientation task with sensory inversion and an object discrimination task

The two results chapters shown up to this point (Chapters 5 and 6) deal with learning in abstract models. This chapter demonstrates the importance of an agent’s situatedness for behaviour, serving as a transition into the rest of the thesis, which gives more emphasis to the role of the agent’s history of interactions with its environment in learning and memory. Although the chapter’s main contribution is a theoretical transition between abstract and situated models in dynamical agents, we provide two concrete examples that illustrate the relevance of this step. We explore the performance of a simple model agent using a reactive controller in situations where, from an external perspective, a solution that relies on internal states would seem to be required. In a visually guided orientation task with sensory inversion and an object discrimination task, a study of the instantaneous response properties and time-extended dynamics explain the non-reactive performance. The results question common intuitions about the capabilities of reactive controllers and highlight the significance of the agent’s recent history of interactions with its environment in generating behaviour. The work in this chapter reinforces the idea that embodied behaviour exhibits properties that cannot be deduced directly from those of the controller by itself. As we will see in Chapters 8 and 9, this will play an important role in understanding situated learning.

7.1 Introduction

Is it possible to deduce the cognitive limitations of an embodied agent from the limitations of its internal dynamics? In particular, is an agent controlled by a reactive system able to perform only reactive behaviours? Questions like these compel us to look carefully at the meaning of now commonly used terms such as embodiment and situatedness, often discussed in the abstract, and try to unravel their implications for concrete systems.

A way of addressing the most specific of these questions is to build agents controlled by reactive systems and evaluate their performance in situations that require non-reactive responses. By a reactive controller we understand a system whose outputs are at each moment only determined by
7.2. Background

In order to make the problem non-trivial we need to define reactive behaviour in terms of the properties of the task and not the controller. For the purpose of this work we adopt a definition of reactive behaviour based on the classification introduced by Clark and Thornton (1997) as the performance of a type-1 task, i.e., a task that requires the agent to exploit regularities which are directly apparent in the current input data. In robotics, obstacle avoidance is typically a type-1 task. In contrast, type-2 tasks require the exploitation of regularities which are ‘hidden’, or whose statistical visibility depends on some ‘systematic recoding’ of the data. Accordingly, we will treat performance of a type-2 task as a form of non-reactive behaviour. Online learning is typically a type-2 task.

In this chapter, evolutionary algorithms are used to design neurocontrollers for the behaviour of model agents that are then analysed dynamically. The goal is to explore the role of embeddedness in space and time in enabling non-reactive performance in systems that can only respond reactively. In particular, we investigate the relation between instantaneous response properties and time-extended performance in orientation tasks, and the time-dependence of responsiveness and ‘decision making’ in shape discrimination. In both cases, embodied agents exhibit properties that cannot be deduced directly from their reactive controllers. The dynamical analysis of these agents allows us to draw some general inferences about the danger of making a priori assumptions about the required properties of internal control mechanisms for a given task.

7.2 Background

We may find classical answers to our opening questions in criticisms of behaviourism. For instance, in the critique of the reflex-arc concept in psychology by Dewey (1896) it becomes clear that action is ongoing and stimuli can only have an effect on the behaving agent because the agent is capable of selecting them actively by the nature of its perceptual systems but also by the nature of its actions. The same point is compellingly made by Merleau-Ponty:

The organism cannot properly be compared to a keyboard on which the external stimuli would play and in which their proper form would be delineated for the simple reason that the organism contributes to the constitution of that form ... it is the organism itself – according to the proper nature of its receptors, the threshold of its nerve centers and the movement of the organs – which chooses the stimuli in the physical world to which it will be sensitive. (Merleau-Ponty, 1967, p. 13)

We find similar views in Varela’s work (e.g., Varela et al., 1991), where the emphasis is on cognition as embodied action wherein the world and the perceiver mutually specify each other. This is closely related to von Uexküll’s functional circles (Lashley & Schiller, 1957), i.e., the formation of a closed unit between the ‘perceptual’ and ‘effector’ worlds that enables an agent to generate its own Umwelt. In robotics, Pfeifer and Scheier (1999) subscribes to a related view, showing the importance of thinking in terms of sensorimotor coordinations. In recent years, we have seen concrete examples of these ideas at work in the area of autonomous robotics. For instance, Nolfi (2002) provides examples in object size and shape classification tasks using reactive controllers, Scheier, Pfeifer, and Kunyioshi (1998) make similar points by studying object-constancy and focus-of-attention problems using hand-coded physical robots as well as evolved simulated agents, and van Dartel, Postma, and van den Herik (2005) discussed perceptual ambiguity in the
context of a reactive agent. Implications of the embodied view in the context of biological neural networks have been summarised in Chiel and Beer (1997).

What we propose to do here is to focus on the opportunities that the ‘neural’ architecture, body and environment offer to the system’s controller. We will show how and why an embodied system can perform non-reactive behaviour (type-2 tasks) even when only endowed with a purely reactive controller. The interesting lessons will be in the details of how the agents work because they reveal the hidden assumptions about the capabilities of embodied and situated systems, even when their internal controllers are very simple.

7.3 Methods

We propose to study the role of the agent’s situatedness using a set-up similar to the one presented in Beer (1996, 2003) with slight variations and extensions on the architecture and tasks. This model has been chosen for two reasons: its simplicity and its potential for sufficiently interesting behaviours that could be called minimally cognitive. As a modification to this set-up, the controller’s architecture is made purely reactive. Two tasks are studied: an approach/avoid task which is made type-2 by the fact that the sensor array may be inverted, and a discrimination task to evaluate the reactivity of the agent at different stages. These are described in the following sections.

The agent has a circular body with a diameter of 30. The agent’s ‘eye’ consists of six rays at $\pm\pi/12$, $\pm\pi/24$ and $\pm\pi/36$ from the centre. An intersection between a ray and an object causes an input to be injected into the corresponding sensory node, with the magnitude of the injected input inversely proportional to the distance to the object along that ray with a maximum length of 220. When the rays are at their maximum length, no input is injected, while the maximum input is injected for rays of zero length.

The agent can move horizontally as objects fall from above (Figure 7.1A) with horizontal velocity proportional to the sum of the opposing forces produced by two motors. The behaviour of the agent is controlled by a network of continuous-time recurrent neural nodes of the form:

$$\tau_i \dot{y}_i = -y_i + \sum_{j=i}^N w_{ji} \sigma(g_j(y_j + \theta_j)) + I_i$$

(7.1)

where $y_i$ is the activation of each node, $\tau$ is its time constant, $w_{ji}$ is the strength of the connection from the $j^{th}$ to the $i^{th}$ node, $\theta$ is a bias term, $g$ is a gain, $\sigma(x) = 1/(1 + e^{-x})$ is the standard logistic activation function, $I$ represents a constant external input (e.g., from a sensor) and $N$ is the number of nodes in the network. In simulation, node activations are calculated forward through time by straightforward time-slicing using Euler integration with a time-step of 0.1.

The network architecture is bilaterally symmetric in the connection weights, biases and time-constants (unless otherwise specified). The architecture consists of six ray sensory nodes projecting to five inter-nodes, which in turn project to two motor nodes controlling horizontal motion (Figure 7.1B). All the sensory nodes share the same time-constant, bias parameter and gain parameter, while the rest of the nodes have a gain of 1.

The controller is made reactive by eliminating the self-connections, changing the connection weights between the inter-nodes to 0, fixing the time-constants for all nodes to 1, and modifying
7.4 Orientation experiments with visual inversion

In the first set of experiments, visually guided agents are evolved to adjust their horizontal position so as to catch or avoid falling objects with normal and inverted vision. On inverting the visual field in the left/right direction an object that appears to the right of the agent will in fact be to its left. This task represents a type-2 problem, for it requires an agent to perform differently for the same stimuli depending on the context.

A simple evolutionary training regime is used. During an evolutionary evaluation 21 circular objects are dropped from the top of the environment straight down with an initial horizontal offset from the centre of the agent uniformly distributed in \([-50, 50]\) and a fixed vertical velocity of \(-3\).
Following Di Paolo and Harvey (2003), this is repeated for objects of different diameter (i.e., 26, 30 and 34). The whole process is then repeated using inverted vision, for a total of 126 trials in a fitness evaluation. At the start of each new trial node activations are initialised to zero.

The performance measure to be maximized is: \[ f = 1 - \frac{\sum_{i=1}^{N} d_i}{N}, \] where \( N \) is the total number of trials and \( d_i \) is the horizontal distance between the centres of the object and the agent when their vertical separation goes to 0 on the \( i \)th trial. Performance is clipped to a maximum of 50, and normalized to run between 0 and 1. Such that, a performance of 1 corresponds to an agent that catches the object, and a performance of 0 corresponds to one that ends up at a distance of 50 or further from the object.

Agents with a reactive controller that could orient to falling circles with normal and inverted vision turned out to be relatively easy to evolve. Over 20 evolutionary runs, the best-evolved agent achieved a mean performance of 0.994 on the 126 evaluation trials after only 100 generations of evolution, with a mean performance of 0.992 using normal vision and 0.990 with inverted sensors on \( 10^4 \) randomly generated trials distributed with initial horizontal positions in \([-50,50]\], and diameter and vertical velocity of the falling object between \([20,40]\) and \([-4,-2]\), respectively.

In Figure 7.2, the behavioural trajectories of the best-evolved agent are shown. For these experiments, the falling object used is a circle of diameter 30. Each figure contains 51 trajectories, depicting each of the different initial horizontal offsets. The agent is tested using normal (left column) and inverted vision (right column). The trajectories are superimposed on the steady-state approach/avoid map. The steady-state map represents the long-term horizontal velocity of the agent relative to the falling object. Each point depicts whether the agent would avoid or approach an object in that location. It is calculated by fixing the object in the same position in the agent’s field of view. This is repeated for every point in its field of view. Regions in which the agent is directed towards the object (centring regions) are white, whereas those in which it is directed away (avoidance regions) are black. The magnitude (i.e., how slow or fast the agent approaches or avoids the falling object) is represented by the intensity of the shade. Accordingly, mid-grey represents little or no movement when an object is in that region.

Figure 7.2A shows the strategy used by the agent to catch falling circles with normal (left) and inverted vision (right). Notice the opposed shading of the velocity fields in the two conditions. As the controller is reactive, a stimulus produces an instantaneous motor effect. This will result in either moving towards or away from the object. Because the object in the inverted vision scenario is on the opposite side of the agent’s field of view, an object in a particular location that would cause the agent to move in, for example, its direction, would now under inverted vision move in the opposite direction. So the steady-state map that are opposite in the case of normal and inverted vision. A real and a virtual object in the same position produce exactly the same instantaneous effect. Yet the situated behaviour of the agent over time results in trajectories that catch both virtual and real objects.

In the normal condition, trajectories are attracted to the centre where the velocity field turns slightly divergent and then ‘trapped’ by the two bright regions of centring velocities which eventually converge on the object’s horizontal position. In the inverted condition, central trajectories become convergent by the nature of the central field, and the rest of the trajectories initially move away from the centre only to be trapped in a different and wider convergent region, reaching the
7.4. Orientation experiments with visual inversion

Figure 7.2: Trajectories and steady-state horizontal velocity fields for normal (left) and inverted (right) vision for the best-evolved agents for objects of diameter 30 and 51 different initial horizontal offsets for: [A] Circle centring task with a reactive symmetrical network; [B] Circle centring task with a recurrent symmetrical network; [C] Circle centring task with a reactive non-symmetric network; and [D] Circle avoidance task using a reactive symmetrical network. Trajectories are superimposed on differently shaded regions representing the long-term horizontal velocity that the agent would eventually adopt if an object were fixed at that location in its field of view at each point as a function of $x$ the horizontal offset in relation to the agent and $y$ the vertical distance of the object. Regions in which the agent is directed towards the object (centring regions) are bright, whereas those in which it is directed away (avoidance regions) are dark. The magnitude is represented by the intensity of the shade (mid-grey is no movement).
centre when the divergent fields no longer have the same effect. The evolved strategy involves taking advantage of the agent’s multiple sensors and most successfully evolved agents relied on a very similar strategy.

Recurrent and time-based networks were evolved as well and analyses of the best-evolved controller yielded the use of a similar strategy to that of the above-analysed reactive network. Figure 7.2B shows the behaviour of the best-evolved recurrent network over 20 evolutionary runs. Again, most successfully evolved agents relied on similar strategies.

Agents with network architectures not constrained to be bilaterally symmetrical seemed to be relatively easier to evolve. The behaviour of the best-evolved agent is shown in Figure 7.2C. The agent’s strategy is to position itself sufficiently to one side of all falling objects, at which point real objects are seen with its right-sided sensors while virtual objects with its left set of sensors. The agent can then centre in on objects with opposite reactions according to which side they appear to be on. The result is a much simpler strategy for centring on both real and virtual objects.

A reactive agent needs to constantly engage with sensory stimuli in order to act which makes avoiding (as opposed to catching) falling objects with normal and inverted vision a counterintuitive task. Agents evolved on this variation of the task seemed to take relatively more generations to achieve high performance, with the best performing agent over 20 evolutionary runs achieving a mean performance of 0.9526 on the 126 evaluation trials after 100 generations of evolution and 0.9012 on $10^4$ random trials. Figure 7.2D shows the behaviour and dynamics of the best evolved reactive controller for such task. From the figure its strategy can be easily understood: under normal vision, the agent avoids objects that are far away and centres on objects that are relatively close. As a result, real objects get avoided as they start falling and disappear from the field of view early, while virtual objects are initially centred, reaching then the point were sufficiently closed objects get avoided.

### 7.5 Categorical perception experiments

In a second set of experiments, we explore agents that can discriminate between circular and horizontal bars of different sizes using normal vision, catching the former and avoiding the latter in a similar task to the one explored in Beer (2003), Di Paolo and Harvey (2003), in this case using a reactive controller. The evolutionary training regime used was similar to that used in the first set of experiments, with the only difference that half of the trials corresponded to circular falling objects and the other half to bar objects (as opposed to sensory inversion).

The performance measure to be maximized is: $f = p_i / N$, where $N$ is the total number of trials and $p_i = 1 - d_i$ for a circular object and $p_i = d_i$ for bars. $d_i$ is defined and normalized as above. This function prevents the avoidance of bars by large distances from dominating the fitness at the expense of accuracy in catching circles. Therefore, the range of fitness is set to vary between 0 and 1, where 1 corresponds to an agent that catches circles and avoids bars, 0.5 corresponds to one that catches or avoids both, and 0 corresponds to one that does the opposite of the task (i.e., avoids circles and catches bars). Following Beer (2003), a class of parameterised hybrid object that continuously interpolates between the bar and the circle is defined as $R(\alpha) = \alpha R_b + (1 - \alpha) R_c$, where $R(\alpha)$ is the radius of the hybrid object, and $R_b$ and $R_c$ are the radii (in polar coordinates) of the bar and circle respectively (see Figure 7.4A). The agent is, nevertheless, exposed only to
7.5. Categorical perception experiments

Figure 7.3: Performance for circle catching [A] and bar avoidance [B] for the best agent evolved using different object sizes between (diameter between [22,40]) and horizontal offset (±50). Bright represents high fitness. Circles represent the actual evolutionary training data.

‘perfect’ circles (\(\alpha = 0\)) and bars (\(\alpha = 1\)) during evolution.

Over 20 evolutionary runs, the best-evolved agent achieved a mean performance of 0.970 on the 126 evaluation trials after 100 generations of evolution, with a mean performance of 0.915 on \(10^4\) randomly generated trials from a broader set (initial horizontal positions between [−50,50], diameter size of the falling object between [20,40] and vertical velocity between [−4,−2]). In Figure 7.3, the catch and avoidance performance as a function of object size and initial position is given for a fixed vertical velocity of −3, as expected there are narrow spikes of poor performance in the centreline for bar avoidance behaviour due to the agent’s symmetrical constraints. The agent is able to generalize reasonably well to untrained examples, except for much smaller circles.

Two major defining characteristics of categorical perception are labelling and discrimination. For this agent, classification is manifested in its approach or avoidance behaviour. In order to demonstrate these, the mean catch or avoid performance was plotted as the shape of the object changed between a circle and a bar (by modifying the parameter \(\alpha\)). Figure 7.4B depicts the average catch performance as a function of \(\alpha\), a sharp sigmoidal curve with a transition from catching to avoidance behaviour at about \(\alpha = 0.55\) is observed. Accordingly, the average difference (see Figure 7.4C) in catch performance for \(\alpha\) values that differ by 0.1 as a function of \(\alpha\) shows a bell shaped function.

How are we to explain the behaviour of the agent? What sort of regularities does it exploit from the environment? The behaviour and steady-state dynamics of this agent are shown in Figure 7.5. The evolved strategy involves positioning all falling objects at a particular angle of view where the difference between the two objects is maximized. This can be appreciated from dominating dark regions in the middle-top field of view of the steady-state velocity. At the point where the object is positioned close to the border of the agent’s field of view, circular objects fall onto a very thin bright region of centring behaviour; bar-shaped objects miss this region, ultimately dropping out of the agent’s field of view. This is further explained from a closer look at what the agent ‘sees’ (data not shown), a circle never stimulates less than 2 sensors, while the bar stimulates only 1 sensor at one point, and this makes it move out of the sensor range.
Figure 7.4: Demonstration of labelling and discrimination. [A] Class of parameterised hybrid object that continuously interpolates between the bar and the circle as a function of $\alpha$. Demonstration of labelling and discrimination in the best-evolved feedforward network. Average [B] and difference [C] in catch performance as a function of $\alpha$. Each point represents the mean value for 101 trials at uniformly distributed horizontal offsets in $\pm 50$.

