Reinforcing Connectionism: Learning the Statistical Way

Peter Samuel Dayan

PhD
University of Edinburgh
1991

Declaration

I have composed this thesis by myself, and it contains original work of my own execution.

Edinburgh, 25th February, 1991

For my family:
Hedi and Anthony, Zoë and Daniel

Abstract

Connectionism's main contribution to cognitive science will prove to be the renewed impetus it has imparted to learning. Learning can be integrated into the existing theoretical foundations of the subject, and the combination, statistical computational theories, provide a framework within which many connectionist mathematical mechanisms naturally fit. Examples from supervised and reinforcement learning demonstrate this.

Statistical computational theories already exist for certain associative matrix memories. This work is extended, allowing real valued synapses and arbitrarily biased inputs. It shows that a covariance learning rule optimises the signal/noise ratio, a measure of the potential quality of the memory, and quantifies the performance penalty incurred by other rules. In particular two that have been suggested as occurring naturally are shown to be asymptotically optimal in the limit of sparse coding. The mathematical model is justified in comparison with other treatments whose results differ.

Reinforcement comparison is a way of hastening the learning of reinforcement learning systems in statistical environments. Previous theoretical analysis has not distinguished between different comparison terms, even though empirically, a covariance rule has been shown to be better than just a constant one. The workings of reinforcement comparison are investigated by a second order analysis of the expected statistical performance of learning, and an alternative rule is proposed and empirically justified.

The existing proof that temporal difference prediction learning converges in the mean is extended from a special case involving adjacent time steps to the general case involving arbitrary ones. The interaction between the statistical mechanism of temporal difference and the linear representation is particularly stark. The performance of the method given a linearly dependent representation is also analysed.

The method of planning using temporal difference prediction had previously been applied to solve the navigation task of finding a goal in a grid. This is extended to compare the qualities of alternative representations of the environment and to accomplish simple latent learning when the goal is initially absent. Representations that are coarse-coded are shown to perform particularly well, and latent learning can be used to form them.

Acknowledgements

I owe an enormous debt to David Willshaw. His influence pervades every page, and his inspiration, tolerance, humour, perceptiveness, and guidance quite made my research, and made returning from exile worthwhile. David Wallace made the return possible, provided more than ample doses of enthusiasm and encouragement, and taught me the value of time. I am very grateful to him for his supervision and help.

My parents and the rest of my family offered their constant support and help, and carefully fostered my academic pretensions. If only it were medical...

Working in David Willshaw's group at the Centre for Cognitive Science has been enjoyable, educational and exciting in equal measure. I am very grateful to Marcus Frean for many fruitful discussions, for my vicarious exercise, and for healthy competition for our trusty steed. Jay Buckingham, whom I also thank for interesting interactions, kept bonzo and its similarly elevated colleagues processing perfectly, and provided the means, if never the excuse, for learning ****. Kate Jeffrey and Toby Tyrrell have more lately been valuable colleagues and friends. Jean Wilson, Leslie Chapman and Rosanna Maccagnano firmly kept us all in check, and greatly enlivened the fourth floor. Both the Centre and the Department of Physics brooked my comings and goings and provided most opulent facilities.

Many people have read and laboured mightily to improve versions of this thesis. Many thanks to Nick Chater, Steve Finch, Carol Foster, Peter Hancock, Kate Jeffrey, Alex Lascarides, Jon Oberlander (in whose illustriousness as co-author I basked), Martin Simmen, Randall Stark, Rich Sutton, Toby Tyrrell, David Wallace, Chris Watkins, and David Willshaw for valiantly providing the irresistible force. The last additionally provided the resistible punctuation.

I have had innumerable discussions with all these people, and many other friends and colleagues – Robert Biegler, Mike Malloch, Richard Morris, Mike Oaksford, Steve Renals, Richard Rohwer, Tom Tollenaere, and all the participants in the Connectionist Workshop at Edinburgh; Andy Barto, under whose ægis, and ultimately care, I made a visit to Amherst which made a lasting impression, and also his group, particularly Vijaykumar Gullapalli and Satinder Singh; Geoff Goodhill, also for light, heavy, and Californian entertainment; and the students and faculty of the 1990 Connectionist Models Summer School, particularly Geoff Hinton and Terry Sejnowski.

Life in Edinburgh was made fondly memorable by Clare Anderson, Shari Cohn, Alan Simpson, Sally Wassell, Hon Yau, Sophie Cormack, who played the anti-heroine and made the large, stuffed, hate-objects, and especially Carol Foster, whose support and encouragement have meant a great deal.

I am very grateful to SERC for their financial assistance.

Contents

Today's theses rap tomorrow's fish.

Chapter 1

Connectionism in Context

To compare AI as mechanised flight with neuroscience as natural aviation is to ignore hot air ballooning. But where, prey, might the hot air come from?

After J Oberlander

1.0 Summary

The vituperative disputes between connectionism and its more traditional alternatives are rather confused between methodology and philosophy, explanation and replication, and theoretical neuroscience and practical computer science. These confusions can be untangled through the careful use of alternative levels of analysis, revealing the interest of connectionism to lie in the cornucopia of non-traditional mechanisms and representations that it offers, and its regard for learning.

Unfortunately, traditional delineations of levels are rather cavalier in the way they treat learning. This chapter argues that learning can be incorporated into such accounts through the mediation of statistical computational theories, and presents a summary of the rest of the thesis in this light.

1.1 Introduction

Many of those people who helped forge modern cognitive science out of behaviourism's sterile mindlessness see connectionism's reappearance as heralding a return to the bad old days. It seems to suffer from its products' apparently uninterpretable and unstructured internal representations, and its producers' seemingly empiricist leanings. The ensuing debate has something of the flavour of the old battles over the philosophical (il)legitimacy of artificial intelligence (AI), with proponents of connectionism talking glibly about systems that cannot be built, quite yet, and opponents, using arguments oft used against them, pointing to a few possible theoretical flaws and to many evident inadequacies of existing systems. A salient paradox is that certain of the strongest critics of traditional AI are actually mildly more favourably disposed.

Arguments over empirical adequacy are set to run and run, ever more fiercely given the sense of *fin de siècle* pervading traditional AI, but their theoretical stablemates have recently come to the fore. It seems that philosophically minded connectionists are trying to steer an uneasy semi-reductionist course between the Scylla of symbolic re-implementation and the Charybdis of biological entrainment. As a number of authors have pointed out, connectionism should refuse to enter this particular Odyssey; it should rather pose questions for its whole field, including AI, based on its own domains of expertise, notably learning. To do this it is necessary to adopt some classification of what it holds dear.

The work contained in this thesis has been motivated by a particular view of the rôle of connectionism in cognitive science. This stresses the importance of statistical notions of learning and the distinction between mechanisms and representations. This chapter attempts to describe the position and relate it to subsequent chapters. The next section looks at theoretical levels of analysis, as applied to classical and connectionist systems, section 1.3 considers how this might be extended to include learning, and section 1.4 introduces the work in the subsequent chapters and relates it to these conclusions. The chapter is purely descriptive – about *how* learning might be incorporated into theoretical accounts. Chapter 6 tries to justify *why* this might be important to cognitive science and AI.

1.2 Levelling the Field

1.2.1 The Three and The Many

Cognitive science now knows well the importance of understanding complex systems at a number of different levels. Marr [96] is usually credited with bringing the issue to the fore, having self-confessedly transgressed earlier in his own career [91, 92, 93]. For information processing systems, he differentiates three levels as follows:

Computational What is the goal of the computation, why is it appropriate, and what is the logic of the strategy by which it is carried out?

Algorithmic How can this computational theory be instantiated? In particular, what is the representation for the input and output, and what is the algorithm for the transformation?

Implementational How can the representation and algorithm be realised physically?

Underlying them is a desire:

> 'to make explicit statements about what is being computed and why, and to construct theories stating that what is being computed is optimal in some sense or is guaranteed to function correctly. The ad hoc element is removed, and heuristic computer programs are replaced by solid foundations on which a real subject can be built.' [96]-p19

Marr was led to posit these levels by the explanatory failures of earlier theories that made a kind of category error in trying to judge computations through algorithms or implementations. All too often, a working *program* in some domain was taken as a complete *theory* of that domain, when in fact it was only an algorithm with no specified range of applicability and many heuristics and hacks. Clearly, one computation can be instantiated by many different algorithms, and one algorithm realised on many different machines. Marr continues by delineating the autonomy of explanation between the levels, each of which:

> 'will have its place in the eventual understanding of perceptual information processing, and ... [all of which] are logically and causally related. But an important point to note is that since the three levels are only rather loosely related, some phenomena may be explained at only one or two of them.' [96]-p25

Marr considered the implementational level to be 'tied' to neuroanatomy and neurophysiology, the algorithmic level to psychophysics. However, he offered no such observable constraint to theorising at the computational level. Rather, the task here is to isolate and mathematise the regularities in the world that make information processing worthwhile at all. The resulting autonomy is precisely that of the FORTRAN programmers who need know nothing about the precise internal workings of the FORTRAN compiler, or the compiler writers who need know nothing about the material composition of the transistors in the target machine.

Foster [43] presents an extensive and critical analysis of the way that levels are used and abused in cognitive science and philosophy, and dissects a number of proposals, including those by Marr, Newell [106], and Pylyshyn [125]. Following a more restricted suggestion due to Haugeland [56], she points out that much of the confusion is caused by these authors considering only one dimension of the levels, rather than two. Two descriptions of a system (*eg* a VAX computer running a program that outputs 1 in response to inputs 01 and 10, and 0 in response to 00 and 11) can certainly differ in their degree of concreteness; *eg* a mathematical description of the task, an algorithm in some language for accomplishing it, or a wiring diagram and a list of the states of all the transistors. However, and particularly at the elusive algorithmic level, they can also differ in their degree of detail; a few succinct lines of FORTRAN compared with a more complete program that includes some checks on the validity of the inputs, compared with the binary numbers representing the resulting VAX machine code.

Having identified the separate dimensions, Foster describes a novel, 'massively multi-level' proposal of her own, focused on the possible levels of detail in the algorithmic description of a system. A particular advantage of this is that it also makes clear exactly what forms of equivalence there can be between two systems. Weak equivalence is extensional equivalence, *ie* when the functions implemented by the systems are the same (modulo a wealth of complications about finiteness and termination). Strong equivalence is a far more severe criterion, requiring the systems to go through the same states in the same order, for all possible paths from input to output.

One of the canonical features of the computational level is its abstraction away from time and space considerations – different algorithms may trade these off in different ways. Much theoretical connectionism, including most of the work in the rest of this thesis, seems to be couched at the computational level. Studies focus on the properties of mathematical objects, independent of their algorithmic or state-by-state realisations; there are many possible implementations of all the mechanisms described here. In fact there is an equivalent multi-level account at the computational level, with varying degrees of detail in the description of the mathematical objects, such as the function approximation schemes.

1.2.2 Battling Connectionism

This much should be ecumenical between most of the warring symbolic and connectionist camps. The disputes really begin when the levels are abused to suggest the ultimate irrelevance of classicism or connectionism to understanding and/or replicating human cognitive powers. For instance, Fodor and Pylyshyn [42] annunciated two criticisms in their recent influential attack on connectionism:

- They claim that connectionist systems are merely methods of implementing a higher level algorithm, and that the *'implementation*, and all properties associated with the particular realisation of the algorithm that the theorist happens to use in a particular case, is irrelevant to the psychological theory' (original emphasis). If true, this would severely restrict connectionism's potential explanatory force.
- They assert that connectionism *qua* connectionism (*ie* other than as a mere implementational means in the above sense) is incapable of handling structured knowledge representation or languages of thought having combinatorial syntax and semantics. Given the (excellent) reasons for believing such representations to be important in human cognition, this would prevent connectionism from offering an account *at a cognitive level* of the mind/brain.

The second of these assertions is actually incorrect – there are a number of radically different schemes ([30, 122, 153, 154] amongst many others) for handling structured knowledge representation in connectionist systems, and much research is underway to overcome their manifest inadequacies. It is also very important not to confuse sufficiency and necessity in Fodor and Pylyshyn's account of combinatorial mental languages and inferential operations (see Fodor's ground-breaking [38]). Their arguments focus almost entirely on the latter, and so say nothing about the possible augmentation or non-formality of their mentalese and (some) of the processes operating on it. Anderson's ACT* [3] is a good example of this, coupling tightly purely formal, structure-sensitive, operations embodied in a production rule interpreter with spreading activation operations in an associative memory.

Chater and Oaksford, in their elegant critique [22] of Fodor and Pylyshyn's article conclude that, in any case, the latter authors are really concerned with the first issue rather than the second. Chater and Oaksford sum up the traditionalist view as an affirmative answer to the question:

> 'Can the cognitive computational level be *formally* specified in an implementation independent way?' (emphasis added)

Dennett [29] neatly encapsulates the intent of the traditionalist approach to the question of implementation independence as a 'triumphant cascade through Marr's three levels', which, like most cascades, flows downwards. For connectionism not to be irrelevant to cognitive science, it is necessary to see why this approach is not the only straw in the wind.

Connectionist Use of the Levels

The trouble with the formulation of the question above is the word *'formally'*. Under the notion of the levels which informed both Fodor and Pylyshyn and Chater and Oaksford, it nullifies the whole issue. Since McCulloch and Pitts [86], it has been known that any classical computational system has a connectionist 'twin' (replicating the finite-state component of a Turing machine operating on an external, infinite, tape), and Pollack [121] has further shown that if the system can represent numbers to arbitrary accuracy, then it can also represent the entire tape in the activation of a set of units. Conversely, it is evident that connectionist systems can be approximated to arbitrary accuracy by traditional systems, since almost all existing connectionist systems are so simulated. This covers anything that can be *formally* specified.

To give the question some bite, therefore, the *formally* must be restricted to something akin to structure sensitive operations on some logic such as first order predicate calculus.[1] This, or some suitable equivalent, is effectively a non-trivial restriction at the computational level of concreteness, and one which requires empirical justification. This is particularly the case in view of AI's success in proving just how radically intractable this form of processing is even for medium sized bodies of knowledge. The original appeal of such a restriction, as the 'obvious' systematisation of folk psychology, is rather diminished by this persistent failure to lead to competent systems and by the advent of alternative systematisations, such as the one developed in [110] and discussed in chapter 6. As mentioned above, Fodor and Pylyshyn only demonstrate that *some* of the operations must be structure sensitive, not that they *all* must be. The remainder of their argument is about how it is theoretically possible to build systems, one of whose computational level descriptionsis *formal* in the above sense,

[1] Which was presumably how Chater and Oaksford intended it to be read.

whilst retaining the supposed advantages of connectionist systems, such as graceful degradation and parallel processing. The gap between claim and actual achievement in this is about as large as it is between the claim and achievement of structure-sensitive connectionist systems.

Fodor and Pylyshyn's claims about implementation obscure the real appeal of connectionism, which is *methodological*. To put it inelegantly, connectionism comprises a set of computational-level tools that seem appropriate for modelling part of cognitive information processing. This claim will be unpacked in two stages; first the notion of computational-level tools, and second their appropriateness.

Marr [96], and most of the proponents of different notions of levels, are interested in describing complete functioning systems, such as the mechanisms responsible for line detection in early vision. However, analysis based on levels can equally well be used to look at methodologies for building systems, and representational and inferential tools, including logics and computer languages such as PROLOG. The difference is that each of these methodologies offers only its own kind of information processing building block, and not a complete building. To extend the analogy further, analysing the nature of the blocks should reveal generalisations about the sort of buildings that each can form. Unfortunately, this kind of analysis is rarely undertaken.

As an example, consider semantic networks, over which much classical blood was spilt for quite some time. Quillian [126] invented them by mechanising certain psychological ideas on associative memory to give increased computational efficiency for the storage and defeasible inheritance[2] of object properties. The initial theory of semantic networks was couched in terms of a particular algorithm. The 'neat' wing of AI (see Hayes [58]) eventually pieced together a logic-based theory at the computational level showing that semantic networks offered no more expressive power than previous theories. More seriously, they also claimed that semantic networks bought their tractability in reasoning at

[2] An example of defeasible inheritance and reasoning is the revision of the initial surmise that Tweety can fly from the information that she is an object of the class BIRD, to the rejection of this conclusion given the extra information that she has clipped wings, is dead, or is additionally an object of the class PUKEKO.

the expense of a delinquency (*eg* Brachman [17, 18] and Woods [173]) that was not evident from the description of the original algorithms. Systems based on semantic networks can easily come to absurd conclusions, as in Brachman's example of one concluding that a non-grey elephant without a trunk must be a giraffe. It is apparent that Hayes and his colleagues performed classic levels-based analyses – inducing a more abstract, computational-level description of a representational medium, and deducing some of its properties based on this abstract description.

Coming back to connectionism, one aspect of the theoretical effort, and in particular the work in all but the last chapter of this thesis, is performed at the computational level – an aspect which includes analysis of the mathematical properties of connectionist techniques and tools in terms of their representational power and statistical convergence.

So much for the tools; the second stage of the claim was that they are appropriate for modelling cognitive information processing. This can be justified by comparing with empirical data the resulting inferred properties, such as associative recall, of systems constructed with these tools. Practical connectionist systems have very different behaviour (even in the weak equivalence or input/output sense) from traditional classical ones (contrary to Fodor and Pylyshyn's first claim), and may therefore be legitimate contenders as models. Of course, these connectionist systems can be realised in traditional hardware, and so in the sense discussed above, have purely formal accounts. However, methodologically, these formal accounts might never have emerged but for their connectionist provenance.

Equally, it is easy to confuse the non-autonomy of levels for the purposes of explanation of some behaviour, with their autonomy for the replication of it. Patricia Churchland [25] is particularly eloquent in her description of co-evolutionary reduction in the explanation of cognitive phenomena, licensing contributions from all the disciplines in the cognitive and brain sciences. However, were there to be a computational level theory of some set of behaviours, it could of course be reproduced by any number of different algorithms, each of which could be implemented in a number of different ways, *without any reference* to the original mechanism that generated them. The explanation, though, will inescapably involve all levels, particularly if, as is common in descriptions of computational systems, the degree of detail gets higher as they become more concrete. So, to re-iterate, the mere fact that the behaviour of some connectionist system can be replicated in a classical way says nothing to the methodological issue of divining the system in the first place.

An example of this last point is the recent development of continuous methods for solving classical combinatorial optimisation problems such as the Travelling Salesman Problem (see [81, 64, 35] for a discussion). Continuous methods had never traditionally been considered, perhaps because of the way the problem is usually posed. However, once the alternatives were realised, through the development of connectionist solutions, links between them and existing bodies of mathematics became clear [34].

Marr himself worked at the interface between levels. As mentioned above, he thought that psychophysics could provide evidence on the adequacy of different algorithms for the same task (indeed choosing between two algorithms for stereopsis on these grounds), and neuroanatomy and neurophysiology could do likewise for different implementations. These concerns then obviously trammel computational theorising, as is evident, for example, from his own work on colour vision. It might be thought that an obvious way to detect colour is to measure the wavelength of incoming light, but there is ample psychophysical evidence that this is not what humans do. The retinex theory (Land and McCann [80]) is an alternative computational theory constrained by this evidence, and Marr actually proposed [94] a neural implementation of an algorithm for retinex due to Horn.[3]

Connectionism therefore survives Fodor and Pylyshyn's [42] attack of implementational irrelevance. It provides a set of computational-level mathematical tools, some of which are investigated later on in the thesis, that are both novel and potentially relevant. This is true whatever the means by which they were originally motivated. It does not license the fallacious conclusion that connec-

[3] In fact, Horn's algorithm does not quite conform to the computational theory, but that is beside the current point.

tionism is the only straw in the wind itself.

Connectionist Abuse of the Levels

The dual to the criticisms that formal equivalence is methodologically irrelevant is the gulf between practical computer science and theoretical neuroscience. Connectionism owes far more to the former than the latter, and can justify little claim to biological realism. Very little is known in a systematic fashion about the computational substrate of the brain in any case, particularly in the area of its dynamical behaviour. Connectionism must stand or fall on its own merits, and not attempt to bask in mere reflected glory.

Another heinous crime perpetrated through ignoring levels of analysis is the doctrine of 'program as theory,' in which some program simulating human or animal behaviour in a domain is taken as a theory of the behaviour in that domain. Connectionism is just as prone to this, as evident from the following suggestion from Clark [26]

> 'The connectionist, however, effectively inverts this strategy [the classicist approach of expecting some high level understanding of a task to *precede* and *inform* the writing of algorithms]. She begins with a minimal understanding of the task, trains a network to perform it, and *then* seeks, in various principled ways, to achieve a higher-level understanding of what it's doing and why. ...This explanatory inversion ... actually constitutes one of the major *advantages* of the connectionist approach over traditional cognitive science. It is an advantage because it provides a means by which to avoid the *ad hoc* generation of axioms and principles.' [26]-p220

This is based on a rather overly optimistic view of the current capabilities of connectionist systems. However, even if it were practical, note that her 'minimal understanding of the task' is intended to inform the assumption that the task is modular (*ie* that it is at all meaningful to model it in isolation), the input/output representations, the design of the network and the functions performed by its nodes, the learning and dynamical algorithms, and the training set. All of these are crucial to a network's performance of a task, and are in any case essentially based on an implicit computational understanding of the task. It is unclear that the *ad hoccery* has been removed.

Consider, for example, Sejnowski and Rosenberg's NETtalk [140], a favourite example of Clark's. This learns a map from an input window of seven letters to a representation of how the middle one of those letters should sound. The use of the input window is crucial to the success of the network, since it allows it to model the fact that the positions of the articulators producing human speech may be partially determined by where they have just been (the forward part of the input window), and where they are due shortly (the backward part). Without this knowledge, either the network would have failed to learn the task, or it would have learnt it in an unilluminating fashion.

In fact, Clark's proposal, far from reducing the *odd hackery* (to use a spoonerism from Dennett), would probably increase it, since the assumptions are obscure in the final result. One is entitled to be nervous about the idea that the performance of a network with (optimistically) thousands of inputs might be that revealing about a cognitive task that is not even known to be modular.

More worrying is the paucity of computational accounts of connectionist systems. Although there is substantial work, particularly in statistical analyses, many of the systems come suspiciously close to being what Marr [95] would classify as Type 2; as involving a 'considerable number of simultaneously active processes, *whose interaction is its own simplest description.*' Too little computational understanding underlies too many proposals in the areas mentioned above, such as the input/output representation, the design of the network, or the method by which it will be constructed incrementally, and the selection of the training set.

Concerns about any (latent) empiricist leanings of connectionist research will re-emerge in chapter 6. No new answers to the old debate on 'nature' *versus* 'nurture' or 'architecture' *versus* 'absorption' are in evidence, although some of the mathematics developed to understand connectionist systems makes the trade-offs rather clearer.

Symbolic *versus* Sub-symbolic Processing

Even if the arguments above succeed in establishing an independent rôle for connectionist systems, this does not imply that symbolic processing is unnecessary in cognitive modelling. Indeed, substantial effort in the connectionist community is focused on the implementation of totally symbolic processing. Again there is confusion about the levels, because there is this possibility that exact or partial analogues of symbolic systems can be created using connectionist mechanisms, as well as the more heterogeneous links that can arise from the implementation of similar functions in different ways (see Foster for a more perspicuous account of this). Clark [26] pits against each other a hybrid account of his own (he calls it a 'rogue' account), the classical cascade downwards through Marr's levels that was critically examined above, and a connectionist 'dam' for the cascade due to Smolensky [143]. A further possibility will also be briefly described.

Smolensky's connectionist dam involves there being a relationship between classical and connectionist systems analogous to that between Newtonian and quantum mechanics. Phenomena at large scales can be described adequately using the techniques and concepts of Newtonian physics, but are really products of quantum scale interactions. These in turn have detectably different small scale properties. As the saying goes, what you see depends on how closely you look. In the Harmony theory [144] embodiment of Smolensky's view of connectionism, global optima correspond to logically complete and correct inference (within a finite domain). Crucially, the system is guaranteed to find such optima if one key parameter is reduced sufficiently slowly. If it is lowered more quickly, which is essential in all practical cases, then the guarantees of finding the overall optima, and consequently of doing correct and complete inference vanish, leaving only some approximation.

There are two troubles with this picture. Firstly that there should even be an approximation to the logical behaviour requires the severe constraints embodied in Harmony theory over and above just the annealing schedule on this parameter. These restrict the architecture, the functions performed by the units, and the learning rules. From the independence of the levels of analysis, there is no top-down pressure that can force compliance with these constraints. Secondly, the way he uses it gives too much to the classicists. In [143] he quotes the

> 'Best-Fit Principle:
> Given an input, a subsymbolic system outputs a set of inferences that, as a whole, give a best fit to the input, in a statistical sense defined by the statistical knowledge stored in the system's connections.' [143]-Manuscript p26

which fits very closely with the view on statistical computational theories developed below. However, in his example of qualitative electrical circuit theory [128] the ideal behaviour is describable purely logically and the statistical capacity is idle. Also, the semantics of the local optima are unclear. This cannot generalise to other intuitive cases, which is what this part of his account is aimed at; unlike the case in Physics, the "Newtonian" behaviour is mysterious too. There is just no set of logical rules that would be followed in some ideal circumstance. In a slightly different context, that of the communication between agents with bounded rationality, Cherniak [23] provides an extensive discussion of limited rationality and how once a logical edifice starts to crumble, *eg* by being intractable, it collapses almost completely. More recent work on connectionist grammar (unaccusativity in French) by Smolensky and his colleagues [83, 84] is essentially an exercise in function fitting, and also neglects learning.

Clark [26] offers a less stark choice. He imagines an architecture in which an exact but slow and serial classical processor trains, as it operates, some form of inexact but quick and parallel connectionist system. This system is responsible for almost all cognitive processing, except when it faces a problem that it cannot immediately solve.[4] As an example of this, consider the garden-path sentence `'The horse raced past the barn fell'`, which would give the connectionist processor pause. At this point, the classical processor, which normally just crunches input idly, intervenes, sorts out the resulting confusion, and possibly trains the connectionist system to avoid its error in the future.

[4] There is an issue here about second order ignorance – whether or not the system can know that it doesn't know an answer. Extra information is available in iterative systems such as the time it takes them to settle.

Although this is an attractive way of looking at the transition from amateur to expert knowledge, and certainly offers succour to both classicists and connectionists, it is not notably parsimonious. Also, it again avoids the main issue of which forms of classical processing are actually feasible, and how the interaction between symbolic and subsymbolic might work. An interesting issue that Clark does not discuss in detail, is how the systems might communicate and/or share a common representation.

In any case, concentrating on how completely classical processing can be achieved, or at least approximately achieved, in connectionist systems is the less interesting half of the story. It presupposes competent classical accounts, which are at best thin on the ground. There are alternatives, such as [110], which looks at the combination of a classical (*eg* PROLOG-like) inference engine with a connectionist memory machine. There, control is shared between the memory, which retrieves the rules and facts for the inference engine based on its current internal context, and the inference engine, which applies them to create new facts, which change the internal context, and therefore the rules that will subsequently be retrieved. This is non-committal on the implementation of the inference engine, but still results in a system that is unlikely to have a natural description in traditional terms. Chapter 6 discusses it in more detail.

Altogether, connectionism should co-exist with more traditional approaches – each adopting its own best focus. With hindsight, it is difficult to credit the fuss.

1.3 Statistical Computational Theories

Even if connectionism does co-exist with traditional systems, it may require, or indeed permit, different types of analysis from them. There are two forms of analysis of the computational tools, referred to above as the 'building blocks', which are used in the construction of complete systems. One looks just at their generic properties, and deduces generalisations about the systems they can comprise. The other considers the process of construction itself. Although this last is of as vital interest to the classical as to the connectionist community, in such areas as knowledge engineering or database update as well as learning, its subtleties seem not to survive traditional analysis in terms of levels. Marr, and most of the other authors, are interested in describing the functioning of a processing system, and are less concerned about its ontogenetic and phylogenetic history, both of which may provide important clues. Those who do not know their history are condemned to repeat it. In traditional terms, there are two separate systems that can be analysed – the end-product processing system, and the learning system which produces this. These must be understood together.

For an intuitive feel, take the picture of a functioning computational system as a collection of embodied symbols pushing each other around on the basis of their shapes (the standard processing-based-only-on-syntax restriction).[5] Consider the complications to the analysis if one of the results of learning is that the embodiments of particular symbols change their shapes, and hence change the way they push each other around and so the computations they perform. At any single moment, one could take a snapshot of the system and construct a standard computational account for it (the analysis of the processing system above). However, this would ignore the regularities in the changes between these accounts during the process of learning. Incidentally, this picture is interesting on other grounds, such as how collections of symbols might progressively get grounded in the world [55] by co-operatively changing their shapes. Appreciation of the historical development of a system may make explaining it far easier.

Another issue this account brings into sharp focus is the traditional view on what might be termed the narrow content [39, 123] of machine states. Smith [142] describes a widely accepted position in his *Knowledge Representation Hypothesis:*

> 'Any mechanically embodied intelligent process will be comprised of structural ingredients that a) we as external observers naturally take to represent a propositional account of the knowledge that the overall process exhibits, and b) independent of such semantical attribution, play a formal but causal and essential role in engendering the behaviour that manifests that knowledge.'

[5] For concreteness at the expense of accuracy, a model of this might be the interactions of proteins, whose shapes define the reactions they catalyse. Since the reactions themselves typically produce new proteins, the whole process can be viewed in computational terms, as in the theory of replication.

In the occurrent descriptions of systems, the structural ingredients can indeed play their rôles independent of any semantical attribution. However, learning, which is sensitive to the relationship between a system and an environment, can determine the nature of this interaction by changing their causal and essential properties (their 'shapes' above).

Cognitive information processing is only sensible at all because there are regularities in the world worth computing over. However, this leaves a trade-off between encoding the regularities in the structure of the system, and getting it to learn them, on the basis of observation of the world, and/or interaction with it. Marr's interest was in early visual processing, where such learning as may happen (for instance during the formation of the connections between the various parts of the brain) is finished by the time the system functions in earnest, leaving a regular structure that is amenable to a direct computational account.

Much of the work in classical symbolic AI also considers supposedly isolated modules of high-level human cognitive behaviour. Competence in these is based on knowledge which is somehow extracted and fed, pre-digested, into the system. Here, the motivation for ignoring learning is less easily justified, partly because it is unclear that the methods of extraction and subsequent pre-digestion are bound to be correct and complete, and partly because these are precisely the domains in which humans are continually learning and improving. More strongly, this continued learning is one important facet of our competence in these domains that such models are bound to omit.

Of course, there is substantial interest in the traditional community in learning and induction. The work on statistical inference (*eg* [119]) fits very neatly into the framework discussed below, although the more symbolic work on induction seems bound to suffer from the severe context sensitivity problems highlighted in chapter 6. A fierce debate rages between the AI and connectionist learning communities as to whether the latter really does offer any mechanisms with powers over and above the former. In such areas as temporal difference, described in chapters 4 and 5, novel and powerful connectionist mechanisms are certainly apparent. How far this conclusion extends is unclear.

Golden [47, 48] pointed the way forward when he provided a unifying probabilistic characterisation for understanding the dynamical retrieval behaviour of a wide range of networks, including those satisfying the Cohen-Grossberg Lyapunov conditions [27], and multi-layer perceptrons under a particular stochastic interpretation.[6] He claimed that 'the primary orientation of [his] article is to provide a computational level of description following Marr [96], of a broad class of neural network models.' It essentially provides a mathematical (*ie* statistical) characterisation of the behaviour of the models, and in a more obscure fashion, their learning capabilities. Such an account is an embryonic statistical computational theory. On this description, connectionist mechanisms have particular statistical performance and learning capabilities, described at the computational level. Similarly, the regularities in the world to which they should become attuned also have statistical characterisation. If these match, then the particular mechanisms are adequate for the task.