An interesting question in the context of this chapter is: at what point during the trial does the agent commit itself to catching or avoiding an object? What is expected from a reactive agent is a strong dependence, throughout the trial, between the shape of the object and the ‘decision’ to catch or avoid. This is explored by switching the shape of the object during the trial and observing the behaviour. In the absence of an internal state to ‘retain’ a previously made decision, one expects the decision to depend mainly on the shape after the switch.

Figure 7.6 shows the performance of the agent when catching a circle or avoiding a bar as a function of the horizontal offset and the distance where the switch from circle to bar, or vice versa, is introduced. The results are contrary to our expectations. Although the agent seems to be completely uncommitted during the initial movements, after passing a more or less well defined ‘decision line’ it becomes largely committed to either catching or avoiding even if the shape is changed afterwards. The ‘decision process’ is very much a discrete event that occurs in the early stages of the trial. This is not necessarily the case in other systems, such as the circle/diamond discriminator described in Beer (2003).
Figure 7.5: Trajectories and steady-state horizontal velocity fields for the best-evolved agent that can discriminate between circles and bars. The trajectories show the movement of the agent relative to the falling object over time for circles [A] and bars [B]. The trajectories are superimposed over the steady-state map (see explanation for Figure 7.2).

The intuition goes wrong because it generalizes from the instantaneous effect of a pattern of stimuli on a reactive controller to the time-extended and situated behaviour of the agent. If, as explained above, discrimination is achieved by a particular correlation between object shape and angle of sensing chosen by the agent, and if after that event, independently of the decision made, the agent is already positioned in either a neutral or a centring velocity field, then any subsequent change of shape will be ignored. This is because behaviour does not depend on the objective shape of the stimulus but more precisely on the sensorimotor correlation brought about by the agent’s behaviour in the presence of the objects.

7.6 Discussion

This chapter has demonstrated the evolution of embodied agents with reactive controllers for visually guided-orientation with sensory inversion and object discrimination. Although the tasks are interesting in themselves, the point of this chapter is not to generate novel behaviour but to probe the intuitions concerning the capabilities of reactive controllers, highlighting the significance of the agent’s recent history of interactions with its environment in generating behaviour.

This work provides concrete examples showing how an embedded system is never purely reactive. From the example of shape discrimination, we show that the evolved agent will exploit
Figure 7.6: Performance as a consequence from switching the object’s identity from a circle into a bar [A] (and vice versa [B]) at different times on the final decision to catch or avoid in the best-evolved agent. Each point represents the centring [A] and avoidance [B] performance as a function of initial horizontal offsets ($x$) and switching times ($y$). Bright means high performance. All figures are for objects of diameter 30 and vertical velocity -3.

state arising from its interaction with the environment and exhibit commitment to a decision. Step by step, the agent’s controller acts reactively, but as a result of the ongoing coupling with its environment, the agent is capable of solving a discrimination task that requires it to have some form of memory or state. We show that the evolved agent-environment system will exploit state arising from the sequence of sensorimotor loops. Agents modify their position with respect to other objects in the environment and, thus, partially determine the sensory patterns they will receive in the next time-step, thereby providing a concrete example of an agent creating the form of the stimulus by its manner of offering itself to actions from the outside, paraphrasing Merleau-Ponty.

For the visual inversion experiment the agent relies on following time-extended dynamics. As the state of the controller depends only on the pattern of inputs, the velocity fields for the normal and inverted conditions are point-by-point, opposed to each other. Which does not mean that the final state of the whole trajectory will be different in each case. This prompts an important conclusion: the limitations of reactive controllers (or generally any given class of controllers) during extended periods of time in a situated system cannot be trivially deduced from the instantaneous, snapshot limitations of the same controllers. Inversion of the sensory array produces an instantaneous reversal of velocities, and yet it results in a conservation, not a reversal, of the end-state of behaviour.
This work reinforces the major advantages of the evolutionary, embodied, and dynamical systems approach to cognition. By grounding the agent’s behaviour in its environment, we illustrate some of the implications of reducing the assumptions about the necessary design of the agent’s internal control mechanisms. In the visual inversion scenario, losing the symmetrical constraints allows the agent to redefine the problem into an easier one: catching objects that fall only to one side of it.

We do not deny the importance of an agent’s internal dynamics in the generation of behaviour. Rather, this work emphasises the importance of the agent’s interaction with its world (even for the extreme case of no internal dynamics). It is obvious that an agent with internal dynamics has a broader set of strategies available for performing a task. It may, nevertheless, be the case that agents with internal dynamics exploit first the strategies available from its situatedness alone. In the visual inversion experiments, agents with internal dynamics have the potential to solve the task in a variety of ways – for example, learning online which way around the sensors are wired up – and then acting accordingly. It is likely, however, that the evolved agents make use of the simpler embodied strategies first, as is shown from the evolved recurrent network with internal state.

In summary, a reactive controller in an embodied system doesn’t imply reactive behaviour: there is a difference between the local, instantaneous state definition of reactivity, and the behavioural definition, i.e., not being able to solve type-2 problems such as approach or avoidance under normal and inverted vision. As a result, whether an embodied agent will behave reactively (i.e., whether it will only be capable of performing behaviours of type-1) cannot be fully determined by the presence of a reactive controller.

The strategy proposed by minimally cognitive tasks for a critical examination of internal representation is straightforward: evolve agents on tasks that are ‘representationally-interesting’, then examine whether the agent is using representations (Beer, 1996). In this case, no internal state is available for manipulation, thus, trivially, nothing is ‘internally represented’, yet behaviours such as commitment to discrete decisions on a categorisation task can still be demonstrated.

Future work should explore extensions to the capabilities of reactive controllers in a variety of directions. In general, it will be interesting to continue to relax a priori assumptions and consider how dynamical, bodily, and environmental constraints can transform ‘cognitively hard’ problems into easier ones (cf. Fine, Di Paolo, & Izquierdo, 2007). Some of these directions include: the effects of physical inertia, non-symmetrical architectures and noisy inter-node interactions.
Chapter 8

Discrete associative learning in embodied and situated agents: A temperature preference task

In this chapter we study associative learning in a fully situated and embodied model. This chapter extends previous work in this area by requiring that a situated agent be capable of re-learning during its lifetime. Agents controlled by neural networks with fixed-weight controllers are evolved to search for food and associate it with one of two different temperatures depending on experience. The task requires either instrumental or classical conditioned responses to be learnt. We analyse the best-evolved agent’s behaviour and explain in some depth how it arises from the dynamics of the coupled agent-environment system.

8.1 Introduction

Associative learning requires responses to be paired with a particular stimulus. Organisms at several levels of ‘complexity’ provide evidence for this, including many extraordinarily simple ones. In the small nematode worm *C. elegans*, evidence for the formation of associations between temperatures and food has been known for quite some time (Hedgecock & Russell, 1975). However, the mechanisms required for the storage and resetting of this memory are still largely unknown.

In animal learning theory there is the idea of the strengthening of a ‘connection’ between a stimulus and a response. This has been directly translated to the strengthening of physical connections between neurons. While this is a good description at the level of the agent’s interaction with the environment (behavioural description), there need not be a direct correspondence of connection-forming processes in the internal behaviour-producing mechanisms of the agent. We believe there is a more fundamental principle underlying learning behaviour at the level of an organism’s internal mechanisms that has to do with dynamics on multiple timescales.

The aim of this work is to: (a) successfully evolve an integrated dynamical system controller with fixed weights in an embodied and situated agent on an associative learning task requiring re-learning, (b) perform a behavioural analysis of the best-evolved agent; and (c) study the coupled agent-environment dynamics of a successful controller and attempt to understand it as implementing a finite state machine (FSM), so as to compare with similar work (Phattanasri et al., 2007).
8.2 Related Work

A number of researchers have used genetic algorithms to evolve, for tasks requiring associative learning, dynamical neural controllers without in-built synaptic plasticity mechanisms or other learning algorithms (see Section 4.3). Yamauchi and Beer (1994a) were the first to explore this idea using a one-dimensional navigation task with a goal and a landmark. Attempts to evolve an ‘integrated’ network failed, so a modular approach was taken.

Blynel and Floreano (2003) evolve controllers on a relatively similar task and environment. In their version, because the light is fixed to one side of the arena and the goal is the only thing that changes, it is possible for the agent to employ a reactive turn left or right strategy, as opposed to approaching or avoiding the light; making it unnecessary to form an association between light and goal. Attempts to remedy those initial difficulties were successfully overcome by Tuci et al. (2002a) in a two-dimensional version of the same task. As the emphasis of that work was on the evolutionary process, no further analysis of the behaviour or internal dynamics was performed.

Fernando (2002) explores the same associative learning task in a slightly more complicated T-maze environment. Despite not being able to evolve an agent that solves the task completely, an analysis of the best performing agent in terms of animal learning theory is attempted. Such work demonstrates that multiple timescale dynamics can exhibit learning-like behaviour without synaptic plasticity mechanisms. However, none of the previous work deals with re-learning during the lifetime of the agent: the agent’s internal state is reset when tested on a different environment. Also, the internal mechanisms of the best-evolved agents have not been explored in much depth or at all in some cases.

Phattanasri et al. (2007) study in-depth the dynamics of an evolved circuit for an associative learning task very similar to the one presented here. In Chapter 6, we extended their work to associative learning on a continuum, but the agent was still disembodied (i.e., a neural network in a vacuum). In the current chapter we extend the analysis of the internal dynamics of an agent performing a discrete associative learning task first considered in Phattanasri et al. (2007) in a different dimension: associative learning in an embodied and situated agent.

The focus of this chapter will be in understanding the dynamics of an embodied and situated agent, that is in constant interaction with its environment, when performing an analogous learning behaviour. For the disembodied scenario, a finite state machine was sufficient to represent the dynamics of the agent performing the associative learning task (see Section 6.3). We would like to know whether the same analysis is possible for an embodied and situated agent. In order to do this, we will have to take into account what we learnt in Chapter 7: evolved agents often exploit their history of interactions with the environment to solve tasks. In this chapter we study how embodiment and situatedness affects our analysis and understanding of the network.

8.3 Methods

We use evolution to synthesize continuous-time recurrent neural networks that display associative learning behaviour when situated. The task is loosely abstracted from the temperature preference behaviour observed in the nematode worm C. elegans (Hedgecock & Russell, 1975). In particular, we would like an agent that is capable of associating temperature with food in two different
We use a 2D arena with a thermal gradient along one of its dimensions containing two types of food: ‘nutritious’ and ‘poisonous’. Each type of food can be found only in regions in a particular temperature range: ‘hot’ between [9,10]; ‘cold’ between [-10,-9]. Which region the nutritious food can be found in depends on the type of environment: \( T \)-env, nutritious food in the hot region; and \( \perp \)-env, in the cold region. For each of the different environment types, the poisonous food can be found in the opposite region to the nutritious food. There are no walls and the thermal gradient extends in all directions.

An example trial of the task is depicted in Figure 8.1A. The task involves placing an agent at random in the central region (between [-2,2]) of the arena (including random orientation) in one of the two environment types, requiring it to find and stay on the food as efficiently as possible. The first challenge involves exploring the whole of the arena in search for food. After a random amount of time (between [80,100] units), the agent is physically displaced back towards the central region of the arena and given a random orientation again. A successful agent should navigate up or down the thermal gradient depending on whether it had found food in the hot or cold region in the previous trial, respectively. This requires that it learn and remember in which of two environment types of environment, and re-learning: modify its temperature preference during its lifetime when required.

Figure 8.1: Task set-up, agent’s architecture and the structure of an individual trial. [A] 2D environment with a thermal gradient. ‘Nutritious bacteria’ can be found in one of the two patches depicted by the spotted regions, ‘poisonous bacteria’ on the other. [B] Agent architecture with 3 fully inter-connected nodes, a food and a temperature sensor, and two wheels controlled by arbitrarily chosen nodes. [C] Example trial: one-dimensional projection of environment with thermal gradient (shades of grey). ‘Nutritious food’ denoted by white bars; ‘poisonous’ with black.
8.3. Methods

Less frequently, the displacement involves changing the environment type as well. This requires that the agent remain sufficiently plastic to change its temperature preference online. Although it is this learning and re-learning phenomena that are central to this chapter, there is also a more basic sensory-motor challenge involved in navigating up and down the thermal gradient which will not be explored.

Agents are modelled as circular bodies of radius 1 with two diametrically opposed motors and two sensors. Agents can move forwards and turn. The mass of the body is sufficiently small so that the motor’s output is the tangential velocity at the point where the motor is located. The agent can sense the local temperature in the environment as well as the food. The food, however, cannot be perceived unless the agent is directly upon it. The food sensor is: 1 for nutritious food, -1 for poisonous food, and 0 when no food is present. The temperature sensor can have any real value.

For the internal dynamics of the agent, we use a continuous-time recurrent neural network (CTRNN) with the following state equation (Beer, 1995b):

$$\tau \dot{y}_i = -y_i + \sum_{j=1}^{N} w_{ji} \sigma(y_j + \theta_j) + s_i T(x) + g_i F(x; e)$$

where $y$ is the activation of each node; $\tau$ is the time constant; $w_{ji}$ is the strength of the connection from the $j^{th}$ to the $i^{th}$ node; $\theta$ is a bias term; $\sigma(z) = 1/(1 + e^{-z})$ is the standard logistic activation function; $T(x)$ is the thermal sensor, a function of the agent’s position along one of the dimensions of the physical space, $x$; $s_i$ is the strength of the connection from the thermal sensor; $F(x; e)$ is the food sensor, also a function of $x$ but parameterized by the type of environment, $e$; $g_i$ is the strength of the connection from the food sensor; and $N$ represents the number of nodes in the network. In simulation, node activations are calculated forward through time by straightforward time-slicing using Euler integration with a time-step of 0.1. The network is fully connected (see Figure 8.1B). There are no additional weight changing or any other parameter changing rules.

The connection weights, biases, and time-constants in Equation 8.1 are encoded in a genotype as a vector of real numbers and evolved using the Microbial genetic algorithm (Harvey, 2001). The size of the population used was 50. We define a generation as the time it takes to generate 50 new individuals.

The fitness of a circuit is obtained by minimising the relative distance away from the food at the beginning of each test ($a$), and maximising the time spent sensing food towards the end of the same phase ($b$), according to

$$a = \int_{t=0}^{50} \left( \frac{20 - d}{20} \right) dt , \quad b = \int_{t=30}^{80} F dt$$

where $F$ is the agent’s sensor for food and $d$ is the absolute distance between the source of food and the position of the agent capped at 20. Both components are normalized to run between 0 and 1. The two components are clearly linked: the first provides emphasis on heading in the direction towards where the food should be at the start of the trial; the second emphasizes staying directly on top of the food once found.

A fitness trial consists of the evaluation of an agent’s performance for the number of times it is displaced in the same environment type, $p$, and for the number of changes of environment type, $k$, all without reinitialising the agent’s state. No evaluation takes place at the start of a trial, nor
immediately after a change of environment type. This is repeated 50 times for each individual and
the fitness taken from the multiplication of their averages, \( f = \bar{a} \cdot \bar{b} \). Each repetition involves the
re-initialisation of the agent’s internal state.

Following Phattanasri et al. (2007), a set of evolutionary stages of increasing complexity are
employed. The changes are in the starting orientation of the agent, \( \phi \), after each start of trial or
displacement; in the number of times an agent is tested (i.e., displaced), \( k \); and the number of
changes of environment type, \( p \); as follows:

<table>
<thead>
<tr>
<th>Stage</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \phi )</td>
<td>( 0, \pi )</td>
<td>( 0,2\pi )</td>
<td>( 0,2\pi )</td>
<td>( 0,2\pi )</td>
<td>( 0,2\pi )</td>
</tr>
<tr>
<td>( k )</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>( p )</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>[1.5]</td>
<td>[1.5]</td>
</tr>
</tbody>
</table>

Transitions occur when the best fitness exceeds 0.8 consistently (i.e., for 5 consecutive generations). At the last stage, the orientation is chosen at random from the full range, the environment
type changes 5 times during the agent’s lifetime, and the changes occur between the first and the
fifth displacement at random.

### 8.4 Results

#### 8.4.1 Evolutionary performance

We attempted evolving 3-, 4-, and 5-node circuits for this task using 15 evolutionary runs with
different seeds for \( 10^4 \) generations each. The proportion of evolutionary runs that reached the
different stages are depicted in Figure 8.2. While no 3-node populations reached the last stage,
several 4 and 5-node populations did. In fact, the majority of 5-node runs were highly successful,
but we will focus our attention on the smallest successful circuit obtained. The interest in evolving
the smallest circuit that solves the task is primarily to make the analysis most amenable to the
mathematics of dynamical systems theory.

An example evolutionary trajectory for the population that produced the best 4-node agent is
shown in Figure 8.3. As can be seen, the fitness drops sharply after every transition except the
Figure 8.3: Fitness vs. generation for the best-evolved 4-node population (best in black and average in grey). Transitions between stages (dashed lines) occur when the best fitness consistently exceeds the horizontal grey line and are labelled accordingly.

Table 8.1: Best 4-node circuit for the embodied temperature preference task

<table>
<thead>
<tr>
<th>y_1</th>
<th>y_2</th>
<th>y_3</th>
<th>y_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>y_1</td>
<td>-1.8126</td>
<td>2.4783</td>
<td>-9.2366</td>
</tr>
<tr>
<td>y_2</td>
<td>4.3620</td>
<td>-3.0999</td>
<td>5.3725</td>
</tr>
<tr>
<td>y_3</td>
<td>9.8654</td>
<td>-8.5512</td>
<td>-2.1634</td>
</tr>
<tr>
<td>T</td>
<td>-0.6210</td>
<td>0.6595</td>
<td>0.6393</td>
</tr>
<tr>
<td>F</td>
<td>-8.7358</td>
<td>-9.5254</td>
<td>1.8491</td>
</tr>
<tr>
<td>θ</td>
<td>-7.3135</td>
<td>8.4150</td>
<td>9.3576</td>
</tr>
<tr>
<td>τ</td>
<td>1.0573</td>
<td>1.0230</td>
<td>9.4312</td>
</tr>
</tbody>
</table>

last: once the circuit is able to generalize to all learning scenarios. It is the best agent of this evolutionary run that will be analysed in some depth in the rest of this chapter.

What are the characteristics of the parameters of the best-evolved 4-node circuit? In Figure 8.4, we show a graphical representation of all evolved parameters (see Table 8.1 for the actual values). There are three things worth noting. First, we can observe a relatively high bilateral inverse symmetry: the way in which neurons 1 and 2 connect to neurons 3 and 4 is of similar strength but inverse effect (i.e., inhibitory or excitatory). Also the bias of node 1 is very low while that of node 2 is high. Both nodes are very fast acting. Node 3 strongly excites neuron 1 and strongly inhibits neuron 2. Their self-connections are both small. The temperature sensor inhibits 1 and excites 2. This symmetry is interesting because, despite imposing no constraint on the parameters, the evolved symmetry is well aligned with the structure of the agent’s body. Second, the food sensor evolved strong inhibitory connections to both of the motor neurons. As we will see ahead, this allows the agent to stay on the food once it is found. Third, and perhaps most importantly, all
8.4.2 Behavioural analysis

The performance of the best circuit was further tested using $10^4$ evaluation trials, each with 10 changes of environment type, between [1,10] displacements, noise in the sensors and motors drawn from a Gaussian distribution ($\sigma=0.05$), and a time-step an order of magnitude smaller (0.01). As we are interested in how well the agent finds the nutritious food in the face of changing environments, only the $b$ component of fitness is considered. The best 4-node circuit obtained 98.81% success rate on this test, meaning that it generalizes well on a broad range of situations. Since the slope of the thermal gradient remains constant throughout evolution, the agent could use the distance instead of the temperature as the relevant factor to remember. We used the same test while varying the slope of the gradient between $\pm 20\%$ with the success rate dropping by only a minor fraction (98.48% success), meaning the agent relies on the temperature and not the distance the food is away from the centre.

How many trials does this agent need to learn an environment? In order to answer this question we can measure the time it takes the agent to reach the food every-time the agent is displaced towards the centre of the environment. Figure 8.5A shows such a learning curve for the best-evolved circuit on the two different possible environments: when food is in the ‘hot’ region (grey squares) and when food is in the ‘cold’ region (black triangles). Each point shows the average time taken over $10^5$ runs. It is worth noting that 19.5 is the minimum amount of time it can take an idealised agent to go from the centre to the food, assuming it is heading in the direction of the food. As is expected time to get to the food increases from the first time an agent is exposed to an environment to the rest. The best performance occurs on the second trial, where the agent approaches the average minimum possible time. On subsequent trials, the time it takes to reach the
Figure 8.5: Memory performance of one environment and re-learning ability. [A] Average time it takes the agent to reach the food in the two different environments: when food is in the ‘hot’ region (grey squares) and when food is in the ‘cold’ region (black triangles). The agent is displaced to a random location every 100 units of time. This is repeated 20 times in the same environment. [B] Average time it takes the agent to get to the food while changing the environment every two trials (dashed vertical lines).

food stabilizes around 21.5. In other words, the evolved circuit shows no sign of forgetting: once it learns where the food is, it always follows the gradient in the appropriate direction. One possible reason for the increased performance on the second trial over the rest is that, due to the shaping protocol employed, agents have experienced the second trial more often (over evolutionary time) than all other subsequent trials.

How plastic is this agent? We know it can learn which environment it is in. But does this plasticity degrade with time? We can address this question by looking at the average time it takes the agent to get to the food while changing the environment every two trials. In Figure 8.5B each data point represents the average time to arrive at the food over $10^5$ runs for 20 consecutive trials. A first trial is allowed for learning and a second one is tested to demonstrate that an improvement with experience has occurred. After this, the environment is changed to the opposite one (vertical dashed line). As can be appreciated, on every second run after a change of environment the time is close to the optimal. On all of the first trials after a change of environment, the time taken is even longer than on the first time. We can summarize this as: not knowing is better than knowing the wrong thing, on average.

Figure 8.6 shows the behaviour of this agent on a typical sequence trial with 2 changes of environment type. At the beginning of the trial, the agent navigates down the thermal gradient but switches to navigating up before reaching the usual region where food could have been located. This is part of the search strategy, as it does not yet know in what type of environment it finds itself. When displaced for the first and second times after reaching the food, however, it navigates more directly up the thermal gradient. Subsequently the environment type is changed, unaware the agent navigates up the thermal gradient as for previous trials, with the difference that negative reinforcement is encountered (but only very briefly$^1$). The agent navigates past this food region and eventually changes behaviour to navigate in the opposite direction of the gradient, until reaching

$^1$Absence of poisonous food in the environment does not affect the learning behaviour in this agent. The reason is that the negative reinforcement is redundant in this task.
Figure 8.6: Activity of the best 4-node circuit on a typical trial sequence. From top to bottom the traces correspond to the food signal (F), the temperature signal (T), and the outputs of the neurons (o₁, o₂, o₃, o₄). The last two neurons control the right (rm) and left (lm) motors. The dark grey horizontal bars in the temperature trace depict where nutritious food is to be found for that trial (⊤ or ⊥). Dotted vertical lines mark different trials (where the agent is displaced). Dashed lines mark transitions between environments.

The nutritious food on the cold region. On subsequent trials, the agent will navigate directly down the gradient, showing that it remembers where the food was last found in the other type of environments as well. A similar pattern is observed in the second change of environment type. This demonstrates the agent’s ability to learn and remember its past behaviour, as well as the flexibility to remain plastic to ongoing changes in the environment type.

We note that all 4 nodes are active at one point or another during the sequence trial; with most of the activity occurring during the navigation phase. Particularly interesting is the activity of node o₃, which seems to be the only one keeping track of which environment type it finds itself in. This is also the node with the largest time constant in the circuit; all other nodes are as fast acting as allowed (see Figure 8.4).

Before any experience, does the agent navigate up or down the thermal gradient? and what does this depend on? We studied the long-term behaviour of the agent when initialised in an environment with nutritious food on both cold and hot regions. As can be seen in Figure 8.7A, what the agent does depends mainly on its starting position: visiting the furthest region first. How does experience affect this pattern? After learning has occurred, the agent will preferably head towards hot or cold regions, even if nutritious food is placed in both areas, depending on where food was found in the previous trial (see Figures 8.7B and 8.7C, respectively). This shows
8.4. Results

Figure 8.7: Experience-dependent behaviour as a function of initial temperature and heading. Points in the map represent the average position (over 20 repetitions) of the agent after 100 units of time with nutritious food on both hot (white) and cold (black) regions while varying its starting position ($p$) and orientation ($\phi$). Points in-between are in shades of grey. Grey dashed lines mark the conditions for which the agent was evolved. Different maps show the agent’s behaviour with different past experiences: Before any experience (A), after a $\top$-environment (B), and after $\bot$-environment (C).

8.4.3 Dynamics of the coupled agent-environment system

We next turn to the dynamics underlying the behavioural phenomena described in the previous section. The primary interest is in understanding how this agent’s dynamics is structured so that where food was encountered in the past affects which direction of the thermal gradient it will navigate towards.

From the equations describing the coupled agent-environment system we can make some general observations. First, the agent is a nonautonomous dynamical system with two inputs, $T$ and $F$. Second, although $T$ varies continuously as a function of $x$, discontinuities are introduced into the dynamics by the food sensor because $F(x; e)$ is a discontinuous function of $x$, making the agent a hybrid dynamical system. Given these two factors, the best way to study its operation is to characterize its autonomous dynamics for all possible combinations and then examine the transient dynamics induced by the agent-environment interaction.

What are the dynamics of the autonomous circuit? In order to answer this, we will consider a set of bifurcation diagrams for different conditions. The dynamics of the circuit change as a function of the local temperature surrounding it. It will also change depending on whether there is food or not. We will first consider the case when food is absent. We will not consider the case of negative reinforcement, because we know that it doesn’t affect the agent’s performance.

First, we would like to know how the equilibria of the system changes as a function of the temperature in the absence of food. This requires that we find out what is the asymptotic state of the neural circuit for fixed temperatures. Given that the circuit has four variables, and the temperature is the parameter that we are varying, then the bifurcation diagram is 5-dimensional. In Figure 8.8 we show four one-dimensional slices of this 5-dimensional diagram, one for each
Chapter 8. Discrete associative learning in embodied and situated agents

Figure 8.8: Bifurcation diagram for the best-evolved circuit in the absence of food. Four one-dimensional slices of this 5-dimensional diagram, one for each of the neurons. The solid curves represent stable equilibria, while the dashed curves represent unstable ones.

Second, we can observe how the equilibria changes as a function of temperature in the presence of food. In Figure 8.9 we depict slices of the bifurcation diagram of the best-evolved circuit. We are only interested in the areas shaded in grey, as food can only be found here. As can be seen, when food is found in the ‘cold’ temperatures the agent can be in one of two possible states, given that it spends enough time there to reach the stable equilibrium. When food is found in the ‘hotter’ region, there is only one stable equilibrium driving the internal state of the system. As we will see later, the two equilibrium points play an important role in understanding the learning as a form of classical or operant conditioning.

The equilibrium points of the system as a function of the temperature and food dictate where the internal state of the system will end up, if given enough time in an unchanging environment. However, as soon as the agent is dropped into its environment the activity in its neurons activate the wheels and make the robot move. As we saw previously, where it heads depends on both the temperature in which it is first placed and the direction which it faces. The agent’s movement changes the local conditions: temperature gets hotter or colder. Without having a chance to reach the attractors in the initial temperature, the temperature changes and this makes the attractors of the system change as well. The internal state of the system is now headed towards a different
8.4. Results

Figure 8.9: Bifurcation diagrams in the presence of food. Four one-dimensional slices of this 5-dimensional diagram, one for each of the neurons. The solid curves represent stable equilibria, while the dashed curves represent unstable ones. The horizontal grey bars show the ranges of temperature where food can be found.

The system is continually attempting to reach its attractor, but in doing so, it changes where the attractor is.

We can observe the relation between the non-autonomous internal dynamics of the circuit and the dynamics of the coupled agent-environment system. In Figure 8.10 we show low-dimensional slices of the trajectories of the internal state of the agent during a sequence trial imposed over its bifurcation diagram. The trajectories are classified into 6 different groups: (Ai) when the agent is first placed in an $\perp$-environment; (Bi) when the agent is first placed in an $\top$-environment; (Aii) when the agent is re-placed into a $\perp$-environment after having experienced the same environment; (Bii) when the agent is re-placed into a $\top$-environment after having experienced the same environment; (Aiii) when the agent is changed into a $\perp$-environment after having experienced a $\top$-environment; and (Biii) when the agent is changed into a $\top$-environment after having experienced a $\perp$-environment. The black solid line represents the stable equilibria, the dashed line represents the unstable equilibria, the black disks represent the bifurcations. The dotted grey lines show the circuit’s trajectory over time as a function of temperature. The rectangle depicts where the trajectories start and the white disk depicts their end state. The white disk is also the stable equilibria in the presence of food (as seen from Figure 8.9).

The bifurcation diagrams depict a manifold of phase-portraits available to the system. In fact, there are five possible bifurcation diagrams to consider: $P_{\pm15}$ (temperature between [-15,15] with no reinforcement), $P_{\top+}$ (cold temp. and positive reinf.), $P_{\top+}$ (hot temp. and positive reinf.), $P_{\top-}$ (cold temp. and negative reinf.), and $P_{\top-}$ (hot temp. and negative reinf.).
Figure 8.10: Trajectories of neural activity for neuron y4 superimposed over its bifurcation diagram. The diagrams on the left column (A) correspond to trajectories when the agent is placed in $\perp$-environment (i.e., food in cold region). The diagrams on the right column (B) correspond to trajectories when the agent is placed in $\top$-environment (i.e., food in hot region). The three rows depict the different possible previous conditions the agent could have experienced before being placed in either environment. Trajectories are shown when the agent is first placed in such environment (i), when it is displaced in the same environment (ii), and when it is changed from the other environment (iii).

Three-dimensional projections of the stable solutions of the first three of these are shown in Figure 8.11, coded in shades of grey as a function of the temperature and labelled accordingly. The portraits corresponding to the negative reinforcements can be left out of the analysis because they do not affect the performance of the agent’s learning behaviour. As can be seen, for mid-temperatures ($P_{\pm9}$) the long-term behaviour of the system is bistable. As the temperature increases or decreases outside of this range, only one attractor is left in opposite ends of the original for cold and hot. Similarly, for $P_{1+}$ the dynamics are bistable and for $P_{1-}$ there is only one stable state.

How do these bifurcation diagrams combine to produce the learning behaviour? We can study
the transient trajectories in the internal state of the agent as it interacts with its environment. In Figure 8.6B we show a set of trajectories from behaviours crucial for the task using the same projection as in the previous plot. Can we interpret the transitions in the internal state of the agent as implementing a FSM? We were unable to do so. The difficulty arises from the agent’s dependence on the temperature sensing as an ongoing and continuous perturbation. We hypothesize that it is the discretisation and non-situatedness of the task in Phattanasri et al. (2007) that facilitates their FSM interpretation. Only when we consider a different form of state machine that allows for ongoing sensori-motor interactions can we summarize the coupled agent-environment system in relation to the agent’s internal dynamics. We will call this an ‘interactive state machine’ (see Figure 8.12). Although similar, strictly speaking the diagram is not a FSM because some of the states include ongoing interactions with the environment. In it, the finite states the system can be in are denoted by circles labelled: $↑+$ or $↓+$, for when nutritious food is found in the hot and cold regions, respectively. The graded ellipses represent the ‘interactive states’: where the agent’s state moves it in relation to the environment, and the change of temperature changes the dynamics of the agent in turn. There are two of these: $↑$ and $↓$, for what results in navigation up or down the gradient, respectively. We denote the starting internal state as $o$. Physical displacements events are depicted with thick arrows. We can characterise the basins of attraction of the bistable dy-
Figure 8.12: [A] 3D projection of the trajectories in internal space state for a typical set of behaviours. [B] Diagram of the coupled dynamics. Interactive State Machine extracted from the agent-environment interaction.

...dynamics in $P_{\pm 9}$ as a function of the agent’s position and orientation from Figure 8.4A as $\langle p, \phi \rangle^z$, where $z$ represents the long-term behaviour (↑ or ↓). The black arrows denote the encountering of nutritious food, +. The thin arrows connecting the ellipses denote the transition from one stable state in $P_{\pm 15}$ to the other in the internal dynamics. This occurs when the agent reaches colder or hotter temperatures. The diagram up to this point is sufficient to fully characterise the observed behavioural phenomena. There is an additional finite state that is never reached during regular associative learning which we denote as ↓+’.

8.4.4 Predictions from the dynamics

The study of the dynamics suggest a number of predictions which we could confirm using behavioural studies. Although a full study of the predictions would require further space, two of them are mentioned briefly.

First, as a result from the bistability of $P_{\pm 9}$, we could predict and confirm that even after experiencing environments with food in the cold regions, if exposed to hot temperatures and food simultaneously for sufficiently long, the agent could be re-conditioned to navigate up the thermal gradient. This was not the case in the opposite scenario, where the agent required doing the down-the-thermal-gradient navigation behaviour to remember. We can describe the agent as employing a mixture of classical (pairing two signals) and operant (pairing an action with a reinforcement) conditioning.

Second, and as a consequence of the geometry of $P_{\pm 15}$, we could predict and confirm that in the total absence of any kind of food, the coupled system falls into a limit cycle, that involves the agent switching between going up and down the gradient modalities. Although this was not a scenario the agent was evolved for, it could be interpreted as a higher level ‘searching for food’ behaviour that emerges from the lower level behaviours selected for.
8.5 Discussion

We successfully evolved situated agents with fixed-weight dynamical neural controllers on an associative learning task requiring re-learning. The observed phenomena can be described as the ability to perform two different behaviours and appropriately switch between them when necessary using feedback from the interactions with the environment. The question of whether such experience-dependent behaviour is actually ‘learning’ has been discussed in Chapters 4 and 6 and will be discussed once more in Chapter 11. The dynamics of the coupled agent-environment is explored in some depth. Attempts to generate a FSM are unsuccessful but a form of ‘interactive state machine’ is provided instead. From the dynamics, two predictions are explored.