That such analysis is possible is due to the particularly simple form of the computations performed by current connectionist systems. Similar conclusions can be derived about traditional statistical inference and learning engines. This approach judiciously mixes occurrent and historical analyses of the connectionist systems, generalising over the particular and unilluminating form of the end-product (*eg* that some hyperplane is at a rather than b) to its historical development. Notions such as the Vapnik-Cervonenkis (VC) dimension [157] of a learning system are applicable. This is a measure of how malleable it is to the training data. If a connectionist system with a low VC dimension manages to capture enough examples, then one can have some statistical confidence that it will generalise correctly, since it could not have been bent and twisted to fit just those particular inputs. Note however, the success of this whole approach and its equivalents depends on the nature of the statistical task faced by the networks – it will not have anything illuminating to say about standard backpropagation confronted with a one-shot learning task, outwith its statistical competence.

[6] Unfortunately this interpretation does not naturally extend to provide an understanding of generalisation, if the training set is generated according to some underlying distribution, or of the relationship between the architecture adopted and the resulting error. See Haussler [57] and Baum and Haussler [12] for an account of these.

Two points remain: inexactitude and representation. It is well known that most forms of complete and correct statistical inference are just as intractable as their symbolic counterparts – Bayesian inference is non-polynomially hard. This means that functioning connectionist systems cannot implement these exactly for other than toy problems, and must be restricted to some inexact and 'cut-down' forms. In this case, there will be a mis-match between the statistical computational level of the connectionist mechanism, and the statistical computational characterisation of the task it has to solve. The gap between desired and actual capabilities should be characterisable statistically. Not only will different types of connectionist system perform differently, but also non-connectionist methods of computation may be better at realising the statistical models of the domains, in which case the replicative autonomy discussed above would justify their adoption in artificial systems.

There are two notoriously difficult and closely related issues in connectionist representation; the codings used both for the inputs and outputs, and for the internal workings of the system. In some cases this is complicated by the question of how to draw the boundary between the system and its environment. For the robotic realists inspired by Brooks [19], who stress behaviour in the 'real' world, and decry simulation, the raw outputs of the available sensors are their inputs, and the raw actuators their outputs. However, different sensors have dramatically different properties. From what is known about early sensory processing, very complicated operations which may well be dependent on the particular sensory coding, are performed even before the information is fed to the higher, more plastic areas. As in the NETtalk example, where the pronunciation of a letter is based on its preceding and succeeding context, adopting a particular input/output coding can amount to an implicit theory about part of the regularity in the domain. If the requisite information is not present in the input (as it would not have been in NETtalk but for the context inputs), then the task is impossible. Although there are a number of specific solutions in specific domains, there are few general principles.

The internal aspect of representation is equally complex. A rough, and equally nebulous, equivalent of the distinction between process and data in traditional systems is that between mechanism and representation. Often, the same connectionist mechanism can be applied to many connectionist representations. Back-propagation, as a stochastic gradient descent minimising mechanism, is a good example of this. Two representations often used with back-propagation are suitably connected collections of sigmoidal hyperplane units (so-called multi-layer perceptrons), which calculate the distance of an input from a hyperplane and pass it through a sigmoid function, and radial basis functions, which calculate the distance of the input from some centre, weighted by some radius function. Albus' CMAC system [2] is another example.

Each representation is effectively a method of function approximation, as indeed are many non-connectionist representations, such as kd-trees [115], and each mechanism can be tailored to the particular function approximation scheme employed. Note how the new version of the levels is necessary to capture this completely. Analytical studies on statistical learning and generalisation theory, such as [12, 28, 57, 156, 157], give measures which determine how particular representations will perform given the regularities they are trying to capture. Some of the these are even distribution-free, and so apply to all possible environments; their results are correspondingly weaker.

As suggested above, the method of function approximation, and its associated tailoring of the mechanisms, are generally mathematical descriptions at the computational level of concreteness, which have multiple possible instantiations in different algorithms. The need for this can be seen even in Marr's own example of a cash register, where the account of the task would be very different, for instance, were each item to come with a bag containing the same number of marbles as it cost. It is hard to reconcile this with Marr's assignment of representation as an algorithmic rather than a computational level issue.

1.4 Introduction to the Thesis

So far this chapter has described a particular viewpoint on the relationship between connectionist and classical systems, and suggested how learning might be described in a consonant fashion with this. The possibilities of various levels of analysis, and of logical differences between understanding human cognitive phenomena and reproducing them on some machine, have given rise to ample confusions. Methodologically, connectionism has already led to new and interesting algorithms, even though, philosophically, these can then be reproduced in non-connectionist form. Also, since most connectionism is not much closer to neuroscience than symbolism, at other than a superficial level, no strong claim to biological relevance can bear much weight. Even if traditional symbol systems can be implemented using connectionist mechanisms, and indeed *vice versa*, this says nothing about the irrelevance of one or other doctrine.

Overall, systems have to respond to the regularities in their environments. In general it is unrealistic to suppose that these regularities can be introspected, so learning will be necessary for any that are not just pre-wired. Sampling the world during learning is inherently a statistical process, and so statistical theories must enter into the accounts of both descriptions of parts of the world, and the descriptions of how computational mechanisms can be suitably attuned to them. These are statistical computational theories. The remaining chapters of the thesis attempt to work within this framework.

Chapter 2 develops a statistical computational theory of a form of one-shot associative matrix memory, tightly coupled with one form of function approximation based on linear hyperplanes. The memory has real-valued synapses as a slightly more accurate idealisation of certain forms of learning than some previous work that has concentrated on binary synapses. Using a statistical criterion of the quality of recall, the signal to noise ratio, it is possible to characterise statistically how the memory will perform based on a strong condition on the nature of the associations it is required to learn. Issues of input/output coding are evident as in the conclusion that the sparser the patterns are, the more of them that can be stored.

The model used in this chapter has been incorrectly analysed in the past, and also a slightly different and somewhat less felicitous model has been fairly prevalent. Although asymptotically optimal learning rules are reminiscent of those suggested by the physiological phenomena of long term synaptic potentiation and depression, the model is not biologically plausible.

Chapter 3 looks at the slightly different case of reinforcement learning. Supervised learning (as in the previous chapter) requires rich information from the environment as to what a system should do (with which outputs it should respond to which inputs). In many circumstances, the environment is only able to criticise a system's response in an unspecific manner, without revealing the nature of its error. This is called reinforcement learning, using the terminology from behaviourist psychology. Note that in the case of just two possible outputs, the two forms of learning differ only in that the supervised learning is typically deterministic, whereas in reinforcement learning, there is typically a statistical relation which has to be sampled between output and reward.

Rich Sutton introduced an extra term into the equations for reinforcement learning systems, creating a learning rule which is very similar to the one found optimal in chapter 2. Although the original motivation for the term was essentially intuitive, statistical analysis of the performance of the algorithm reveals both how it might work, and suggests a possible improvement. Simulations confirm the efficacy of the new term, and demonstrate that it performs better than the old one in some tasks, and, statistically, is never significantly worse.

Chapter 4 looks at reinforcement learning in temporally extended tasks. It considers how learning to make accurate predictions of performance in certain problems might aid learning the optimal controls, and focuses on Sutton's temporal difference (TD) method of learning the predictions in the first place. TD is designed to handle regularities over extended time intervals. Sutton has proved the statistical convergence of the algorithm to the correct predictions for a special case of the method which involves only one-step time dependencies and for a particular representational form. Chris Watkins identified the close connections between TD and dynamical programming, and this chapter applies the resulting understanding of the workings of TD to prove its statistical convergence in the general case involving arbitrary time dependencies and the same representation. Another of Watkins' results is used to show a stronger form of convergence in a further special case.

The interplay between the TD method and the various representations is particularly evident in this chapter, as the same mechanism, which is not connectionist, can be applied to a number of different function approximation schemes, which need not be connectionist either. Proving convergence naturally involves the properties of both the mechanism and the function approximator.

Chapter 5 applies the TD prediction-for-control approach to the task of navigating around a small grid. Part of the intent is to test whether the statistical characterisation of the problem maps the capabilities of the TD mechanism. The method is shown to be robust to different representations of the environment, so long as they are not ambiguous. Also, latent learning in the absence of any primary reinforcement is accommodated through the self-supervised learning of appropriate representations. Again this interplay between representation and mechanism is evident, as the various different representations result in demonstrably different speeds of learning. The results for the grid task are evaluated against those from a similar problem presented to rats – the open field water maze – and certain suggestions about rat cognitive mapping are reviewed.

The motivation for incorporating learning in the manner suggested has not yet been provided. Chapter 6 considers why learning might be important through a consideration of the benefits and drawbacks of context sensitivity in inference, and tells a fanciful story about a system that combines both classical and non-classical components.

Finally, the key elements of the tale of two paradigms are related in the coda – in a more historical setting.

Hades awaits – nascent potholes pepper this work. A prime example is in Watkins' identification that the underlying functioning of the TD mechanism is essentially dynamical programming. Dynamical programming is a totally general technique for planning, which, like all other similarly general techniques, including AI's, suffers from its own version of exponential intractability. Richard Bellman identified this in the present context as the curse of dimensionality. Increasing the number of dimensions in the state space increases the number of unique states exponentially (assuming fixed accuracy); calculating the appropriate actions for each state is then equally exponentially intractable. Using sophisticated function representation, effectively allowing variable accuracy, is one approach, but it is only well founded relative to strong assumptions about the nature of the prediction and control spaces. Another approach is to use some form of hierarchical structure, in which the dynamical programming has to solve a collection of small problems, rather than one large one. Again, this requires substantial knowledge about the problem. Connectionism is not notably forthcoming about how this meta-level information might be acquired.

For all the current distance of the armchair goals expressed in the previous sections, these chapters indicate that steps along the way are feasible. For simple non-dynamical systems such as these, complete statistical computational theories beckon.

Chapter 2

Optimal Plasticity

Maths best.

2.0 Summary

A statistical computational theory can be constructed for the traditional model of connectionist associative memory [167, 168]. A simple extension to this model is to allow real valued synapses. Under these circumstances, the learning rule that optimises the signal/noise ratio, which is a measure of the potential fidelity of recall, turns out to have a covariance form. Two other learning rules, which bear loose similarities to ones proposed in the neurophysiology literature, are asymptotically optimal in the limit of sparse coding. All three have the automatic property that the expected value of a single synapse is zero.

The results appear to contradict a line of reasoning particularly prevalent in the physics community. In fact, it turns out that the apparent conflict is due to the adoption of different underlying models. Ironically, they perform identically at their co-incident optima.

2.1 Introduction

The previous chapter concluded that part of the task of understanding a connectionist mechanism is constructing a statistical computational theory for it – a statistical theory of what it does faced with which inputs. Unfortunately, the state of statistical knowledge does not extend this far for general systems, rather derailing the overall programme. However, associative memories have long been ripe for such theorising, and there is a distinguished tradition ([170, 93] to name just two) along these lines. This chapter extends the analysis to a more general case.

More utilitarian motivation comes from trying to understand the theoretical implications of certain biological results on synaptic plasticity. For instance, the immense body of work on the neurophysiology of synaptic plasticity is severely tantalising the theoreticians. The incidence of long term potentiation (LTP) [15] in the hippocampus and neocortex is generally thought to support the Hebb hypothesis [60] about the facilitation of synapses due to coincident pre- and post-synaptic activity. However, even on theoretical grounds, it is clear that there also has to be some mechanism for reducing their efficacies, and the more recent discovery of long term depression (LTD) [145] points to this. There are various hypotheses about how LTD might work, and the intent of this chapter is to analyse the consequence of some of these, based on a highly simplified account.

A slight relaxation of the standard Willshaw associative network [170, 167] will be adopted. As well as having binary inputs and outputs, the Willshaw net also has binary synapses, which, once blown (set to be 'on'), are blown forever. Although this is a severe restriction, it actually leads to very efficient information storage. The obvious relaxation of the constraints is to allow real-valued rather than binary valued synapses, but to retain the binary input and outputs to make the analysis tractable. The most straightforward learning rule for these synapses is then a linear one, in which the contributions from each association are just summed. This removes any dependency of the synaptic values on the order in which the associations are presented, and is a further significant simplification.

For the saturating (two-state) synapses in the Willshaw net, it is hard to justify any learning rule other than the Hebb-like one which blows them on the conjunction of pre-synaptic and post-synaptic activity. In the linear case, though, there is no such intuition. Ignoring the rôle of time, there are four possible conjunctions of activity or quiescence on the input and output fibres, and, in principle, the efficacy of the synapse linking them could change by a different amount for each of these. These four numbers define a learning rule. The obvious questions are which rule is optimal, and how far from the optimum are other interesting possibilities.

However, determining the optimal learning rule requires some way of judging the quality of the unit. One such metric is the signal/noise ratio (S/N), which has its roots in engineering and has proved useful in a large number of applications. Consider a single unit that is to discriminate between two classes of outputs, the 'lows' and the 'highs', based on a scalar 'return', the *dendritic sum*. If the distributions for the two classes are both approximately Gaussian,[1] $\mathcal{G}(\mu_l, \sigma_l^2)$ and $\mathcal{G}(\mu_h, \sigma_h^2)$, say, then it will be easy to separate the two classes if the signal, $\mu_h - \mu_l$, is large (informally, if the peaks are far apart) and/or if the two contributions to the noise, σ_l^2 and σ_h^2, are small (informally, if the peaks are very narrow). Figure 2.1 shows the two distributions. The S/N is defined as:

$$\rho \equiv \frac{(\mu_h - \mu_l)^2}{\frac{1}{2}(\sigma_l^2 + \sigma_h^2)}. \tag{2.1}$$

and so incorporates both these effects. Maximising the S/N should enhance separability.

Note that the S/N is entirely independent of any threshold θ the unit might actually set to make the discrimination. This is desirable, since it factors out an issue which typically arises that one of the classes will occur more frequently than the other. Such imbalance might happen, for instance, if the output patterns are sparsely coded having many more lows than highs. Then, it may be more important to set θ either to preserve the few of the latter, or to make fewer errors by getting the bulk of the former correct. If high and low patterns occur with equal frequency, then it is likely to be wise to set $\theta = (\mu_l + \mu_h)/2$. For very

[1] As they will be for real valued synapses, but are *not* for the rather anomalous Willshaw model [167].

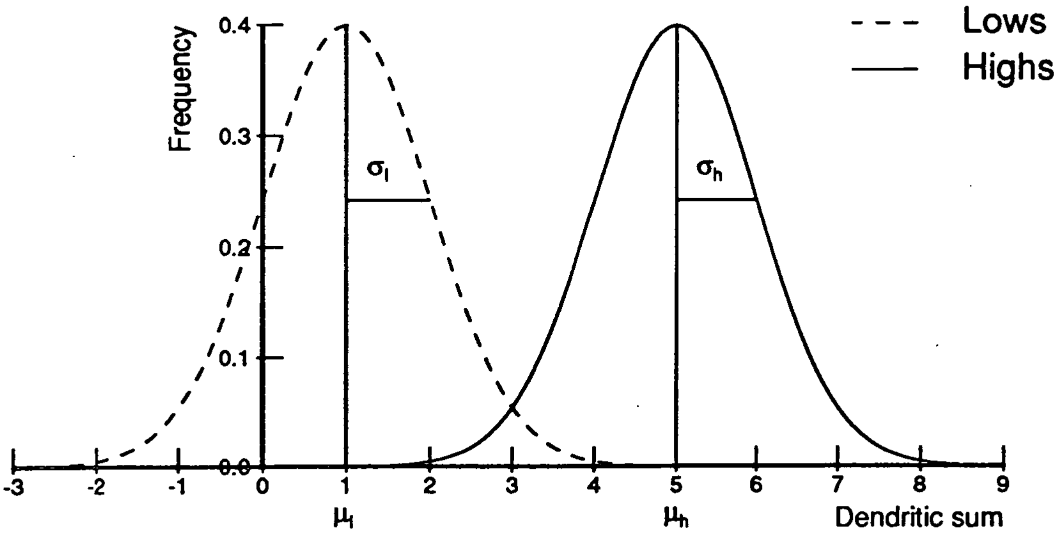


Figure 2.1: Distributions of 'Low' and 'High' dendritic sums

large systems, the limit studied in the physics community, the classes will either be perfectly discriminated or perfectly confused, so the threshold is essentially irrelevant.

In general, the S/N will be some function of both the learning rule and the input and output patterns. *Pace* a strong statistical assumption about these patterns, it is possible to work out a theoretical value for the S/N, and to optimise it with respect to the learning rule. It turns out that care is necessary over exactly how the S/N is defined. At least one incorrect and two different correct values for it are quoted in the literature.

The next section describes the model due to Palm [116, 117], section 2.3 demonstrates how each of the three possible expressions for the S/N and associated optimal rules, arise, and section 2.4 discusses their properties. Section 2.5 considers how thresholds might be set within the framework of this chapter, section 2.6 compares the results with those current in the physics community, and section 2.7 locates other, related, work in this area.

2.2 The Model

The underlying model is due to Palm [116, 117], who developed it from the original Willshaw binary, associative, memory in [170, 167]. A matrix memory, of the form shown in figure 2.2, is intended to store Ω associations, indexed by ω, between the patterns, each component of which can take one of two values:

$$\begin{aligned} a_i(\omega) &\in \{c, 1\}, & i &= 1\ldots m, \quad c \in \Re, \text{ and} \\ b_j(\omega) &\in \{l, h\}, & j &= 1\ldots n. \end{aligned}$$

This is called the $\{c, 1\}$ model for the low and high values of the input respectively. All the patterns are statistically independent and within each set are distributed identically, with probabilities:

$$\begin{aligned} p &= \mathcal{P}[a_i = 1], & 1 - p &= \mathcal{P}[a_i = c], \\ r &= \mathcal{P}[b_j = h], & 1 - r &= \mathcal{P}[b_j = l]. \end{aligned}$$

Patterns for which $b_j = l(h)$ will be called low (high). For pattern ω^*, define

$\#_c(\omega^*)$ as the number of $i \in [1, m]$ for which $a_i(\omega^*) = c$, and
$\#_1(\omega^*)$ as the number of $i \in [1, m]$ for which $a_i(\omega^*) = 1$.

The j^{th} unit has synaptic weights, or efficacies, $e_{ij} \in \Re$, $i = 1 \ldots m$, and consequent dendritic sum output in response to pattern ω^* input:

$$d_j(\omega^*) = \sum_{i=1}^{m} e_{ij} a_i(\omega^*). \tag{2.2}$$

The synaptic efficacies are set by the learning rule as:

$$e_{ij} = \sum_{\omega=1}^{\Omega} \Delta_{ij}(\omega),$$

where $\Delta_{ij}(\omega)$ is given in table 2.1. The linear dependency on the associations learnt is clear. Any more interesting case, for instance where the synaptic elements saturate, is more difficult to analyse because the effect of a particular association can depend on when it is learnt.

Hence, from equation (2.2),

$$d_j(\omega^*) = \sum_{i=1}^{m} a_i(\omega^*) \left[\sum_{\omega=1}^{\Omega} \Delta_{ij}(\omega) \right]. \tag{2.3}$$

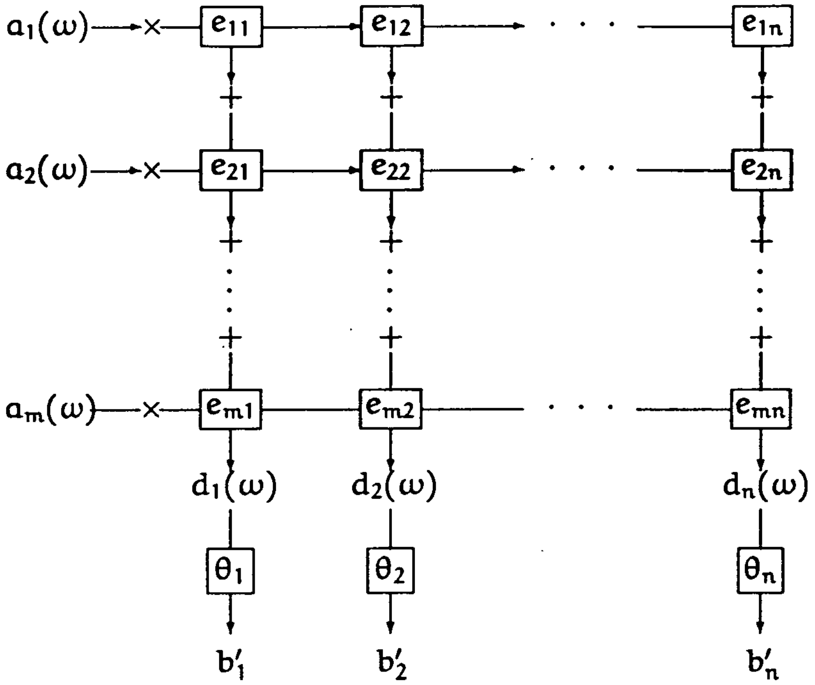


Figure 2.2: The matrix shows the steps taken in the retrieval of the pattern $b(\omega)$ that was previously stored in association with $a(\omega)$. For good recall, the calculated output b', the result of thresholding the dendritic sum output by θ, should closely resemble the desired output $b(\omega)$.

$\Delta_{ij}(\omega)$		Output $b_j(\omega)$	
		low	high
Input	c	α	β
$a_i(\omega)$	1	γ	δ

Table 2.1: Local synaptic learning rule.

Since the learning rule is local, each unit learns separately. The following discussion concerns only one such unit, and the subscript j will be dropped.

For the given pattern ω^*, equation (2.3) can be separated into two parts:

$$d(\omega^*) = \sum_{i=1}^{m} a_i(\omega^*)\Delta_i(\omega^*) + \sum_{i=1}^{m} a_i(\omega^*)\left[\sum_{\omega=1,\omega\neq\omega^*}^{\Omega} \Delta_i(\omega)\right]. \qquad (2.4)$$

The first of these terms, $S(\omega^*)$, determines the *signal* for pattern ω^*, and the second, $N(\omega^*)$, determines the *noise.*

The central limit theorem implies that the dendritic sums $d(\omega^*)$ for both classes of patterns (those to which the unit should respond low and high) will be approximately Gaussian. Figure 2.1 above gives a possible frequency graph showing the distribution of the dendritic sums. The two peaks, corresponding to the two classes, are clearly evident, as is the fact that there is no threshold θ that would not result in either errors of commission or errors of omission, or both. To see this last point, observe that no vertical line could be drawn that entirely separates the two peaks. As discussed in the introduction, the signal/noise ratio (S/N), defined in equation 2.1, is a measure of the average potential fidelity of recall for the unit.

Note also that c, the value contributed by a 'low' input, is a parameter of the system. This is to allow an evaluation of certain claims being made about how dramatically the $\{0,1\}$ model (*ie* $c = 0$) outperforms the $\{-1,1\}$ model (*ie* $c = -1$) for sparse patterns. *A priori,* this seems unlikely, since there is a formal equivalence between these two models. To see this, consider $a_i(\omega^*) = \lambda\hat{a}_i(\omega^*) + \mu$, where $\hat{a}_i(\omega^*) \in \{0,1\}$ are the 'canonical' inputs for a pattern. By varying λ and μ it is possible to generate any of these models–*eg* $\mu = c, \lambda = 1-c$, gives the $\{c,1\}$ model. Then

$$\begin{aligned} d(\omega^*) &= \sum_{i=1}^{m} e_i a_i(\omega^*) \\ &= \lambda\left\{\sum_{i=1}^{m} e_i \hat{a}_i(\omega^*)\right\} + \mu\left\{\sum_{i=1}^{m} e_i\right\}. \end{aligned} \qquad (2.5)$$

where e_i are the unit's weights. The μ term is purely additive, and so cannot affect the S/N. The λ term is multiplicative, but expanding the size of the

Gaussian curves for both classes also fails to change the S/N. To see this note that although the distance between the means goes up by the multiplicative factor, so does the breadth of each of the curves. Operationally, for any given value of θ for, say, the $\{0, 1\}$ model, there is some other threshold for, say, the $\{-1, 1\}$ model, which allows the unit to make *identical* errors. Changing c is essentially a formal step. The apparent contradiction between these results and those enjoying currency in the physics community will be explored in section 2.6.

2.3 The Ugly, the Bad and the Good

Under normal circumstances, calculating ρ, the S/N, is fairly straightforward. Having separated the dendritic sum for both low and high patterns into *Signal* + *Noise*, as in equation 2.4, the numerator for ρ (see equation 2.1) would be the expected difference between the signals for the two classes:

$$\mu_h - \mu_l = \mathcal{E}_h[S(\omega^*)] - \mathcal{E}_l[S(\omega^*)],$$

and the denominator would be the average variance of their noises

$$\frac{1}{2}(\sigma_l^2 + \sigma_h^2) = \frac{1}{2}(\mathcal{V}_l[N(\omega^*)] + \mathcal{V}_h[N(\omega^*)]).$$

where $\mathcal{E}_h$ implies that the expectation is taken over those patterns for which the output $b(\omega^*) = h$, and similarly for $\mathcal{V}_h$.

This amounts to making two assumptions:

Expectation of the noise: that $\mathcal{E}_h[N(\omega^*)] = \mathcal{E}_l[N(\omega^*)]$;

Variance of the noise: that it is this quantity rather than some other measure of the spread of the dendritic sums that determines the ability of the unit to perform its discrimination accurately.

Neither assumption is true, although it is possible to reconstruct results in the literature on the S/N using either or both of them. The following three subsections demonstrate the effects of accepting and rejecting them.

2.3.1 The Ugly

Consider a high pattern, ω_h. The signal $S(\omega_h)$ is the contribution due to terms $\Delta_i(\omega_h)$ for all the input lines and so:

$$S(\omega_h) = \delta\#_1(\omega_h) + c\beta\#_c(\omega_h)$$

The expectation value of the signal is therefore

$$\mathcal{E}_h[S(\omega_h)] = m[p\delta + (1-p)c\beta].$$

Similarly for a signal ω_l for which the value of the unit should be l,

$$\mathcal{E}_l[S(\omega_l)] = m[p\gamma + (1-p)c\alpha]$$

Assuming that the expectation of the noise is the same for high and low cases,

$$\mu_h - \mu_l = m[p(\delta-\gamma) + (1-p)c(\beta-\alpha)]. \tag{2.6}$$

For calculating the noise, there is a lemma that if:

$$\Gamma = \begin{cases} \Phi & \text{with probability } a, \\ \Psi & \text{with probability } 1-a, \end{cases}$$

where Φ and Ψ are random variables, then the variance $\mathcal{V}$ of Γ is

$$\mathcal{V}[\Gamma] = a\mathcal{V}[\Phi] + (1-a)\mathcal{V}[\Psi] + a(1-a)(\mathcal{E}[\Phi] - \mathcal{E}[\Psi])^2. \tag{2.7}$$

Now consider in equation 2.4, the inner sum in the noise term:

$$\sum_{\omega=1,\omega\neq\omega^*}^{\Omega-1} \Delta_i(\omega)$$

This is made up from contributions from each of $\Omega - 1$ patterns, where, for each pattern,

$$\Delta_i(\omega) = \begin{cases} \begin{cases} \delta & \text{with probability } r, \\ \gamma & \text{with probability } 1-r, \end{cases} & \text{with probability } p, \\ \\ \begin{cases} \beta & \text{with probability } r, \\ \alpha & \text{with probability } 1-r, \end{cases} & \text{with probability } 1-p. \end{cases}$$

$\Delta_i(\omega)$		Output $b(\omega)$	
		low	high
Input	c	$f - g(1 - \tau + \tau c)$	$f - g(1 - \tau)(1 - c)$
$a_i(\omega)$	1	$f - g$	f

Table 2.2: Optimal $\hat{\rho}_1$ local synaptic learning rule.

Applying equation 2.7 twice, the variance of $\Delta_i(\omega)$ is:

$$\begin{aligned}\mathcal{V}[\Delta_i(\omega)] &= p[\tau(1-\tau)(\delta-\gamma)^2] + (1-p)[\tau(1-\tau)(\beta-\alpha)^2] + \\ &\quad p(1-p)[\tau\delta + (1-\tau)\gamma - \tau\beta - (1-\tau)\alpha]^2. \end{aligned} \qquad (2.8)$$

Equation 2.4 involves the sum of $\#_c(\omega^*)$ copies weighted by c and $\#_1(\omega^*)$ copies weighted by 1. Under the apparently plausible assumption of independence between $a_i(\omega^*)$ and $\sum_{\omega=1,\omega\neq\omega^*}^{\Omega-1} \Delta_i(\omega)$, over all the patterns,

$$\mathcal{V}[N(\omega^*)] = (\Omega - 1)(c^2\#_c(\omega^*) + \#_1(\omega^*))\mathcal{V}[\Delta_i(\omega)], \qquad (2.9)$$

and making the assumption that the variance for each pattern can be averaged over all patterns ω^* would produce

$$\begin{aligned}\sigma_h^2 &= \sigma_l^2 \\ &= m(\Omega-1)[p + c^2(1-p)]\mathcal{V}[\Delta_i(\omega)] \\ &= (\Omega-1)[p + c^2(1-p)]\tau(1-\tau)\times \\ &\left\{p(\delta-\gamma)^2 + (1-p)(\beta-\alpha)^2 + \tfrac{p(1-p)}{\tau(1-\tau)}[\tau(\delta-\gamma) - \tau(\beta-\alpha) + (\gamma-\alpha)]^2\right\}.\end{aligned} \qquad (2.10)$$

The S/N, ρ, can now be calculated from expressions 2.6 and 2.10. Maximising it with respect to α, β, γ and δ determines the conditions for an optimum.

Table 2.2 sets out the consequent rule, where δ has been arbitrarily set to f and β to $f - g$. The optimal S/N is:

$$\hat{\rho}_1 = \frac{m}{\Omega - 1}\frac{1}{\tau}\frac{1}{1-\tau}$$

which, oddly enough, is correct in the general case, as shown later.

For $p = r$, one of the special cases of the rule is the one quoted by Palm [117],

$$\begin{aligned} \alpha &= cp & \beta &= -c(1-p) \\ \gamma &= -p & \delta &= 1-p. \end{aligned}$$

Palm [117] also gives the S/Ns for two rules which are not, in general, instances of the optima. They are:

The Hopfield rule: $\rho_1^{\text{Hopfield}} = \frac{m}{\Omega - 1}\frac{1}{2p(1-p)}\frac{1}{1-2p(1-p)}$

$$\begin{aligned} \alpha &= 1 & \beta &= -1 \\ \gamma &= -1 & \delta &= 1 \end{aligned} \quad p = r,\ c = -1$$

(This is optimal for $p = r = 1/2$):

The Hebb rule: $\rho_1^{\text{Hebb}} = \frac{m}{\Omega - 1}\frac{1}{p}\frac{1}{1-p^2}$

$$\begin{aligned} \alpha &= 0 & \beta &= 0 \\ \gamma &= 0 & \delta &= 1 \end{aligned} \quad p = r,\ c = 0$$

Although these results are identical to those in [117], it remains unclear to what extent this derivation, and the general expression for the S/N, mirrors that of Palm.

That something is amiss may be appreciated by considering the behaviour of ρ_1^{Hopfield} as $p = r \rightarrow 0$. One might expect that the S/N should decrease under these circumstances, since the learning rule is incorrectly symmetrical in $a(\omega)$. However, ρ_1^{Hopfield} actually increases. Simulations confirm this point; table 2.3 shows theoretical and empirical values of ρ_1^{Hopfield} and ρ_1^{Hebb} for various values of p.[2] It is apparent both that the simulations diverge substantially from the theoretical expression, ρ_1, and that the Hopfield rule does indeed get worse for smaller p. The Hebb rule is not optimal for any values of p or r, but it is asymptotically optimal for sparse patterns, as $p = r \rightarrow 0$.

Interestingly, Palm actually makes the assumption from the very outset that the S/N will be unaffected if all of α, β, γ, and δ are multiplied by the same

[2] For this and the other simulations in this chapter, the $n = 20$ units have $m = 512$ input lines and are fed $\Omega = 200$ patterns. Figures are averages over 50 runs. The observed values of the variance are based on a unit-by-unit calculation.