This work raises a number of issues we believe deserve to be further studied. First, in the case of a situated agent, how useful is the conventional distinction drawn between operant and classical conditioning? Our work suggests that the distinction arises from the discretisation of the task or the minimisation of the coupling between agent and environment. Second, in such ‘representationally-hungry’ tasks, correlations between the activity of internal components and that which the agent has to remember are trivial to spot. Could they be interpreted as symbols the agent can manipulate to perform computations? Further work unravelling what is meant by ‘internal representations’ from minimal model systems such as the one presented here should be of interest. Finally, an important next step will be to extend this work to an agent that can associate any temperature along a continuum with food, as is the case in the phenomena observed in C. elegans from which this task was abstracted.
Chapter 9

Learning from a continuum in embodied and situated agents: An experience-dependent categorisation task

In Chapter 6 we studied associative learning on a continuum in a disembodied model. We then explored the role of the history of interactions in a reactive agent to produce non-reactive behaviours in Chapter 7. In the previous chapter, we studied associative learning in an embodied model, where the associative learning was discrete (food can be in the cold or hot region). This chapter combines and extends the previous three chapters, by providing a first set of experiments in learning on a continuum in an embodied and situated model. The task is loosely abstracted from imprinting in birds. The analysis of successful agents demonstrates how memory behaviour can arise from the agent’s situatedness, provided the environment has sufficient invariants that the agent can exploit.

9.1 Introduction

Much of the intelligence that goes on in living organisms appears to be in the interface between the body and the world\(^1\). This is particularly the case in human cognition. Ballard and colleagues have shown that in tasks where people are asked to rearrange arrays of squares, they ‘offload’ their short-term memory to the world when they can (Ballard, Hayhoe, Pook, & Rao, 1997). Smith argues that this offloading is critical to the development of higher-level cognitive functions (Smith & Gasser, 2005). Passive dynamic walking (McGeer, 1990) is one of the most obvious example of offloading, but the role of such agent-environment interaction\(^2\) need not be restricted to very simple (reactive-like) cognitive functions (cf. Chapter 7) nor to purely physical-coordination tasks (e.g., walking). In this chapter we explore how agents can use their interaction with the world to learn.

\(^1\)We believe not just much, but all of intelligence is at such interface. Whether an agent that is not situated and embodied can be called ‘intelligent’ is questionable or likely to involve a definition of intelligence that is not compatible with generating adaptive behaviour. William James famously said “My thinking is first and last and always for the sake of my doing” (James, 1901, Ch. 22). In other words, thinking is for doing. We agree.

\(^2\)We will not refer to the phenomena of using the agent-environment history of interactions as offloading. We think the term is confusing, because it suggests there is some process that would usually be carried out in the agent’s brain but is ‘offloaded’ to the interaction. As we said in the previous footnote, we consider that all cognitive activity is in the interaction. Because it is never really ‘loaded’ purely in the agent’s brain, there is never a need to offload it either.
Learning and evolution are two of the most fundamental adaptive processes in nature. Evolution modifies the behaviour of populations of individuals over many generations, whilst learning modifies the behaviour of organisms during their lifetime. These are two processes that operate at very different time-scales but whose interaction is crucial for the generation of adaptive behaviour. The advantages of learning behaviour for organisms are manifold. In particular, the ability to learn enables an organism to cope with changes in the environment, even if their ancestors have never before experienced such changes.

Over the last decade, various people have evolved dynamical system controllers to perform different behaviours according to their environment. Except for the work shown in Chapter 6, all of such work has in practice been limited to a discrete set of environments, generally two. This chapter focuses on demonstrating for the first time that situated and embodied agents can be evolved to learn to discriminate on some environmental variable that has a continuum of possible values. The question of interest is whether successful agents employ their interaction with the environment to remember the continuous signal (in a manner analogous to how the reactive agents used their interaction to perform a simpler discrimination task in Chapter 7), or if they rely purely on the internal dynamics to remember (as in the example from Chapter 6).

When the changes that can happen to the environment are a finite known set then an agent which is evolutionarily trained on each of these different scenarios can evolve a similar finite number of modes of interaction that enable it to cope with them. The learning behaviour in this case would be the swapping between the different modes of interaction driven by the changes in the environment. This is the case in the examples of learning shown in Chapters 5 and 8. However, organisms can also sometimes adapt to challenges and scenarios that have not previously been experienced. This is the case when, for example, the set of possible scenarios is infinite (on a bounded continuum). A good example of this is imprinting in birds (Lorenz, 1956). In such challenges, agents have to cope with remembering a feature that they may never have trained on previously (e.g., identifying their particular parents). This requires a successful agent to be able to generalize from the set of training examples given during evolutionary time. In this chapter we address the question of how embodied agents may cope with such a challenge.

Imprinting phenomena can be viewed as a way of adaptively modifying behaviour based on the history of interaction with the environment and on the generation of certain environmental regularities such as, in the case of the young of many birds, the first large moving thing they see after hatching. It is obvious in this scenario that building a set of modes of interaction for each possible parent is not satisfactory while evolutionary training with all the possible parents is impossible.

In this chapter we develop a situated and embodied minimal version of an imprinting-inspired scenario, in order to begin to answer for the first time questions regarding learning continuous stimuli phenomena in situated and embodied agents. The idea of the proposed task and model agent follows the minimally cognitive behaviours approach (Beer, 1996), on the one hand simple enough to be analytically tractable, while at the same time bearing a resemblance to learning challenges faced in the real world by living organisms.
9.2 Related work

A framework for studying minimally cognitive tasks was first introduced in Beer (1996). The main idea is to study the simplest behaviours that raise issues of genuine cognitive interest using an evolutionary, situated, embodied, and dynamical systems approach on visually-guided agents. The framework has been extended in several other papers (e.g., Slocum et al., 2000; Goldenberg, Garcowski, & Beer, 2004; Beer, 2003; Di Paolo & Harvey, 2003), addressing a variety of different cognitive tasks: orientation, reaching, object shape discrimination, perception of body-scaled affordances, self/nonself discrimination, short-term memory, and selective attention. All of the tasks share a common methodology (described in the methods section).

One application of this framework has been to “explore the nature and necessity of the notion of representation in cognitive science” (Beer, 1996). Cognitive science is dominated by the use of a notion of “internal representations in the brain”. But the meaning of this notion has remained elusive. One reason for this is the lack of empirically driven case studies in embodied, situated and dynamical agents. Such an approach could be used to illustrate unambiguously how an “internal representation in the brain” should be understood. Also, what the advantage of using it is when understanding embodied adaptive behaviour.

A motivation for the framework of minimally cognitive behaviours has been to study tasks that are “representation-hungry” (Clark & Toribio, 1994). Intuitively, these are tasks that cognitive scientists would claim require an “internal representation in the brain”. More specifically, however, we take “representationally hungry” tasks to mean behaviours that require an agent to “transcend its immediate environment by allowing past experiences to influence its future actions” (Slocum et al., 2000). This requires experience-dependent behaviour. Of the tasks explored so far, the last two have involved situations that could be said to require a ‘memory’. Despite this work, there has been a line of criticism that argues that while representations may not be necessary for simple tasks they are essential for some sufficiently representationally hungry set of tasks. Clark and Toribio (1994) put it this way:

The empirically driven anti-representationalist invokes superficially compelling case studies of complex but representation-free behaviour. But these case studies, on closer examination, are compromised by a failure to address the right type of problem viz., the range of cases in which ambient environmental information is (prima facie) insufficient to guide behaviour. (Clark & Toribio, 1994, p. 402)

In this chapter we extend the work on minimal cognitive learning behaviours, to include an experience-dependent categorisation task. The task that we study meets exactly Clark and Toribio (1994)’s “right type of problem” (see above specification): imprinting in birds. Experiments by ethologist Konrad Lorenz provided good examples of a form of learning behaviour which requires the subjects to be able to discriminate between different objects (i.e., who is the parent and who is not) using features on a continuum in their environment (i.e., what the parent looks like) and to adapt their behaviour according to this history of interaction (Lorenz, 1956). For example, geese, in order to keep up with their moving parents, must first learn to recognise them. This is a phenomenon that is exhibited by several species when young; they will follow and become

---

The approach is not restricted to agents that use vision. What is important is that the agent have some means of sensing its environment.
attached to the first moving object they encounter, which usually, but not necessarily, is their mother (Horn, 1985; Bateson, 1990). Other examples include song learning in birds (Marler, 1990), where for various reasons (but mainly mate selection), birds learn to sing their species-specific song from what they hear during their first days of life. An example of how parental imprinting can evolve from sexual selection as an adaptive learning mechanism is provided in Todd and Miller (1993).

Here we develop a minimalistic learning task loosely inspired by imprinting in birds designed to address a behaviour in which the environmental information at the time of discrimination is insufficient to guide behaviour. We would like to understand the role of the agent’s embodiment and situatedness in remembering features from the environment. The role of the body and environmental interaction in learning has also been observed in living organisms. In geese, for example, it is the physical act of following that triggers the learning; a goose carried passively behind its parent, for example, will not learn (Lorenz, 1956). From Chapter 6 we know that a disembodied neural network with fixed weights can remember a feature on a bounded continuum. From Chapter 7 we know that the history of agent-environment interactions can sometimes ease the burden on the complexity of the internal dynamics. Finally, from Chapter 8 we know embodied and situated agents can perform a discrete associative learning task. This chapter combines and extends the continuous and embodied dimensions in a minimally cognitive learning experiment.

9.3 Methods

We use an agent and environment set-up similar to Beer (1996), using a real-valued genetic algorithm to evolve dynamical systems controllers with slight variations on the agent’s architecture and environment. We require that the agents be able to identify the first-presented object (their ‘parent’) by discriminating between it and other objects (‘strangers’) when a parent or stranger is subsequently presented; the identifying feature for a parent is changed between trials. The model has been chosen for two reasons: its simplicity, and its potential for sufficiently interesting behaviours that could be called minimally cognitive.

9.3.1 Agent’s body and environment

The agent is situated and embodied, meaning that its sensory input is at all times co-determined by itself as well as by the interaction with its environment. The agent has a circular body with a diameter of 30 units. The agent’s ‘eye’ consists of seven rays distributed uniformly between $\pm \pi/12$. An intersection between a ray and an object causes an input to be injected into the corresponding sensory node, with a magnitude inversely proportional to the distance. For distances over 220, the input is set to zero, and zero distance gives the maximum input. The agent can move horizontally under the influence of ‘left’ and ‘right’ motors as objects fall from above (see Figure 9.1A); the horizontal velocity is proportional to the sum of the opposing forces produced by these two motors. Circle-shaped objects of radii between [10, 20] fall straight down from the ‘sky’ at constant speed (-3); these represent possible parents or strangers.
Figure 9.1: Environment and agent experimental set-up. [A] The agent (black circle) can move horizontally while objects (the grey disc) of varying diameters fall from above. The range of sizes can vary between the inner circle (solid line) and the outer circle (dashed line). The agent has an array of 7 distal sensors (grey lines). [B] The network architecture has a layer of 7 sensory nodes connected to a layer of $N$ inter-nodes, in turn connected to a layer of two motor nodes. The inter-node layer is fully interconnected, including self-connections. Experiments were carried out for $N$ from 3 to 6.

### 9.3.2 Agent’s controller

We use a dynamical system implemented by a continuous-time recurrent node network (CTRNN) with the following state equation (Beer, 1995b):

$$
\tau_i \dot{y}_i = -y_i + \sum_{j=i}^{N} w_{ji} \sigma(g_j(y_j + \theta_j)) + I_i
$$

(9.1)

where $y_i$ is the activation of each node, $\tau$ is its time constant, $w_{ji}$ is the strength of the connection from the $j^{th}$ to the $i^{th}$ node, $\theta$ is a bias term, $g$ is a gain, $\sigma(x) = 1/(1 + e^{-x})$ is the standard logistic activation function, $I$ represents an external input (e.g., from a sensor) and $N$ represents the number of nodes in the network. In simulation, node activations are calculated forward through time by straightforward time-slicing using Euler integration with a time-step of 0.1.

The network’s architecture consists of a layer of seven sensory nodes projecting to a layer of $K$ inter-nodes, which in turn projects to a layer of two motor nodes that control horizontal motion (see Figure 9.1B). The total amount of nodes in the network is $N = 7 + K + 2$. The inter-node layer is fully interconnected, including self-connections. The neurons in the sensory layer share one genetically specified bias and gain and the neurons in the motor layer share one genetically specified bias. The gain parameters for the motor layer and the inter-node layer are set to one. The network architecture is allowed to evolve without any constraints (e.g., symmetry). Time-parameters are genetically specified with a minimum possible value 1.
9.3.3 Task
The task consists of two phases, in each of which a circle-shaped object falls. Before the second circle falls, there is a delay of random duration (between \([10, 20]\) units of time). The first circle presented is the ‘parent’, to be identified and recognised by its size. After the delay (and without reinitialising the state of the agent), a second circle (or ‘test individual’) is presented. The circles always fall from the same horizontal absolute position. It can be the same or different size to the first circle (‘parent’ or ‘stranger’, respectively): the agent should centre in on a parent, but move away from a stranger. For each set of trials, the sizes of parents and strangers are drawn from a range of possible values. The task is loosely inspired on imprinting-like learning in that, for the case of geese, the first thing that it sees is its parent and it must remember it, with its particular features from a continuum of possible features, so that later, when presented again it can recognise it, or in the case of being a different individual from his parent, to avoid it (or get eaten!).

9.3.4 Artificial evolution
All parameters of the agent’s neural controller are evolved using a Microbial genetic algorithm (Harvey, 2001). There are a total of \(N^2 + 11N + 5\) parameters, where \(N\) is the number of inter-neurons. These are encoded in a genotype as a vector of real numbers over the range \([0, 1]\). Offspring of Microbial tournaments are generated as a mutation of the winner of the tournament (without recombination). The mutation is implemented as a random displacement on every gene drawn uniformly from a Gaussian distribution with mean 0 and variance 0.01. Each gene is forced to be in \([0, 1]\): when a mutation takes a gene out of this range it is reflected back. The offspring replace the loser of the tournament. Genes are mapped to network parameters using linear maps from \([0, 1]\) to \([-10, 10]\) for biases and connection weights and to \([1, 10]\) for the gain parameter. Time-constants are exponentially mapped to \([e^0, e^4]\). The size of the population used is 50 and we define a generation as the time it takes to generate the same number of new individuals. A minimal 1D wrap-around geography with neighbourhoods of size 10 was used: such that only individuals 10, or less than 10, positions away from each other could compete in tournaments. Finally, because the fitness is noisy (described below), agents are re-evaluated every time they participate in a tournament.

9.3.5 Fitness evaluation
Agents are evaluated by performing 200 fitness trials. Each trial consists of the presentation of the parent individual followed by a delay of random duration and then the presentation of the test individual. Half of the trials involve scenarios where the test circle is the same size as that of the parent. The other half involves scenarios where the test circle is different. Half of those (50 trials) involve tests circles that are smaller than the parent, and the other half involves bigger test circles than the parent. This ensures a balanced presentation of the three situations that can arise during the agent’s lifetime. For each of the trials, however, the actual size of the parent and test circle are chosen at random from the range \([10, 20]\). This is to allow successful agents to generalize over any of the possible parent-size/test-size configurations.

The fitness for a trial is determined by the absolute horizontal distance, \(|d|\), between the agent and the falling object by the time the test individual reaches the vertical position of the agent,
Chapter 9. Learning from a continuum in embodied and situated agents

9.4 Results

9.4.1 Evolutionary performance

Evolutionary searches using agents with 3, 4, 5, and 6 inter-neurons were performed. In Figure 9.2 we show whisker boxes for the mean fitness of the best individual over the last 50 generations of evolution for each of the 20 evolutionary runs performed per size of circuit. Notably, the addition of internal neurons doesn’t seem to improve the performance. This is unlike the results presented for all other tasks in this thesis, where increasing the number of neurons in the network resulted in increased fitness. It will become clearer as we understand the most successful agents why this is the case for this particular task.

9.4.2 Learning performance

Given that evolutionary runs with all of the different sizes of circuits achieved a relatively high fitness, we will consider the behaviour of the best-evolved agent for each size. The best way to capture the agent’s ability to learn the size of its parent and generalize over all possible combinations of parent and test size is to visualise the distance between the agent and the test individual at the end of its fall. Figure 9.3 shows this learning generalisation map for the best circuit for agents with: 3 (A), 4 (B), 5 (C), and 6 (D) internodes. We have already shown this type of map in Chapter 6 (e.g., Figures 6.7 and 6.10). On the horizontal axis is the size of the test circle, and on the vertical axis is the size of the parent circle. The shade of grey represents the distance between the agent and the test circle at the end of its fall: lighter shades represent larger distances (i.e., test circle avoided) and darker shades represent closer distances (i.e., test circle approached). Each point in the map depicts an average over 50 trials, with delays chosen at random in the range [10, 20].
Figure 9.3: Performance of best-evolved agent with 3- [A], 4- [B], 5- [C] and 6-interneurons [D]. Shade of grey represents catching (black) or avoiding (white) the test individual. The vertical axis represents the radius of the parent; the horizontal axis represents the radii of the test individuals. The diagonal corresponds to those test individuals that are parents (i.e., the same size); strangers will lie off this diagonal.

In the maps shown in Figure 9.3, perfect learning behaviour would appear as an all white map except for a thin black line along the diagonal. This would correspond to an agent that catches only test sizes of the exact same size to the original parent and avoids all others, including very similar ones. Successful agents, however, show some misclassification errors. As can be seen in Figure 9.3, most of the misclassifications correspond to approaching test circles whose sizes are very similar to that of the actual parent, but not the same. This is expected; very small differences are not always correctly identified. In other words, when the test circle is sufficiently similar to the parent circle, there is an increased chance that it will be mistaken for it.

The best 3-interneuron agent’s slightly lower fitness at the end of its evolutionary run is evident from its generalisation map. Despite being the best of its size, this network performs the worst out of the other three networks shown (compare Figure 9.3A with Figures 9.3B through 9.3D). The poorer generalisation performance is manifest in the amount of darker shades outside
of the diagonal. The result is an increased width of the approaching-behaviour diagonal. This
 corresponds to an agent that cannot, for example, remember whether its parent was of size 15
 or 16, so it approaches all in that range. The other, larger, networks have a much more refined
 approaching diagonal. Notably, even though the approaching diagonal of the 6-interneuron agent
 is very well defined (see Figure 9.3D), some unusual catching mistakes are made for test circles
 that are much larger than the to-be-remembered parent sizes (i.e., black shades in the bottom right
 of Figure 9.3D).

9.4.3 Behaviour

Let us turn now to how the agents interact with the falling objects, starting with the first circle.
Remember that the fitness function only evaluates where the agent is at the moment the second
 circle ends its fall. Although, in principle, the agent could ignore the first circle entirely, if it
 is to categorize successfully parents from strangers during the second phase, then it must ‘pay
 attention’ to the first falling circle. In particular, to its size as this is the only distinctive feature of
 the falling objects.

One approach to learning the size of the circle could involve an agent that remains stationary
 below it as the object falls, sensing it passively yet ‘recording’ (i.e., putting inside the agent’s
 brain) the to-be-remembered feature, in this case its size. As the task is configured, this is a
 possibility that is available to the evolving agents. This is because: (a) the agents are not re-
 quired to move (or do anything) for the fall of the first circle, and (b) the internal dynamics of
 the agent have multiple-timescales and recurrent structures that allow them to evolve sufficiently
 complex internal state functionalities (as we saw from the results of Chapter 7). Furthermore, such
 a ‘passive recording’ solution is what traditional cognitive science and artificial intelligence would
 predict/assume arises. Is this what the evolved agents do? As can be seen from Figure 9.4, suc-
 cessful agents do not remain stationary; they tend to move while sensing the falling circle instead.
 More importantly, the movement is not irrelevant to the learning process. On the contrary, it is a
 crucial tool that it will exploit to aid it in remembering, as we will see ahead.

In Figure 9.4, we show the motion over time in object-centred coordinates for each of the
 best-evolved agents for the four different sized circuits. The two solid vertical lines denote the
 first phase (while the parent circle is falling). During this phase, the motion of the agent is shown
 while circles of 5 different sizes (across the full range) fall. The darker trajectories correspond to
 smaller parent circles and lighter grey trajectories correspond to larger ones. As can be appreciated
 for all four of the agents under analysis, towards the end of the first circle’s fall the agents move
 in a way such that they separate systematically away from the falling circle depending on its size
 (e.g., farther away from bigger circles than from smaller circles or vice versa). The result is that
 the second circle starts to fall in a uniquely different position relative to the agent. Thus, where
 the second circle appears in the field of view of the agent has the potential to ‘remind’ the agent of
 the size of the parent circle and therefore whether it should catch or avoid the current one. While
 this is the case for the 3 and 5 interneuron agent (Figures 9.4A and 9.4C), it is most pronounced
 in the 4 and 6 interneuron agents (see Figures 9.4B and 9.4D, particularly the areas pointed by the

---

4Cognitivists view perception as a passive process. Walter Freeman describes their view appropriately: “They hold
 that perception begins when a stimulus gives information that is transduced by sensory receptors into a burst of neural
 activity that cascades through the brain stem and thalamus into a sensory cortex.” (Freeman, 2000)
arrow labelled i). It is also important to note that the only section where the different sized parent circle trajectories seem to collide are for the small circles (darker trajectories) in the 5-interneuron agent (see Figure 9.4C, particularly the area pointed by the arrow labelled ii). We will explore these features further in the next sections.

Figure 9.4: Behaviour of the best-evolved agent for different sized circuits: 3 (A), 4 (B), 5 (C) and 6 (D) internodes. Each trajectory depicts the agents motion over time in object-centered coordinates for different sized objects. The two solid vertical lines denote the first phase. During this phase, the motion of the agent is shown for 5 different sized circles. The darker trajectories correspond to smaller parent circles and lighter grey trajectories correspond to larger ones. There is a delay between the fall of the first and second circle (trajectories between the second vertical solid line and the first vertical dashed line). The two dashed vertical lines denote the second phase. The motion of the agent is shown for 25 different conditions (for 5 test circle sizes for each of the original 5 parent sizes). Green trajectories correspond to test circles of the same size as the parent circles. Blue and red trajectories correspond to test circles smaller and larger than the parent, respectively. The labeled arrows are referred to from the main text.

After the first circle falls, there is a delay of random duration before the second circle appears in the field of view of the agent. In order to ease the analysis, for the figures shown in 9.4 the delays are fixed and of the same duration. The two dashed vertical lines denote the second phase (while the test circle is falling). During this phase, the motion of the agent is shown again for 5 different sizes of test circuit, for each of the 5 parent sizes (25 trajectories in total). The five trajectories that correspond to test circles of the same size as the parent circles are coloured green; the 10 trajectories that correspond to test circles larger than the parent are coloured red; and the 10
trajectories for which test circles are smaller are coloured blue.

As can be seen from Figure 9.4: (a) the green trajectories culminate close to 0, meaning that test circles of the same size as the parent are approached; and (b) blue and red trajectories finish further away from 0, meaning smaller and larger test circles are avoided. This is what is expected from successful agents. Although agents are only selected to identify the circle that is the same as the previous one, a classification of ‘smaller-than-the-parent’ and of ‘larger-than-the-parent’ circles emerges from the interaction. We can tell this is the case from Figure 9.4 because the red and blue trajectories can be grouped into separate bundles; although, of course, in some agents this separation is less clear-cut than in others. The larger versus smaller classification of the strangers is most notable in the 5-interneuron agent (see Figure 9.4C, arrow iii).

9.4.4 The role of the agent’s situatedness versus the agent’s internal dynamics in remembering

We have established that all agents shown have the ability to learn the size of the first falling circle and to remember it long enough to discriminate between similar and different parents, in the absence of environmental stimuli that allow for the categorical relationship to be made. In some cases, the agents can also discriminate between smaller or larger (than the original) circles - an additional relational categorisation that was not selected for. So, how can these agents remember? Is the memory ‘stored’ only in the agent’s ‘brain’? Or can we talk about it ‘remembering’ through the agent-environment interaction as well? For a traditional cognitivist the answer to this question would be obvious. If considered at all, they would most likely answer the question without considering situations where either options are possible. But let’s ask that question for the evolved agents in this task. It will be useful to rephrase it first: does the agent have (i.e., in the variables of the agent’s internal state) all the information that is required to remember the size of the circle? Is all such information ‘stored’ in the agent’s history of physical interactions with the falling circle? Or is it a mixture of both? We will first measure the correlation between the feature to be remembered (and no longer present in the environment) and the state of the agent at the start of the fall of the second circle. As we have already suggested, there are 2 different sources of state that the agent can exploit: the first is the state from its internal dynamics; the second is from the history of interactions with the falling object. Note that while high correlation does not imply that such component will be employed in the decision making process (i.e., not a sufficient requirement), it will be a necessary requirement for such component to be ‘used’ at all.

In Table 9.1 we show the Pearson product-moment correlation coefficient between the feature to be remembered (i.e., size of parent circle) and (a) the sigmoided output of the internodes ($o_i$), and (b) the distance between the agent and the falling circle ($|d|$). This is calculated at the end of the delay (and before the second circle appears in the agent’s field of view). We also show the highest absolute correlation coefficient among the internodes ($|o|$). As can be observed, both the physical interaction of the agent with the falling object ($|d|$) and its most correlated internal variable ($|o|$) are highly correlated with the to-be-remembered feature. For all of the agents except

---

5There are several reasons why this is often the case: first, they may not have the accessibility to all of the variables (e.g., cognitive scientists observing learning behaviour in humans); second, the experiment may not have been devised to include the possibility of using the agent-environment interaction to remember; and thirdly, from the assumptions about what intelligence is and where learning happens (e.g., building adaptive systems in artificial intelligence).
9.4. Results

Table 9.1: Correlation between the to-be-remembered feature (i.e., size of the parent circle) and the state of the agent before the start of the fall of the second circle. The correlation is given for the best-evolved agent for networks of different size (according to the number, $K$, of nodes in the inter-layer).

| $K$ | $o_1$  | $o_2$  | $o_3$  | $o_4$  | $o_5$  | $|p|$   | $|d|$   |
|-----|--------|--------|--------|--------|--------|--------|--------|
| 3   | -0.9910| 0.9094 | -0.9454|        |        | 0.9910 | 0.9885 |
| 4   | 0.9406 | 0.9630 | 0.2033 | 0.9848 |        | 0.9848 | 0.9997 |
| 5   | 0.0    | 0.8984 | -0.9584| -0.8062| 0.8844 | 0.9584 | 0.4977 |
| 6   | -0.9947| 0.2060 | 0.9589 | 0.9951 | 0.9269 | 0.9940 | 0.9951 | 0.9945 |

The 5-interneuron one, the correlation between both components is very high. Correlation alone would lead us to assume that agent-environment interaction and internal dynamics are equally important for the remembering process. We will revise this premature conclusion ahead. For the 5-interneuron agent, however, the correlation of the physical interaction component is much lower than that of the most correlated neuron, suggesting that the internal dynamics of this agent are likely to play a more important role in the remembering behaviour. Note that the dependency between two variables over an extended period of time can be relatively complex, and thus need not be captured by the linear correlation between the two variables frozen at an instantaneous point in time. In practical terms, to claim that two variables are not correlated, we would have to study their dependencies in more depth (e.g., mutual information, total correlation, polychoric correlation, etc). This is not necessary in this case for two reasons: (a) the dynamics of the variables under analysis are sufficiently simple (i.e., not oscillating nor moving rapidly up and down) and (b) for all of the variables we discover that there is a linear dependency.

Yet, remembering is not merely about correlations between the to-be-remembered feature and the states of the agent while the feature is absent from its field of view. The agent must perform some behaviour (in this case approach or avoid the circle) to realize the remembering process fully. This would require the ‘use’ of one, some, or all of the components whose states are correlated with the feature to be remembered. As we suggested previously, a highly correlated component may or may not affect the agent’s ability to discriminate between which circles to approach or avoid.

With this in mind, we should ask: what is the role of the agent’s situatedness versus the agent’s internal dynamics in remembering performance? We can address this question with the following experiment: we replace the agent’s state between the fall of the first and second circles with what it would be if it had experienced a substitute first circle of a certain size different from the original. First, we run a set of possible sizes of parent circle (at intervals of 0.5 in the full range [10, 20]), recording at the end of its fall: (a) the relative position between the object and the agent, and (b) the full internal state of the agent (i.e., activation of the neurons). This information is what we will use as the substitute parent. For each of the saved states from the substitute parent $k$, we test the agents with parent circle $i$ and test circle $j$, also for a set of all possible combinations ($i$ and $j$ sampled at intervals of 0.5 between [10, 20]). During this test we replace either: (a) what the
position would have been if it its parent had been $k$ instead of $i$, (b) what the internal state of the agent would have if it its parent had been $k$ instead of $i$, or (c) both.

In Figure 9.5 we show the performance of each of the agents at approaching test circles that are the same as the substitute parent $k$ and avoiding those that are different. Approaching the substitute parent contributes to 50% of the performance and avoiding the test circles different from the substitute parent contribute the other 50%. We show this performance when only the physical state is affected (stars), when only the full internal state of the agent is affected (boxes), and when both are affected (diamonds). The case when both are affected is effectively the regular performance of the agent in approaching/avoiding the appropriate sized circles spread across the continuum of the feature to be remembered. The figure shows the effect of the state of the physical placement versus that of the internal state in causing an agent to change the ‘memory’ of the parent circle.

![Figure 9.5](image)

Figure 9.5: Role of situatedness versus internal dynamics in remembering performance for different sized circuits: 3 (A), 4 (B), 5 (C) and 6 (D) internodes. Each point corresponds to the performance (vertical axis) of each of the agents at approaching test circles that are the same as the substitute parent (horizontal axis) and avoiding those that are different when: only the physical state is affected (stars), only the full internal state of the agent is affected (boxes), and both are affected (diamonds). The labelled arrows are referred to from the main text.

The first thing to note from Figure 9.5 is that the contributions are not linearly additive: the performance of replacing the physical state and the internal state individually does not necessarily add up to the performance when both states are replaced simultaneously. In at least one case (i.e.,
circles of size 18.5 in the 3-interneuron agent, see Figure 9.5A, arrow labelled i), the performance when replacing the internal state only (box) causes the agent to perform better than replacing both (diamond). More generally, however, replacing only one state and not the other causes deterioration in performance. Furthermore, considering that 0.5 performance is the baseline (i.e., random behaviour), the sum of each of the components (the agent’s situatedness and its internal state) on their own result in a lower fitness than the performance when both of them are included. This suggests a synergetic effect between the agent’s situatedness and its internal dynamics, where the combined effects are greater than the sum of the separate parts. An example of this agent-environment interaction and internal dynamics synergy is most evident in the 5-interneuron agent (see Figure 9.5C, arrow ii). Catching the substitute parent and avoiding different ones when only the agent’s position is substituted provides a 0.02 increase over random behaviour, and 0.05 when only the internal state is substituted. Individually, both are very near purely random behaviour. Yet, when both are substituted a 0.47 increase is observed. This corresponds to near optimal behaviour.

Secondly, it is important to note that the effect of the body versus that of the internal dynamics of the agent is not only different for every individual, but it is also different in the same agent for different ranges of the feature to be remembered. An analogy of this phenomena in the context of humans would be the use of an incomplete shopping list: where some of the items (e.g., the unusual ones) are written down on the list, while others are left out (e.g., the more obvious ones), relying on our internal dynamics to remember them. We will first consider how the body and internal dynamics affect the memory across the different agents. We will then consider differences within the same agent.

Out of the four agents analysed, the 4-interneuron agent relies the most on its physical state (Figure 9.5B). Just changing the position, regardless of what the state of the neurons are (yet still in the context of having seen a previous but different parent circle), will change the ‘memory’ of the system. This can be observed from the similarity between the performance when only the agent’s position is substituted (stars) to when both are substituted (diamonds), and the near random performance when only the internal state is substituted (boxes). Which sizes of test circle this agent avoids or approaches depends almost entirely on where it is falling from. Furthermore, given: (a) the high correlation between the agent-object distance and the size of the parent circle (0.9997 in Table 9.1) and (b) the agent’s movement (Figure 9.4B), we know that where the second circle appears first in the field of view of the agent is determined by the size of the parent circle. In other words, this agent uses its interaction with the falling object (i.e., horizontal offset) as a source of external memory. Also, remember from Table 9.1 that this agent showed a high correlation between some of the neuron activations and the to-be-remembered (but now absent) feature (e.g., 0.98). Yet we have shown that the neural states are not actually employed to make the categorisation; instead the relative position is the key factor. This provides a clear example of the insufficiency of only considering correlations to understand the remembering process.

Although the agent with 6-interneurons also relies mostly on its physical state, there are some features of the internal state which are crucial in remembering as well (Figure 9.5D). However, as can be seen from the effect of just replacing the internal state (i.e., performance falls to near random behaviour), the position is absolutely crucial. Thus, while this agent also uses its interaction
with the falling object as an external memory, it needs some help from the internal state to fully disambiguate the parents from non-parents.