Hebb rule

		Predicted S/N			Actual	
p,r	c	ρ_1	ρ_2	ρ_3	S/N	$\pm\,\sigma$
0.5	0	6.9	1.7	0.050	0.10	± 0.11
0.4	0	7.7	2.8	0.12	0.11	± 0.090
0.3	0	9.4	4.6	0.32	0.34	± 0.15
0.2	0	13	8.6	1.1	1.2	± 0.47
0.1	0	26	21	7.7	7.1	± 1.0
0.05	0	52	47	32	28	± 18

Hopfield rule

		Predicted S/N			Actual	
p,r	c	ρ_1	ρ_2	ρ_3	S/N	$\pm\sigma$
0.5	0.5	1.0	1.0	10	11	± 1.3
0.5	0	5.1	5.1	10	11	± 1.3
0.5	−0.5	9.3	9.3	10	11	± 1.3
0.5	−1	10	10	10	11	± 1.3
0.4	−1	10	9.5	7.5	8.3	± 1.5
0.3	−1	11	7.5	1.4	1.3	± 0.40
0.2	−1	12	4.8	0.25	0.32	± 0.22

Table 2.3: Theoretical and empirical values of the S/N for the Hebb and Hopfield rules.

non-zero number, or if the same number is added to them all. *A priori,* and, as indeed is borne out by simulations, the last invariance is most unlikely to hold. If a large enough quantity is added to each element in the rule such that all the weight values are large and positive, then the signal which determines the classification of a particular pattern as low or high is likely to be entirely swamped by the noise due to the uncertainty in the number of the inputs that are c or 1. Palm uses this incorrect assumption to reduce the number of free variables on which the learning rule depends.

2.3.2 The Bad

The first assumption given above was that the expected values of the noise obscuring high and low patterns are the same. This is not true, and so the difference between the expected value of $d(\omega^*)$ for high and low patterns cannot be taken to be equal to the difference between the expected value of the signal $S(\omega^*)$ in the two cases. In equation 2.4, the noise term

$$N(\omega^*) = \sum_{i=1}^{m} a_i(\omega^*) \left[\sum_{\omega=1,\omega\neq\omega^*}^{\Omega-1} \Delta_i(\omega) \right]$$

excludes pattern ω^*, and there is a difference between excluding a pattern for which $b(\omega^*) = h$ and one for which $b(\omega^*) = l$. If $\mathcal{N}_h$ patterns have $b(\omega^*) = h$ and $\mathcal{N}_l$ have $b(\omega^*) = l$, so $\mathcal{E}[\mathcal{N}_h] = \Omega\tau$ and $\mathcal{E}[\mathcal{N}_l] = \Omega(1-\tau)$, then:

$$\begin{aligned} \mathcal{E}_h[N(\omega^*)] &= m\mathcal{E}_h[a_i(\omega^*)]\mathcal{E}[(\mathcal{N}_h - 1)(p\delta + (1-p)\beta) + \mathcal{N}_l(p\gamma + (1-p)\alpha)], \\ \mathcal{E}_l[N(\omega^*)] &= m\mathcal{E}_l[a_i(\omega^*)]\mathcal{E}[\mathcal{N}_h(p\delta + (1-p)\beta) + (\mathcal{N}_l - 1)(p\gamma + (1-p)\alpha)]. \end{aligned}$$

and therefore:

$$\mathcal{E}_h[N(\omega^*)] - \mathcal{E}_l[N(\omega^*)] = -m[p + c(1-p)][p\delta + (1-p)\beta - p\gamma - (1-p)\alpha].$$

Using this contribution to amend the expression for $\mu_h - \mu_l$ in equation 2.6 yields

$$\begin{aligned} \mu_h - \mu_l &= m[\ p(\delta-\gamma) + c(1-p)(\beta-\alpha) - \\ &\qquad (p + c(1-p))(p\delta + (1-p)\beta - p\gamma - (1-p)\alpha)] \qquad (2.11) \\ &= mp(1-p)(1-c)[(\delta-\gamma) - (\beta-\alpha)]. \end{aligned}$$

$\Delta_i(\omega)$		Output $b(\omega)$	
		low	high
Input	c	$h - g\frac{1-p-r}{1-p}$	$h - g\frac{1-r}{1-p}$
$a_i(\omega)$	1	$h - g$	h

Table 2.4: Optimal $\hat{\rho}_2$ local synaptic learning rule.

Using 2.11 and the old expression 2.10 for the noise gives:

$$\rho_2 = \xi \frac{p^2(1-p)^2(1-c)^2[(\delta-\gamma)-(\beta-\alpha)]^2}{p(\delta-\gamma)^2+(1-p)(\beta-\alpha)^2+\frac{p(1-p)}{r(1-r)}[r(\delta-\gamma)-r(\beta-\alpha)+(\gamma-\alpha)]^2}, \tag{2.12}$$

where $\xi = \frac{m}{\Omega - 1}\frac{1}{(p + c^2(1-p))r(1-r)}$.

Maximising this with respect to α, β, γ and δ gives the optimal rule shown in Table 2.4, where, for comparison, $\delta = h$ and $\gamma = h - g$. The optimal S/N is now:

$$\begin{aligned} \hat{\rho}_2 &= \frac{m}{\Omega - 1}\frac{1}{r(1-r)}\frac{p(1-p)(1-c)^2}{p + c^2(1-p)} \\ &= \hat{\rho}_1 \frac{p(1-p)(1-c)^2}{p + c^2(1-p)}. \end{aligned}$$

This derivation has removed the dependence on c of the learning rule, but leaves us free to maximise the S/N with respect to c. The maximum occurs at $\hat{c} = -p/(1-p)$, where the average value of each input is zero. Then, $\hat{\rho}_2 = \hat{\rho}_1$.

Not only is this rule somewhat inelegant, but it also violates two empirical principles outlined earlier; the S/N should actually be independent of c, the numerical value of a low input, and the rule should not be additively invariant, *ie* it should not be the case that any number can be added to the rule without affecting its S/N. Table 2.3 also compares the theoretical ρ_2 and actual S/Ns for the Hebb and Hopfield rules for various values of $p = r$. It is apparent that ρ_2 is indeed fallacious. Note again that the Hopfield rule is a special case of the optimum for $p = r = 1/2$ and $h = 1, g = 2$.

2.3.3 The Good

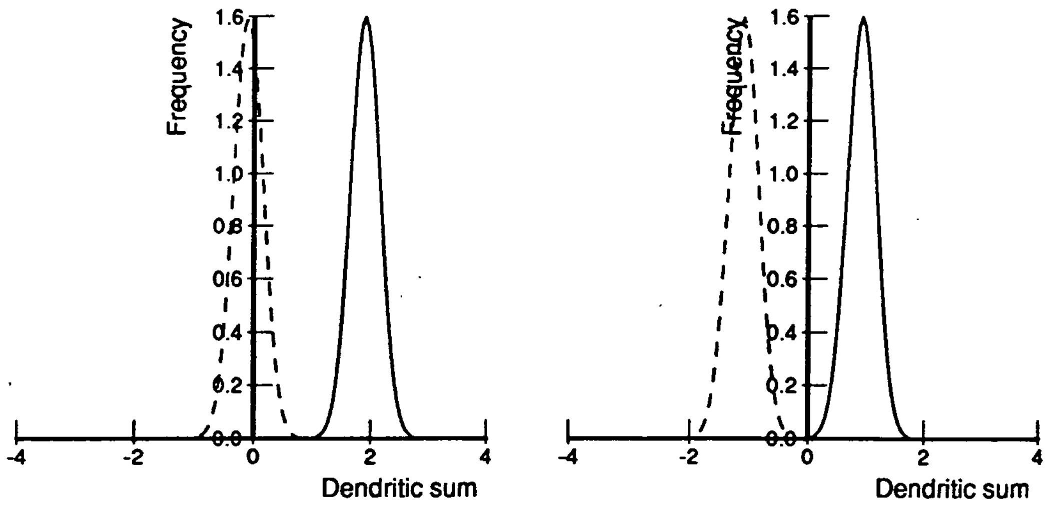
The first pointer to a resolution of these problems came from the simulations. There are two possible ways of calculating the mean dendritic low and high dendritic sums; either over the whole set of output units, or on an individual, output-unit by output-unit basis. The estimated sample variance will obviously depend on which of these is adopted, and should be higher for the first method than for the second. However, under the second assumption, that it is the variance of the noise that determines the theoretical discriminability, they would not differ in the limit of large numbers of inputs. Simulations confirmed that this was not the case.

It was then obvious that it is not enough to calculate the variances of the dendritic sums – the correlations between two dendritic sums are important too. The analysis based solely on the variance ignores the fact that the efficacies e_i are quenched, *ie* although they are determined during learning by the statistics of the patterns, they are fixed by the time of recall. Also, the units can take advantage of this by setting their thresholds independently, each according to its own quenched weights. The correlations in the dendritic sums come about because the synaptic efficacies are determined by the *actual* numbers of low and high patterns the units have learnt rather than just the *mean* numbers.

For instance, using the Hebb rule with $\{0, 1\}$ patterns, a unit that happens to have learnt a large number of high patterns will tend to have dendritic sums that are greater than those for a unit that happens to have learnt only a few. The variance analysis for ρ_2 just balances these cases out, whereas it is clear that the threshold for a unit of the first type will optimally be larger than the threshold for one of the second.

The following simple didactic example of the effects of correlation between noise terms demonstrates the class of phenomenon that occurs. Imagine that signals $\phi(t) \in \{-1, 1\}$ are corrupted by additive noise $\psi(t)$. There are two possible processes generating $\psi(t)$:

$$\begin{aligned} \psi_1 &\sim \mathcal{G}((1-\pi), \quad \sigma^2), \text{ and} \\ \psi_2 &\sim \mathcal{G}(-\pi, \qquad \sigma^2) \end{aligned}$$



Distribution under ψ_1 Distribution under ψ_2

Figure 2.3: Distributions under ψ_1 and ψ_2 for $\pi = 0.1$, $\sigma = 0.25$ – dotted lines the low signals, solid lines the high ones. Translation is the only difference.

where each collection is independent and identically distributed. It is not known before the experiment which process will generate the noise; all that is known is that

$$\begin{aligned} \pi &= \mathcal{P}[\psi \text{ is given by } \psi_1], \text{ and} \\ 1-\pi &= \mathcal{P}[\psi \text{ is given by } \psi_2]. \end{aligned}$$

Figure 2.3 demonstrates the two possibilities. Rather similarly to the effect of changing μ in the analysis of the rôle played by c (see equation 2.5), the only difference between the two cases is that the frequency graphs are shifted with respect to each other. The S/Ns are identical, and indeed an appropriate choice of threshold would result in no more and no fewer errors being made.

However, performing the formal analysis as for ρ_2 gives that

$$\begin{aligned} \mathcal{E}[\psi(t)] &= 0, \text{ and} \\ \mathcal{V}[\psi(t)] &= \pi(\sigma^2 + (1-\pi)^2) + (1-\pi)(\sigma^2 + \pi^2) \\ &= \sigma^2 + \pi(1-\pi). \end{aligned}$$

But this is clearly an overestimate of the 'operative variance', which is here defined as the expected dispersion of the corrupted signals about their *actual* means, rather than their *expected* means. So long as the unit can set its own threshold according to which of ψ_1 and ψ_2 occurred, this is the appropriate quantity to calculate, being the factor that disposes it to err. Its value is obviously σ^2, the individual variance of both ψ_1 and ψ_2.

In the simple example, the noise terms are correlated, because one choice (based on the probability π) determines the distributions for them all. Ignoring this, by calculating the true variance rather than the dispersion of the corrupted signals, leads to an incorrect measure of how well the unit will be able to do its job of discriminating between the two possible classes, $\phi(t) = -1$ and $\phi(t) = 1$.

In the case of the associative memory, this issue is slightly more complicated. Here, the distribution of the noise terms is also determined in advance of the operation of the unit as a discriminator, in this case by the quenched weight values that emerge from the particular set of input/output associations it learns. However, the effects of the noise are mediated through the actual $\{c, 1\}$ input values for the patterns. If

$$c = -\frac{p}{1-p}$$

then the expected value of any input is zero. This nullifies any effect from the differences between the actual efficacies of the synapses and their expected values, which are normally the cause of the whole problem. If c does not take this value, then there will be an effect due to the quenching, that will make the variance of the dendritic sums diverge from the dispersion. To re-iterate, it is the dispersion rather than the variance that determines the unit's ability to discriminate, and so it is the dispersion that is the appropriate measure for the S/N.

The mean dispersion is defined as:

$$s_h^2 = \mathcal{E}\left[\frac{1}{\mathcal{N}_h}\sum_{\{\omega|b(\omega)=h\}}[d(\omega)]^2 - \left(\frac{1}{\mathcal{N}_h}\sum_{\{\omega|b(\omega)=h\}}d(\omega)\right)^2\right], \tag{2.13}$$

$\mathcal{N}_h$ (and $\mathcal{N}_l$) being the number of ω for which $b(\omega) = h(l)$. s_l^2 is defined similarly as the expected dispersion for low patterns. Symbolically,

$$\texttt{Dispersion} = \texttt{Variance} - \texttt{Correlation},$$

and it is the interaction of the quenched weights with c that introduces the correlations.

Calculating the expected value of the dispersion explicitly by writing out the squares of the sums in equation 2.3 and taking expectations produces:

$$\begin{aligned} s_h^2 \simeq\ & mp(1-p)[(1-c)^2(1-2p)^2(\delta-\beta)^2 - 2p(1-p)(1-c)^2(\delta-\beta)^2] + \\ & mp(1-p)\mathcal{E}[p(1-p)(1-c)^2(\mathcal{N}_h(\delta-\beta)^2+\mathcal{N}_l(\gamma-\alpha)^2)] + \\ & mp(1-p)\mathcal{E}[2(1-c)^2(1-2p)(\delta-\beta)(\mathcal{N}_h\phi+\mathcal{N}_l\psi)] + \\ & mp(1-p)\mathcal{E}[(1-c)^2(\mathcal{N}_h\phi+\mathcal{N}_l\psi)^2]. \end{aligned} \tag{2.14}$$

where $\phi = p\delta + (1-p)\beta$ is the average contribution to the synaptic efficacy from a high pattern, and $\psi = p\gamma + (1-p)\alpha$ is the average contribution from a low one.

For large Ω, the last of these terms

$$\mathcal{E}[(\mathcal{N}_h\phi+\mathcal{N}_l\psi)^2] = \Omega r(1-r)(\phi-\psi)^2 + \Omega^2(r\phi+(1-r)\psi)^2 \tag{2.15}$$

will dominate the noise and swamp the signal unless $r\phi + (1-r)\psi = 0$. In practice this removes the additive degree of freedom in the rules for ρ_1 and ρ_2, ensuring that the average value of the efficacy of a synapse must be 0. The component that remains arises from the uncertainty in the values $\mathcal{N}_h$ and $\mathcal{N}_l$:

Ignoring the first terms in s_h^2, which are dominated by the terms in Ω and Ω^2, gives

$$\begin{aligned} s_h^2 &\sim s_l^2 \\ &\simeq m\Omega p(1-p)(1-c)^2[\ p(1-p)(r(\delta-\beta)^2+(1-r)(\gamma-\alpha)^2) + \\ &\qquad r(1-r)(\phi-\psi)^2 + \Omega(r\phi+(1-r)\psi)^2] \\ &= m\Omega p(1-p)(1-c)^2[\ r(1-r)(p(\delta-\gamma)^2+(1-p)(\beta-\alpha)^2) + \\ &\qquad p(1-p)(r\delta+(1-r)\gamma-r\beta-(1-r)\alpha)^2 + \\ &\qquad \Omega(r\phi+(1-r)\psi)^2] \end{aligned}$$

and so:

$$\rho_3 = \frac{m}{\Omega}\frac{p(1-p)[\delta-\gamma-\beta+\alpha]^2}{p(1-p)[r(\delta-\beta)^2+(1-r)(\gamma-\alpha)^2]+r(1-r)[\phi-\psi]^2+\Omega(r\phi+(1-r)\psi)^2}$$

Comparing the form of ρ_3 with that of ρ_2, it turns out that, excluding the term in Ω^2, they have the same dependence on α, β, γ and δ, but that the dependence on c has finally been excised.

$\Delta_i(\omega)$		Output $b(\omega)$ low	 high
Input	c	pr	$-p(1-r)$
$a_i(\omega)$	1	$-(1-p)r$	$(1-p)(1-r)$

Table 2.5: Optimal $\hat{\rho}_3$ local synaptic learning rule.

Maximising with respect to α, β, γ and δ, the optimal rule is just as for ρ_2 apart from the important constraint that $r\phi + (1-r)\psi = 0$. This gives one true optimum:

The Covariance rule **R1**:

$$\begin{array}{rclrcl} \alpha & = & pr & \beta & = & -p(1-r) \\ \gamma & = & -(1-p)r & \delta & = & (1-p)(1-r) \end{array} \qquad \rho_3^{\text{Covariance}} = \frac{m}{\Omega}\frac{1}{r}\frac{1}{1-r}$$

Two other sub-optimal rules have previously been proposed (see the next section for a discussion). As Alessandro Treves has pointed out (personal communication), our original classification in [171] of them as being locally optimal was incorrect. In fact, they are not even optimal under the additional condition that $\alpha = 0$. They are:

The Heterosynaptic rule **R2**:

$$\begin{array}{rclrcl} \alpha & = & 0 & \beta & = & -p \\ \gamma & = & 0 & \delta & = & 1-p \end{array} \qquad \rho_3^{\text{Hetero}} = \frac{m}{\Omega}\frac{1}{r}$$

The Homosynaptic rule **R3**:

$$\begin{array}{rclrcl} \alpha & = & 0 & \beta & = & 0 \\ \gamma & = & -r & \delta & = & 1-r \end{array} \qquad \rho_3^{\text{Homo}} = \frac{m}{\Omega}\frac{1}{r}\frac{1-p}{1-r}$$

Table 2.5 gives the covariance rule $\hat{\rho}_3$ for comparison with the others.

Table 2.3 shows the close agreement between the theoretical prediction, ρ_3, of the S/N and the empirical result for the Hebb and Hopfield rules. Table 2.6 shows the theoretical S/N for the optimal, sub-optimal, and the Hebb and Hopfield rules for various values of $p = r$, based on ρ_3. The Hopfield rule is optimal for $p = r = 1/2$, but rapidly tails off as the patterns get more sparse. Even though

	Signal/Noise Ratios for			
p,r	R1	R2, R3	Hebb	Hopfield
0.5	10	5.1	0.050	10
0.4	11	6.4	0.12	7.5
0.3	12	8.5	0.32	1.4
0.2	16	13	1.1	0.25
0.1	28	26	7.7	0.045
0.05	54	51	32	0.015

Table 2.6: Theoretical ρ_3 predictions of the S/N for the optimal (R1), sub-optimal (R2 and R3), Hebb, and Hopfield rules for various values of $p = r$. Note that R2 and R3 are very close to R1 as the sparsity increases, but the Hebb rule is significantly worse.

the Hebb rule is asymptotically optimal as p gets small, it is significantly worse than all three optima even for quite tiny but finite values.

2.4 The Optimal and Sub-Optimal Rules

The optimal and two sub-optimal rules in the previous section can be identified with ones suggested in various places in the literature. The covariance rule was originally proposed by Sejnowski [138, 139], and has since been widely used in connectionist systems. For instance, the Hopfield rule [167, 63] is a special case of it when $p = r = 1/2$. In fact, in the physics models, [155, 21, 120] discussed in section 2.6, it is taken as read. The motivation behind it is even clearer from the equivalent form

$$\Delta_i(\omega) \propto (a_i(\omega) - \bar{a})(b(\omega) - \bar{b}).$$

where $\bar{a}$ is the average value of an input $(p + (1 - p)c)$ and $\bar{b}$ is the average value of an output.

Note that if the environment were stochastic – *ie* the unit had to work out from a number of presentations of an input pattern with contradictory output information whether or not it should fire, then this term is no longer optimal. This case is treated in the next chapter. None of the ρ_3 rules described is biologically plausible, for reasons discussed below, but $\rho_3^{\mathrm{Covariance}}$ is particularly difficult to justify because $\alpha > 0$. α is the change in efficacy of a synapse in the absence of either pre- or post-synaptic activity. One could imagine some form of decay process, which would tend to eliminate unused synapses, but for the efficacy actually to rise is counterintuitive. α's 'rôle' is to keep the expected value of a synapse zero, which is the non-additivity condition that Palm ignored.

Various parts of the brain show synaptic plasticity, including the visual system (during development), the cerebellum and the hippocampus. Different underlying mechanisms are believed to be responsible – for instance the analogue of long term potentiation (LTP) in the hippocampus seems to be long term depression (LTD) in the cerebellum [66] – and the extent to which the plasticity is merely an artefact of the procedure is also in doubt. The hetero- and homo-synaptic rules are so called because of their similarities with the eponymous biological rules for LTD. Heterosynaptic LTD has been known about for some time in various parts of the brain, and a theoretical rule like this has been suggested by Stent [146], Singer [141], and others. The evidence for homosynaptic LTD in the hippocampus is rather more recent [145], and disputes remain about its reality and properties. Bienenstock, Cooper and Munro [14] made an early proposal along the lines of the homosynaptic rule for plasticity in the visual system.[3]

Both the hetero- and homo-synaptic rules perform worse than the covariance rule; ρ_3^{Hetero} by a factor $1-\tau$, and ρ_3^{Homo} by a factor $1-p$. However, since the regime in which any of the rules work well is where the patterns are sparse (*ie* p and τ are small), these factors are relatively small. The nervous system is known to employ sparse coding. For $p = \tau$, ρ_3^{Hetero} and ρ_3^{Homo} are equal. The homosynaptic rule has also been used for connectionist systems, such as Kanerva's sparse distributed memory (SDM) [72]. The original version of SDM only considers patterns with $\tau = 1/2$, and for it to be used optimally with different activity ratios, the analysis here would suggest that the equivalents of γ and δ ought to be suitably juggled.

[3] Note that the optimal rule under the additional condition that $\alpha = 0$ specifies decreases in efficacy under both hetero- and homo-synaptic conditions. The latter are an order of magnitude greater than the former for sparse patterns.

One notable feature of all the rules is that for sparse patterns, the absolute value of the increment δ is an order of magnitude larger than the decrements β or γ. If this were also true of the real rules, it would make LTD significantly more difficult to detect than LTP. This would require careful experimental design, to ensure the frequency of non-stimulation of input and output fibres was sufficiently high.

All the rules involve both increases and decreases in synaptic efficacy. Unfortunately for their biological relevance, they also require the synapse to take both positive and negative values. The whole scheme works by ensuring that the expected value of every change to a synapse is zero – otherwise the Ω^2 factor lurking in equation 2.15 will swamp the signal entirely. Dale's law, that almost no synapse can change its spots from being excitatory to inhibitory, or *vice-versa*, has the status almost of a theoretical *pons asinorum* – one that these rules cannot cross. The obvious solution to this, which is adopted in the Hancock, Smith and Phillips [54] paper discussed in section 2.7, is to regard each unit as a composite of two mutually inhibiting units; one which sums up the excitatory inputs, and the other which sums up the inhibitory ones. For this to work in practice, there would have to be a high degree of anatomical specificity in connections and connection types, for which there is no evidence.

A further problem with these rules is that they ignore the crucial rôle of time in the learning, and they rely too heavily on the convenient availability of the $\mathbf{b}$ patterns with which inputs are associated. It is ironic that the hippocampus is one of the main regions in which the 'static' phenomena of LTP and LTD are studied, since it is known to be important for a variety of temporal tasks such as delayed matching or non-matching to sample [44]. Any model of learning, such as these, that allows no temporal influence, is unlikely to be very accurate. However, even given these constraints, and the added fact of the highly complex time-course of real LTP [127], the model does provide a theoretical maximum discriminability for any associative memory built along these lines.

2.5 Threshold Setting

As seen above, the S/N is a threshold-independent measure of the quality of the unit. The unit is susceptible to the two types of error (commission and omission) and the threshold can be set optimally according to how each of these is weighed. Essentially the problem reduces to the standard statistical one of class discrimination [36] with so called 'Type 1' and 'Type 2' errors. As an example, consider the problem of minimising the probability of the unit erring, given that the two distributions are distributed as Gaussians with a common variance, $\mathcal{G}(\mu_l, \sigma^2)$ and $\mathcal{G}(\mu_h, \sigma^2)$ respectively, and with the relative frequencies of low and high patterns being $(1 - \tau) : \tau$. This is a fair approximation, as discussed above. Then, for a threshold θ, the overall probability of a misclassification is

$$\mathcal{P}_M = \frac{1-\tau}{\sqrt{2\pi\sigma^2}} \int_{\theta}^{\infty} e^{-\frac{(x-\mu_l)^2}{2\sigma^2}} dx + \frac{\tau}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\theta} e^{-\frac{(x-\mu_h)^2}{2\sigma^2}} dx$$

where the first term is the probability of getting a low pattern wrong and the second is the probability of misclassifying a high one.

Differentiating $\mathcal{P}_M$ with respect to θ gives

$$\frac{d}{d\theta}\mathcal{P}_M = \frac{1}{\sqrt{2\pi\sigma^2}} \left\{ \tau e^{-\frac{(\theta-\mu_l)^2}{2\sigma^2}} - (1-\tau) e^{-\frac{(\theta-\mu_h)^2}{2\sigma^2}} \right\}$$

which is zero at

$$\hat{\theta} = \frac{\mu_h + \mu_l}{2} - \frac{\sigma}{\sqrt{\rho}} \ln\left[\frac{\tau}{1-\tau}\right],$$

where ρ is the S/N. This makes intuitive sense since $\lim_{\tau \to 0} \hat{\theta} = +\infty$, *ie* if virtually every pattern is low, the threshold will be large and positive, and so almost every pattern will be classed as a low. Equivalently, $\lim_{\tau \to 1} \hat{\theta} = -\infty$, which arranges for the opposite effect. Note also that the larger the S/N, the smaller the effect of any difference between the two frequencies.

Table 2.7 shows the result of using the Hopfield rule in conjunction with this threshold for various values of $p = \tau$, demonstrating the close agreement between theory and simulation. Recall that this rule is only optimal (as an example of **R1**) for $p = \tau = 1/2$. The normal criterion adopted for the Willshaw associative net [167] is that the expected number of errors across all the outputs should

Hopfield rule

		Expect	Actual		Expect	Actual
p, τ	c	S/N	S/N	$\pm\sigma$	Errors	Errors
0.5	−1	10	11	± 1.3	1.1	1.1
0.4	−1	7.5	8.3	± 1.5	1.7	1.6
0.3	−1	1.4	1.3	± 0.40	4.6	4.5
0.2	−1	0.25	0.32	± 0.22	4.0	4.2

Table 2.7: Using threshold $\hat{\theta}$, the expected and actual numbers of errors across $n = 20$ output lines.

be 1. Since the expected number of errors rises with the number of output units, achieving this criterion for a network of units requires a higher S/N than for a single output unit; either there will need to be more input lines, or more sparse patterns must be used, or else fewer patterns can be stored to the same accuracy.

One interesting feature of table 2.7 is that the expected and actual number of errors both decrease between $p = 0.3$ and $p = 0.2$, despite the fact that the S/N also decreases, and so the unit might be expected to behave less well. This is because the expected error rate using the above threshold is

$$\mathcal{P}_M = (1-\tau)\Phi\left(-\frac{\sqrt{\rho}}{2}+\frac{1}{\sqrt{\rho}}\ln\frac{\tau}{1-\tau}\right)+\tau\Phi\left(-\frac{\sqrt{\rho}}{2}-\frac{1}{\sqrt{\rho}}\ln\frac{\tau}{1-\tau}\right).$$

where $\Phi(x)$ is the area up to x under a standard Gaussian curve. Figure 2.4 plots this as a function of τ and ρ, and it is apparent that for small S/N, the unit will be expected to make more errors for values of τ away from 0 and 1, even at the same S/N. An intuitive feel for this is given by the observation that the maximum error rate is bounded by $\min\{\tau, 1-\tau\}$, as the threshold could be set at ∞ or $-\infty$. Note the waxing and waning bimodality of this function.

All this analysis is based on the assumption that each unit can set its own threshold. Section 2.3 showed that this is only necessary when $c \neq -p/(1-p)$, as otherwise the average value of any input is zero, and so the quenching of the weights cannot affect the overall positioning of the two distributions. This is true for the standard Hopfield rule ($p = \tau = 0.5, c = -1$), but not for any $\{0, 1\}$ version of the Hebb rule. Also, the effect of the different values of c is not confined to this particular model of associative memory. Buckingham's [20]

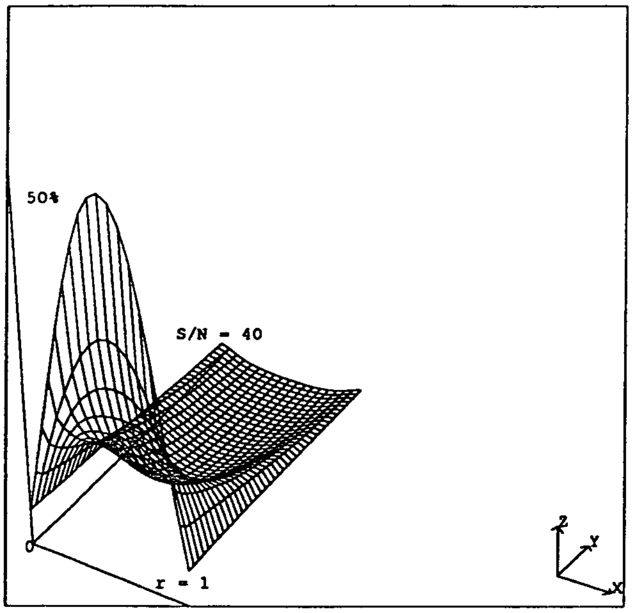


Figure 2.4: Expected error rates using the optimal threshold as a function of r and the S/N.

work, following Marr's model of the hippocampus [93], on sparsely connected Willshaw nets, shows exactly the same phenomenon, and indeed its extinction when this particular value of c is used. Marr was apparently not aware of this effect.

2.6 Physics Models

The physics connectionist community became interested in exactly the sort of issues aired in this chapter, at about the same time as Palm was introducing his model. They were essentially responding to the poor performance of the Hopfield learning rule ($\alpha = \delta = 1, \beta = \gamma = -1$) for values of p and r (which are simply related to a quantity called the magnetisation) other than 1/2. The first papers were by Tsodyks and Feigel'man [155] and Buhmann, Divko, and Schulten [21], who studied the case of asymptotic sparsity $p = r \rightarrow 0$, followed by Perez-Vincente and Amit [120], who published on the case of general $p = r$. Another important contribution came from Gardner [46], who showed how many patterns any such network can store, and how this depends on the magnetisation.

Having assumed a covariance rule, Tsodyks and Feigel'man claimed that:

> It should be borne in mind that the "old" theories of associative memory were formulated in terms of the [$\{0, 1\}$] model, which seems to be most natural. Then, however, it was replaced by the [$\{-1, 1\}$] model without careful analysis of their equivalence. The results of our paper give rise to an amazing conclusion that in some cases such "obvious" simplification may drastically affect the performance of the neural networks. [155]-p105

However, section 2.3 showed that changing the value of c from 0 to -1, which is equivalent to changing from the $\{0, 1\}$ to the $\{-1, 1\}$ model, makes no difference in the Palm model to the ability of the unit to discriminate between low and high patterns. There is thus something rather uncertain about any such remarkable performance in terms of the number of patterns the unit can store. Also, note that the mere number of patterns is not necessarily the appropriate way to judge a learning rule. Sparse patterns inherently contain less information than dense ones (there is less uncertainty as any element is more likely to be a 0), so storing more of them may not increase the informational efficiency. Gardner [46] showed just this – although as $p = r \to 0$, the theoretical maximum increases without bound of the number of potentially retrievable patterns that can be stored, the total information stored by the network actually decreases *whatever* the learning rule.

Perez-Vincente and Amit also assume a covariance rule, and conclude that, in the notation of this paper, the variance of the noise is (Amit, personal communication)

$$\sigma^2 = \left\{p(\delta-\gamma)^2 + (1-p)(\beta-\alpha)^2 + \tfrac{p(1-p)}{r(1-r)}\left[r(\delta-\gamma) - r(\beta-\alpha) + (\gamma-\alpha)\right]^2\right\},$$

that the S/N is essentially ρ_2, and that it is essential to take $c = -p/(1-p)$. This also seems to contradict the findings above.