Unlike the case for how the physical state is used in the agent with 4-interneurons, none of the evolved agents use only the internal state to remember. In fact, the only agent for which the internal state plays a more important role than the physical state is, unexpectedly, the agent with fewest neurons (Figure 9.5A). Replacing the internal state for that of a substitute parent results in a switch of the agent’s behaviour to what if would be if it had previously experienced the substitute parent better than if the physical state had been replaced. This suggests the internal dynamics of this agent play a more important role in the memorisation process than its situatedness. This agent, however, performs poorer than all of the other circuits considered. Hence, from the understanding of the other agents’ behaviour, we suspect this agent could have achieved a better performance if it had evolved to exploit its situatedness more fully.\(^6\)

Finally, the 5-interneuron agent provides an example where which component (physical interaction versus internal state) plays the most important role changes as a function of the feature to be remembered (Figure 9.5C). For circles smaller than 13, the performance of the agent is better when the internal dynamics are substituted than when the agent’s relative position is substituted instead. For bigger circles, the inverse is true, with the agent’s situatedness playing the more important role. This agrees with the agent’s behaviour as observed in Figure 9.4C), where the darker trajectories (smaller circles) bundle together - making the agent’s situatedness less useful in the discrimination between parents and non-parents. This result shows that embodied and situated agents will use a synergistic mixture of their internal dynamics and their history of interaction with their environment to remember, and that how each of the components affect the remembering process can also change within the dimension of the feature to be remembered.

### 9.5 Discussion

In summary, this chapter has: (a) developed a situated and embodied imprinting-like learning task; (b) demonstrated the ability to evolve dynamical system controllers (without additional plastic mechanisms) that can modify their behaviour according to continuous features from the environment; (c) emphasised the importance of the agent’s interaction with the environment in the generation of learning behaviour, as a source of plasticity or ‘memory’ on a continuum.

The results shown emphasise the role of the agent’s situatedness and history of interaction with the falling object in the generation of the learning behaviour. This points towards plasticity being a feature not only of the controller but of the interaction itself. It is important to remember that the evolved agents were not at any point selected to maximize the use of their body and interaction with the environment, as opposed to internal state. While an agent may make use of its situatedness as an ‘external memory’, it is not constrained to do so. It can similarly use its potentially rich internal dynamics as a source of memory. This suggests that (a) when available the plasticity arising from the interaction will be exploited and (b) that this plasticity may be a simpler way to solve ‘memory-related’ problems than using internal state.

A possible objection of the work presented in this chapter is that the task does not actually

---

\(^6\)Although it is also possible that in order to exploit its situatedness, the circuit required already a sufficiently rich internal dynamics. This is an interesting possibility that has not been considered further in this thesis.
meet Clark and Toribio (1994)’s sufficiently representation-hungry task specification where environmental information must be insufficient to guide behaviour. Their argument would be that, as the circles fall always from the same absolute position, the task allows for a solution whereby the agent exploits its position after the parental presentation. This is indeed the case. Yet, we argue that the specification is nevertheless met. At the point when the classification (parent or stranger) must be made there is no information in the environment alone to guide behaviour. The ‘information’ is in the agent-environment interaction. In fact, it is the agents themselves that create (through evolution) this regularity from their interaction with the falling objects. Hence, it will be different for every agent. Swapping two agents will not produce satisfactory solutions (even though it is the same task) precisely because the ‘information guiding behaviour’ is dependent on the ‘personal’ history of each agent’s interactions with the environment.

Finally, imprinting phenomena have been used to study developmental processes. We believe the model presented in this work has the potential to investigate issues relating to developmental robotics. Much of the work in this area consists of adding ad hoc morphogenetic-like mechanism to the existing control systems and, in some cases, testing it on a range of different tasks to see whether performance or ‘evolvability’ improves (e.g., Mjolsness, Sharp, & Alpert, 1987; Kitano, 1990; Gruau, 1994; Dellaert, 1995; Bongard & Pfeifer, 2003; Balaam, 2005; Wood, 2006). Our work suggests a radically different approach: choosing a task that seems to require a developmental process and then evolving dynamical systems that perform the task. The interest in this case will be in the analysis of the evolved networks. Either agents with developmental processes evolve, in which case it will be useful to understand how this happens, or agents without developmental processes solve the task, in which case we must either (a) rethink whether the task actually ‘requires’ a developmental process, or (b) redefine what a developmental process is from the analysis of successful systems.
Chapter 10

Learning with your body: Single neural circuit with fixed weights performing two qualitatively different tasks

In the previous chapters we have studied the role of the agent’s internal dynamics and of its situatedness in learning behaviour. Learning behaviour, however, is a result of the interaction between the agent’s brain, its body and its environment. In this penultimate chapter, we explore the role of the body in learning. Living organisms perform a broad range of different behaviours during their lifetime. It is important that these be coordinated such as to perform the appropriate one at the right time. This chapter extends previous work on evolving dynamical recurrent neural networks by synthesizing a single circuit that performs two qualitatively different behaviours: orientation to sensory stimuli and legged locomotion. We demonstrate that small fully interconnected networks can solve these two tasks without providing a priori structural modules, explicit neural learning mechanisms, or an external signal for when to switch between them. Dynamical systems analysis of the best-adapted circuit explains the agent’s ability to switch between the two behaviours from the interactions of the circuit’s neural dynamics, its body and environment.

10.1 Introduction

Multifunctionality refers to the ability of neural circuits to generate multiple behavioural patterns and it is widespread among vertebrate and invertebrate species (Briggman & Kristan Jr., 2008). This is reflected in their ability to coordinate multiple behaviours. The nematode worm Caenorhabditis elegans, for example, with ‘only’ 302 neurons shows a remarkable ability to perform a broad range of different behaviours (Hart, 2006; Rankin, 2004). Although our understanding of the neural basis for most of them is still at an early stage (de Bono & Maricq, 2005), it is known that overlap exists between some of the neural circuits responsible for these behaviours (Hobert, 2003). Also, it is well known that the morphology of living organisms is in constant change, both throughout evolution and during the lifetime of the organism. This work investigates how a single neural network that is not structurally divided into separate circuits can produce different behaviours in different bodies.

When modelling adaptive behaviours, assumptions have to be made with regard to the structure of the organism studied in order to simplify the modelling process or the analysis of the model’s
behaviour. One such assumption that has been made in the past is that functional modularity, the existence of several qualitatively different behaviours in the same organism or agent, should be mirrored by structural modularity in its neural controller. Complex systems are thus often divided into small parts that are synthesized in isolation. Such a divide-and-conquer approach can be very useful for engineering robots that need to perform multiple complex tasks, not least because it simplifies the understanding of how the robot works. But it is less useful in the context of developing the tools and language to understand biological organisms, as these may not necessarily have evolved to be easily decomposable.

In this chapter we investigate whether a single neurocontroller can exhibit qualitatively different behaviours without imposing constraints on its structure. We use artificial evolution to synthesize a recurrent neural network that when coupled to two different simulated bodies, namely a one legged insect and a two-wheeled robot with a chemical sensor, has to perform legged locomotion in the former and chemotaxis in the latter case. A successful agent has to detect which body it inhabits and generate the appropriate behaviour. It must do this in the absence of an external signal and without any online changes in the parameters of the controller. We aim to find the smallest network that can solve the task. Although the structure of the network is under evolution, we do not investigate whether the evolved networks exhibit a degree of structural or "functional"1 modularity.

Second, using the mathematical tools of dynamical systems theory, we explain how the circuit in interaction with its body and environment can generate distinct behaviours. We characterize the autonomous dynamics of the best-evolved circuit and how its dynamics vary with inputs. We then study how the observed behavioural patterns are generated through the closed-loop interaction of the neural dynamics with the body and environment, for the two different tasks. Finally, we show how the evolved agent makes use of context-dependent feedback to shape the different transients using the same dynamical landscape. This leads us to suggest a dynamical systems perspective on adaptive behaviour that goes beyond attractors.

10.2 Related work

Synthesizing neural controllers to generate multiple qualitatively different behaviours is a challenge that has been posed by many. However, the focus has been on the role of modularity. Toigelius (2004) showed how subsumption architecture models could be merged with an evolutionary robotics approach for a simulated robot on a learning task. Nolfi (1997) investigated modularity for evolution of a garbage collecting robot that had to cope with subtasks such as recognizing, picking up and disposing of desired objects. Although the networks had a hard-wired modular architecture, evolution was free to choose how these modules were used. Calabretta, Nolfi, Parisi, and Wagner (1999) addressed the same task, but used a system in which neural modules could evolve from a population of non-modular networks through gene duplication. Although both reported improved performance relative to monolithic networks, Ziemke, Carlsson, and Bodén (1999) showed that in a more difficult version of the task, a monolithic recurrent network outperformed all modular architectures.

Watson and Pollack (2005) argue that structural descriptions of the network are not sufficient to determine dependence or independence in the dynamics of different subsets of the network.
Many animals can rapidly change between different modes of locomotion. In Ijspeert, Crespi, Ryczko, and Cabelguen (2007), the problem of designing the neural controller for switching between swimming and walking in a salamander-like robot is presented. In Buckley, Fine, Bullock, and Di Paolo (2008), an agent is evolved to do phototaxis with the sensor in two different positions (front and back of the body) while constraining the dynamical system controller to use a single basin of attraction. In both papers, however, the two behaviours share a large range of qualities. Our work is different from theirs in that the two tasks (chemotaxis and walking) were chosen to be as different as possible, while sharing sensor and effectors.

Yamauchi and Beer (1994a) evolved a simulated robot that had to learn which of two environments it was placed in and take an appropriate action such as to approach a desired position. They only succeeded after dividing the network into separate modules with explicitly assigned roles that were evolved separately. They then evolved a classifier network to determine which of the modules is to control the agent. Tuci et al. (2002a) later successfully evolved a controller for a very similar task without dedicated modules. Finally, Beer and Gallagher (1992) evolved agents for chemotaxis and walking, but not with the same dynamical system controller.

10.3 Methods

10.3.1 Walking task
The walking task employed follows very closely the simple one-legged body described and analysed in Beer and Gallagher (1992), Beer, Chiel, and Gallagher (1999). Three variants of this model have been studied in Beer (1995a), differing in whether sensory feedback is available constantly, only occasionally, or absent. Of the corresponding controllers, namely reflexive pattern generators (RPGs), central pattern generators (CPGs), and mixed pattern generators (MPGs), we focus on the first type only. The leg is controlled by three effectors: one specifies whether the foot has contact with the ground while the other two control clockwise and counter-clockwise torques for the leg’s hinge-joint with the body (Figure 10.1A). The opposing torques model antagonistic muscles, commonly found in animal limbs. Sensory feedback is provided continuously by the leg’s joint angle. At the beginning of each walking trial, the state of the leg (i.e., its angle with respect to the body) is initialised at random. The agent is then given 220 units of time to walk. The total distance covered during the trial measures performance.

10.3.2 Chemotaxis task
For the chemotaxis task we also follow a methodology similar to that employed in Beer and Gallagher (1992). A food patch, placed at arbitrary locations and orientations with respect to the agent, emits a chemical signal \( s \), whose intensity falls off as a function of the distance from the center of the patch \( d \): \( s = \lambda^d \), where \( \lambda \) is a constant, \( -0.0138 \). The agent moves freely in an environment without walls and must find and remain in the vicinity of the food patch.

The agent has a circular body and possesses a sensor that can detect the intensity of the chemical signal at its location (Figure 10.1B). Additionally, it is equipped with two effectors located on opposite sides of its body. These effectors\(^2\) can apply forces that move the body forward and

\(^2\)Although the mechanics of the body correspond closer to a khepera-like robot, similar physics have been used in idealised models of the nematode worm’s movement in Ferree and Lockery (1999). Instead of ‘wheels’, the effectors...
10.3. Methods

Figure 10.1: Task set-up. The same neural network circuit is used to control two different bodies: a one-legged insect-like walking agent (A) and a khepera-like chemotaxis agent (B). The circuits are fully inter-connected. The effector neurons control the antagonistic muscles and the foot for walking and the two effectors on the opposite sides of the body for chemotaxis. All of the neurons receive sensory perturbations: from the leg angle during walking, and from the proximity to the food during chemotaxis.

rotate it. In the simplified physics of this environment, the velocity of movement is proportional to the force applied.

During a chemotaxis trial, a food patch is placed in a random direction from the agent, anywhere between 10 and 15 units of space apart. This is repeated after 100 units of time. Three food patches are shown in total. Performance is given by: \( f_c = (d_i - \bar{d}) / d_i \), where \( d_i \) and \( \bar{d} \) are the initial and average Euclidean distance between the agent and the food patch, respectively.

10.3.3 Neural model

We use continuous-time recurrent neural networks as a model of the agent’s internal dynamics. Each component in the network is governed by the following state equation:

\[
\tau_i \dot{y}_i = -y_i + \sum_{j=1}^{N} w_{ji} \sigma (y_j + \theta_j) + sw_i S(t)
\]  

(10.1)

where \( y \) is the activation of each node; \( \tau \) is its time constant; \( w_{ji} \) is the strength of the connection from the \( j^{th} \) to the \( i^{th} \) node; \( \theta \) is a bias term; \( \sigma(x) = 1 / (1 + e^{-x}) \) is the standard logistic activation function; and \( N \) represents the number of nodes in the network. All nodes have access to the sensory perturbations via a set of connection weights: \( sw_j \). The sensory input is normalized to run between 0 and 1 for both tasks. This prevents a solution that switches behaviour reactively as a response to different sensory input ranges. The network is fully connected (including self-connections) and no symmetry is imposed on its weight matrix. In simulation, node activations are calculated forward through time by straightforward time-slicing using Euler integration with a time-step of 0.1.

located on opposite sides model the ventral and dorsal neck muscles of the worm.
10.3.4 Evolutionary algorithm

The parameters of each circuit (i.e., biases, time-constants, inter-neuron and sensor-neuron weights for each node) are evolved using a version of the Microbial genetic algorithm (Harvey, 2001). There are $N^2 + 3N$ parameters in total. These are encoded in a genotype as a vector of real numbers over the range $[0, 1]$. Offspring of Microbial tournaments are generated as a mutation of the winner of the tournament (i.e., no recombination). The mutation is implemented as a random displacement on every gene drawn uniformly from a Gaussian distribution with mean 0 and variance 0.01. Each gene is forced to be in $[0, 1]$: when a mutation takes a gene out of this range it is reflected back. The offspring replace the loser of the tournament. Genes are linearly mapped to network parameters in the range $[-10, 10]$ for biases, inter-node and sensory weights and to the range $[1, 20]$ for time constants. The size of the population used is 50 and we define a generation as the time it takes to generate the same number of new individuals. A minimal 1D wrap-around geography with neighbourhoods of size 10 was used: such that only individuals 10, or less than 10, positions away from each other could compete in tournaments. Finally, because the fitness is noisy, agents are re-evaluated every time they participate in a tournament.

A successful circuit must maximize: (a) the distance walked when embodied in the insect-like body and (b) the time spent around the chemical-emitting food patch when in the khepera-like body. A fitness evaluation consists of 2 trials of the walking task and 15 trials of the chemotaxis task. At the start of each trial the state of the neural controller and the body is randomised. The performance on each task is averaged over all trials and normalized in the range $[0, 1]$. The fitness of an individual is calculated by multiplying the performance on both tasks.

10.4 Results

10.4.1 Evolutionary performance

Evolutionary searches with 3-, 4-, and 5-node circuits were performed. We examined the ability of populations to evolve for both tasks, and conducted control experiments in which either task was evolved on its own. For each condition, 20 evolutionary experiments with different initial random seeds were carried out.

First we compare networks of different size. Figure 10.2 shows the performance of the set of best circuits grouped according to size on the walking (10.2A) and chemotaxis task (10.2B). For each, two whisker plots show the performance of circuits on the task at hand. The grey whisker boxes correspond to populations evolved for only that task. White whisker boxes correspond to populations evolved on both tasks. The first thing to note was that success increased with the number of nodes in the circuit. This was true for the individual evolutionary runs as well as the combined ones. The average performance of the best individual per population on the combined tasks was 31.4%, 58.4%, and 70.2% for 3-, 4-, and 5-node circuits respectively. Second, when tested on each task individually, same-sized circuits performed (on average) better if they had been evolved exclusively for that task than if they had been evolved on both tasks. The best-evolved circuit on the combined task achieved 75.6%, 82.0%, and 93.6% of the score of that of the best-evolved circuit on the chemotaxis task, and achieved 92%, 97.4%, and 101.9% of that of the best-scoring circuit on the walking task (for 3-, 4-, and 5-node circuits, respectively). This suggests that the networks evolved on both tasks dedicate some part of their “dynamic capacity”
to the other task that would otherwise be free to improve the performance on the first. And that this difference is lessened as the size of the circuit increases. This is also as expected.

Nevertheless, sufficiently successful circuits that performed both tasks did evolve. Finally, one question of interest is whether, within the populations evolved on the combined task, some individuals specialized at performing one of the tasks at the expense of the other one. In Figure 10.2C we show chemotaxis versus walking fitness for each of the best individuals from all evolutionary runs on the two tasks. We observed no such trade-off between walking and chemotaxis performance in the best-evolved agents. That is, there are no clusters of best-evolved circuits in regions of high performance on one task and poor performance on the other. This is most likely the result of calculating fitness as the multiplication of performance on each task. In other words, individuals tended to be proportionally good (or bad) at both tasks.

Figure 10.2: Evolutionary performance. Whisker plots (25% to 75% quantiles and outliers as points) comparing the fitness achieved by 3-, 4- and 5-node networks after having evolved for only one task (grey) and both tasks (white) in the case of walking [A] and chemotaxis [B]. [C] Chemotaxis versus walking fitness on the best 3- (diamonds), 4- (stars), and 5-node (squares) circuits evolved for both tasks.
10.4.2 Performance and behaviours

The best-evolved agent achieved a walking fitness of 93.59% and a chemotaxis fitness of 91.18%. This individual corresponds to the red square from Figure 10.2C. In Figure 10.3 we show an example trial with this circuit performing both tasks. It is relevant to note that all neurons are active during both tasks.


Walking

The optimal walking pattern for this one-legged model has been studied in Beer et al. (1999). As can be seen from Figure 10.3B, the evolved pattern is almost perfectly aligned with the optimal pattern, at least geometrically. The different sections in this pattern correspond to particular stages of the walking cycle (labelled in grey): (1) foot up and swing, (2) foot down, (3) stance power, and (4) stance coast. This agrees with results in Beer et al. (1999). Yet, we know the performance is only 93.59% of the optimal asymptotic velocity that the walking agent can achieve (0.627). The difference is in the timing. The best-evolved circuit has 2 units of time delay between the moment it sets its foot down (phase 2) and the moment it starts to move the leg backwards (phase 3).
This unnecessary foot ‘rest’ causes the degradation in performance, nevertheless maintaining the optimal geometry of the walking pattern. Thus, the best-evolved circuit completes the full cycle at a slightly slower rate than the optimal. If we turn the sensory feedback off, the circuit cannot walk. This is expected from the RPG.

Chemotaxis

Given that the agent has only one sensor\(^3\), it has to move in order to detect the chemical gradient. The only successful strategy is to use subsequent sensory signals to estimate the chemical gradient in time rather than space.

We can identify four relatively different phases in the best-evolved agent during a chemotactic run: (1) circling search, (2) decreased turning in direction towards the gradient, (3) straight run, (4) circling around food patch. However, this is only a simplified observer-perspective heuristic and the phases are not always clear-cut. The full story is provided only by a geometrical account.

For the initial problem of finding the gradient the agent employs a heuristic that involves circling on the spot while the distance from the food is either constant or decreasing, and moving straight otherwise (i.e., while it is increasing). To confirm this hypothesis, we performed a series of experiments in which we allowed the agent to move while controlling the sensory information arbitrarily. We considered the initial transient behaviour under two conditions: when the sensor value is fixed or decreasing, and when it is increasing at a constant rate. During the former, the agent circles around a small region of about 2 units of space. During the latter, the agent reduces the turning behaviour as a function of the rate of increase of the sensor activity.

The chemotaxis behaviour of the best-evolved circuit employs a strategy similar to that observed in very simple organisms. In *E. coli*, for example, chemotaxis is achieved by modifying the frequency of ‘tumbling’ (Macnab & Koshland, 1972). In *C. elegans*, the turning behaviour is referred to as a ‘pirouette’ but the heuristic is similar (Pierce-Shimomura, Morse, & Lockery, 1999).

Switching behaviours

We know the agent can perform well doing each task independently. In order to test whether it can also switch between them during its lifetime we change the circuit’s body without resetting the state of the neurons and evaluate the circuit’s performance. Although populations were not evolved to cope with this transition, most of the successful circuits managed to switch between tasks in both directions, including the best one analysed here. The example shown in Figure 10.3A is for a successful transition in one of the directions: from walking to chemotaxis. We will answer why this is possible in the last section of the results.

10.4.3 Dynamics of the decoupled circuit

As a first step towards understanding the evolved behaviours, we consider the dynamics of the circuit when decoupled from the environment. We do this by examining the asymptotic behaviour of the circuit after replacing the time-varying sensory input with a fixed parameter, thereby reducing it to an autonomous system.

\(^3\)Even if it had more than one sensor, then it would be too small to sense the gradient along the length of its own body.
Bifurcation diagram

In Figure 10.4 we show the asymptotic behaviour of the circuit as a function of the possible sensory perturbations that it can receive. Solid black trajectories represent attractors, dashed black trajectories represent saddle-nodes and grey dots correspond to limit cycles. Three bifurcations can be observed and are shown in the figure as colored disks. From left to right, the first bifurcation is a saddle-node bifurcation (red disk), from which a fixed point ($a_2$) and the saddle node ($sn$) arise. Fixed point $a_1$ is a stable spiral point for $s < 0.38$. This spiral is weakened and a stable limit cycle ($lc$) arises near the origin in what is likely to be a Hopf bifurcation (green disk). The size of the cycle first increases slowly and then comes crashing inwards until it reverts to a stable spiral point for $s > 0.77$.

**Figure 10.4:** Bifurcation diagrams. Two-dimensional slices of the 6D bifurcation space (5 neurons + sensor). How the equilibrium points change as a function of the sensor value, projected on the two effector neurons $y_2$ (A) and $y_1$ (B). See main text for an explanation of the labels.

Phase portraits

The bifurcations divide the space of qualitatively different dynamics available to this circuit in four. The dashed vertical lines in Figures 10.4A and 10.4B represent the slices of parameter space that are studied in Figure 10.5, which shows two-dimensional slices of phase-portraits $P1$
10.4. Results

through $P_4$. The portraits are shown only for neurons 4 and 5, which control forward and backward movement in both the insect and khepera bodies. $P_1$ corresponds to the family of phase portraits available when $s < 0.28$ before the first bifurcation occurs. As there is only attractor ($a_1$) in the system all trajectories are drawn to it. The paths taken to get to it are not direct, but follow a spiral in its vicinity. Prior to this, however, a subset of the trajectories follow a much longer transient involving a loop near the region labelled $t$.

Figure 10.5: Example phase portraits. Two-dimensional slices of phase-portraits $P_1$ through $P_4$ for the two effector neurons, $y_1$ and $y_2$. See main text for an explanation of the labels.

$P_2$ corresponds to the family of phase portraits available between the first and second bifurcations ($0.28 < s < 0.38$). It comprises two stable fixed points $a_1$ and $a_2$, and a saddle-node (not shown). Trajectories starting in the top-left corner approach the newly created stable point, $a_2$, whose basin of attraction (not shown) is smaller than $a_1$’s. Hence, most trajectories approach $a_1$. The transient towards it is, again, not direct. In fact, as we will see, this is always the case for this attractor. What varies is the spatial extent of the spiral loop. When looking through the perspective of neurons 4 and 5, any trajectory bound for $a_1$ will first navigate towards $t$. In $P_3$ the spiral attractor becomes a stable cycle. The transient remains similar.

In $P_4$ the cycle disappears and gives way to a stable fixed point. Also, $a_2$’s basin of attraction becomes larger, with certain initial configurations ending up in $a_2$ that previously ended in $a_1$. Also, the effect of the saddle-node ($sn$) becomes more obvious in this portrait. The transient loop ($t$) still exists, but it is relatively closer to $a_1$.

Finally, approximations of the turning point of the transient loop are incorporated into our
bifurcation diagram as disks labelled $t$ in Figures 10.4A and 10.4B. As these are not real limit sets of the system, they do not show up in our bifurcation analysis. They will play, however, a fundamental role in the agent’s autonomous behaviour. If the phase-portrait of the system is changing sufficiently fast (due to rapidly varying input), and if the neural state falls in the basin of attraction of $a_1$, then we can predict that it will most likely be seen around $t$ and never actually reach $a_1$.

10.4.4 Brain-body-environment coupled dynamics

Let us now consider the behaviour of the agent when coupled to the environment and how it relates to the underlying dynamical landscape described in the previous section. Figure 10.6 depicts the trajectories of the controller when driven by the agent’s sensor, which is itself influenced by the circuit’s effectors and the corresponding changes to how the agent perceives the environment. Red lines correspond to the walking task, blue lines to chemotaxis. The trajectories are imposed over a simplified version of the circuit’s autonomous dynamics from Figures 10.4A and 10.4B, using the same projections. While the dynamics of the two tasks are significantly different, they share the same underlying dynamical landscape.

During the walking task, the dynamics of the circuit are constantly switching between approaching attractor $a_1$ when swinging the leg forward, and approaching attractor $a_2$ during the stance power and coast. However, while the system gets close enough to $a_2$, it ends up relatively far from $a_1$. In fact, the cycle that arises from the coupled system is observed to switch between $a_2$ and $t$ (the longer transient towards $a_1$). This agrees with our prediction from the circuit’s autonomous dynamics, which suggested that it is being driven at a relatively fast rate. We test this hypothesis in the next section. We also note that the cycles in $y_2$ and $y_1$ follow opposite directions, clockwise and counterclockwise, respectively. This reflects the antagonistic muscle cooperation necessary to produce the swinging of the leg.

The trajectory during chemotaxis is more subtle and is produced solely within the basin of attraction of $a_1$. The circling search behaviour is produced by the longer transient towards $t$, in combination with the spiral shape in the vicinity of $a_1$. However, the state of the neural controller doesn’t really reach $a_1$ until the agent gets close to the food patch, at which point the sensor gets maximally activated. As soon as the gradient towards the food is found, the sensory value increases and the phase-portrait shifts, leaving the state of the effector neurons in a region of space where the power of the opposing effectors are balanced, which corresponds to moving straight. The spiral attractor and indeed the limit cycle around $a_1$ ensure that if the gradient ceases to increase, the agent will circle on the spot until it increases again. This agrees with our observation of the agent’s chemotactic heuristic. Finally, once the agent reaches the top of the gradient, the dynamics come cycling in towards $a_1$, which ultimately leaves the agent turning on the spot near the food patch.

10.4.5 Behaviour coordination: driven circuit

How does the neural controller perform the appropriate behaviour at the appropriate time? What our analysis shows is that it is not the neural controller itself that coordinates the change of behaviours. Instead, different patterns of feedback are created when the neural system is coupled to
Figure 10.6: Brain-body-environment coupled dynamics for the two different tasks: walking (red) and chemotaxis (blue). Trajectories are imposed over the bifurcation diagram of the non-autonomous nervous system (grey). Two slices of the 6-dimensional space are shown. The same two slices shown for Figure 10.4A and 10.4B, respectively.

a different body, and it is these patterns that ultimately produce the distinct transient behaviours. It is worth emphasizing that brain, body and environment form a closed loop such that no single part is the sole cause for the difference in the dynamical patterns. The shape of the feedback is as much the result of neural output and body dynamics as the neural activity itself is the result of environmental feedback.

Is a particular feature of the feedback signal associated with the change of behaviours? From the previous section, we observed that the walking task is generated by the system’s movement between two basins of attraction, in such a way that it never actually settled on any one of them. This suggests that fast switching between the two basins generates walking. During chemotaxis, on the other hand, the dynamics stay within the basin of $a_1$ and movement is sufficiently slow to allow for it to draw close to the attractor, settling only when sufficiently close to the food patch.

In summary, the essential feature of the feedback is its time-scale. While the sensory feedback from the insect-like body is relatively fast, the sensory feedback from the khepera-like body is
Chapter 10. Learning with your body

much slower. We test this hypothesis by driving the neural system with fast and slow sine waves, and compare the observed dynamics in internal space (Figure 10.7A) to the dynamics during walking and chemotaxis (Figures 10.7B and 10.7C). We find that, depending on the frequency, it will either: (i) jump from one attractor to the other, which is relevant to the walking behaviour, or (ii) stay on the central attractor, which is relevant to chemotaxis. Finally, this provides an explanation for why the evolved agent can switch between behaviours during its lifetime. The behaviours do not depend on where in neural space the state of the system is, but on the rate at which it is being driven by the feedback from its interactions with the environment, as a product of the mechanics of its body.

10.5 Discussion

Multifunctionality in biological circuits is understood to be the result of either: (a) multistable neural dynamics or (b) modular organization (Briggman & Kristan Jr., 2008). This follows directly from the assumption that the nervous system is solely responsible for the generation of behaviour. In order to demonstrate the problematic nature of this assumption, we evolved model neural circuits that exhibit either chemotaxis or walking behaviour depending on the body in which they are embedded. The coordination of both behaviours occurs in the absence of any adaptive process built into the neural circuits themselves. Rather, the differing feedback through the body and environment in the two cases serves to produce distinct behavioural dynamics. No amount of analysis of the neural circuit alone could reveal the mechanisms of these behaviours, because they are a property only of the coupled dynamics of the entire brain-body-environment system.

We have shown that small dynamical neural networks are able to implement qualitatively different behaviours as distinct transients on a single dynamical landscape. Specifically, we evolved an agent that could perform locomotion when coupled to a one-legged body and chemotaxis when controlling a khepera-like robot. We demonstrate this is possible without imposing structural modules on the controller, and without employing complicated fitness functions or evolutionary shaping protocols. Neither was it necessary to introduce parameter changes in the controllers or to provide a signal for when the swap of bodies and corresponding behaviour was to occur. The interactions of neural controller, body and environment alone are sufficient to create distinct transient dynamics appropriate for solving both tasks.

The divide-and-conquer approach championed in engineering would suggest that separate modules should be evolved to produce the two tasks independently. This however wouldn’t necessarily simplify the problem, as the main challenge would then be to design a mechanism of coordination and a sophisticated sensing machinery to detect when to switch between the modules. More importantly however, while modular structures and synaptic-plasticity exist in living organisms, they were selected based on the adaptiveness of their behaviour and not on how easily-understandable their internal mechanisms are. We therefore argue that understanding networks whose structure is not imposed from the top down will help us develop the tools to understand how multiple behaviours are generated in living organisms.

Our dynamical systems account of the evolved agent indicates that it is misleading to associate a behaviour with an attractor or a basin of attraction in the decoupled internal dynamics of the controller. In fact, there can be many behaviours in the same basin of attraction, as shown in Buckley
Figure 10.7: System driven by sinusoidal waves of different frequencies. [A] Trajectories of the neural state of the system are imposed over the circuit’s bifurcation diagram (grey, limit cycles not shown). When the system is driven slowly \( f = 0.272 \), thin black trajectory), the trajectory remains near the \( a_1 \) attractor. As the system is driven faster \( (f = 0.136 \text{ and } f = 0.068 \), thicker trajectories), the state of the system starts oscillating between attractors \( a_1 \) and \( a_2 \), but because of the longer transient towards \( a_1 \), the oscillation is effectively between \( t \) and \( a_2 \). [B] Two-dimensional slice through the space of neural activity during walking (red) and chemotaxis (blue) imposed over one of the phase-portraits (P3). [C] Neural activity for the system when driven by a fast \( (f = 0.272) \) and slow \( (f = 0.068) \) sinusoidal wave.

et al. (2008), or single behaviours that require several different attractors, as in the RPG shown here and in Beer (1995a). Furthermore, as this chapter demonstrates, multiple behaviours may use an overlapping set of diverse attractors and their basins. This provides an example of the importance of understanding behaviours as a result of the interactions between brains, bodies and environments, where transients play an equal, if not more important, role than attractors.

Finally, an important feature of recurrent neural networks is that their history of activations allows them to respond to otherwise identical stimuli in a context-dependent fashion. In other words, a system with internal state, when embodied and situated, is not constrained to a single
sensori-motor mapping (as was shown in our example). In von Uexkull (1957)’s terms, such systems could be said to “bring forth their own Umwelt”. But while the act of interpreting sensory input contextually is usually attributed wholly to the agent, the example presented here shows that “meaningful” behaviour is the result of interactions in the brain-body-environment system as a whole.
Chapter 11

Concluding remarks

The aim of this thesis has been to better understand how learning behaviour can be generated from a situated, embodied, and dynamical agent. We employed existing evolutionary techniques to synthesize dynamical system neural controllers on tasks that required learning behaviour. We varied the experimental conditions on several dimensions. First, the stimulus to be remembered was in some tasks drawn from a discrete set and in other tasks it was drawn from a continuum. Second, the level of embodiment and situatedness of the model agent varied from completely abstract ‘neural networks in a vacuum’ to simulations that increasingly take into account the role of the agent’s interaction with the environment and the feedback generated through its body. Third, the scope of the cognitive tasks was also varied. We explored Hebbian learning, associative learning, object discrimination, coping with visual inversion, imprinting, and coping with changes to body morphology. Yet we added no learning algorithm to the internal dynamics of the agent. Evolution had to come up with the mechanisms that would produce the learning behaviour on its own, starting from continuous-time recurrent neural-like components as its building blocks. We succeeded in artificially evolving continuous-time recurrent neural networks without synaptic plasticity on all of the tasks that we set out to study. For each of the most successful networks we then went into some depth trying to understand how the learning behaviour was produced using dynamical systems theory.

This chapter contains an overview of the work presented in this thesis. The first section comprises a review of the simulation experiments conducted and discusses what conclusions we may draw from the results obtained. The second section discusses whether we should really consider the experiments in this thesis to be learning. The third section proposes a set of directions for future research that have opened up as a result of our work. The last section provides the overarching conclusions of this thesis.
11.1 Contributions

Hebbian mechanisms in nonlinear recurrent neural circuits do not necessarily result in Hebbian-like learning behaviour

In Chapter 5 we demonstrate that increasing the strength of the connection linking two model neurons does not always increase their firing correlation. We also show that neural networks without synaptic plasticity can be evolved to perform Hebbian-like learning behaviour. Our experiments suggest that the smallest networks must have at least four components, one of which must be an order of magnitude slower than the rest. We provide a geometric explanation for the Hebbian behaviour, as switching between two basins of attraction whose shape determines the correlation of firing. The importance of this chapter is that such a demonstration, because of its simplicity, lays the foundation for much of the rest of the work on ‘learning without synaptic plasticity’.