Three possibilities for explaining this divergence spring to mind; the difference between heteroassociative and autoassociative nets or between S/N and mean field analyses, or the non-isomorphism of the underlying models. Although Palm's model is heteroassociative, and the physics models autoassociative, it turns out that this should make no difference so long as the 'identity' synapses, *ie* the diagonal terms in the connection matrix, are absent. In the S/N analysis, they introduce correlations that swamp out all the contributions from the other synapses, whereas for autoassociation their absence is also required for there to be an energy function describing the trajectory of the system as it stabilises to a memory.

S/N studies are generally used as a preliminary to the more exact mean field analyses, and also to confirm their results. The mean field analysis itself is only true in the limit of very many inputs, whereas the S/N can be calculated for finite systems. An example of where this might be important is the threshold; in the limit, the two distributions shown in figure 2.1 are either infinitely far apart or totally indistinguishable, and so the threshold is essentially irrelevant. This turns out to be the case for the mean field analysis too, but is obviously not true for any finite system.

As became evident in a series of discussions with David Willshaw and Daniel Amit, the results really differ because the models do too. The three physics papers mentioned above, apart from Gardner's, all consider the effects of inputting one pattern into a whole set of output units, each of which has learnt its own associations independently of the others, but is entrained to have the same threshold as all the others. The Palm model considers the effects of inputting many patterns into a single output unit. The rationale behind this is that there is no necessary connection between one unit and the next, and so no *a priori* reason to tie the threshold for one unit to that of another. The S/N measures the theoretical capability of a single unit to discriminate between its output, not the capability of some 'average' unit, which, in principle, cannot exist.

This difference between the models explains the divergence of the results. Lumping together a whole set of output units forces one to measure the variance rather than the dispersion of the dendritic sums, and so to ignore the helpful correlations between them which would be evident for any single unit in isolation. Setting $c = -p/(1-p)$, the 'optimum' identified by Perez-Vincente and Amit, eliminates the helpful correlations, and so makes the Palm and physics models perform equally well. Likewise, a common threshold can be set across all the units, determined only by the statistics of the associations, rather than their actual values. The 'amazing conclusion' has been reduced to something more mundane. Equivalently, it is well known that the $\{0, 1\}$ model can apparently store roughly half as many patterns as the $\{-1, 1\}$ model in the standard Hopfield case (where $p = r = 1/2$), but again this is almost an artifact.

Furthermore, the original Willshaw net [167] uses the same threshold for all the units, namely the number of on bits in the input. This is again due to its rather anomalous form.

Tsodyks and Feigel'man analyse the case in which $p \rightarrow 0$. For this case, the results here would predict the optimal value of c to be $-p/(1-p)$, which also tends to 0. As seen in the quotation, they actually use $\{0, 1\}$ patterns, which are only asymptotically optimal, but find that this is adequate given the particular manner in which the limit is approached.

Interestingly, Gardner used the Palm model for her analysis. She therefore also treats the threshold in a different manner from Perez-Vincente and Amit, not needing to introduce it in the first instance. This allowed her rather more elegant results.

2.7 Further Developments

This work is being extended in two main directions. As mentioned above, Buckingham and Willshaw [169, 20] are applying similar methods to Marr's model of the hippocampus [93], and Hancock, Smith, and Phillips [54] have developed an error-correcting version of it based on a biological learning rule suggested for the visual cortex by Artola, Bröcher, and Singer [4].

Marr's was the first, and is still the most complete, model of the hippocampus. Although it willfully ignores neuroanatomical and neurophysiological detail that was known in his day [169], Marr's analysis indicates quite precisely both how and for what inputs and outputs it should work, and how and why it should fail. The model specifies a three-layer associative net, with the deepest layer being autoassociative, and the others heteroassociative. The process of recall is for a (possibly noisy) pattern to be presented to the first layer, for it to be processed by the second layer, and fed through to the third layer, which would remove any errors and fill in any gaps through its iterative process. Each layer is just like the Willshaw net, with binary synapses that remain blown when once blown. However, they are only sparsely connected, and there are no explicit b patterns for training. Instead, some unspecified competitive process generates the targets.

The interest in Marr's proposal really lies in the sparse connectivity. As mentioned above, for a fully connected Willshaw net with patterns with a fixed number of inputs set to 1 (rather than having this number determined randomly based on probability p, as in the Palm model) the distribution of dendritic sums for the high patterns is anomalous, being just a single spike. Once the network is sparsely connected though, *ie* some of the synapses are missing, this upper distribution becomes approximately Gaussian. However, unlike figure 2.1, the variances of the two distributions will not generally be the same.

The key point that Marr missed is that the quenching of the weights after presentation of the associations leads to important correlations in the dendritic sums, just as in the Palm model. This affects the settings of the thresholds, an issue which Marr anyway rather sidestepped. Just as in the previous sections; the system can perform far better if each unit can set its own threshold based on the patterns it alone has learnt. Once again, setting $c = -p/(1-p)$ would statistically eliminate this but this option was not open to Marr, since he used the $\{0, 1\}$ model. Buckingham [20] has confirmed this, and has developed ways of setting thresholds in this sparse case.

Marr allowed a more complicated threshold mechanism than the one discussed above. He assumed that each unit could measure not only the dendritic sum due to an input, but also the total activity impinging on it from that input. This information is provided by a biologically unrealistic feed-forward inhibitory mechanism. In the case of binary patterns and weights and sparse coding, this is useful, since it allows the unit to estimate more accurately the likelihood that it has learnt something like the input pattern, based on the fraction of active synapses have been modified. There is no simple analogue of this for the case of real valued inputs and synapses.

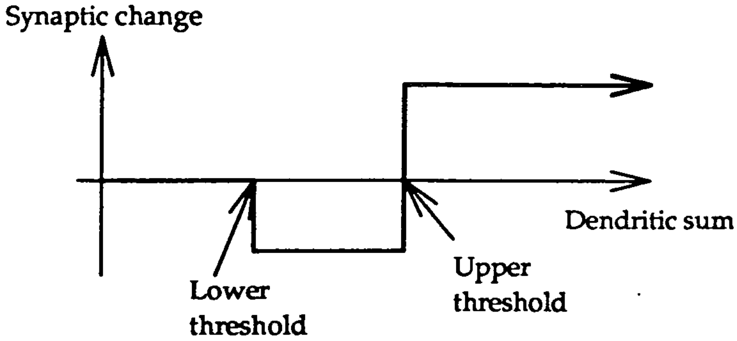


Figure 2.5: Hancock, Smith and Phillips' synaptic learning rule.

One other possible criterion for setting the threshold is the information content of the output. Following [105], the entropy of the output has two contributions; a positive one

$$-\tau \ln \tau - (1-\tau)\ln(1-\tau)$$

from the initial uncertainty in whether an output is on or off, and a negative one arising from the possibility that the unit might err. The threshold appears in this last term, and an optimal value for it can be found through differentiation. In this case, for sparse patterns, there is a penalty incurred for setting the threshold towards $+\infty$ (to reduce the number of errors amongst the more numerous lows), as a substantial fraction of the information would be lost if the system underestimates the (small) number of highs.

Artola, Bröcher and Singer [4] found a learning rule in the development of visual cortex which specifies the heterosynaptic changes in synaptic efficacy shown in figure 2.5. For very low dendritic sums, below the lower threshold, no synapses change; for very large dendritic sums, above the upper threshold, the efficacies of activated synapses increase, whereas for sums of an intermediate size, the efficacies of activated synapses decrease. Hancock, Smith and Phillips speculate that this can execute a simple form of heterosynaptic error correction.

Imagine that the training signal is delivered to the output units in the form of a large depolarisation (*ie* a large positive contribution to the dendritic sum). Then imagine presenting an input pattern to one unit that should and one that should not fire. For the former, the teaching depolarisation is supposed to be sufficient to force the changes to the synaptic efficacies to be in the positive region of

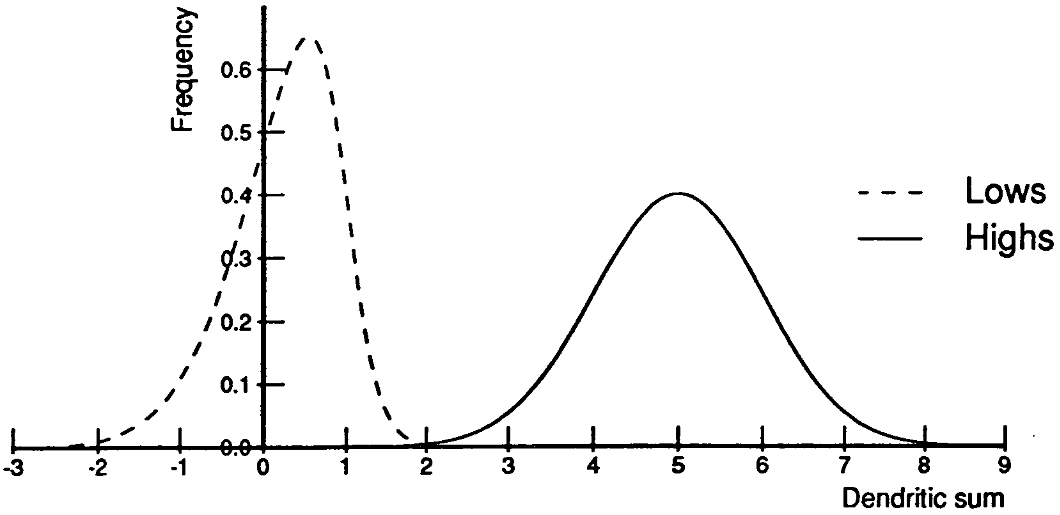


Figure 2.6: Effect of the Hancock, Smith and Phillips learning rule on the distribution of the lows.

figure 2.5. This will increase the likelihood that the unit will fire in response to a subsequent presentation of the input pattern, this time without the teaching signal. For the unit that should not fire, there are two choices. If it is only barely excited (*ie* it has a low dendritic sum), then there will be no change in the efficacy of any synapse. If, however, it is slightly more excited, so the dendritic sum is in the middle region of figure 2.5, then the efficacies of all activated synapses will actually decrease, making the unit less likely to respond to that particular input pattern in the future. Figure 2.6 shows how the frequency graph of dendritic sums will change from the original in figure 2.1. The marked skew in the distribution of the lows aids the unit's powers of discrimination. Note that it also invalidates the use of the S/N, as one of the curves is no longer approximately Gaussian.

The rule implements a form of error correction, because it is differentially sensitive to false firing. It is not complete error correction, since it is not sensitive to correct firing, and will continue to increase the efficacies of synapses even when the unit is performing perfectly. Hancock, Smith and Phillips present results to show how well it outperforms the covariance rule, but they only present the patterns a few times, and so would not run into this problem. They

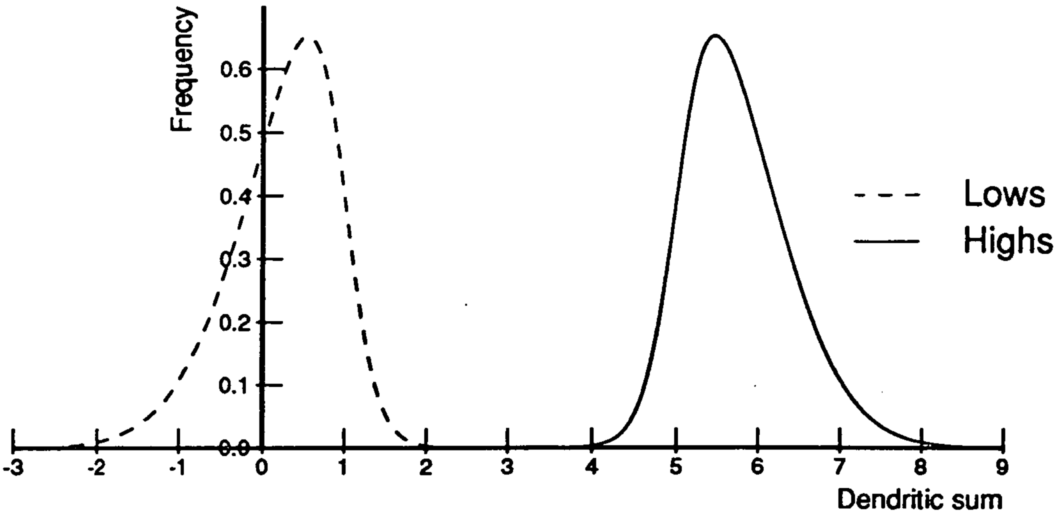


Figure 2.7: Distributions of the paired Hancock, Smith and Phillips learning rule.

also claim (personal communication) that it works very much faster than back-propagation [160, 118, 131, 82], since its underlying associative nature allows it to get the weights vector approximately correct very quickly, before its error correcting aspect takes over to perfect the memorisation.

This rule still requires synapses to be alternately excitatory and inhibitory, which was one reason for concern as to the biological plausibility of the previous optimal rules. In an attempt to improve on this, Hancock, Smith and Phillips considered the paired architecture, discussed above, in which two mutually inhibiting units combine to have the effect of one. If both units adopt the error-correcting learning rule, the resulting frequency distributions of dendritic sums will look rather like those in figure 2.7. This improves the performance of the rule still further, because both the curves are skewed. Unfortunately, the required degree of specificity is extremely unlikely to occur naturally.

2.8 Conclusions

Adopting the criterion of maximising the signal/noise ratio (S/N) for a class of very simple associative matrix memories, leads to one optimal, and two sub-optimal learning rules. Each of these, a covariance, a heterosynaptic, and a homosynaptic rule, has previously been proposed, but they have not previously been analysed in a common fashion. The covariance rule performs better than the other two, but only negligibly so in the limit of sparse coding. Unlike the other two rules, it also requires synapses to increase in efficacy even if their pre- and post-synaptic units are silent. All the rules have the automatic consequence that the average value of a synapse should be zero, to suppress noise, and so require synapses to take both positive and negative values. The threshold may be set according to an additional criterion, such as minimising the probability of an error, but certain of these criteria may not be monotonic in the S/N.

The rules here, and the lack of dependency of the S/N on the input values for patterns, differ from previous analyses. Some of these analyses are incorrect, ignoring vital correlations in the noise terms. Other analyses are correct, but are based on a different model. The key characteristic discussed here is that each unit is evaluated independently, and so can set its own threshold to allow for its particular quenched weights. Other analyses have lumped collections of output units together, and awarded them the same thresholds. This reduces the apparent quality of the memory quite markedly, unless one particular relationship holds between the high and low values of the input patterns. In that case they perform identically.

Deliberately, neither the model, nor the rules it suggests are biologically plausible – of particular concern is their violation of Dale's law. The models are suggestive though, as in the difference of an order of magnitude between the effects of LTP and LTD. Natural processing is by no means constrained to follow an optimal path, but it is nevertheless instructive to understand the consequences of suggested synaptic mechanisms, as providing theoretical limits to performance.

Chapter 3

Reinforcement Comparison

The reason we try that hard to keep up with the Jones is because they are so accurately the reflection of ourselves.

A Smith

3.0 Summary

In this chapter, efficient rules for a second type of learning scheme, reinforcement learning, are examined. Sutton [147] introduced a reinforcement comparison term into the equations governing the development of certain stochastic learning automata, the traditional mechanisms addressing this scheme. He argued that it should speed up learning, particularly for unbalanced reinforcement tasks. Williams' subsequent extensions [165] to the class of algorithms demonstrated that they were all performing approximate on-line, stochastic, gradient ascent, but that, in terms of expectations, the comparison term has no first order effect.

This chapter analyses the second order contribution, and uses the criterion that its modulus should be minimised to determine an optimal value for the comparison term. This value turns out to be different from the one Sutton used, and simulations confirm its efficacy.

3.1 Introduction

For most purposes, reinforcement learning is defined operationally, in terms of the nature of the judgement visited by the environment on the output of a learning system. For the case of supervised learning, as seen in the previous chapter on associative memories, the system is told which response it should make (the b patterns) to which input. In reinforcement learning, though, it is only given some reward, which at most may indicate whether or not it was correct. Reinforcement learning therefore appears to be more general. However, as Mackintosh [88] points out in his discussion about the differences between classical and instrumental conditioning, the system need not be quite so entrained to the experimenter's view.

For only two outputs (and, strictly, two possible rewards), Barto and Sutton [9] pointed out that the information provided by the environment is essentially the same for both reinforcement learning and the supervised associative learning described in the previous chapter. Unlike the cases studied there, though, reinforcement learning systems (equivalently stochastic learning automata) almost always operate in stochastic environments. Although these systems also build weight matrices that determine which outputs their units associate with which inputs, they have to sample the statistics of the environment rather than becoming set on the first sight of a pattern. For the associative memories, there was a tacit assumption of some training or 'now-(im)print' signal that selects between storage and retrieval.

That such reinforcement learning systems depend on their input makes them associative, an extension to the traditional range of stochastic learning automata due to Barto and Sutton [9, 7, 147]. Albeit with some interesting anomalies, animals are capable of learning under stochastic classical and instrumental conditioning paradigms [31, 88]. The question addressed here is which learning rules make the systems learn more accurately and faster. Signal to noise analysis is no longer appropriate as a way of judging the rules, because the 'correct' output value can only be determined by sampling. An alternative criterion is adopted which judges the rules on how much they increase the expected amount of reinforcement.

Sutton [147] introduced the notion of reinforcement prediction as a way of speeding up the learning of a class of stochastic learning automata. Most previous methods made assumptions about the independence of the learning of the automata from all aspects of their reinforcement history that were not 'compiled' into their current action probabilities. This is a particular kind of Markov property which makes analysis more straightforward, since it obviates the need for handling different possible histories. Sutton reasoned that comparing the amount of reinforcement a system has just received with some function of its frequency of delivery in the past, might be helpful for determining whether or not its actions were making things worse or better. He expected particular utility for such comparisons in the difficult cases in which reinforcement delivery is unbalanced - for instance when all actions tend to be rewarded or all punished.

Williams [165] analysed a related set of algorithms, which includes Sutton's, and demonstrated that they all perform on-line stochastic gradient ascent in the expected amount of reinforcement. The surprising part of this paper is that, for the case treated by Sutton, the comparison term may be eliminated from the analysis at an early stage. The result on stochastic gradient ascent is unaffected by its value. Sutton's simulations, however, demonstrated that different comparison terms perform very differently.

Williams essentially looked at the first order term in the Taylor expansion of the function that relates expected reinforcement to the weights determining the probability of performing the actions. Since the comparison term vanishes to first order, it is essential to examine higher order terms to detect its effects. Second order analysis might reveal a rôle for it, and, potentially, an optimal value.

The next section looks at Williams' theory, and the derivation of the optimal second-order comparison term, section 3.3 compares the empirical performance of the new term with that of the old one, and section 3.4 considers recent developments in the field of associative reinforcement learning.

3.2 Theory

3.2.1 Williams' Analysis

Williams treats a very general problem. At any time, each of a collection of n units receives an input $x^i \in \Re^p, 1 \leq i \leq n$ from some environment, and uses its weight vector $w^i \in \Re^p$ to determine whether to fire or not; $y_i = 1$ or $y_i = 0$ respectively. Before it chooses its actions, and before the environment evaluates the combined set of actions, every unit also chooses a reinforcement comparison value $b_{ij}, 1 \leq i \leq n, 1 \leq j \leq p$, for each component of each weight. The environment returns a global reinforcement value r that is stochastically related to the quality of the actions of the units, and each unit then updates its weight vector according to the reinforcement, its chosen action, and its reinforcement comparison values.

A simple example of such a reinforcement learning system is the two armed bandit problem, which will be discussed later. For this, the automaton has no inputs, but chooses, stochastically on the basis of a stored weight, to pull either the left arm ($y = 0$) of the bandit or the right arm ($y = 1$). The machine delivers reinforcement of $r = \pm 1$ with different probabilities for the two arms, and the automaton has to learn, by changing the weight, which arm it is best to pull.

More formally, Williams proves that if:

$$\Delta w_{ij} = \alpha_{ij}(r - b_{ij})e_{ij}, \tag{3.1}$$

where,

- r is the reinforcement ($\in \Re$),
- α_{ij} is the learning rate parameter for w_{ij},
- b_{ij} are reinforcement baselines, which are conditionally independent of the actions y_i given the weights $W \equiv [w^i] \equiv [w_{ij}]$ and the inputs x^i,

$g_i(\xi, w^i, x^i) \;=\; \mathcal{P}[y_i = \xi | w^i, x^i]$ is the probability the i^{th} unit emits action ξ given its weights w^i and its input x^i.

$e_{ij} \;=\; \frac{\delta \ln g_i}{\delta w_{ij}}$ is the so-called eligibility of the weight w_{ij}, a measure of how influential it was in choosing the action,

then:

$$\mathcal{E}[\Delta w_{ij} | W] = \alpha_{ij} \frac{\delta \mathcal{E}[r|W]}{\delta w_{ij}}. \tag{3.2}$$

where W is the matrix of all w^i.

One attractive feature of Williams' analysis is the way in which he teases apart the rule and the representation. This arises from the form of his eligibility term e_{ij}, which incorporates the latter automatically through the process of differentiation. Were a different representation or a different activation function to be adopted, then e_{ij} would depend differently on x and W, but the properties of the underlying rule would not change.

Equation 3.2 implies that these algorithms are all performing stochastic gradient ascent in an averaged sense. The dependence on the values of the b_{ij} drops out at an early stage, since:

$$\begin{aligned}
\mathcal{E}[\alpha_{ij} b_{ij} e_{ij} | W, x^i] &= \sum_{\xi_i} \alpha_{ij} b_{ij} \mathcal{P}\left[y_i = \xi_i | w^i, x^i\right] \frac{\delta}{\delta w_{ij}} \left\{\ln \mathcal{P}\left[y_i = \xi_i | w^i, x^i\right]\right\} \\
&= \alpha_{ij} b_{ij} \sum_{\xi_i} \frac{\mathcal{P}\left[y_i = \xi_i | w^i, x^i\right]}{\mathcal{P}\left[y_i = \xi_i | w^i, x^i\right]} \frac{\delta}{\delta w_{ij}} \left\{\mathcal{P}\left[y_i = \xi_i | w^i, x^i\right]\right\} \\
&= \alpha_{ij} b_{ij} \frac{\delta}{\delta w_{ij}} \left\{\sum_{\xi_i} \mathcal{P}\left[y_i = \xi_i | w^i, x^i\right]\right\} \\
&= \alpha_{ij} b_{ij} \frac{\delta}{\delta w_{ij}} \{1\} \\
&= 0.
\end{aligned}$$

However, looking at equation 3.1, it is apparent that changing the b_{ij} is likely to affect at very least the stability of the algorithm. Consider what would happen if b_{ij} were large and positive; Δw_{ij} would also tend to be large, but alternately positive and negative, based on the sign of e_{ij}. Therefore, although Williams' proof guarantees that the average behaviour will be suitable, it can provide no such comforting assurance about the particular behaviour that will be observed. Sutton [147] empirically compared his algorithm with existing ones, and found faster convergence across a range of problems for b_{ij} as estimators of the average amount of reinforcement received, than for $b_{ij} = 0$.

3.2.2 The Second Order Term

Unfortunately, treating higher order terms at the same level of generality as Williams is not fruitful. Consider instead one of the simpler cases that Sutton takes: there is just one unit, weights w_i, inputs x_i, reinforcement r, and:

$$\Delta w_i = \alpha(r - b)(y - \pi)x_i$$

where $\pi = \mathcal{E}[y|x, w]$. b can depend on x and w, but not on the output y. Note that in general

$$(y - \pi)x_i \neq \frac{\delta}{\delta w_i}\{\ln \mathcal{P}[y = 1|w, x]\}$$

as required for this analysis. However, as Williams points out, it is proportional when

$$\mathcal{P}[y = 1|w, x] = \frac{1}{1 + e^{-w.x}} \tag{3.3}$$

In any case, as shown below, $\mathcal{E}[\Delta w_i|w, x]$ still does not depend on b.

A rather different way of looking at his result is through the Taylor expansion of $\mathcal{E}[r'|w, x]$, using the prime ' to indicate that it is the expected value of the reinforcement that will be received at the <u>next</u> time step, just given the statistics of the possible happenings at the current time step.

$$\begin{aligned}\mathcal{E}[r'|w, x] &= \mathcal{E}[r|w] + \sum_i \mathcal{E}[\Delta w_i|w, x]\frac{\delta\mathcal{E}[r|w]}{\delta w_i} + \\ &\quad \frac{1}{2}\sum_{ij} \mathcal{E}[\Delta w_i \Delta w_j|w, x]\frac{\delta^2\mathcal{E}[r|w]}{\delta w_i \delta w_j} + \ldots. \end{aligned} \tag{3.4}$$

Williams deals with the first order term, showing that the first term in the product is proportional to the second, and that b makes no contribution whatsoever.

Just as above,

$$\begin{aligned}\sum_{i} \mathcal{E}[\alpha b(y-\pi)x_i|w,x]\frac{\delta\mathcal{E}[r|w]}{\delta w_i} &= \alpha b\sum_{i}\frac{\delta\mathcal{E}[r|w]}{\delta w_i}x_i\mathcal{E}\left[y-\mathcal{E}[y|w,x]|w,x\right] \\ &= 0.\end{aligned}$$

because $\mathcal{E}\left[y-\mathcal{E}[y|w,x]|w,x\right]=0$.

Setting $\mathcal{F}(z)=\mathcal{E}[r|z]$, the second order term is:

$$\sum_{i,j}\frac{\delta^2\mathcal{F}}{\delta w_i\delta w_j}x_i x_j\alpha^2\sum_{\xi,\rho}\mathcal{P}[y=\xi|w,x]\mathcal{P}[r=\rho|\xi,x](\rho-b)^2(\xi-\pi)^2. \tag{3.5}$$

Note that the inner sum does not depend on the value of i or j. This is a positive quadratic in b, and differentiating to find the value, $\hat{b}$, at which it is a minimum gives:

$$\begin{aligned}\hat{b}\sum_{\xi}\mathcal{P}[y=\xi|w,x](\xi-\pi)^2 &= \sum_{\xi,\rho}\mathcal{P}[y=\xi|w,x]\mathcal{P}[r=\rho|\xi,x]\rho(\xi-\pi)^2, \\ \textit{ie} \qquad \hat{b} &= \frac{\mathcal{E}[r(y-\pi)^2|w,x]}{\mathcal{V}[y|w,x]}\end{aligned}$$

However, y can only take on two values; 0 or 1. Let p be the probability that $y=1$, and r_0 and r_1 be the expected rewards for choosing actions 0 and 1 respectively, so

$$\begin{aligned}p &= \mathcal{P}[y=1|w,x], \\ r_0 &= \mathcal{E}[r|y=0,w,x], \\ r_1 &= \mathcal{E}[r|y=1,w,x].\end{aligned}$$

Then $\pi=p$, and

$$\begin{aligned}\hat{b} &= \frac{p(1-p)^2r_1+(1-p)p^2r_0}{p(1-p)} \\ &= (1-p)r_1+pr_0.\end{aligned}$$

in which, counterintuitively, the expected reward for emitting action 1 is paired with the probability of emitting action 0, and *vice-versa*. Williams (personal communication) derived the same expression for $\hat{b}$ on the grounds of minimising the variance of the Δw_i.

The reinforcement comparison algorithm favoured by Sutton involves teaching an extra unit to predict the future reinforcement level. He defines $s=\sum_i v_i x_i$, where v are the prediction weights. These are changed according to:

$$\Delta v_i=\beta(r-s)x_i.$$

This makes s an estimator of sorts of $\mathcal{E}[r|w,x]$, or b^*, where:

$$b^* = pr_1 + (1-p)r_0.$$

which, *a priori*, is the more natural pairing. Symbolically, the rule that results from Sutton's expression has

$$\Delta w_i = \alpha(r - \bar{r})(y - \bar{y})x_i$$

where $\bar{r} \equiv b$ is the long term average of the reinforcement, and $\bar{y} \equiv \pi$ is the expected value of the action y. This is just the equivalent of the covariance rule R1 from chapter 2 (see table 2.5), which was seen there to be optimal in the associative memory case.

The difference here lies in the $(\xi - \pi)^2$ contribution in equation 3.5 – if this were not there, then the optimal value of b would indeed be $\mathcal{E}[r|w,x]$. However, the weight change rule would not work properly without it, as, amongst other things, it cures the imbalance caused by the agent choosing one action more frequently than the other.

3.2.3 Choosing b

Although $\hat{b}$ minimises the inner sum in the second order term of equation 3.5, it is not yet clear that this is appropriate. Indeed, Williams, who also derived $\hat{b}$, decided that minimising the variance of the Δw_i, which was his *modus operandi*, was unwise. He reasoned that one of the contributions to this variance arises from the weight change being of the correct sign, but being possibly either small or large. It would be most unwise to minimise this contribution through introducing $\hat{b}$, if it were at the expense of forcing some of the changes to be of the wrong sign.

The derivation above, however, provides a different reason for choosing $\hat{b}$. This is particularly clear in the one dimensional case. Since the reinforcement is bounded both above and below, the second derivative $\delta^2\mathcal{F}/\delta w^2$ in equation 3.5 is likely to be positive for some values of w and negative for others. Since the optimal weight is either $\pm\infty$,[1] according as the optimal action is 0 or 1, the contribution from this term is bound both to speed and hinder the learning on occasion. Setting $b = \hat{b}$ minimises the second half of equation 3.5, and so minimises both the harmful and helpful contributions of this term.

Another way to look at this is that gradient ascent works perfectly on linear functions – whose second and higher order derivatives are all zero. Under the assumption that the significance of the derivatives decreases with order, choosing $\hat{b}$ to minimise the contribution of the second, makes the function 'as linear as possible'. Therefore it makes as appropriate as possible the stochastic gradient ascent that Williams himself shows the algorithm to be performing.

The same will be true in higher dimensions too, in that the second order term will be alternately a hindrance and a help. Minimising its modulus should therefore increase the overall efficiency of the stochastic gradient ascent.

Note, though, that it is less clear that maximising the expected amount of reinforcement at the next time step is itself appropriate. For instance, under certain circumstances *exploring* to improve the statistical certainty of the estimate of the reinforcement comparison term might be wiser than *exploiting* the current situation to maximise immediate reinforcement. Too much premature exploitation could harm the chances of the system ever learning the optimal action at all. This issue is also related to the distinction Watkins makes in his thesis [159] between learning optimal behaviour by any means, and learning it optimally (*eg* as quickly as possible).

As an example of the effect of the second order term, consider the first task Sutton investigated, which is one example of a two-armed bandit problem. In this case, there are two possible actions $y = 0, 1$, and based on these:

$$\begin{array}{rclrcl} y = 0 \;\Rightarrow\; \mathcal{P}[r = 1] & = & 0.8 \qquad & \mathcal{P}[r = -1] & = & 0.2 \\ y = 1 \;\Rightarrow\; \mathcal{P}[r = 1] & = & 0.9 \qquad & \mathcal{P}[r = -1] & = & 0.1 \end{array}$$

so the optimal action is $y = 1$. Choose $\mathcal{P}[y = 1|w] \equiv f(w) = 1/(1 + e^{-w})$, then:

$$\mathcal{E}[r|w] \;=\; 0.6 + 0.2f(w)$$

[1] If $\mathcal{P}[y|w, x]$ depends in a natural way on w as, for instance, through the logistic function in equation 3.3.

$$\begin{aligned}\frac{\delta}{\delta w}\mathcal{E}[r|w] &= 0.2f(w)(1-f(w)) \\ \frac{\delta^2}{\delta w^2}\mathcal{E}[r|w] &= 0.2f(w)(1-f(w))(1-2f(w))\end{aligned}$$

So, with $\Delta w = \alpha(r - b)(y - \pi)$, the changes are:

y	r	$\mathcal{P}$	$\Delta w/\alpha$
0	-1	$0.2(1-f(w))$	$(1+b)f(w)$
0	1	$0.8(1-f(w))$	$-(1-b)f(w)$
1	-1	$0.1f(w)$	$-(1+b)(1-f(w))$
1	1	$0.9f(w)$	$(1-b)(1-f(w))$

Then $\mathcal{E}[\Delta w] = \alpha \times 0.2f(w)(1 - f(w))$, which, as expected, is independent of b.

However, let $g(w) = \mathcal{E}[r|w] = 0.6 + 0.2f(w)$, then:

$$\begin{aligned}\mathcal{E}[r'|w] = {} & 0.2(1-f(w))g(w+\alpha f(w)(1+b)) + \\ & 0.8(1-f(w))g(w-\alpha f(w)(1-b)) + \\ & 0.1f(w)g(w-\alpha(1-f(w))(1+b)) + \\ & 0.9f(w)g(w+\alpha(1-f(w))(1-b)),\end{aligned}$$

where r' is the reinforcement received after the automaton's next choice. b will not drop out of this. Figure 3.1 shows a graph of the second order term, for three values of b. It is apparent that it helps learning for $w < 0$ and hinders it for $w > 0$. Setting $b = \hat{b}$ minimises both these effects, although it is apparent just how close $\hat{b}$ and b^* are.

There are two types of imbalance that can afflict problems like the two-armed bandit:

- Imbalance in the probabilities – in which both the better and the worse action usually lead to the same value of reinforcement, the only difference being in the precise frequency. It is difficult to select the optimal action, since, as Barto [7] says 'an action can more frequently appear to be the

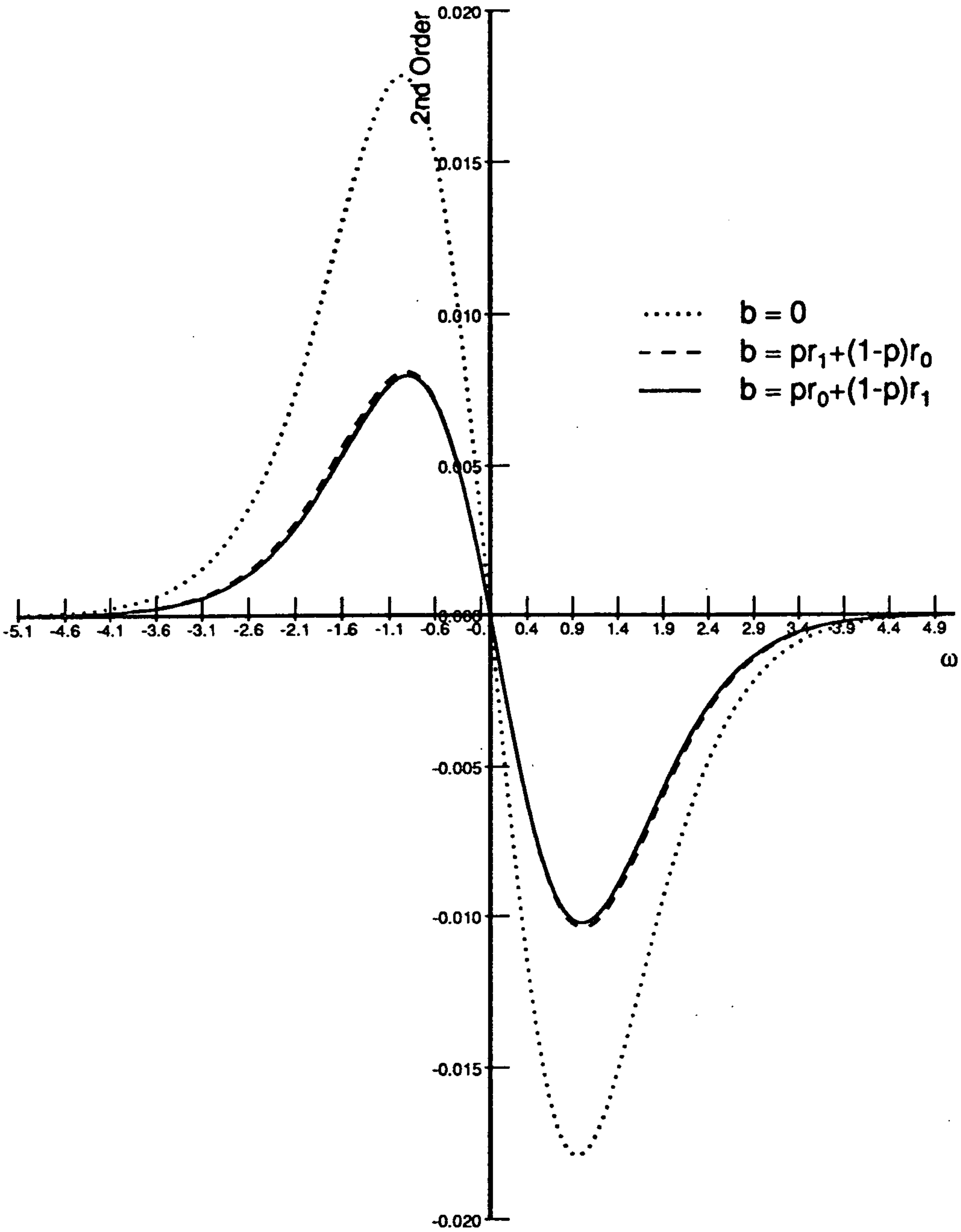


Figure 3.1: Second order contribution to the change in weights

desired action just because it is being performed more [or less] often than the other action' with the more or less depending on whether the actions usually lead to positive or negative reinforcement.

- Imbalance in the reinforcement values – in which the actual reinforcement values received are not centred around 0. This can make learning substantially more difficult by making the sign of the changes in the weights on any one occasion independent of the reinforcement received.

Reinforcement comparison deals specially with the second type of imbalance. Williams (personal communication) has pointed out that the term $r - b_{ij}$ in the formula for the weight change, equation 3.1, will take both positive and negative values if the b_{ij} lie between the maximum and minimum reinforcement values. Barto [7] provides some reasons why the term $y - \pi$ in the learning rule helps mitigate the effects of the imbalance in the probabilities.

3.3 Results

Calculating the 'optimal' $\hat{b}$ is more difficult than calculating Sutton's b^*, because of the cross-pairing of the average reinforcement for action 1 with the probability of doing action 0. It is possible to develop an estimator $s^t(x) = \sum v_i^t x_i$ with weights v^t, as in Sutton's algorithm, and to change them according to:

$$\Delta v_i^t = \beta \left[r^t \left\{ y^t \frac{1 - \pi^t}{\pi^t} + (1 - y^t) \frac{\pi^t}{1 - \pi^t} \right\} - s^t \right] x_i.$$

where π^t is an approximation to $\mathcal{E}[y^t | w^t]$. s^t then estimates $\hat{b} = (1-p)r_1 + pr_0$. Since y is never 1 if $\pi^t = 0$, the first term is never infinite – similarly for the second term.

Unfortunately, this scheme would not be expected to converge. Consider what happens as $\pi^t \rightarrow 1$, for example. Most frequently, $y^t = 1$, in which case the changes to v^t will be very small (as the $y^t(1-\pi^t)/\pi^t \sim 0$). Rarely, though, $y^t = 0$, in which case the change to v^t will be very large. This is the mechanism by which the cross-pairings can balance out, but it introduces significant instability.

Task Number	Reinforcement Type	r range	r condition Action 1	Action 0
1	Binary	$\{1,-1\}$	0.90	0.80
2	Binary	$\{1,-1\}$	0.20	0.10
3	Binary	$\{1,-1\}$	0.55	0.45
4	Continuous	$\Re$	0.90	0.80
5	Continuous	$\Re$	−0.80	−0.90
6	Continuous	$\Re$	0.05	−0.05

Table 3.1: The Tasks (From Sutton [147]-p18).

The alternative way, suggested by, but not discussed in, Sutton's thesis, is to develop separate predictions of r_0 and r_1, using two sets of weights.[2] These would then be combined with π^t as $(1-\pi^t)r_1 + \pi^t r_0$. Both methods were simulated.

The problems that Sutton developed for his thesis [147] will be adopted. Those used here are the non-associative ones from his Chapter II, although the new comparison term will work for associative tasks too. Table 3.1, copied from P18, shows the problems. The binary tasks produce reinforcement of ±1, with the probability that it is +1 given for each action in the last two columns of the table. The continuous tasks produce reinforcement spread uniformly within ± 0.1 of the means given in the last two columns.

Formal descriptions of the algorithms compared are given in table 3.2, using Sutton's notation. Algorithms $\mathcal{A}$ and $\mathcal{A}'$ are Sutton's algorithms 8 and 9, which he found to be the best. $\mathcal{B}$, $\mathcal{B}'$, $\mathcal{C}$ and $\mathcal{C}'$ all make s^t estimate the quantity recommended by the analysis above. $\mathcal{B}$ and $\mathcal{B}'$ do this through a single term, whereas $\mathcal{C}$ and $\mathcal{C}'$ also employ u_1^t and u_0^t, which are designed to predict r_1 and r_0 respectively. Sutton's caveat that it is dangerous to extrapolate from only a few values of α should be remembered.

Figures 3.2-3.7 show how the algorithms performed on each of the various tasks, for differing values of the learning rate α. Figure 3.8 shows how the

[2] He considered using these for his estimator b^* rather than $\hat{b}$.

Table 3.2: The Algorithms (After Sutton [147]-p21).

Algorithm	Update Rule
$\mathcal{A}$	$w[t+1] = w[t] + \alpha(r[t+1] - s_{\mathcal{A}}[t]) \quad \times \quad (y[t] - \frac{1}{2})$
$\mathcal{A}'$	$w[t+1] = w[t] + \alpha(r[t+1] - s_{\mathcal{A}'}[t]) \quad \times \quad (y[t] - \pi[t])$
$\mathcal{B}$	$w[t+1] = w[t] + \alpha(r[t+1] - s_{\mathcal{B}}[t]) \quad \times \quad (y[t] - \frac{1}{2})$
$\mathcal{B}'$	$w[t+1] = w[t] + \alpha(r[t+1] - s_{\mathcal{B}'}[t]) \quad \times \quad (y[t] - \pi[t])$
$\mathcal{C}$	$w[t+1] = w[t] + \alpha(r[t+1] - s_{\mathcal{C}}[t]) \quad \times \quad (y[t] - \frac{1}{2})$
$\mathcal{C}'$	$w[t+1] = w[t] + \alpha(r[t+1] - s_{\mathcal{C}'}[t]) \quad \times \quad (y[t] - \pi[t])$

Where: $w[0] = 0, \pi[0] = \frac{1}{2}, y[t] \in \{1, 0\}, \alpha > 0$,
and $\pi[t]$ is the probability that $y[t] = 1$.

$$\text{For all algorithms, } y[t] = \begin{cases} 1, & \text{if } w[t] + \eta[t] > 0; \\ 0, & \text{otherwise,} \end{cases}$$

where $\eta[t]$ is distributed as a Gaussian $\mathcal{G}[0, 0.3]$.

For $\mathcal{A}$ and equivalently $\mathcal{A}'$, $s_{\mathcal{A}}[0] = r[1]$, and

$$s_{\mathcal{A}}[t+1] \;=\; s_{\mathcal{A}}[t] + \beta(r[t+1] - s_{\mathcal{A}}[t]),$$

For $\mathcal{B}$ and equivalently $\mathcal{B}'$, $s_{\mathcal{B}}[0] = r[1]$, and

$$s_{\mathcal{B}}[t+1] \;=\; s_{\mathcal{B}}[t] + \beta\left\{r[t+1]\left(\frac{(1-y[t])\pi[t]}{1-\pi[t]} + \frac{y(t)(1-\pi[t])}{\pi[t]}\right) - s_{\mathcal{B}}[t]\right\},$$

For $\mathcal{C}$ and equivalently $\mathcal{C}'$, $s_{\mathcal{C}}[0] = r[1]$, $u_1[0] = r[1]$, $u_0[0] = r[1]$, and

$$\begin{aligned} s_{\mathcal{C}}[t+1] &= s_{\mathcal{C}}[t] + \beta(u_1[t+1](1-\pi[t]) + u_0[t+1]\pi[t] - s_{\mathcal{C}}[t]), \\ u_1[t+1] &= u_1[t] + \beta(r[t+1] - u_1[t])y[t], \\ u_0[t+1] &= u_0[t] + \beta(r[t+1] - u_0[t])(1 - y[t]), \\ \text{and } \beta &= 0.2. \end{aligned}$$

All algorithms are run for 200 iterations, and each mark on the graphs in figures 1-7 is the average over 500 runs.

algorithms performed across the entire range of tasks, choosing for each its best result. The y-axis shows the terminal probability of choosing action 1, which is the better action for all of the tasks. It is apparent that $\mathcal{C}$ which uses the new estimator, does indeed perform better than $\mathcal{A}$ and $\mathcal{A}'$ which use the original one, although not by much. The differences between $\mathcal{C}$ and $\mathcal{A}$ are statistically significant using the one-tailed t−test at the 5% level for tasks 2, 3, 4, 5 and 6, but the algorithms are statistically indistinguishable for task 1. $\mathcal{B}$ and $\mathcal{B}'$ are particularly bad on the two tasks for which reinforcement is generally negative whichever action is taken. It is unclear why the instability mentioned above should affect these particular tasks more than the others. Figures 3.5-3.7 are particularly striking on how $\mathcal{C}$ differs from the other algorithms, as it achieves better terminal probabilities for far lower values of the learning rate α.

It is also unclear why $\mathcal{C}$ should outperform $\mathcal{C}'$, since Sutton generally found algorithms with eligibility terms of the form $y - \pi$ were preferable to those employing $y - 1/2$. Williams (personal communication) has some results to suggest that it is preferable to use the sample mean of y in the learning rule, rather than π or $1/2$. This may just be because by being *less* accurate, it allows the weight to change faster.

An alternative to the methods used for algorithms $\mathcal{C}$ and $\mathcal{C}'$ is to develop explicit estimators of $(1 - p)r_1$ and pr_0, and to use their sum. In the non-associative case the resulting algorithms would not differ greatly from $\mathcal{C}$ and $\mathcal{C}'$. They would differ in the associative case, however, since the learning rule for these estimators would not change, whereas the equivalents of $\mathcal{C}$ and $\mathcal{C}'$ would involve estimators of $r_1(x)$, $r_0(x)$ and $p(x)$, the probability of choosing output 1 for input x, which do depend on the input x.

In a further experiment, the standard deviation σ of the distribution of $\eta[t]$ was set to 0.5. This value determines the balance between the exploitation of the current weight $w[t]$, and the exploration for a better one. Figure 3.9 is the equivalent of figure 3.8 for this case, showing the best performance of the algorithms, and again algorithm $\mathcal{C}$ can be seen to be somewhat superior. Indeed, it affords enough improvement in this case to make $\mathcal{C}$ significantly better than $\mathcal{A}$ and $\mathcal{A}'$ in the first task.

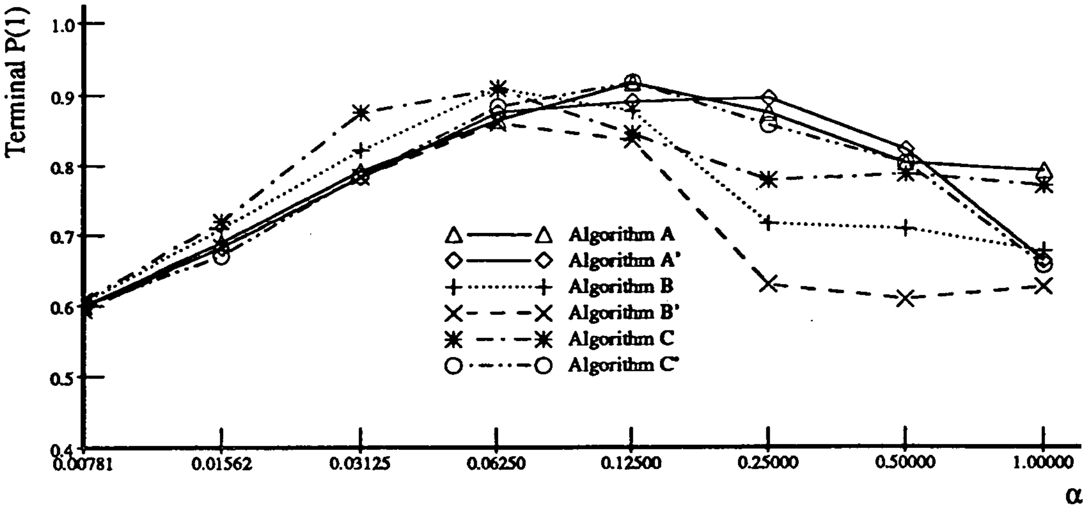


Figure 3.2: Task 1

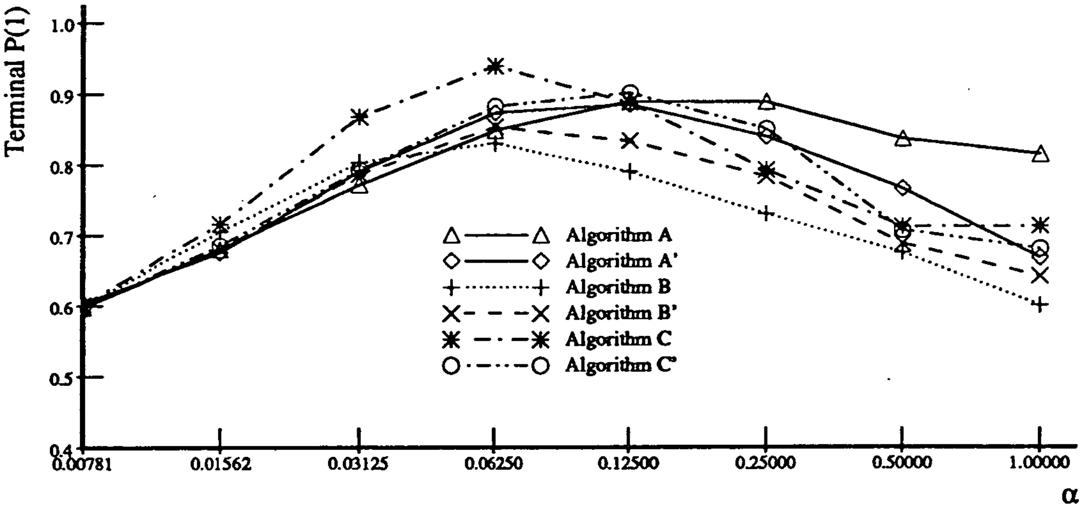


Figure 3.3: Task 2

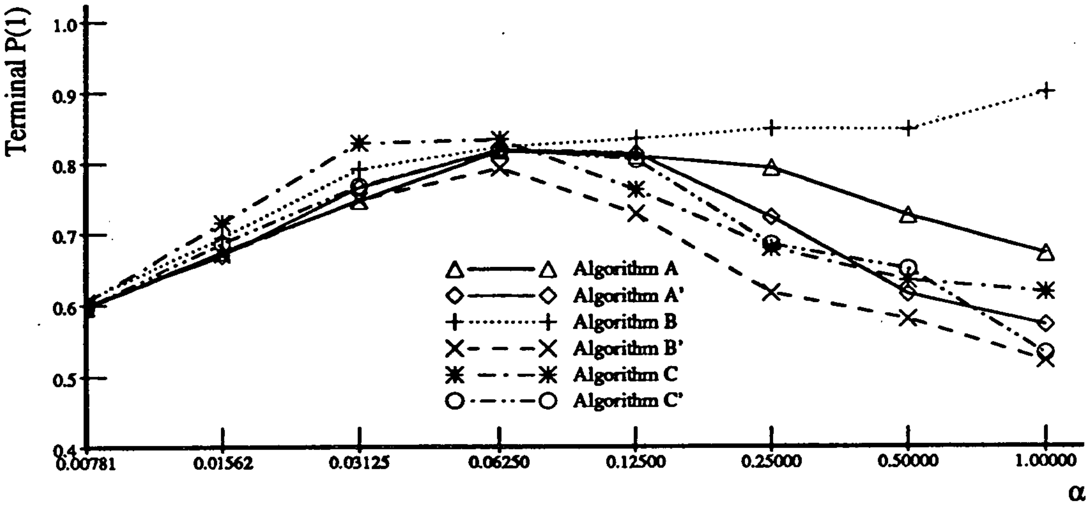


Figure 3.4: Task 3

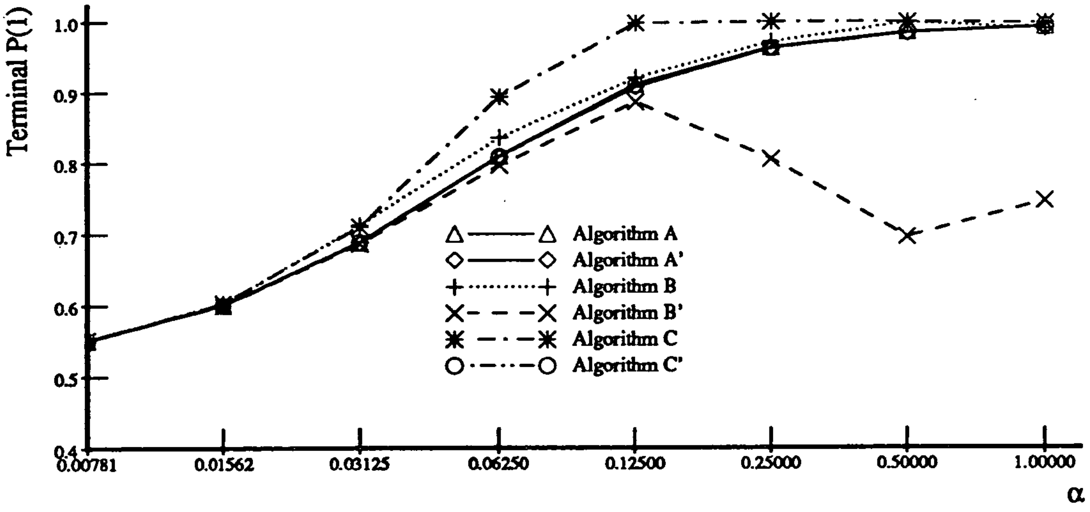


Figure 3.5: Task 4

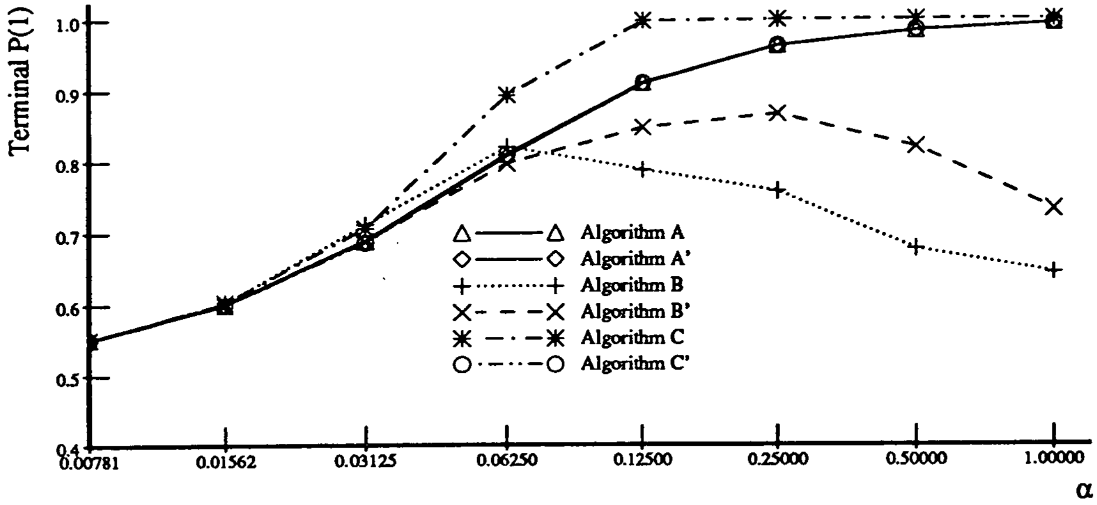


Figure 3.6: Task 5

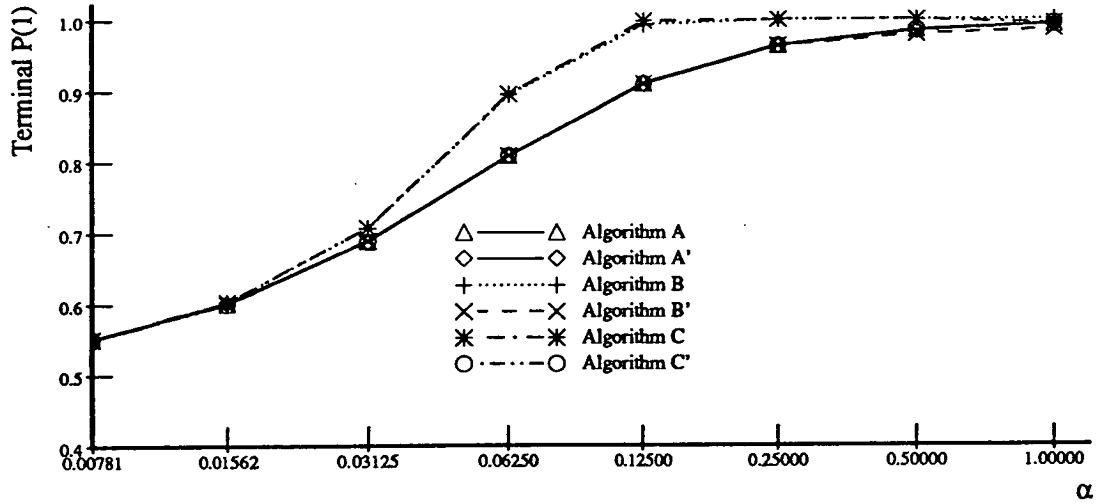


Figure 3.7: Task 6

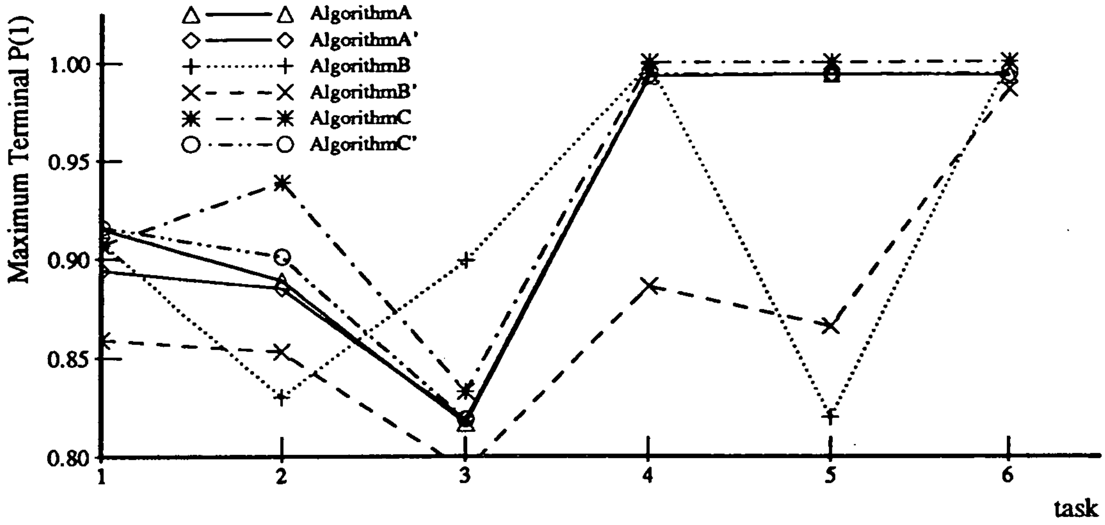


Figure 3.8: All tasks – best performing α

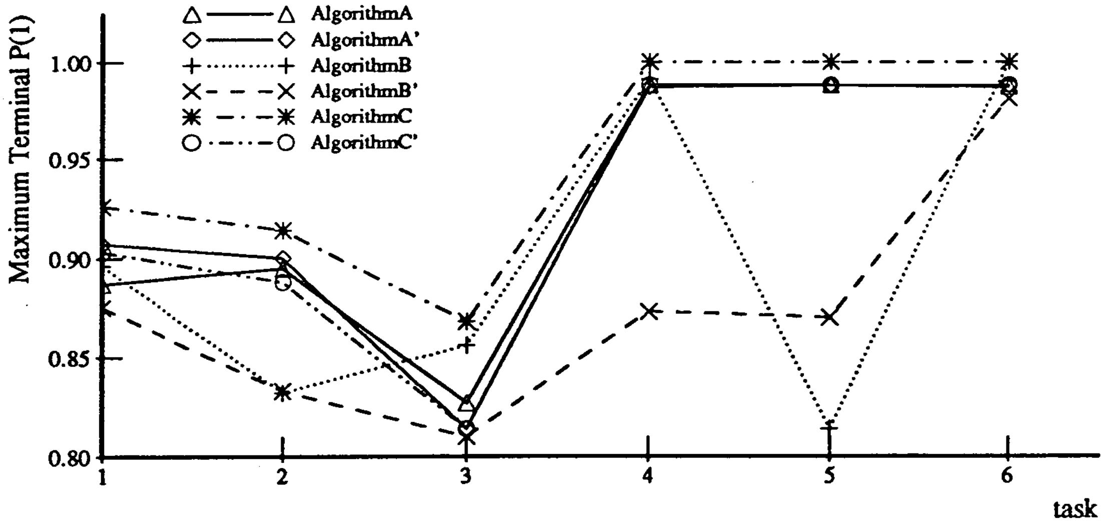


Figure 3.9: All tasks – best performing α, $\eta[t] \sim \mathcal{G}[0, 0.5]$

3.4 Further Developments

Kaelbling, in her thesis [71], adopts a still more statistical approach to reinforcement learning problems. Consider the two-armed bandit problem. If the learning system can record how often it has tried each of the arms as well as some measure of how well it has done in pulling them, then, given a model of the environment, it can calculate the statistical certainty of this measure. This would allow it explicitly to trade off exploration and exploitation. Kaelbling develops and extends such an algorithm, and shows that it out-performs Sutton's reinforcement comparison algorithm.[3]

The focus of most of the rest of the work in reinforcement learning has shifted either to showing how it relates to supervised learning, or to the case of temporally extended tasks, as discussed in the chapters that follow. Additionally, Gullapalli [50] has developed an extension (called SRV) of Barto's A_{RP} [7] to the case of real-valued outputs. Gullapalli has demonstrated the power of the SRV in a number of applications, including a version of the pole balancer and a replication [49] of some results due to Zipser and Anderson [174] which show how the firing of computational units can match those measured from nerve cells.

Both the SRV and the temporal difference (TD) reinforcement learning algorithms, which will be discussed in the next chapter, use something akin to reinforcement comparison. The SRV operates by manipulating two parameters; the mean and variance of a normally distributed random variable, which it uses as its action. It adjusts the mean according to the reinforcement received, and alters the variance according to how well the unit is doing, measured against an absolute standard which is known *a priori*. The variance acts as an explicit arbiter between exploration and exploitation for the unit. The formula which Gullapalli uses for changing the mean is exactly like the one described above, *ie* $(r - b^*)(y - \pi)$. Using $\hat{b}$ in place of b^* should improve SRV too, for the same reason.

[3] Her method of judging the algorithms is actually slightly different from the one here, in that it includes the choices the system makes on the way to deciding which arm is optimal, and so favours fast learning.

Reinforcement comparison plays a somewhat more complex rôle in TD. For secondary reinforcement, in which the system tries to 'associate stimuli with forthcoming reinforcement and then use the information provided by their occurrence to better [*sic*] assign temporal credit' ([147] p 121), the key quantity is the difference between the predicted reinforcements at two successive time steps. Here, it is important that the predicted quantities really are expected reinforcements rather than the 'cross-paired' equivalents of $\hat{b}$, so the improvements obtained in this chapter would not be expected to be available.

The links between reinforcement learning and supervised learning are rather complex. Consider a task such as teaching a computer with a parametrically controlled voicebox to speak. It used to be a commonplace that this could only be done using a form of reinforcement learning; the system could attempt to produce some appropriate sounds, but because it was unclear exactly how the inputs produced the outputs, the information about exactly how (*eg* in which frequencies) the actual output differed from the desired output was useless. It was mysterious how this *distal* error could be related to the *proximal* choice of parameters. The alternative is just 'plain' reinforcement learning, in which the system gets criticised for making the wrong noise, but has to figure out for itself what is amiss. This self-imposed impoverishment of the available information seems bound to slow learning down.

Particularly through the work of Jordan [68, 70], it has since become clear that the conclusion that no characteristics can be used of the distal error other than its mere presence, is too pessimistic. It is possible to learn a forward model of the voicebox, and to use this as the supervisor for an inverse model which is trained so that the composition of the two models is the identity.[4] The forward model provides the mapping from proximal to distal coordinate schemes that allows fully supervised learning to function.