Learning without synaptic plasticity is not restricted to switching between two modes of interaction

In Chapter 6 we propose an associative learning task inspired on thermal-preference behaviour observed in the nematode worm. The task is modelled at an abstract level: non-embodied and minimally situated. The dynamical system controller is required to learn the temperature associated with food and then remember it for further testing. With this work we demonstrate that neural networks without synaptic plasticity are not restricted to learning behaviour in tasks that require swapping between one of two modes of interaction. We show that they are also capable of learning, remembering, and recalling continuous stimuli. Therefore, that they can be evolved to modulate their dynamics on a continuum as well. We show this with agents constrained to use only their internal dynamics to remember. Our experiments suggest that successfully evolved networks must integrate components acting on three different timescales. We show that these networks can remain sufficiently plastic to relearn new stimuli while forgetting previously learnt stimuli during their lifetime.

The cognitive limitations of a situated agent cannot be deduced from the limitations of its internal dynamics alone

In Chapter 7 we introduce the importance of studying embedded networks when attempting to understand their behavioural capabilities. We show that it is not possible to deduce the cognitive limitations of a situated agent from the limitations of its internal dynamics alone. We provide two examples of reactive controllers (i.e., without internal state) performing non-reactive behaviours (i.e., demanding some state) by exploiting their history of interactions with the objects in the environment. In the first example we study an agent that has to catch falling circles with and without inverting vision. In the second example we study an agent that has to distinguish between bar-shaped and circle-shaped objects. In both cases, the reactive agents evolve to exploit their history of interactions with the falling object to succeed. Although there is no such thing as a living organism with purely reactive internal dynamics, the extremity of the conditions in the examples suggest that, for situated and embodied agents, we must also consider a broader range of available sources of state than just their internal dynamics.
The usefulness of the FSM interpretation is reduced as the ecological context of the task is increased

In Chapter 8 we study an extension to the associative learning task in Chapter 6. For the abstract case presented there, the internal dynamics instantiated a state machine of some kind. This chapter explores the differences in required internal dynamics when the agent has the potential to be active in choosing its own stimuli by being fully embodied and situated. Our analysis suggests that as the ecological context of the task is increased, the usefulness of understanding the evolved dynamical system as a finite state machine is reduced. Although the analysis of embedded networks becomes harder, we have shown that it is still tractable. We also show how the relevance of the transient behaviour arising from the multiple timescales and from the embedding in the environment becomes as important as the asymptotic behaviour of the circuit.

The agent’s history of interaction with its environment can be used as a source of memory

In Chapter 9 we provide the first set of experiments of learning on a continuum in an embodied model. The task is loosely abstracted from imprinting in birds. An agent is first presented with a falling circle of a certain size. A second circle is then presented, the agent has to catch it if it is of the same size as the first and avoid it otherwise. The task requires that agent remember the size of the first circle and compare it with that of the second circle to make a decision to catch or avoid. We demonstrate that the agent’s history of interaction with the environment has also a role in learning behaviour. We illustrate that agents, if situated, can use their interaction with the environment as a source of memory to remember continuous stimuli in a learning task. We also show that this is often the case, suggesting that agents will exploit it when and if available. Furthermore, we show that several of the successful agents use, in fact, a combination of both internal dynamics and their history of interaction to remember. This provides a concrete example of how plasticity is a feature not of the controller but of the complete agent-environment system.

The feedback loop through the agent’s morphology can modulate the circuit’s dynamics

In Chapter 10 we demonstrate that also the body has an important role to play in learning. We illustrate that neural networks, when embodied, can switch between performing two qualitatively different tasks: orientation to sensory stimuli and legged locomotion. We demonstrate that small fully interconnected networks can solve these two tasks without providing a priori structural modules, explicit neural learning mechanisms, or an external signal for when to switch between them. We show how the difference in the form of the feedback that is generated from the movements of the different bodies generates the appropriate two behaviours and the switching, when in interaction with an attractor landscape of the internal dynamics that does not change.

From a broader perspective, our work has continued to extend the analysis of dynamical agents performing minimally cognitive behaviours towards more “representation-hungry” tasks. It has been often argued for the need for more sophisticated evolutionary algorithms and particularly for more sophisticated dynamical system controllers with special capabilities in order to achieve more complex behaviours. In this thesis we set out to increase the complexities and representation-hungliness of the tasks, and to do so without resorting to the addition of specialised learning algorithms, parameter-changing mechanisms, or training techniques of any sort. We succeeded to extend the complexity of the tasks using the existing methodological techniques.
11.2 Discussion: Is it really learning?

Our work assumes that the definition of learning being considered is not purely mechanistic (i.e., tied to a particular kind of change in a particular kind of system). Instead, we consider learning as a behaviour. More specifically, as a change in behaviour. But which changes in behaviour deserve to be labelled learning and which do not? Our work also assumes that learning has to be considered in the broader context of evolution. In conditions where (a) the environment tends to be subject to changes within the lifetime of an agent and (b) some organisms have the potential to modulate their behaviour to adapt to such environmental changes, if adapting provides the organism with a selective advantage over non-adapting organisms, mechanisms that facilitate learning will evolve in the population. It is in this context that we wish to understand learning: those modifications of behaviour that allow the organism to re-adapt to changes in the environment that occur within its lifetime (and which as a consequence improve its reproductive success).

Previously, work on evolving learning behaviours using neural circuits without synaptic plasticity had focused on tasks that required choosing between two possible modes of interaction based upon a single experience (see Chapter 4). Some would argue that this is not really “learning”. For us, what is important about these studies is that it provides a clear example of how an agent can change its behaviour within its lifetime in response to a change in the environment in the absence of changes in the synapses. But even if we consider the sceptics’ position, by arguing that organisms are ‘usually’ able to change between more than just two behaviours. The question then becomes, how many choices are enough to be regarded as “real” learning? In order to address this concern, in this thesis we considered scenarios where the agent had to change behaviours between an infinity (in a bounded continuum) of possibilities. So, could an agent without synaptic plasticity be evolved to manage beyond changes between two modes of interaction? In two of our experiments (Chapter 6 and 9), we demonstrated that an agent can indeed be evolved to learn a particular feature of the environment within a bounded continuum, and change its behaviour according to it. And that it can do this using non-synaptic forms of plasticity.

This does not mean, however, that the learning that evolves in these agents is ‘universal’ (i.e., that the circuit can now adapt to any changes in the environment). In fact, it does not even mean that a successful agent should adapt to changes outside the range that its predecessors experienced during evolution. Here again some would like to argue that this is not really learning. As we discussed previously (see Section 4.1.1), learning tends to be treated as a “general problem solving” ability. However, we do not think learning is ever universal (not even for humans). For any organism, we should be able (at least in theory) to find out the set of changes of environment that it would be able to cope with successfully, and those for which it would fail to adapt. Most certainly, that such an organism fails to adapt to certain changes to the environment should not mean that it could not learn. On the contrary, the questions of interest for any agent that can adapt to changes to its environment are: (a) how is the failure-to-adapt boundary related to the scenarios experienced over evolutionary time? And (b) which mechanisms facilitate the expansion of the set of changes that it can adapt to? Most importantly, both of these questions must be tackled within the context of the organism’s evolutionary niche. This is partly the reason why we studied our successful circuits outside of the range of scenarios that they had been evolved for. It was not because we expected that the evolved agents would employ their non-synaptic plasticity to adapt to
everything. Instead, testing the boundaries where learning fails (e.g., under which conditions the agent stops being able to relearn? under what conditions does it start forgetting what was learnt?) helps us understand how the circuit works.

Finally, it is generally assumed that the nervous system is primarily responsible for the production of behaviour, while the body and environment play supplementary roles. This assumption has affected the traditional perspective of what learning is. Previously, work on evolving learning behaviours using neural circuits without synaptic plasticity had only considered that plasticity could arise from within the agent’s ‘brain’. However, if we seriously accept that behaviour is the product of the interaction between brain, body, and environment, then we must also consider that behaviour can be modulated by three different means: the internal dynamics of the brain, the feedback through the body, and the interactions with the objects in the environment. In this thesis, we considered all three different ways of generating plasticity for the first time.

In the worst case scenario, if all that this thesis achieves is to encourage the reader to question the assumptions about what learning is and how it can arise in a situated, embodied, and dynamical agent, then it will have served its purpose.

11.3 Future work

The methodology used in this thesis should be extended towards other learning behaviours. The most obvious next step could be to study second and higher-order conditioning, where the initially associated stimulus can consequently be used to learn about some new stimulus. Another useful next step should be to study the blocking effect, a phenomenon observed whereby conditioning to a stimulus is blocked if the stimulus has been reinforced in compound with a previously conditioned stimulus. Both phenomena are discussed in most textbooks on learning (e.g., Squire, 1992; Shanks, 1995; Pearce, 2008). It will also be important to relate the dynamical system explanations with the existing understanding from animal learning theories.

Cognitive science is dominated by the use of a notion of internal representations in the brain. But the meaning of this notion has remained elusive (Harvey, 2008). One reason for this is the lack of empirically driven case studies in embodied, situated and dynamical system agents that illustrate unambiguously what this notion means and what the advantage of its use when understanding embodied adaptive behaviour is. There have been criticisms that argue that while representations may not be necessary for simple tasks they are essential for some sufficiently “representation-hungry” set of tasks, as described by Clark and Toribio (1994). Thus, one of the motivations for the work presented in this thesis has been to increase the level of “representational hunginess” in the existing minimally cognitive tasks, where the agent has to make decisions based on a stimulus that is no longer present in its immediate environment. One useful direction of research would be to engage with those cognitive scientists that would predict that in any of the tasks presented in this thesis a representation of some sort was required, and then to make precise what those entail for such situated, embodied, dynamical agents.

This thesis deliberately avoided using synaptic plasticity in learning in order to focus on the rest of the aspects of the brain-body-environment interaction that can be involved in the generation of learning behaviour. But of course, synaptic plasticity does exist and it has been shown to be involved in learning behaviours for many living organisms. Unfortunately, most work that
includes synaptic plasticity assumes that the changes to the weights are in charge of modulating
the behaviour of the network. Studying what the role is of the synaptic plasticity observed in
evolved networks in tasks that require learning behaviour should be useful. An interesting next
step could be to analyse networks with synaptic plasticity on the same learning tasks as the ones
we presented here. If, as expected, the behavioural plasticity is not always related to the changes
of the synaptic strengths, then it should be of interest to explore under which conditions (e.g., nature
of the selection process, or the range of parameters used for the neural and synaptic dynamical
laws) synaptic plasticity does or does not perform the role of modulating the dynamics of the
network.

Another direction of interest should be to ‘translate’ some of the learning mechanisms that are
observed in the evolved networks in this thesis into rules for synaptic plasticity. Let us illustrate
what we mean by considering a fully connected three-node network. There are two ways in which
any two nodes are connected. Node \( A \) is connected to node \( B \) via the synaptic weight that goes
from \( A \) to \( B \). But \( A \) is also connected to \( B \) via the set of synapses that pass through node \( C \). Changes
in the state of \( C \) will affect the interaction between \( A \) and \( B \) (as much as changes in the strength
of the synapse from \( A \) to \( B \) will affect it). Therefore, it should be possible to deduce a dynamical
law such that when applied to the synaptic connection between \( A \) and \( B \), it serves to substitute
node \( C \) altogether. Such a weight-changing rule could be either variations of existing rules or an
altogether novel one. Both situations would be of interest.

Almost all of the networks analysed in this thesis achieved learning by evolving components
with intrinsic multiple timescales (i.e., the time-constant of each node was independent and un-
der evolution). It should be possible for networks to perform learning behaviours by using their
recurrent connections only, even in the absence of components that can act slower than others.
One question of interest that our work opens up is: Why did networks evolve multiple time-scales,
rather than, for example, reverberative dynamics? One possibility is that this was a side effect
of limiting search to relatively small networks. Future work could, for example, constrain the
dynamical recurrent neural networks to have equally fast time-constants while performing the
same learning tasks explored in this thesis. A question of interest is, for example, can networks
comprised of components that act over the same timescale remember features on a continuum?
A similar direction could be taken if we consider that components with slower timescale have
a higher metabolic cost on the organism, with selection pressure towards networks with mostly
fast-acting components.

Another direction of interest should be to explore whether there is a fundamental difference
between the dynamics of evolved circuits on tasks requiring an irreversible change that occurs
during a critical developmental period, and another one requiring that the circuit remain plastic
throughout its lifetime. Except for the imprinting task in Chapter 9, all of the experiments in this
thesis involved tasks that required the circuit to remain plastic throughout its lifetime.

At any point in time, a dynamical system can be reactive (at the mercy of its environment)
or it can be fully autonomous (all actions being generated by activity generated from within,
completely ignoring the sensations from its environment). There is in fact a full spectrum of
environmentally-determined versus internally-determined behaviours. Let us call this the agent’s
connection with their environment. An example of a system at the environmentally-determined
end of the spectrum is a leaf floating downstream in a river. If we choose the boundaries of our system of interest as ‘the leaf’ then we can say that its future behaviour is largely determined by external variables, its environment (in this case the water in the river). On the other end of the spectrum there are systems that are purely internally-determined. Their environment, on the contrary, does not determine or influence the future behaviour of these systems. These are solely self-governing systems. Although autonomous, in the traditional use of the term, these systems are not of interest because they cannot readily adapt to their changing environments. GOFAI robots with internal models are good examples (e.g., Nilsson, 1984; Moravec, 1982; Giralt, Chatila, & Vaisset, 1984). Each step that Honda’s Asimo robot takes, for example, is largely determined by internal calculations based on the positions of its limbs and the distribution of the weight of its body as internally modelled (Chestnutt, Lau, Cheung, Kuffner, Hodgins, & Kanade, 2005). Dynamical systems can be anywhere in the spectrum. But any dynamical system that remains at a constant level of connectedness with the environment is likely to be of little interest (e.g., leaves, rocks, GOFAI robot). Our intuition is that dynamical systems that can learn are those whose connection to the environment varies over time. Adaptive dynamical systems will at times require the ability to be strongly influenced by its environment and less so at other (imagine a person kayaking, sometimes following the current, sometimes rowing). An interesting next step would be to develop a measure for how environmentally-driven or internally-driven an agent is over time, and to evaluate the dynamical recurrent neural networks performing learning tasks. Also comparing them with systems evolved for tasks that do not require learning behaviour.

Finally, and given certain developments in the neural sciences, I believe the time is appropriate to begin to understand the neural basis of learning behaviour in real living organisms using evolutionary robotics techniques and dynamical system analysis. The nematode worm *C. elegans* is probably one of the organisms best suited for this endeavour. The anatomical connectivity of its nervous system is known (White, Southgate, Thomson, & Brenner, 1976). There are reasonable models of its body and biomechanics (e.g., Niebur & Erdös, 1991). The repertoire of behaviours that the worm is capable of continues to expand in interesting directions (see Hobert, 2003, for a review). The technologies for recording neural activity live in freely moving worms is also beginning to be feasible (Faumont & Lockery, 2006). A novel technology for creating environments with particular shapes, which will make possible studying the worm on a wide variety of learning assays, is underway (Lockery, Lawton, Doll, Faumont, Coulthard, Thiele, Chronis, McCormick, Goodman, & Pruitt, 2008). We can use the anatomical knowledge to constrain the circuit, use the behavioural data to design a fitness function and then let the evolutionary algorithm find different parameter sets that accomplish the desired behaviour. Some work in this direction has been carried out (Dunn, Lockery, Pierce-Shimomura, & Conery, 2004; Dunn, Conery, & Lockery, 2007). In fact, a long term goal would be to use such an approach to generate a complete model *C. elegans*. Also because the worm’s neurons are non-spiking, continuous-time recurrent neural networks are a biologically reasonable model. And so it should be possible to begin to use the dynamical analysis employed for our artificial agents in such real biological networks.

---

1. Although important details are still missing. Neurotransmitter interactions are, for example, less well known.
11.4 Conclusions

The work in this thesis has demonstrated that there are several non-synaptic forms of plasticity in an embodied, situated, and dynamical agent. It has shown the ability of small continuous-time recurrent neural networks to perform learning behaviour under a series of different conditions. All of the work on evolving agents that learn without synaptic plasticity has focused on tasks where the agent is required to behave differently in a discrete number of different environments, in practice two. We extended the approach from remembering discrete to continuous stimuli. Also, all of the work has focused on the role of the internal dynamics of an agent in learning behaviour. By analysing networks evolved in abstract tasks as well as more ecological versions of those same tasks, we showed how plasticity could switch from being generated purely as a result of the internal dynamics to arising from the full brain-body-environment interaction.

From our work we can conclude that: (a) synaptic plasticity is not necessary to produce learning behaviour, (b) continuous-time recurrent neural networks can produce learning behaviours that go beyond switching between two modes of interaction, (c) learning behaviour can be (and is often) produced from the agent’s history of interactions with the environment and from the feedback that the morphology of its body provides, not just from the within its brain, and (d) we can employ the mathematics of dynamical systems theory to understand the operation of such evolved brain-body-environment systems.
Bibliography


Bibliography


Bibliography