Such methods erode the traditional boundary between reinforcement and supervised learning, and indeed have been demonstrated to be superior in certain applications. Unfortunately, learning the forward model, and particularly

[4] As Jordan and Rumelhart [70] point out, learning an identity map in this way avoids some of the pitfalls of attempting to acquire a direct inverse model as described in Jordan and Rosenbaum [69].

learning it in such a way that error propagation is meaningful[5] is not always easy. In any case, temporally extended tasks still require reinforcement learning methods.

3.5 Conclusions

Reinforcement learning occupies the middle ground between unsupervised and supervised learning techniques. In the simple case covered in this chapter it acts like a supervised learning technique under a particularly difficult condition for learning – noise in the teaching signal. Sutton developed the method of reinforcement comparison as a way of speeding up learning, and demonstrated its efficacy. Williams' subsequent result that the algorithm is performing stochastic gradient ascent, whatever the comparison term, was based solely on a first order term.

Adopting the criterion that the second order term should be minimised as a way of optimising the efficacy of stochastic gradient ascent, it is possible to derive an optimal value for the comparison term. This is justified both empirically, although not strikingly so, and theoretically, since the second order term can be seen to be harmful in certain cases. Rather than being an average of the reinforcement received in the past, which is similar to the covariance rule discussed in the previous chapter, the new term pairs the average reinforcement for doing one action with the probability of doing the other.

This analysis, like Williams', says nothing at all about the convergence of the algorithms. Not only is stochastic gradient ascent merely an average criterion, but also the algorithm cannot be guaranteed to avoid terminating in some local optimum. Gullapalli [51] proved a theorem about the convergence to the correct values of a slight modification of the SRV algorithm. This depends on certain assumptions about decreasing learning rates and regularity conditions.

[5] It is essential that the forward model depends on the system's action selection system and not just its state, otherwise the path between action and model is uninformative for the error propagation. This is non-trivial (Brody, personal communication) if the actions are essentially deterministic functions of the state of the system.

It would be surprising if some such result were not true of these algorithms too. SRV should benefit from the new reinforcement term as well, although it is unclear what impact this would have upon Gullapalli's proof.

Previous assumptions that reinforcement and supervised learning techniques are distinct are being overturned – there are ways of re-casting problems that appear designed for one in terms of the other. The advantages lie with the supervised methods, but only if the environment can provide sufficiently fulsome criticism. In any case there are circumstances, such as temporally extended tasks, in which reinforcement methods are essential. These are the focus of the succeeding chapters.

Chapter 4

Temporal Difference: TD(λ) for General λ

"Academic Mathematics": $\left\{\begin{array}{l}\textit{Obfuscating by numbers,}\\ \textit{Numbing by degrees.}\end{array}\right.$

4.0 Summary

The work in the previous chapter ignored the rôle of time in reinforcement learning. One facet of this issue is making consistent predictions about the future – this lies at the heart of dynamical programming. Watkins [159] analyses Sutton's [148] temporal difference (TD) methods as ways of doing dynamical programming in an incremental fashion. Sutton proves a weak form of convergence for a special case of prediction learning, TD(0), which only involves adjacent time steps. This chapter puts these two pieces of analysis together to prove the same weak form of convergence, but in the general case. This involves arbitrary time steps, but weights more distant ones exponentially less.

Watkins further proves that Q-learning, his closely related prediction and action learning mechanism, converges with probability one. This proof can be applied to show convergence with probability one for TD(0) prediction under slightly different conditions, but only for one particular representation.

4.1 Introduction

Many systems act in temporally extended circumstances, where whole sequences of actions may be performed or states observed. They face three related problems. The first is *prediction;* how they can foresee what some future outcome might be, given some (possibly stochastic) relationship between it and the current state. The second is *temporal credit assignment;* how they can learn exactly which actions are deleterious, if it only becomes apparent after an extended time that some unspecified ones out of an extended sequence were mistaken. The third is *action selection;* if one out of a set of actions can be chosen at the current time step, and, depending on which is selected, one out of a different set of actions can be chosen at the next time step, and so on, then there is an explosion in possible choices. Acting appropriately in such a domain would seem to be computationally intractable.

The associations between these problems may not be obvious. However, the engineering method of dynamical programming (DP) [13] solves them all together. DP comes in many different forms, but it essentially involves two key tricks; the consistency of a *value function* which assigns numerical values to the states, and the evaluation of an implicit policy, which defines the actions the agent performs at each state, giving the value function. Each of these forces a restriction on the scope of the method.

Consider a reinforcement learning system wending its way through some Markovian state space collecting rewards and/or punishments as it moves to a goal. For concreteness, consider the task (which is described in detail in the next chapter) for an agent moving about a finite, but possibly complicated, world trying to get to a terminal position in the shortest time. Given any policy the agent might have, the value function at any state is just the amount of reward/punishment the agent will get on the way to the goal. So if it is punished one unit for every step it makes until it reaches the goal, for instance, then the value of any state is just the number of steps between it and the goal. Crucially, the value function will be consistent between states; imagine the case in which the agent and its environment are deterministic (and furthermore that it always gets to the goal). Each time it enters a state it always makes the same move to the same destination. Therefore, if it happens to go from state S_1 to state S_2, then the value of S_1 should be exactly one more than that of S_2. This is the notion of consistency that is employed.

Were the agent's choices of actions and/or the environment's responses to them stochastic (and stationary), then the value of a state would be the mean number of steps from it to the goal.

Note that it is unnecessary for the policy to be represented explicitly, or indeed be available to the agent in any represented form over which computations are possible. The policy obviously affects the value function in a holistic manner, but if this function is accurate, then it contains all the information about the policy that is relevant to getting to the goal.

Conventional DP consists of a set of methods for calculating value functions and improving policies on the basis of them. For instance, given a value function for a policy, consider the agent trying action B at some state where the policy specifies action A. Consider further if action B takes it to a state whose value is unexpectedly greater than before. Even though this value is based on the existing, sub-optimal, policy, a new policy which is the same as the old one, except suggesting action B rather than A at the original state, will be an improvement.

This links the prediction problem, which is to work out the value function, to the other problems; temporal credit assignment and action selection. If the evaluations of states are appropriately related to the actions that can be chosen, then they can act like secondary reinforcement values to solve the temporal credit assignment problem. Even if the agent is far from a goal, an action can be judged at the time it is made according to whether it moves the agent to states which are more or less highly valued. Similarly, an agent pondering a number of actions can choose to go to the state that is evaluated as closest to the goal. This is discussed in more extensive detail in Watkins' thesis [159], where the links between AI tree-searching techniques and DP are also highlighted. The value function can be seen as cache-ing the results of tree searches, obviating the need for new ones to be done on each occasion.

Unfortunately, it is difficult to represent multiple goals in one value function, except by some more or less *ad hoc* numerical combination or chicanery with the state space. This results from the two 'tricks' mentioned above; consistency in the value function requires it to be unique, and the merely implicit presence of the policy in the value function makes it hard to factor out the different goals. It leads to problems in domains such as navigation, as seen in the next chapter. A further difficulty with DP is that it is too general. It is well known that general methods will be too slow to solve reasonably sized problems. Also, although it is easy to incorporate certain sorts of prior information, such as rough initial estimates of distances from the goal, information about the structure of the environment is harder to code. The next chapter shows one approach at this, through the representation of the states and value function.

The remaining task is prediction, which is the focus of temporal difference methods. Reinforcement comparison, discussed in the previous chapter, is based on something like consistency across trials – the quality of an action is judged relative to how well actions generally do. Temporal difference, though, concentrates on consistency across time within one trial – using the difference between the predictions at one state and the predictions at the next to provide an error. Generally, this error is used to tailor the parameters governing the representation of the value function. Consistency is therefore used as the instrument of instruction. This learning can happen incrementally, as the system observes its environment and its rewards and punishments, and need not be based on a model of the transition structure of the world. By contrast, traditional methods of DP require such a model, and calculate their value functions in iterative fell swoops.

Watkins calls the resulting method (and variants, such as Q-learning, which is discussed below) Incremental Dynamic Programming (IDP). Sutton calls it relaxation planning, since such systems handle the exponential complexities of planning in such a way that they can react instantaneously (if possibly incorrectly) at all times. They are never lost in garbage collection and/or thought.

Some of the earliest work in temporal difference and reactive planning is due to Samuel [132, 133]. His checkers (draughts) playing program tried to learn a consistent value function for board positions, using the discrepancy between the predicted values at each state based on limited depth games-tree searches, and the subsequently predicted values after those numbers of moves had elapsed. A particularly interesting feature of this is the possibility of a trade-off between the complexity of the value function, which might be considered a sub-symbolic issue, and the size of the look-ahead search, which is more symbolic. Samuel was obviously limited in both these domains by the hardware of his day.

Many other proposals along similar lines have been made; Sutton acknowledges the influence of Klopf [74, 75] and in [148] discusses Holland's bucket brigade method for classifier systems [62], and a procedure by Witten [172]. Hampson [52, 53] presents empirical results for a quite similar navigation task to the one described above and in chapter 5. Barto, Sutton and Anderson [10] describe an early TD system which learns how to balance an upended pole, a problem introduced in a further related paper by Michie and Chambers [98]. Watkins [159] also gives further references.

The next section defines TD(λ), shows how to use Watkins' analysis of its relationship with DP to extend Sutton's theorem, and makes some comments about unhelpful state representations. Section 4.3 looks at Q-learning, and uses a version of Watkins' convergence theorem to demonstrate in a particular case the strongest guarantee known for the behaviour of TD(0).

4.2 TD(λ)

Sutton [148] develops the rationale behind TD methods for prediction, and proves that TD(0), a special case with a time horizon of only one step, converges in the mean for observations of an absorbing Markov chain. Although his theorem applies generally, he illustrates the case in point with the example shown in figure 4.1. Here, the chain always starts at state D, and moves left or right with equal probabilities from each state until it reaches the left absorbing barrier A or the right absorbing barrier G. The problem facing TD is predicting the probability it absorbs at the right hand barrier rather than the left hand one, given any of the states as a current location.

The raw information available to the system is sets of sequences of states and

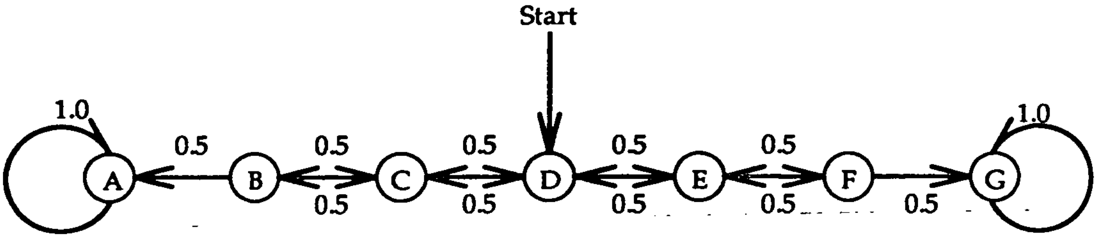


Figure 4.1: Sutton's Markov example. Transition probabilities given above (right to left) and below (left to right) the arrows.

their terminal locations – it initially has no knowledge of the transition probabilities. Sutton describes the supervised Widrow-Hoff [164] technique, which learns to evaluate the states by making the estimates of the probabilities for each place visited on a sequence closer to 1 if that sequence ended up at the right hand barrier, and closer to 0 if it ended up at the left hand one. He shows that this technique is exactly TD(1), one special case of TD, and contrasts it with TD(λ) and particularly TD(0), which tries to make the estimate of probability from one state closer to the estimate from the next, without waiting to see where the sequence might terminate. The discounting parameter λ in TD(λ) determines exponentially the weights of future states based on their temporal distance – smoothly interpolating between $\lambda = 0$, for which only the next state is relevant, and $\lambda = 1$, the Widrow-Hoff case, for which all states are equally weighted. As described in the introduction, it is its obeisance to the temporal order of the sequence that marks out TD.

The following subsections describe Sutton's result for TD(0) and separate out the algorithm from the vector representation of states. They then show how Watkins' analysis provides the wherewithal to extend it to TD(λ), and finally re-incorporate the original representation.

4.2.1 Sutton's Theorem

Following Sutton [148] we consider the case of an absorbing Markov chain, defined by sets and values:

$\mathcal{T}$		the terminal states
$\mathcal{N}$		the non-terminal states
$q_{ij} \in [0,1]$	$i \in \mathcal{N}, j \in \mathcal{N} \cup \mathcal{T}$	the transition probabilities
$x_i \in \Re^c$	$i \in \mathcal{N}$	the vectors representing non-terminal states.
$\bar{z}_j$	$j \in \mathcal{T}$	the expected return for ending at state j
μ_i	$i \in \mathcal{N}$	the probabilities of starting at state i, where $\sum_{i \in \mathcal{N}} \mu_i = 1$.

The chain described above is particularly simple; the returns from the terminal states A and G are deterministically 0 and 1 respectively. This makes the expected return from any state just the probability of absorbing at G.

The estimation system is fed complete sequences $x_{i_1}, x_{i_2}, \ldots x_{i_m}$ of observation vectors, together with their scalar terminal return z. It has to generate for every non-terminal state $i \in \mathcal{N}$ a prediction of the expected return $\mathcal{E}[z|i]$ for starting from that state. If the transition matrix of the Markov chain were completely known, these predictions could be computed as:

$$\mathcal{E}[z|i] = \sum_{j \in \mathcal{T}} q_{ij}\bar{z}_j + \sum_{j \in \mathcal{N}} q_{ij} \sum_{k \in \mathcal{T}} q_{jk}\bar{z}_k + \sum_{j \in \mathcal{N}} q_{ij} \sum_{k \in \mathcal{N}} q_{jk} \sum_{l \in \mathcal{T}} q_{kl}\bar{z}_l + \ldots.$$

Again, following Sutton, let $[M]_{ab}$ denote the ab^{th} entry of any matrix M, $[u]_a$ denote the a^{th} component of any vector u, Q denote the square matrix with components $[Q]_{ab} = q_{ab}, a, b \in \mathcal{N}$, and with h, the vector whose components are $[h]_a = \sum_{b \in \mathcal{T}} q_{ab}\bar{z}_b$, for $a \in \mathcal{N}$, then:

$$\mathcal{E}[z|i] = \left[\sum_{k=0}^{\infty} Q^k h\right]_i = \left[(I - Q)^{-1} h\right]_i \tag{4.1}$$

where the existence of the limit and the equation are proved by Sutton's Appendix A.1. As he states there, they follow from the fact that Q is the transition matrix for the non-terminal states of an absorbing Markov chain, which, with probability one will ultimately terminate.

During the learning phase, linear TD(λ) generates successive vectors $w_1, w_2, \ldots$, changing w after each complete observation sequence. Define $V_n(i) = w_n.x_i$ as the prediction of the terminal reward starting from state i, at stage n in learning. Then, during one such sequence, $V_n(i_t)$ are the intermediate predictions of these terminal values, and, abusing notation somewhat, define also $V_n(i_{m+1}) = z$, the observed terminal reward. Note that in [148], Sutton uses P_t^n for $V_n(i_t)$. TD(λ) changes w according to:

$$w_{n+1} = w_n + \sum_{t=1}^{m} \alpha[V_n(i_{t+1}) - V_n(i_t)] \sum_{k=1}^{t} \lambda^{t-k} \nabla_{w_n} V_n(i_k). \tag{4.2}$$

where α is the learning rate.

Sutton shows that TD(1) is just the normal Widrow-Hoff estimator [164], and also proves the following theorem (theorem 2 in his paper):

> **Theorem** For any absorbing Markov chain, for any distribution of starting probabilities μ_i, for any outcome distributions with finite expected values $\bar{z}_j$, and for any linearly independent set of observation vectors $\{x_i | i \in \mathcal{N}\}$, there exists an $\epsilon > 0$ such that, for all positive $\alpha < \epsilon$ and for any initial weight vector, the predictions of linear TD(0) (with weight updates after each sequence) converge in expected value to the ideal predictions (4.1); that is, if w_n denotes the weight vector after n sequences have been experienced, then
>
> $$\lim_{n \to \infty} \mathcal{E}[w_n . x_i] = \mathcal{E}[z|i] = \left[(I - Q)^{-1} h\right]_i, \forall i \in \mathcal{N}.$$

4.2.2 Subtracting the Representation

Equation 4.2 rather conflates two facets of TD(λ); the underlying temporal algorithm and the representation of the prediction functions V_n. Even though these will remain tangled in the ultimate proof of convergence, it is beneficial to separate them out, since it makes the operation of the algorithm clearer.

The easiest way to do this is to represent V_n as a look-up table, with one entry for each state. This is equivalent to choosing a set of vectors x_i for which just one component is 1 and all the others are 0 for each state, and no state has the same representation. This trivially satisfies the conditions of Sutton's theorem, and also makes the w_n easy to interpret, as each component is the prediction for just one state. In this circumstance, the terms $\nabla_{w_n} V_n(i_k)$ in the sums

$$\sum_{k=1}^{t} \lambda^{t-k} \nabla_{w_n} V_n(i_k)$$

just reduce to counting the number of times the chain has visited each state, exponentially weighted in recency by λ. In this case, as in the full linear case, these terms do not depend on n, only on the states the chain visits. So define the characteristic function for state j:

$$\chi_j(k) = \begin{cases} 1 & \text{if } i_k = j \\ 0 & \text{otherwise} \end{cases}$$

and the prediction function $V_n(i)$ as the entry in the look-up table for state i at stage n during learning. Then equation 4.2 can be reduced to its elemental pieces

$$V_{n+1}(i) = V_n(i) + \sum_{t=1}^{m} \alpha[V_n(i_{t+1}) - V_n(i_t)] \sum_{k=1}^{t} \lambda^{t-k}\chi_i(k) \tag{4.3}$$

in which the value for each state is updated separately.

To illustrate this process, consider the punctate representation of the states B, C, D, E and F in figure 4.1:[1]

$$\begin{array}{ccccc} \chi_B & \chi_C & \chi_D & \chi_E & \chi_F \\ \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \end{array}$$

Then if the observed sequenced is D, C, D, E, F, E, F, G, then the sums

$$\sum_{k=1}^{t} \lambda^{t-k} \nabla_{w_n} V_n(i_k)$$

after each step are:

$$\begin{array}{ccccccc} D & C & D & E & F & E & F \\ \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 1 \\ \lambda \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ \lambda \\ 1+\lambda^2 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ \lambda^2 \\ \lambda+\lambda^3 \\ 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ \lambda^3 \\ \lambda^2+\lambda^4 \\ \lambda \\ 1 \end{pmatrix} & \begin{pmatrix} 0 \\ \lambda^4 \\ \lambda^3+\lambda^5 \\ 1+\lambda^2 \\ \lambda \end{pmatrix} & \begin{pmatrix} 0 \\ \lambda^5 \\ \lambda^4+\lambda^6 \\ \lambda+\lambda^3 \\ 1+\lambda^2 \end{pmatrix} \end{array}$$

[1] States A and G are absorbing and so are not represented.

and component i in this sum is clearly

$$\sum_{k=1}^{t} \lambda^{t-k} \chi_i(k)$$

at time t.

4.2.3 Watkins' Analysis

Watkins [159] shows that a fruitful way of looking at TD estimators is through dynamical programming and its associated contraction mappings. The method starts from the current prediction function $V_n(i), \forall i \in \mathcal{N}$ and shows how to define a whole collection of statistically better estimators $V_{n+1}(i), \forall i \in \mathcal{N}$ based on an observed sequence. Imagine the chain starts at some state i_0, and runs forward through states $i_1, i_2, \ldots$, ultimately absorbing. Define the r-state estimate of i_0 as *either* the estimate $V_n(i_r)$ of state i_r, if the chain has not absorbed after r steps and so $i_r \in \mathcal{N}$, *or* the terminal value z of the sequence, if the chain has absorbed before this time.

Formally, define the random variables

$$\begin{aligned}
V^1_{n,i_0} &= \begin{cases} V_n(i_1) & \text{if } i_1 \in \mathcal{N} \\ z & \text{otherwise} \end{cases} \\
V^2_{n,i_0} &= \begin{cases} V_n(i_2) & \text{if } i_2 \in \mathcal{N} \\ z & \text{otherwise} \end{cases} \\
&\vdots \\
V^r_{n,i_0} &= \begin{cases} V_n(i_r) & \text{if } i_r \in \mathcal{N} \\ z & \text{otherwise} \end{cases} \qquad (4.4) \\
&\vdots
\end{aligned}$$

where i_1 is the first state accessed by the Markov chain in one particular sequence starting from i_0, i_2 is the second and so on, and z is the actual return delivered if the chain gets absorbed before state r is reached. These are random variables, since they depend on the particular sequence of states which will be observed. This is obviously undetermined at i_0. Naturally, they also depend on the initial values $V_n(i)$.

Then,

$$\begin{aligned}\mathcal{E}\left[V^{\tau}_{n,i_0}\right] &= \sum_{t_1\in\mathcal{T}} q_{i_0t_1}\bar{z}_{t_1} + \sum_{i_1\in\mathcal{N}} Q_{i_0i_1} \sum_{t_2\in\mathcal{T}} q_{i_1t_2}\bar{z}_{t_2} + \ldots + \\ &\quad \sum_{i_{\tau-1}\in\mathcal{N}} Q^{\tau-1}_{i_0i_{\tau-1}} \sum_{t_\tau\in\mathcal{T}} q_{i_{\tau-1}t_\tau}\bar{z}_{t_\tau} + \sum_{i_\tau\in\mathcal{N}} Q^{\tau}_{i_0i_\tau} V_n(i_\tau)\end{aligned}$$

whereas it can easily be shown that

$$\begin{aligned}\mathcal{E}[z|i_0] &= \sum_{t_1\in\mathcal{T}} q_{i_0t_1}\bar{z}_{t_1} + \sum_{i_1\in\mathcal{N}} Q_{i_0i_1} \sum_{t_2\in\mathcal{T}} q_{i_1t_2}\bar{z}_{t_2} + \ldots + \\ &\quad \sum_{i_{\tau-1}\in\mathcal{N}} Q^{\tau-1}_{i_0i_{\tau-1}} \sum_{t_\tau\in\mathcal{T}} q_{i_{\tau-1}t_\tau}\bar{z}_{t_\tau} + \sum_{i_\tau\in\mathcal{N}} Q^{\tau}_{i_0i_\tau} \mathcal{E}[z|i_\tau].\end{aligned}$$

So,

$$\mathcal{E}\left[V^{\tau}_{n,i_0}\right] - \mathcal{E}[z|i_0] = \sum_{i_\tau\in\mathcal{N}} Q^{\tau}_{i_0i_\tau}\left(V_n(i_\tau) - \mathcal{E}[z|i_\tau]\right) \tag{4.5}$$

Watkins actually discusses a slightly different case, in which the target values of the predictors are based on *discounted* future returns whose contribution diminishes exponentially with the time until they happen. In this case it is easier to see how the reduction in error is brought about. His analogue of equation 4.5 is effectively

$$\mathcal{E}\left[V^{\tau}_{n,i_0}\right] - \mathcal{E}[z|i_0] = \gamma^{\tau} \sum_{i_\tau\in\mathcal{N}} Q^{\tau}_{i_0i_\tau}\left(V_n(i_\tau) - \mathcal{E}[z|i_\tau]\right)$$

where $\gamma < 1$ is the discount factor. Since $\sum_{i_\tau\in\mathcal{N}} Q^{\tau}_{i_0i_\tau} \leq 1$, as Q is the matrix of a Markov chain, Watkins can guarantee that

$$\max_{i_0}\left|\mathcal{E}\left[V^{\tau}_{n,i_0}\right] - \mathcal{E}[z|i_0]\right| \leq \gamma^{\tau} \max_{i_\tau}\left|V_n(i_\tau) - \mathcal{E}[z|i_\tau]\right|$$

which provides a guarantee that the error of V^{τ}_n will be less than that of V_n, in this weak sense.

The same is not quite true in the non-discounted case, when $\gamma = 1$. The use of the predictor here is motivated by the non-zero probability that the chain will absorb before finishing τ further steps from i_0. In this case, the value of V^{τ}_{n,i_0}, being z, will be unbiased, and so should provide for error reduction. Even if it does not absorb, its value can be no further from what it should be than is the most inaccurate component of V_n. So now, although there is no error reduction due to γ, it is guaranteed that

$$\sum_{i_r \in \mathcal{N}} Q^r_{i_0 i_r} \leq 1$$

with inequality for all those states from which it is possible to absorb within r steps. This does not ensure that

$$\max_{i_0} \left| \mathcal{E}\left[V^r_{n,i_0}\right] - \mathcal{E}[z|i_0] \right| < \max_{i_r} |V_n(i_r) - \mathcal{E}[z|i_r]|$$

since it could be that the maximum is achieved pathologically. However, the estimates for the states that are within r steps of absorption will, on average, improve, and this should, again on average, filter back through to the states that are 'hidden' from termination. In the special case that there is a non-zero probability that the chain can absorb in one step from any non-terminal state, using the norm

$$q \equiv \|Q\| = \max_{\{x:|x|=1\}} |Qx| < 1$$

implies that:

$$\max_{i_0} \left| \mathcal{E}\left[V^r_{n,i_0}\right] - \mathcal{E}[z|i_0] \right| \leq q^r \max_{i_r} |V_n(i_r) - \mathcal{E}[z|i_r]|$$

Following Watkins, we see that if, for $0 < \lambda < 1$, we define a further random variable:

$$V^\lambda_{n,i_0} = (1-\lambda) \sum_{a=1}^{\infty} \lambda^{a-1} V^a_{n,i_0} \tag{4.6}$$

then, in the special case,

$$\max_{i_0} \left| \mathcal{E}\left[V^\lambda_{n,i_0}\right] - \mathcal{E}[z|i_0] \right| \leq q \frac{1-\lambda}{1-q\lambda} < q. \tag{4.7}$$

Watkins points out that in choosing the value of λ, there is a trade off between the bias caused by the error in V_n, and the variance of the real return z. The higher λ, the more significant are the V^r_n for higher values of r, and the more effect the unbiased terminal values will have. This leads to higher variance and lower bias. Conversely, the lower λ, the less significant are the contributions from higher values of r, and the less the effect of the unbiased terminal values. This leads to smaller variance and greater bias.

Now, expanding out the sum in equation 4.6,

$$\begin{aligned} V^{\lambda}_{n,i_0} - V_n(i_0) &= [V_n(i_1) - V_n(i_0)] + \\ &\quad \lambda\,[V_n(i_2) - V_n(i_1)] + \\ &\quad \lambda^2\,[V_n(i_3) - V_n(i_2)] + \ldots \end{aligned} \tag{4.8}$$

defining $V_n(i_s) = z$ for $s > \max\{t | i_t \in \mathcal{N}\}$.

The whole point of defining V^{λ}_{n,i_0} is so that it can be used to make V more accurate. The obvious incremental update rule to achieve this has

$$V_{n+1}(i_0) = V_n(i_0) + \alpha\left[V^{\lambda}_{n,i_0} - V_n(i_0)\right] \tag{4.9}$$

Now, from equation (4.8) we can see that the changes to $V_n(i_0)$ involve summing future values of $V_n(i_{t+1}) - V_n(i_t)$ weighted by ever larger powers of λ. Again following Watkins, this can be calculated through an activity trace based on the characteristic functions $\chi_i(t)$ that were defined earlier as a way of counting how often and how recently the chain has entered particular states.

Then, using index t for the members of the observed sequence, the on-line version of the TD(λ) rule has

$$V_{t+1}(i) = V_t(i) + \alpha\,[V_t(i_{t+1}) - V_t(i_t)] \sum_{k=1}^{t} \lambda^{t-k}\chi_i\,(k). \tag{4.10}$$

Note that Watkins' expression on page 91 of his thesis is identical, except that he defines the activity traces explicitly by $C(x,t) = \sum_{k=1}^{t} \lambda^{t-k}\chi_x(k)$.

For the problem that Sutton treats, the change to V_n is applied off-line, after a complete sequence through the chain. Therefore, if the states through which the chain passes on one sequence are $i_0, i_1, \ldots, i_{m-1} \in \mathcal{N}$, and $i_m \in \mathcal{T}$, it absorbs with return $V_n(i_m) \equiv z$, and V_{n+1} is the new estimator after experiencing the sequence, then

$$\begin{aligned} V_{n+1}(i_0) &= V_n(i_0) + \sum_{t=1}^{m} \alpha\,[V_n(i_{t+1}) - V_n(i_t)] \sum_{k=1}^{t} \lambda^{t-k}\chi_{i_0}(k) \\ V_{n+1}(i_1) &= V_n(i_1) + \sum_{t=2}^{m} \alpha\,[V_n(i_{t+1}) - V_n(i_t)] \sum_{k=1}^{t} \lambda^{t-k}\chi_{i_1}(k) \\ &\vdots \\ V_{n+1}(i_{m-1}) &= V_n(i_{m-1}) + \alpha\,[z - V_n(i_{m-1})] \sum_{k=1}^{m} \lambda^{t-k}\chi_{i_{m-1}}(k), \end{aligned}$$

summing over terms where $i_a = i_b$ (so $\chi_{i_a} \equiv \chi_{i_b}$). Note that these expressions are exactly the same as the TD(λ) weight change formula in equation 4.3.

Thus, the actual TD(λ) algorithm is based on the exponentially weighted sum defined in equation 4.6 of the outcomes of the V_i^r random variables. The mean contraction properties of these variables will therefore determine the mean contraction properties of the overall TD(λ) estimator.

4.2.4 RePresentation

The previous subsection considered the TD(λ) algorithm isolated from the representation Sutton used. Although a number of different representations might be employed, the simplest is the linear one he adopted. Identifying the vectors x with the states they represent gives

$$V_n(x) = w_n.x$$

where w_n is the weight vector at stage n of learning.

The basic algorithm is concerned with the V_n^λ predictor random variables rather than how their values can be used to change the initial predictor V_n. Under the new representation, equation 4.9 no longer makes sense since the states cannot be separated in the appropriate manner. Rather, the information about the error has to be used to update all the weights on which it depends. The appropriate formula, derived from the delta-rule is

$$w_{n+1} = w_n + \alpha \left[V^\lambda_{n,i_0} - V_n(i_0)\right] \nabla_{w_n} V_n(i_0)$$

weighting the error due to state i_0 by the vector representation of i_0. Then the equivalent of equation 4.10 is just Sutton's main TD(λ) equation 4.2.

4.2.5 Proving TD(λ) for General λ

The strategy for proving TD(λ) for general λ in the case of the linear representation is to follow Sutton in considering the expected value of the new prediction weight vector given the observation of a complete sequence, and to follow Watkins in splitting this change into the components due to the equivalents of the V^r random variables, and then summing them.

So define the $V^r_{\cdot,\cdot}$ random variables as in equation 4.4 as

$$V^r_{n,i_0} = \begin{cases} w^r_n.x_{i_r} & \text{if } x_{i_r} \in \mathcal{N} \\ z & \text{otherwise} \end{cases}$$

where x_i are identified with the states in the observed sequence, w^r_n is the current weight vector defining the estimated return, and z is the actual return. Then, after observing a whole sequence of values, w^r_n is updated as:

$$\begin{aligned} w^r_{n+1} &= w^r_n + \alpha \sum_{x_i \in \mathcal{N} \text{ visited}} [V^r_{n,i} - V_n(i)] \nabla_{w_n} V_n(i) \\ &= w^r_n + \alpha \sum_{x_i \in \mathcal{N} \text{ visited}} [V^r_{n,i} - w_n.x_i] x_i . \end{aligned} \tag{4.11}$$

An exact parallel of Sutton's proof procedure turns out to apply to w^r. Define η^s_{ij} as the number of times the s-step transition

$$x_i \to x_{k_1} \to x_{k_2} \ldots \to x_{k_{s-1}} \to x_j$$

occurs, for any intermediate states $x_{k_t} \in \mathcal{N}$. We can then regroup the sum in equation (4.11) in terms of source and destination states of the transitions:

$$\begin{aligned} w^r_{n+1} = w^r_n + \; & \alpha \textstyle\sum_{i \in \mathcal{N}} \sum_{j_r \in \mathcal{N}} \eta^r_{ij_r} [w^r_n.x_{j_r} - w^r_n.x_i] x_i \; + \\ & \alpha \textstyle\sum_{i \in \mathcal{N}} \sum_{j_r \in \mathcal{T}} \eta^r_{ij_r} [z_{j_r} - w^r_n.x_i] x_i \; + \\ & \alpha \textstyle\sum_{i \in \mathcal{N}} \sum_{j_{r-1} \in \mathcal{T}} \eta^{r-1}_{ij_{r-1}} \left[z_{j_{r-1}} - w^r_n.x_i\right] x_i \; + \\ & \vdots \\ & \alpha \textstyle\sum_{i \in \mathcal{N}} \sum_{j_1 \in \mathcal{T}} \eta^1_{ij_1} [z_{j_1} - w^r_n.x_i] x_i \; + \end{aligned} \tag{4.12}$$

where z_j indicates that the return is generated from the distribution due to state j, and the extra terms are generated by the possibility that, from visiting any $x_i \in \mathcal{N}$, the chain absorbs before taking r further steps.

Taking expected values over sequences, we have, for $i \in \mathcal{N}$

$$\begin{aligned} \mathcal{E}[\eta^r_{ij}] &= d_i Q^r_{ij} && \text{for } j \in \mathcal{N} \\ \mathcal{E}[\eta^r_{ij}] &= \textstyle\sum_{k \in \mathcal{N}} d_i Q^{r-1}_{ik} q_{kj} && \text{for } j \in \mathcal{T} \\ \mathcal{E}[\eta^{r-1}_{ij}] &= \textstyle\sum_{k \in \mathcal{N}} d_i Q^{r-2}_{ik} q_{kj} && \text{for } j \in \mathcal{T} \\ &\vdots \\ \mathcal{E}[\eta^1_{ij}] &= d_i q_{ij} && \text{for } j \in \mathcal{T} \end{aligned}$$

where d_i is the expected number of times the Markov chain is in state i in one sequence. For an absorbing Markov chain, it is known that the dependency of this on the probabilities μ_i of starting in the various states is:

$$d_i = \sum_{j\in\mathcal{N}} \mu_j(I-Q)^{-1}_{ji} = [\mu^T(I-Q)^{-1}]_i \tag{4.13}$$

So, substituting into equation (4.12), after taking expectations on both sides, noting that the dependence of $\mathcal{E}[w^r_{n+1}|w^r_n]$ on w^r_n is linear, and using $\bar{w}$ to denote expected values, we get:

$$\begin{aligned}
\bar{w}^r_{n+1} &= \bar{w}^r_n + \alpha \textstyle\sum_{i\in\mathcal{N}} d_i x_i \Bigg[\displaystyle\sum_{j_r\in\mathcal{N}} Q^r_{ij_r}(x_{j_r}.\bar{w}^r_n) - \\
&\quad (x_i.\bar{w}^r_n)\left\{ \sum_{j_r\in\mathcal{N}} Q^r_{ij_r} + \sum_{\substack{j_r\in\mathcal{T},\\ k\in\mathcal{N}}} Q^{r-1}_{ik} q_{kj_r} + \ldots + \sum_{j_1\in\mathcal{T}} q_{ij_1} \right\} + \\
&\quad \sum_{\substack{j_r\in\mathcal{T},\\ k\in\mathcal{N}}} Q^{r-1}_{ik} q_{kj_r}\bar{z}_{j_r} + \sum_{\substack{j_{r-1}\in\mathcal{T},\\ k\in\mathcal{N}}} Q^{r-2}_{ik} q_{kj_{r-1}}\bar{z}_{j_{r-1}} + \ldots + \sum_{j_1\in\mathcal{T}} q_{ij_1}\bar{z}_{j_1} \Bigg].
\end{aligned}$$

Now, define X to be the matrix whose columns are x_i, so $[X]_{ab} = [x_a]_b$, and D to be the diagonal matrix $[D]_{ab} = \delta_{ab} d_a$, where δ_{ab} is the Kronecker delta. Remembering that $h_i = \sum_{j\in\mathcal{T}} q_{ij}\bar{z}_j$, and converting to matrix form, we have

$$\bar{w}^r_{n+1} = \bar{w}^r_n + \alpha XD\left[Q^r X^T \bar{w}^r_n - X^T\bar{w}^r_n + (Q^{r-1} + Q^{r-2} + \ldots + I)h\right] \tag{4.14}$$

since

$$\sum_{j_r\in\mathcal{N}} Q^r_{ij_r} + \sum_{\substack{j_r\in\mathcal{T},\\ k\in\mathcal{N}}} Q^{r-1}_{ik} q_{kj_r} + \ldots + \sum_{j_1\in\mathcal{T}} q_{ij_1} = 1$$

as this covers all the possible options for r−step moves from state i.

Now define $[\bar{e}^*]_i = \mathcal{E}[z|i]$, then, from equation 4.1, we also have that

$$\begin{aligned}
\bar{e}^* &= [\mathcal{E}[z|i]] \\
&= h + Qh + Q^2h + \ldots
\end{aligned}$$

$$
\begin{aligned}
&= (I + Q + Q^2 + \ldots + Q^{r-1})h + Q^r(I + Q + Q^2 + \ldots + Q^{r-1})h + \ldots \\
&= \sum_{k=0}^{\infty} [Q^r]^k (I + Q + Q^2 + \ldots + Q^{r-1})h \\
&= (I - Q^r)^{-1} (I + Q + Q^2 + \ldots + Q^{r-1})h \qquad (4.15)
\end{aligned}
$$

where the sum converges by the argument in Sutton's appendix A.1.

Now multiplying equation (4.14) on the left by X^T, we get

$$
\begin{aligned}
X^T\bar{w}^r_{n+1} &= X^T\bar{w}^r_n + \\
&\quad \alpha X^T XD\left[(I + Q + Q^2 + \ldots + Q^{r-1})h + Q^r X^T \bar{w}^r_n - X^T\bar{w}^r_n\right] \\
&= \left[I - \alpha X^T XD(I - Q^r)\right] X^T\bar{w}^r_n + \\
&\quad \alpha X^T XD(I + Q + Q^2 + \ldots + Q^{r-1})h
\end{aligned}
$$

So, subtracting $\bar{e}^*$ from both sides of the equation, and noting that from equation 4.15 $(I - Q^r)\bar{e}^* = (I + Q + Q^2 + \ldots + Q^{r-1})h$, we get

$$
\begin{aligned}
\left[X^T\bar{w}^r_{n+1} - \bar{e}^*\right] &= \left[I - \alpha X^T XD(I - Q^r)\right] X^T\bar{w}^r_n + \alpha X^T XD(I - Q^r)\bar{e}^* - \bar{e}^* \\
&= \left[I - \alpha X^T XD(I - Q^r)\right] \left[X^T\bar{w}^r_n - \bar{e}^*\right].
\end{aligned}
$$

Now the Watkins construction of TD(λ) discussed in the previous section tells us that, starting from $w^r_n = w^\lambda_n, \forall r$,

$$
w^\lambda_{n+1} = (1 - \lambda) \sum_{r=1}^{\infty} \lambda^{r-1} w^r_{n+1}
$$

Therefore, since $(1 - \lambda) \sum_{r=1}^{\infty} \lambda^{r-1} = 1$,

$$
\begin{aligned}
\left[X^T\bar{w}^\lambda_{n+1} - \bar{e}^*\right] &= \left\{(1 - \lambda) \textstyle\sum_{r=1}^{\infty} \lambda^{r-1} \left[I - \alpha X^T XD(I - Q^r)\right]\right\} \left[X^T\bar{w}^\lambda_n - \bar{e}^*\right] \\
&= \left\{I - \alpha X^T XD\left(I - (1 - \lambda)Q\left[I - \lambda Q\right]^{-1}\right)\right\} \left[X^T\bar{w}^\lambda_n - \bar{e}^*\right]
\end{aligned}
$$

where $\bar{w}^\lambda$ are the expected weights from the TD(λ) procedure. The sum

$$
(1 - \lambda) \sum_{r=1}^{\infty} \lambda^{r-1} Q^r = (1 - \lambda)Q\left[I - \lambda Q\right]^{-1} \qquad (4.16)
$$

converges since $0 < \lambda < 1$.[2]

[2] Note also the similarity between equations 4.7 and 4.16. The latter is just the matrix equivalent of the former.

Define

$$\Delta_\lambda = \left\{ I - \alpha X^T X D \left(I - (1-\lambda) Q \left[I - \lambda Q \right]^{-1} \right) \right\}$$

then the convergence of TD(λ) in the mean will be shown, if it can be demonstrated that $\exists \epsilon > 0$ such that for $0 < \alpha < \epsilon$, $\lim_{n \to \infty} \Delta_\lambda^n = 0$. In the case that $\lambda = 0$ (for which this formula remains correct), and X has full rank, Sutton proves this on pages 26-28 of [148], by showing successively that $D(I - Q)$ is positive, and that $X^T X D(I - Q)$ has a full set of eigenvalues all of whose real parts are positive, and finally that α can thus be chosen such that all eigenvalues of $I - \alpha X^T X D(I - Q)$ are less than 1 in modulus.

Almost all of Sutton's proof applies *mutatis mutandis* to the case in which $\lambda \neq 0$. However, to make this chapter complete, a full account of the proof will be given, highlighting the single altered section.

The equivalent of $D(I - Q)$ is $D\left(I - (1-\lambda)Q\left[I - \lambda Q\right]^{-1}\right)$. This will be positive definite, according to a lemma by Varga [158] and an observation by Sutton, if

$$S_\lambda = D\left(I - (1-\lambda)Q\left[I - \lambda Q\right]^{-1}\right) + \left\{ D\left(I - (1-\lambda)Q\left[I - \lambda Q\right]^{-1}\right) \right\}^T$$

can be shown to be strictly diagonally dominant with positive diagonal entries. This is the part of the proof that differs from Sutton, but even here, its structure is rather similar.

Define

$$S_r = D(I - Q^r) + \left\{ D(I - Q^r) \right\}^T .$$

Then

$$\begin{aligned} [S_r]_{ii} &= [D(I - Q^r)]_{ii} + [\{D(I - Q^r)\}]_{ii} \\ &= 2 d_i [I - Q^r]_{ii} \\ &= 2 d_i (1 - [Q^r]_{ii}) \\ &> 0, \end{aligned}$$

since Q is the matrix of an absorbing Markov chain, and so Q^r has no diagonal elements ≥ 1. Therefore S_r has positive diagonal elements.

Similarly,

$$[S_r]_{ij} = d_i [I - Q^r]_{ij} + d_j [I - Q^r]_{ji}$$

$$
\begin{aligned}
&= -d_i[Q^r]_{ij} - d_j[Q^r]_{ji} \\
&\leq 0, \ i \neq j
\end{aligned}
$$

since all the elements of Q, and hence also those of Q^r, are positive.

S_r will be strictly diagonally dominant if $\sum_j [S_r]_{ij} \geq 0$, with strict inequality for some i.

$$
\begin{aligned}
\textstyle\sum_j [S_r]_{ij} &= \textstyle\sum_j d_i[I - Q^r]_{ij} + \sum_j d_j[I - Q^r]_{ji} \\
&= d_i \textstyle\sum_j [I - Q^r]_{ij} + [\mathbf{d}^T(I - Q^r)]_i \\
&= d_i(1 - \textstyle\sum_j [Q^r]_{ij}) + [\mu^T(I - Q)^{-1}(I - Q^r)]_i \text{ by equation (4.13)} \\
&= d_i(1 - \textstyle\sum_j [Q^r]_{ij}) + [\mu^T(I + Q + Q^2 + \ldots + Q^{r-1})]_i \\
&\qquad \text{since } I - Q^r = (I - Q)(I + Q + Q^2 + \ldots + Q^{r-1}) \\
&\geq 0
\end{aligned}
$$

since $\sum_j [Q^r]_{ij} \leq 1$, as the chain is absorbing, and $[Q^s]_{ij} \geq 0, \forall s$. Also, since $\exists i : \mu_i > 0$, the inequality is strict for that i.

Now, since S_r is strictly diagonally dominant for all $r \geq 1$,

$$
\begin{aligned}
S_\lambda &= (1 - \lambda) \sum_{r=1}^{\infty} \lambda^{r-1} S_r \\
&= D\left(I - (1 - \lambda)Q[I - \lambda Q]^{-1}\right) + \left\{D\left(I - (1 - \lambda)Q[I - \lambda Q]^{-1}\right)\right\}^T
\end{aligned}
$$

is strictly diagonally dominant too, and therefore $D\left(I - (1 - \lambda)Q[I - \lambda Q]^{-1}\right)$ is positive definite.

Next, $X^TXD\left(I - (1 - \lambda)Q[I - \lambda Q]^{-1}\right)$ should have a full set of eigenvalues all of whose real parts are positive. X^TX, D and $\left(I - (1 - \lambda)Q[I - \lambda Q]^{-1}\right)$ are all non-singular, which ensures that the set is full. So, let ψ and u be any eigenvalue-eigenvector pair, with $u = a + bi$ and $v = (X^TX)^{-1}u \neq 0$, so $u = (X^TX)v$. Then

$$
\begin{aligned}
u^*D\left(I - (1 - \lambda)Q[I - \lambda Q]^{-1}\right)u &= v^*X^TXD\left(I - (1 - \lambda)Q[I - \lambda Q]^{-1}\right)u \\
&= v^*\psi u \\
&= \psi v^*(X^TX)v \\
&= \psi(Xv)^*Xv
\end{aligned}
$$

where '*' indicates the conjugate transpose. This implies that

$$\mathrm{Re}\ \left(u^* D \left(I - (1-\lambda)Q\,[I-\lambda Q]^{-1}\right) u\right) = \mathrm{Re}\ (\psi (X\nu)^* X\nu)$$

or equivalently,

$$\begin{aligned}\{(X\nu)^* X\nu\}\,\mathrm{Re}\,(\psi) &= a^T D \left(I - (1-\lambda)Q\,[I-\lambda Q]^{-1}\right) a + \\ &\quad\; b^T D \left(I - (1-\lambda)Q\,[I-\lambda Q]^{-1}\right) b.\end{aligned}$$

Since the right side (by positive definiteness) and $(X\nu)^* X\nu$ are strictly positive, the real part of ψ must be too.

Furthermore, u must also be an eigenvector of

$$I - \alpha X^T X D \left(I - (1-\lambda)Q\,[I-\lambda Q]^{-1}\right)$$

since

$$\begin{aligned}\left[I - \alpha X^T X D \left(I - (1-\lambda)Q\,[I-\lambda Q]^{-1}\right)\right] u &= u - \alpha\psi u \\ &= (1-\alpha\psi)u.\end{aligned}$$

Therefore, all the eigenvalues of $I - \alpha X^T X D \left(I - (1-\lambda)Q\,[I-\lambda Q]^{-1}\right)$ are of the form $1 - \alpha\psi$ where $\psi \equiv \upsilon + \phi i$ has positive υ. Take

$$0 < \alpha < \frac{2\upsilon}{\upsilon^2 + \phi^2}$$

for all eigenvalues ψ, and then all the eigenvalues $1 - \alpha\psi$ of the iteration matrix are guaranteed to have modulus less than one. So, by another theorem of Varga [158]

$$\lim_{n\to\infty} \left[I - \alpha X^T X D \left(I - (1-\lambda)Q\,[I-\lambda Q]^{-1}\right)\right]^n = 0.$$

This implies that the expected values of the estimates will converge to their desired values as more sequences are observed.

4.2.6 Non-Independence of the x_i

In moving from Watkins' representation-free proof to Sutton's treatment of the linear case, one key assumption was that the x_i, the vectors representing the states, were all independent. In the case that they are not, *ie* matrix X does not have full rank, the proof breaks down. Even though $D\left(I-(1-\lambda)Q\left[I-\lambda Q\right]^{-1}\right)$ is still positive, $X^TXD\left(I-(1-\lambda)Q\left[I-\lambda Q\right]^{-1}\right)$ will no longer have a full set of eigenvalues with positive real parts, since

$$Y=\left\{y|XD\left(I-(1-\lambda)Q\left[I-\lambda Q\right]^{-1}\right)y=0\right\}\neq\{0\}.$$

Saying what will happen to the expected values of the weights turns out to be easier than understanding it. Choose a basis:

$$\{\mathbf{b}_1,\ldots,\mathbf{b}_p,\mathbf{b}_{p+1},\ldots,\mathbf{b}_n\}\text{ for }\Re^n,$$

with $\mathbf{b}_i\in Y$, for $1\leq i\leq p$ being a basis for Y.

Then the proof above applies exactly to $\mathbf{b}_{p+1},\ldots,\mathbf{b}_n$; that is there exists some $0<\alpha<1$ such that:

$$\lim_{n\to\infty}\left[I-\alpha X^TXD\left(I-(1-\lambda)Q\left[I-\lambda Q\right]^{-1}\right)\right]^n\mathbf{b}_i=0,\text{ for }p<i\leq n.$$

Also,

$$\left[I-\alpha X^TXD\left(I-(1-\lambda)Q\left[I-\lambda Q\right]^{-1}\right)\right]^n\mathbf{b}_i=\mathbf{b}_i,\text{ for }1\leq i\leq p$$

by the definition of Y.

So, writing

$$X^T\bar{\mathbf{w}}_0^\lambda-\bar{\mathbf{e}}^*=\sum_{i=1}^n\beta_i\mathbf{b}_i$$

we have that

$$\begin{aligned}X^T\bar{\mathbf{w}}_n^\lambda-\bar{\mathbf{e}}^* &= \left[I-\alpha X^TXD(I-Q^r)\right]^n\left[X^T\bar{\mathbf{w}}_0^\lambda-\bar{\mathbf{e}}^*\right]\\ &= \left[I-\alpha X^TXD(I-Q^r)\right]^n\left[\sum_{i=1}^n\beta_i\mathbf{b}_i\right]\\ &\to \sum_{i=1}^p\beta_i\mathbf{b}_i,\text{ as }n\to\infty\end{aligned}$$

and so

$$XD\left(I-(1-\lambda)Q\left[I-\lambda Q\right]^{-1}\right)\left[X^T\bar{\mathbf{w}}_n^\lambda-\bar{\mathbf{e}}^*\right]\to 0\text{ as }n\to\infty. \qquad (4.17)$$

To help understand this result, consider the equivalent for the Widrow-Hoff estimator, TD(1). There we have

$$\mathrm{XD}\left[\mathrm{X}^{\mathrm{T}}\bar{w}_{n}^{1}-\bar{e}^{*}\right]\rightarrow 0 \text{ as } n\rightarrow\infty. \tag{4.18}$$

and so, since D is symmetric,

$$\begin{aligned}
\frac{\delta}{\delta\bar{w}_{n}^{1}}\left[\mathrm{X}^{\mathrm{T}}\bar{w}_{n}^{1}-\bar{e}^{*}\right]^{\mathrm{T}}\mathrm{D}\left[\mathrm{X}^{\mathrm{T}}\bar{w}_{n}^{1}-\bar{e}^{*}\right] &= \mathrm{X}(\mathrm{D}+\mathrm{D}^{\mathrm{T}})\left[\mathrm{X}^{\mathrm{T}}\bar{w}_{n}^{1}-\bar{e}^{*}\right] && (4.19)\\
&= 2\mathrm{XD}\left[\mathrm{X}^{\mathrm{T}}\bar{w}_{n}^{1}-\bar{e}^{*}\right] && (4.20)\\
&\rightarrow 0 \text{ as } n\rightarrow\infty, && (4.21)
\end{aligned}$$

by equation 4.18. For weights w, the square error for state i is $\left|[\mathrm{X}^{\mathrm{T}}w-\bar{e}^{*}]\right|_{i}^{2}$, and the expected number of visits to i in one sequence is d_i. Therefore the quadratic form

$$\left[\mathrm{X}^{\mathrm{T}}w-\bar{e}^{*}\right]^{\mathrm{T}}\mathrm{D}\left[\mathrm{X}^{\mathrm{T}}w-\bar{e}^{*}\right]$$

is just the loaded square error between the predictions at each state and their desired values, where the loading factors are just the expected frequencies with which the Markov chain hits those states. The condition in equation 4.21 implies that the expected values of the weights tend to be so as to minimise this error. This seems ideal behaviour.

This is false in general for $\lambda \neq 1$. Intuitively, the problem is that the trade-off between bias and variance has returned to haunt. For the case where X is full rank, Sutton shows that it is harmless to use the inaccurate estimates from the next state $x_{i_{t+1}}.w$ to criticise the estimates for the current state $x_{i_t}.w$. Where X is not full rank, these successive estimates become biased on account of what might be deemed their 'shared' representation. The amount of extra bias is then related to the amount of sharing, and the frequency with which the transitions happen from one state to the next.

Formalising this leads to a second problem; the interaction between the two statistical processes of calculating the mean weight and calculating the expected number of transitions. Comparing equations 4.17 and 4.18, one might expect

$$\lim_{n\rightarrow\infty}\frac{\delta}{\delta\bar{w}_{n}^{\lambda}}\left[\mathrm{X}^{\mathrm{T}}\bar{w}_{n}^{\lambda}-\bar{e}^{*}\right]^{\mathrm{T}}\mathrm{D}\left(\mathrm{I}-(1-\lambda)\mathrm{Q}\left[\mathrm{I}-\lambda\mathrm{Q}\right]^{-1}\right)\left[\mathrm{X}^{\mathrm{T}}\bar{w}_{n}^{\lambda}-\bar{e}^{*}\right]=0 \tag{4.22}$$

However, the key step in proving equation 4.21 was the transition between equations 4.19 and 4.20, which relied on the symmetry of D. Since Q is not in general symmetric, this will not happen. Defining

$$\begin{aligned} g(w') &= \frac{\delta}{\delta w}\left[X^T w - \bar{e}^*\right]^T D\left(I - (1-\lambda)Q\left[I - \lambda Q\right]^{-1}\right)\left[X^T w' - \bar{e}^*\right] \\ &= XD\left(I - (1-\lambda)Q\left[I - \lambda Q\right]^{-1}\right)\left[X^T w' - \bar{e}^*\right] \qquad (4.23) \end{aligned}$$

all that will actually happen is that $g(\bar{w}_n^\lambda) \to 0$ as $n \to \infty$.

One could arrange for equation 4.22 to hold by completing the derivative, *ie* by having a learning rule whose effect is

$$\begin{aligned} &\left[X^T \bar{w}_{n+1}^\lambda - \bar{e}^*\right] = \\ &\left\{I - \alpha X^T X\left(D - \tfrac{1-\lambda}{2}\left[DQ(I-\lambda Q)^{-1} + (I - \lambda Q^T)^{-1} Q^T D^T\right]\right)\right\}\left[X^T \bar{w}_n^\lambda - \bar{e}^*\right] \end{aligned}$$

The Q^T term effectively arranges for backwards as well as forwards learning to occur, so that not only would state i_t adjust its estimate to make it more like state i_{t+1}, but also state i_{t+1} would adjust its estimate to make it more like state i_t.

Werbos [161] and Sutton (personal communication) both discuss this point in the context of the gradient descent of TD(λ) rather than its convergence for non-independent x_t. Werbos presents an example based on a learning technique very similar to TD(0), in which completing the derivative in this manner makes the rule converge away from the true solution. He faults this procedure for introducing the unhelpful correlations between the learning rule and the random moves from one state to the next which were mentioned above. He pointed out the convergence in terms of functions g in equation 4.23 in which the w' weights are fixed.

Sutton presents an example as an intuition pump to help explain the result. At first sight, augmenting TD(λ) seems quite reasonable; after all it could quite easily happen by random chance of the training sequences that the predictions for one state are more accurate than the predictions for the next at some point. Therefore, training the second to be more like the first would be helpful. However, Sutton points out that time and choices always move forward, not backwards. Imagine the case shown in figure 4.2, where the numbers over the arrows represent the transition probabilities, and the numbers at the terminal nodes represent terminal absorbing values.

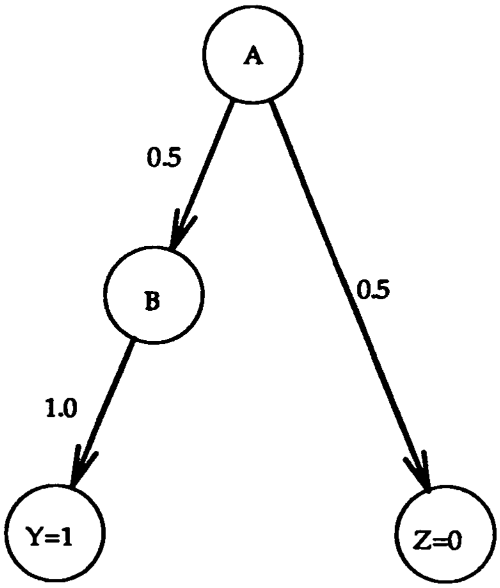


Figure 4.2: Didactic example of the pitfalls of backwards training. If Y and Z are terminal states with values 1 and 0 respectively, what value should be assigned to states A and B respectively?

Here, the value of state A is reasonably 1/2, as there is 50% probability of ending up at either Y or Z. The value of state B, though should be 1, as the chain is certain to end up at Y. Training forwards will give this, but training backwards too will make the value of B tend to 3/4. In Werbos' terms, there are correlations between the weights and the possible transitions that count against the augmented term. Incidentally, this result does not affect TD(1), because the training values, being just the terminal value for the sequence, bear no relation to the transitions themselves, just the number of times each state is visited.

Coming back to the case where X is not full rank. TD(λ) for $\lambda \neq 1$ will converge, but away from the 'best' value, to a degree that is determined by the matrix

$$\left(I - (1-\lambda)Q\,[I - \lambda Q]^{-1}\right).$$

4.3 Q-Learning

Both Sutton's proof and the proofs in the previous section are weak. They accomplish only the nadir of stochastic convergence, *viz* convergence of the mean, rather than the zenith, *viz* convergence with probability one. Watkins [159] was able to prove the latter form of convergence for a form of prediction and action learning he called Q-learning. Fortunately, his theorem applies almost directly to the discounted predictive version of TD(0), albeit without the representation, and so provides the first strong proof for a temporal difference method.

Q-learning wraps up prediction and control, rather like the case described in the introduction to this chapter, and the navigation example in the next. A TD-like method is used to estimate the value, called the Q-value, of doing a particular action in a particular state. These are learnt on the basis of exploration through the space of actions and states. Once they have been acquired, the optimal action is just the one whose Q-value is highest in a state. Control using TD directly is slightly different, as will be described in the next chapter. However, Watkins' proof can be applied directly to prediction, by imagining the special case in which the Markov chain described in the previous section is controlled, but there is only one action possible in every state. This makes the value of doing that action in that state just the prediction value of the state itself.

Theorem (after Watkins [159])

If $\mathcal{N}$ is a finite set of non-terminal states of an absorbing Markov process, $\mathcal{T}$ is a set of terminal states with rewards with expected values $\bar{z}_j, j \in \mathcal{T}$, $V_0(i) \in \Re, i \in \mathcal{N}$ is a collection of starting values, $0 < \gamma < 1$ is a discount rate, $q_{ij} = Q_{ij}$ is the transition matrix, and $V_*(i)$ is the optimal value function for state $i \in \mathcal{N}$, where

$$V_*(i) = \sum_{j \in \mathcal{N}} q_{ij} V_*(j) + \sum_{j \in \mathcal{T}} q_{ij} \bar{z}_j = \mathcal{E}[z|i].$$

Then for any sequence of observations $\{i_n, j_n, z_n\}, n \geq 1$ during learning, where the chain starts at state $i_n \in \mathcal{N}$, ends up at $j_n \in \mathcal{N} \cup \mathcal{T}$, and

$$z_n = \begin{cases} 0 & \text{if } j_n \in \mathcal{N} \\ z & \text{if } j_n \in \mathcal{T}, \text{ where } z \text{ is the actual terminal value} \end{cases}$$

define recursively

$$V_n(i) = \begin{cases} (1 - \alpha_n) V_{n-1}(i) + \alpha_n [z_n + \gamma V_{n-1}(j_n)] & \text{if } i = i_n, \\ V_{n-1}(i) & \text{otherwise,} \end{cases}$$

where $\alpha_n \in \Re$, and let $n_i(a), a \geq 1$ be the a^{th} time at which $i_{n_i(a)} = i$.

Then, if the z have finite variance, $0 < \alpha_n < 1, \forall n$, and

$$\sum_{a=1}^{\infty} \alpha_{n_i(a)} = \infty, \quad \sum_{a=1}^{\infty} \alpha^2_{n_i(a)} < \infty, \forall i \in \mathcal{N}, \tag{4.24}$$

then, with probability one,

$$\lim_{n \to \infty} V_n(i) \text{ exists, and equals } V_*(i).$$

There are various differences between this version of TD prediction and the one discussed in the previous section. Here, learning is on-line, that is the V values are changed for every observation. Also, learning need not proceed along an observed sequence – there is no requirement that $j_n = i_{n+1}$, and so uncoupled or disembodied moves can be used.[3] The conditions in equation 4.24 have as a consequence that every state must be visited infinitely often. Also note that Sutton's proof, since it is confined to showing convergence in the mean works for a fixed learning rate α, whereas Watkins', in common with all other stochastic convergence proofs, requires α_n to tend to 0. The discount factor $\gamma < 1$ plays a very central rôle in Watkins proof, in ensuring that bounds can be placed on the effects of early V_n values. The proof should still work, though, for an absorbing Markov chain with $\gamma = 1$, as the ever increasing probability of absorption should achieve the same effect.

4.4 Further Developments

Watkins' proof requires two principal extensions; to the case of TD(λ) for $\lambda \neq 0$, and to the case of more interesting representations. Neither of these looks obvious; the first because altering λ will bring back sequence-based rather than transition-based observation, and will disrupt Watkins' proof method. Proving convergence with even the simplest forms of more powerful representations, such as Sutton's linear one, will also require more complicated methods, perhaps along the lines of Kushner and Clark [77] or White [162].

[3] This was one of Watkins' main motivations, as it allows his system to learn about the effects of actions it believes to be sub-optimal.

Given this, though, TD(λ) will 'inherit' a similar theoretical base to other methods of stochastic approximation. For these also there are proofs of convergence in special cases under more or less restrictive conditions, most of which are violated for practical purposes. For example, on account of its harmful effects on time to convergence, it is rare for applications to use decreasing values of the learning rate α, even though this is a key condition for convergence. Also, it is possible to use more sophisticated representations, such as general non-linear approximators (for example back-propagation networks), or exotic computer science techniques, such as kd-trees [115], even though again there may be no proofs of convergence.

Coming back to how an agent should act in a temporally extended domain, Williams and Baird [166] have recently done some work on other asynchronous forms of IDP. Under Q-learning the policy the agent will follow, *ie* the function relating states to actions is (possibly stochastically) determined by the values of the states. Williams and Baird consider the circumstance in which policy and value function are separately represented, and define two kinds of update operator; one for the value function, which makes it closer to evaluating the current policy at the current state, and one for the policy, which tends to make it more optimal with respect to the current value function at the current state. The second of these is the asynchronous form of another traditional method of DP called policy iteration, which finds the optimal policy by iteratively calculating the value function for the current one and choosing better local actions. These operators can potentially be applied in different orders in different states.

Unfortunately this process is prone to cycling – it is only guaranteed to converge under fairly strong conditions, whose satisfaction would be hard to determine *a priori*. If it does converge, though, then it will be correct. It may simply be too general a decomposition of the prediction and action problems. In terms of this chapter, there is nothing essentially new about the prediction method Williams and Baird use, their focus is rather on the interaction during learning between the value function and the policy.

4.5 Conclusions

Watkins' original analysis in his thesis of the relationship between temporal difference (TD) estimation and dynamic programming has been used to extend Sutton's proof that TD(0) prediction converges in the mean to the case of TD(λ) for general λ. The linear function approximation that Sutton used does not function correctly if the states are represented by non-independent vectors, and indeed only the Widrow-Hoff predictor TD(1) gives well motivated answers in this case. TD(λ) still converges, but to a sub-optimal value.

Sutton's proof and the extension here are only of convergence in the mean; Watkins proved that Q-learning, his method of incremental dynamical programming, converges with probability one, and this proof has been applied to a special case of TD(0). It is unclear how to alter it further to prove convergence for TD(λ) or for more complicated representations of the state-space.

Chapter 5

Temporal Difference in Spatial Learning

"Maze": A diversionary tool for finding and finding out.

5.0 Summary

This chapter applies temporal difference (TD) methods to navigation. The traditional emphasis on map building is criticised for ignoring the implicit and notoriously intractable planning problem, and the TD method for relaxation planning is motivated and described. A task due to Barto, Sutton and Watkins [11] is shown to be similar to Morris' open field water maze [101], and the solution to the former is pondered in the light of the latter. Extensions are discussed in the areas of goal-free learning and alternate representations of the world, including ones that code places differently according to the orientation of the agent.

5.1 Introduction

A computational agent's map-building and navigation system faces two key questions:

where are the agent and the things it needs to find in its environment,
how can it get to them?

Representations neglecting one of these in favour of the other are unlikely to suffice. As is very well described in [113], philosophy has traditionally concentrated on notions of the 'where', trying to choose between relative space, constructed from the relationships between objects, and absolute space, in which locations exist independently of the objects they may contain. From the experience of AI, on the other hand, it is apparent that the 'how' will be computationally very difficult to address. It suffers from essentially all the problems of computationally intractable planning. This problem is particularly acute when the agent has to learn the map which it is to use for navigation.

We can delude ourselves into overlooking the close connection between these two questions because of the way in which we seem able to use maps. Not only do they provide a very compact representation of the contents of space, helping answer the 'where', but also they permit us to use an apparently special form of visual inference, with which we have particular facility. Unfortunately, the mechanistic grounding of this skill is still mysterious, and in any case, for many tasks, the problem remains of learning such a rich representation by exploration, as opposed to just using a pre-existing map.

The mammalian hippocampus has long been suspected of being involved in mapping and navigation. The initial evidence was the discovery of place cells in this structure in the rat [112]. These are cells that fire when the rat is at particular locations in its environment. The hypotheses about the rôle of the hippocampus have been both empirically and theoretically honed since then, resulting, on the way, in O'Keefe and Nadel's monumental book [113]. They reviewed almost all the then available evidence on the hippocampus, and fitted it into their theory that it is a cognitive mapping system. They considered data from many different species; for instance bats, who are able to construct and use maps of their living areas, have large hippocampi. Incidentally, O'Keefe and Nadel also suggested that in humans, the right half of this structure performs this spatial mapping function, whereas the left half is responsible for semantic mapping. Gallistel [45], in a seminal tome on animal representation, presents substantial bodies of evidence that many species have the ability to construct

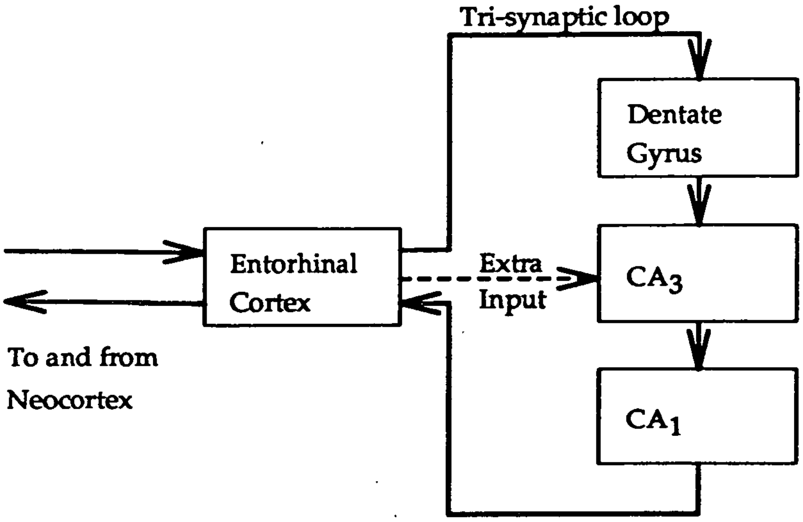


Figure 5.1: Schematic diagram of the hippocampus showing dual inputs

and use maps.

For reference, figure 5.1 shows a block diagram of the structure and some of the connections of the hippocampus. It receives input from all the sensory modalities, and sends its output to neocortex. Place cells are seen in the CA_3 and CA_1 regions, and the entorhinal cortex. The tri-synaptic loop, which has long been known, is augmented on the diagram by an additional path from the entorhinal cortex to CA_3.

Unfortunately, the content of the O'Keefe and Nadel's theory, and the representation of space it supposes, are aimed almost exclusively at answering the 'where', ignoring the 'how'. For instance, on the latter, they suggest that:

> 'Let us imagine our animal has explored an environment containing food, while sated, and has built a map of that environment including the location of the food. At a later time the animal, now hungry, finds itself in the same environment... How might the map guide the animal's behaviour towards the food? Let us assume that, in addition to a generalised subliminal excitation of all place representations connected together into one map which ensues whenever two or more parts of a map are excited, hunger specifically excites those place representations where food has been experienced. These two will sum, bringing the representation of the place containing food in that environment close to activation. Since the place that the animal

> occupies will also be receiving afferent drive [the place cells will be firing based on the animal's current location] there will be two sets of place representations which are potentiated. The programming system can now search for the appropriate motor programme which activates both of these representations simultaneously; this programme will take the animal from where it is to where the food is.' [113]-p227

This rather overlooks the complexity of the 'how'. Just starting from arbitrary distributions of firing over two sets of place cells – one representing the current location, the other the goal location – it is unclear how the whole sequence of actions that may be necessary to get from one to the other may be programmed. There may well be many alternative choices for each element of the sequence, each choice opening some further possibilities and closing others. The complexity of these issues, even for a simple maze task, seems to render such theories unmechanisable. At issue is how the rat might avoid being caught in Guthrie's trap – being left buried in thought because the planning problem underlying navigation is too difficult to solve – representation of space is not all. Gallistel's theory is similarly reticent on the subject of the 'how'.

O'Keefe and Nadel consider animals to have two systems for navigation; the *locale* system, their rôle for the hippocampus, with its place cells, and the *taxon* system, which is elsewhere in the brain, and which generates lists of guidance and orientation hypotheses, which determine routes. Guidance involves the animal associating positive or negative valences with cues or items, and choosing actions to approach or avoid them appropriately. Orientations are more specific hypotheses about the types of behaviours appropriate in the presence of particular cues, for example 'right turn at the corner'.

A more recent proposal concerning the function of the hippocampus is due to McNaughton [90]. His 'Hebb-Marr' theory has this structure learn a function of the form:

$$\texttt{place} \times \texttt{action} \rightarrow \texttt{next place.}$$

based on place cell representations of locations. For this function to be well defined, the same place must be represented differently according to which direction the animal is facing in it – a purely allocentric coding of space could not be used with anegocentric coding of actions. McNaughton consistently finds this – *eg* in [89], when rats run through a maze with radial arms, different place cells fire when they run inwards from when they run outwards. However, O'Keefe's evidence, *eg* [114], does not support this conclusion.

Two problems are apparent with McNaughton's proposal; firstly there are technical difficulties in persuading the particular Hebb-Marr memories he posits to learn such a conjunctive function, in which many places are associated with the same action, and many actions, each with different consequences, with the same place. Secondly, and more seriously, the planning side of navigation has also been overlooked. At most, the map can directly tell its owner how to get from `Start` to `Goal` only if some single action performs this step. How a sequence of actions might be compiled together is still undetermined. Sutton's DYNA systems, described in section 5.5, show one way that this might be done, but his is more an engineering than a neurobiological proposal.

From the discussion above, it would seem that some combination of systems that carry out locale and taxon functions might be appropriate for handling both the 'where' and the 'how', to the extent, of course, that O'Keefe and Nadel's distinction carves the world at some natural joint. This was the initial impetus for considering temporal difference (TD) methods. Independent motivation comes from their emphasis on the rôle of time, and also from the TD model Sutton and Barto present of general classical conditioning [151]. As far as the mechanisms of a general classical conditioning model are capable of solving a navigation problem, one is forced to reconsider the oft mooted suggestion that the spatial mapping system has radically different properties from the ones underlying other complex forms of learning and memory.

One paradigmatic example of a spatial learning task is the open field water maze, devised by Richard Morris [101]. Rats are placed in a large circular pool of water, to which some milk powder has been added to make it opaque. A platform is hidden under the surface, and the rats must learn how to get to it from wherever they are initially placed in the pool. Although the strains of rats used are powerful swimmers, they prefer dry land, and so have an incentive to get to the platform as quickly as possible. When first put in the pool, the rats swim rather randomly, but they rapidly learn fast paths to the platform. One attractive feature of the water maze is that it severely hinders the use of olfactory cues, to which rats are far more sensitive than humans.

Since the pool itself is as radially symmetrical as possible, the salient cues to which the rats can attend to navigate by are external to it. Indeed, when the pool is surrounded by uniform curtains, which obscure the rats' views of the rest of the room, they are either unable to find the platform reliably, or use interestingly different strategies, such as circling at the distance from the walls of the pool at which the platforms are always to be found.

Barto, Sutton and Watkins [11] developed a system that learns using TD to navigate to a goal in a grid. Although their intent was to demonstrate and discuss the workings of reactive planning, their example actually makes plain the sheer complexity of both the task and the decisions underlying theoretical models of map making and map use. These apply to biological systems as much as the engineering ones that are developed below. Among the issues that unavoidably arise are the manner of exploration, the nature of the rewards and punishments for reaching or failing to reach the goal, and the representation of the environment. This chapter focuses mainly on representations, but all these concerns impinge on observable behaviour.

The next section looks at Barto, Sutton and Watkins' navigation example and describes some of its choices on these issues, section 5.3 evaluates the effects of changing the representation of the environment, section 5.4 considers what might be learnt in the absence of reinforcement, and section 5.5 reviews some of the recent work related to this in connectionist planning and mapping.

5.2 Barto, Sutton and Watkins' Model

As a didactic example of TD methods and their relation to dynamical programming, Barto, Sutton and Watkins [11] present a system which learns how to find a goal point in a small grid. The task is shown in figure 5.2. Unless prevented by the barrier (represented by the thick lines), or the edge of the grid, the agent can move in any of the four directions. The 'blob' at grid location (2,2) is the goal, but its location is not known to the agent at the start. There are some similarities

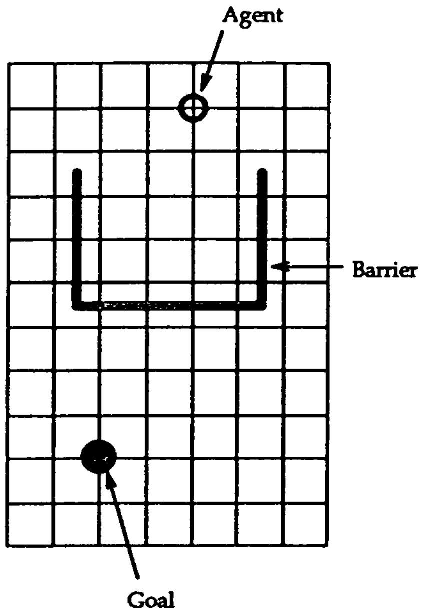


Figure 5.2: The Grid Task.

between this and the open field water maze – in both, there is a goal whose location is unknown, and the agent or rat has to work out how to navigate there from anywhere, based on attaining it repeatedly during exploration. However, it would, of course, be unsupportable to claim that rats solve the water maze using TD methods, and the mechanisms and representations developed below are more of engineering than biological interest.

The model and the flow of information through it are shown in figure 5.3. The environment provides the agent with two sources of information – the thick line representing a stimulus vector, which is a function of the agent's location in the grid, and the thin line representing a scalar reinforcement value. The agent is awarded a reinforcement of -1 every time it moves in the grid, unless it moves onto the goal, in which case it receives no reinforcement at all. Once it has moved onto the goal, the next trial begins. Its task is to choose actions so as to maximise total reinforcement, *ie* to minimise the lengths of the paths to the goal (since the reinforcement is negative).

The obvious problem is that the reinforcement provided by the environment is very uninformative between particular actions – it is almost always -1. As was evident in chapter 4, the system could try to overcome this handicap by

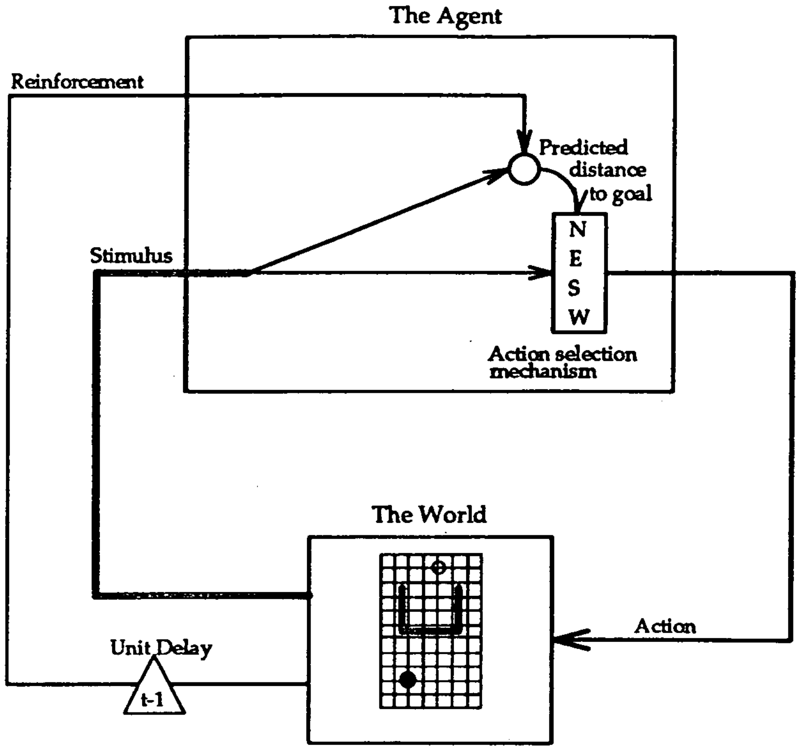


Figure 5.3: Barto, Sutton and Watkins' agent.

learning the function that relates locations in the grid to their distances from the goal. If this were possible, then the selection of an action could be rewarded if it leads from a location believed to be far from the goal to one thought to be close, and punished otherwise.

For this particular case, one could dispense with the γ discount factor altogether, since the grid is so small, but it will be retained for convenience. The desired discounted distance function should satisfy:

$$V^\lambda(x_t) = r_{t+1} + \gamma V^\lambda(x_{t+1})$$

where γ is the discount factor, λ is the TD learning decay rate, x_t is the representation of the location of the agent at time t, x_{t+1} is the representation at time $t + 1$, and r_{t+1} is the reinforcement awarded for the intervening move. This can be seen by looking at a complete path to the goal – the system ought to evaluate each location as being one step farther away than the next. Again as in the previous chapter, the inequality in this expression is turned into an error measure:

$$\epsilon_{t+1} = r_{t+1} + \gamma V^\lambda(x_{t+1}) - V^\lambda(x_t)$$

and this is used to alter the parameters generating V^λ.

Barto, Sutton and Watkins use the simplest possible representation – a punctate one for which each of the 96 locations in the grid is represented by the firing of a single unit. Coupled with a linear form for $V_t^\lambda(x) = v_t^\lambda.x$, this amounts to a table look-up for the distance from the goal. v^λ is changed according to:

$$\Delta v_{t+1}^\lambda \propto \epsilon_{t+1}\bar{x}_{t+1} \tag{5.1}$$

where

$$\bar{x}_{t+1} = (1 - \lambda)\bar{x}_t + \lambda x_t$$

is the usual TD trace function.[1]

The action selection system operates similarly. Associated with each of the four possible actions (North, South, East or West) is a vector (n_t, s_t, e_t, and w_t). At time t, the system chooses the largest of:

$$n_t.x_t + \eta_t^n,\ s_t.x_t + \eta_t^s,\ e_t.x_t + \eta_t^e \text{ and } w_t.x_t + \eta_t^w \quad .$$

where the η_t are identically distributed random variables, and then tries to move in that direction. V^λ is used to criticise the action according to the difference between the estimated value (or equivalently cost) of the move and the estimated value of the current location. The cost of the move is:

$$r_{t+1} + \gamma V^\lambda(x_{t+1})$$

which is the sum of the reinforcement actually received and the estimated value of the new location. The estimated value of the current location x_t is $V^\lambda(x_t)$. The difference between these two is just ϵ_{t+1}, the same quantity that was used in equation 5.1 for learning V^λ itself. The action vectors are then updated according to:

$$\Delta e_{t+1} \propto \epsilon_{t+1}\bar{x}_{t+1}^e$$

[1] For consistency with Sutton and Barto's formulation of TD for classical conditioning [150], which is cited in Barto, Sutton and Watkins [11], the definition of the trace decay term is slightly different from the one adopted in chapter 4. That would have been:

$$\bar{x}'_{t+1} = \lambda\bar{x}'_t + x_t.$$

(using $\mathbf{e}_t$ as an example), where each action has its own trace decay eligibility term, which is updated according to

$$\bar{\mathbf{x}}^e_{t+1} = \begin{cases} (1-\lambda')\bar{\mathbf{x}}^e_t + \lambda'\mathbf{x}_t & \text{if East was the chosen action at time t,} \\ (1-\lambda')\bar{\mathbf{x}}^e_t & \text{otherwise} \end{cases}$$

All equations for this section and section 5.4 are given in appendix A, along with the values of the parameters that were used in the simulations.

Note that in this representation, the environment does not explicitly tell the agent about the barrier or the sides of the grid. It has to find these by trying invalid actions, which still earn a reinforcement of -1 but leave it fixed. This swiftly causes the action selection system to prefer some alternative.

A typical learning curve for this system is shown in figure 5.4.[2] Points on the graph are averages over 200 complete runs, and are calculated by switching off the learning after the number of runs given on the x–axis, starting the agent from every location on the grid, and recording how many more steps it takes to reach the goal from there than the optimal strategy would. It is apparent that the system rapidly learns fast paths to the goal, despite the relative paucity of the information it receives. Figure 5.5 shows the development of V^λ across the grid as the system learns to solve the task. The barrier can clearly be seen in the sharp jump in this value function. Figure 5.6 shows the final actions the system chose at every location, for one particular run of the program.

Three key issues identified in the introduction for theoretical models like this are the manner of exploration, the nature of the rewards and punishment, and the representation of the environment. The randomness provided by the η_t is responsible for the balance between exploitation of the existing set of actions and exploration for new and better sets. As the agent learns appropriate actions, the effect of the η_t smoothly diminishes, since these quantities are outweighed by the differences between the evaluations of the actions.

This is a form of implicit and incomplete annealing. In normal annealing [73], the randomness, governed by a computational temperature, is explicitly reduced as the system learns. Once the temperature is reduced to zero, the system

[2] The axes of the graphs differ from those in [11].

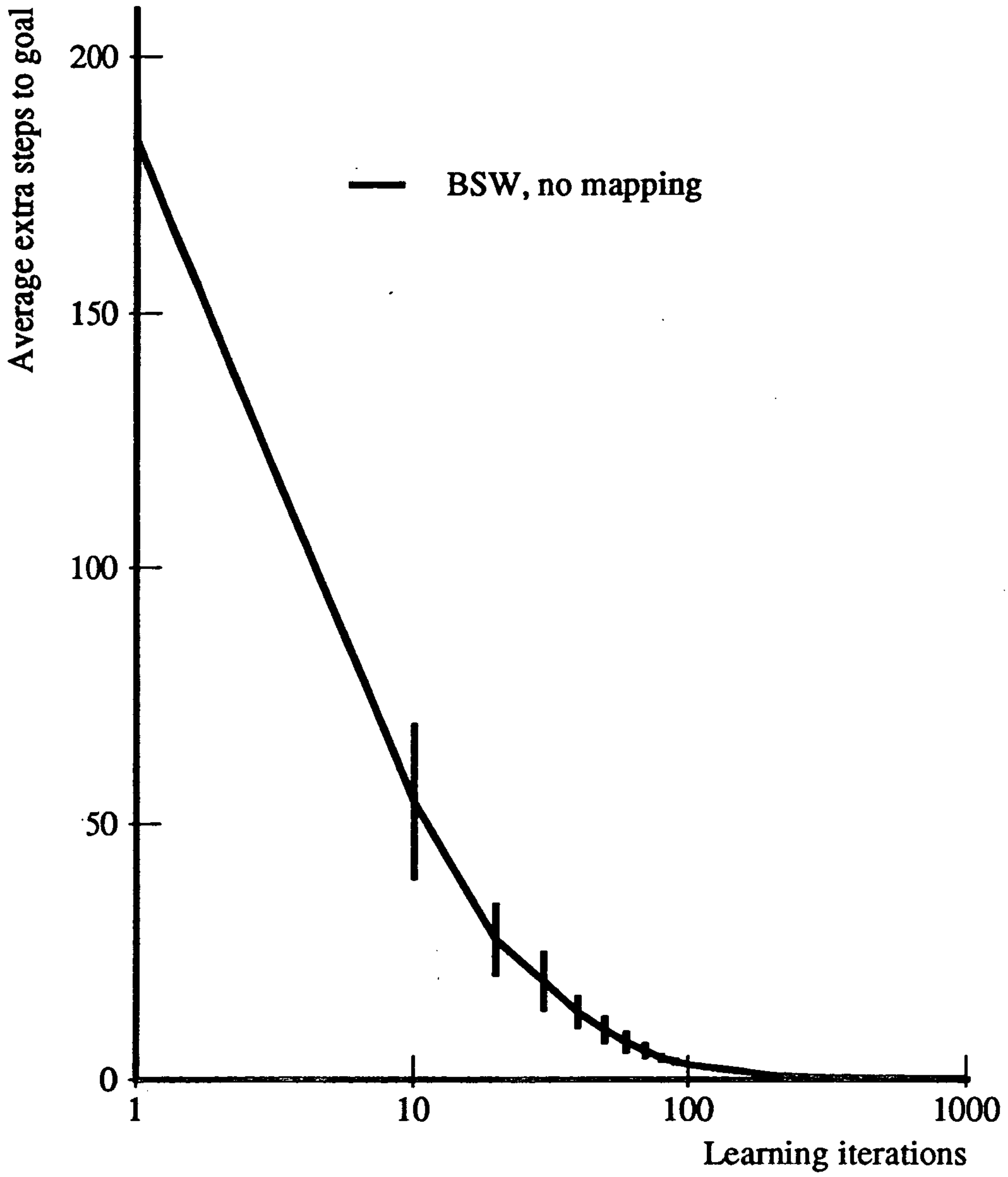


Figure 5.4: Average learning curve for BSW. Note the log-linear scale.

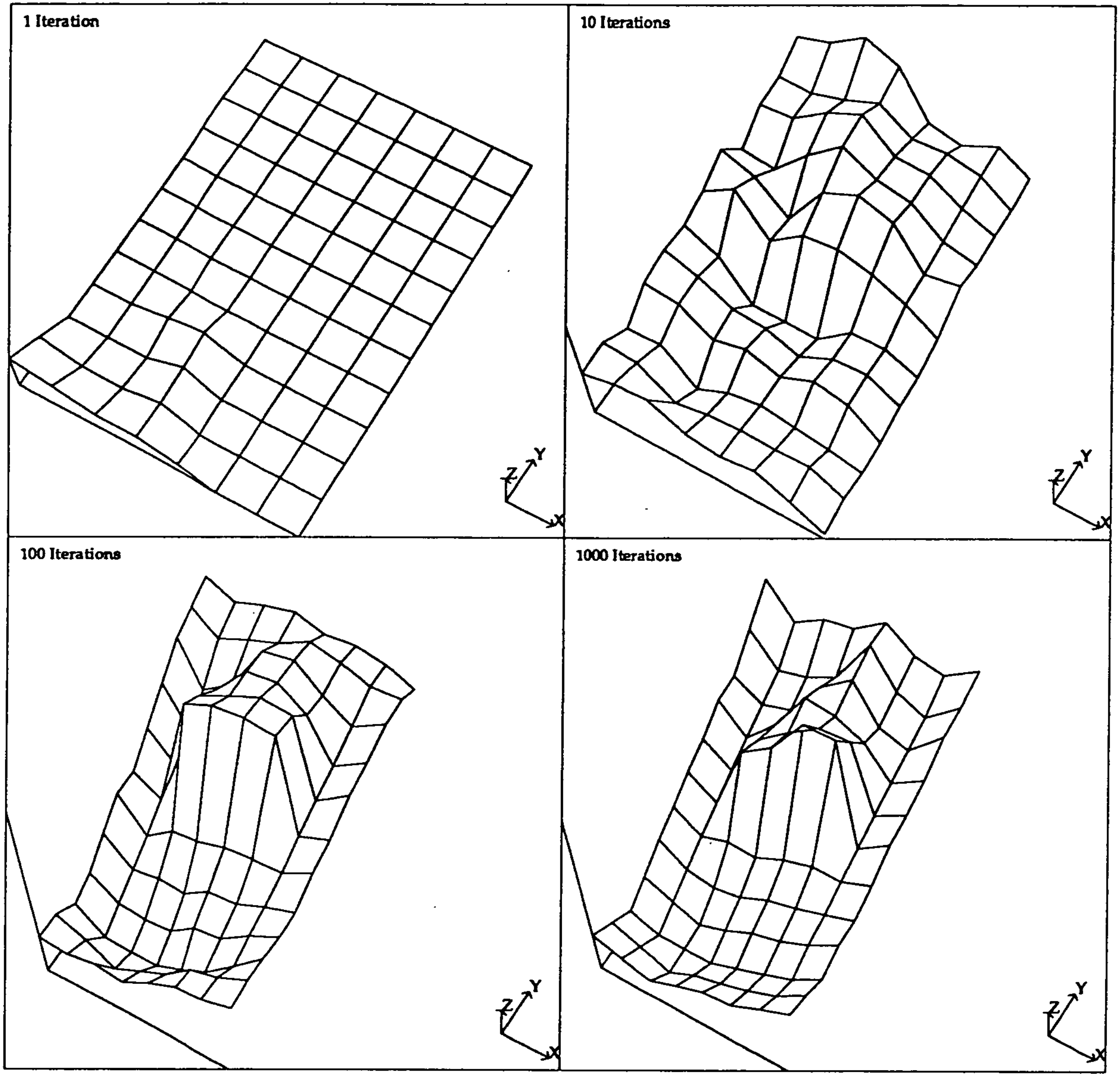


Figure 5.5: Development of V^{λ} after 1, 10, 100, and 1000 iterations.

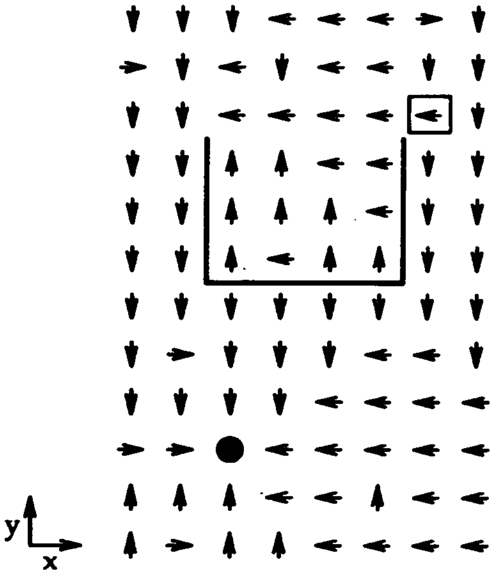


Figure 5.6: The favoured actions at every grid point. Note the error at (6, 9).

solidifies, and is incapable of responding to manipulations in the environment. Here, the solidity is controlled by the magnitude of the differences between the evaluations of the actions. If anything happens to change these evaluations (for instance, if a path becomes blocked) then the randomness, which was lurking under the surface, typically re-emerges to continue the search for the best action under the new circumstances. The process only works in this case because there are convenient automatic bounds on the value function which, in normal circumstances, stop it growing without bound.

On the structure of the rewards,[3] there is an interesting trade off between the initial estimates of the V^λ and the reinforcement values awarded after each move. An alternative reward schedule to the one described above is to award no reinforcement for any move, until the agent moves onto the goal, and then for that last step to set $r_{t+1} = 1$. If the discount factor is $\gamma < 1$, then the ideal value of any location is γ^n, where n is the number of steps it takes to get to the goal. The incentive on the agent to produce shorter paths is provided by the ever decreasing γ^n.

This system is not obviously equivalent to the one Barto, Sutton and Watkins

[3] This arose during a discussion with Tony Prescott.

used. Consider what happens if V^λ starts out at 0 when the agent is moving around the grid without having ever reached the goal. If the reinforcement values are all 0, then all its evaluations of V^λ will appear to be correct, and so it will not alter the evaluation of grid points it has been through. Contrast this with what happens when the reinforcements are -1. Now, passing through a grid point is immediately deleterious, and the agent will learn to avoid places it has already visited, even before it has ever reached the goal. One way to restore the equivalence is to arrange for the 0 reinforcement case that the initial values of the V^λ are constant, but not zero, as then the

$$\gamma V^\lambda(x_{t+1}) - V^\lambda(x_t)$$

component to ϵ_{t+1} will have the same avoidance effect. Although giving certain locations higher initial evaluations is one way to bias the exploration of the agent[4] it is difficult to do in general, depending on the nature of the representation (or equivalently function approximation scheme) adopted. In this particular case, the table look-up makes it easy, but this will not be true of all representations, *eg* $\mathcal{R}_A$, which is discussed below.

The representation of the environment employed makes this way of doing navigation something of a hybrid between locale and taxon systems. On one hand the punctate, position-coded, units somewhat resemble place cells (however, although they fire only at certain locations in the environment, the representation is not distributed across the firing of a number of units, as seems actually to be the case) and the action system is not based on the evaluation of any particular cue. On the other, the agent does not construct a map which is meaningful independent of the particular goal at (2,2), and so would have to relearn, painfully, were the goal to be moved. Section 5.3 considers alternative representations that are more closely related to the information directly available from the environment. Section 5.4 considers how latent learning, in the initial absence of a goal, can be used to develop 'hidden-unit' representations that speed subsequent goal-based learning.

Sutton [149] calls the stimulus-response map that the system generates a 'compiled plan' in honour of the absence in this method of any interpretation process

[4] Indeed, as described below, this is the means by which the 'imagination' of Sutton's DYNA system [149] ensures that alternative paths are continually revisited to see if they are better, even when the agent has already settled on a good path.

on a map. His advocacy of such approaches is based on their continually swift responses, and the incremental nature of their learning – the agent does not have to stop and crunch substantial dynamic programs as it learns, which would be computationally unwise early on in learning and for a continually changing world. In fact, in a rather different problem, Barto and Singh [8] found exactly this; Watkins' Q-learning system, which uses compiled plans, substantially outperformed an alternative system due to Sato, Abe and Takeda [134] that continually improves a complete Markov model, and calculates, using DP, the optimal actions this model would specify.

The similarities between the grid task and the open field water maze should be clear. In both, the problem solvers are placed at various locations in an initially unknown environment, and have to learn how to get to one particular location in it in the shortest time. Some differences should also be clear – particularly in that there are no cues in the grid task, rather, the punctate representation is provided at the outset. Also, whereas other experiments would suggest the rats in the water maze would have the capacity to learn about the environment in the absence of the platform, this would not happen in the present model.[5] The next sections look at these two issues.

Other manipulations are also possible – for instance partial barriers could be installed, which the agent would just be penalised for breaching. Experimentally, the system handles these admirably, choosing to breach barriers only if it would benefit from doing so.

5.3 Alternative Stimulus Representations

Stimulus representations, the means by which the agent finds out from the environment where it is, can be classified along two dimensions; whether they are punctate or distributed, and whether they are directionally sensitive or in register with the world. Over most of the grid, a 'sensible' distributed representation, such as a coarse-coded one would be expected to make learning

[5] Apparently, technical difficulties have prevented this experiment from being done, but such 'latent' learning is well established in more conventional, dry, mazes [16].

faster. It would allow adjacent grid points to share information for the V^λ. This is appropriate since points that are near each other are nearly the same distance from the goal. Information could also be shared for the n, s, e and w vectors which determine the action selection. This is frequently wise given the simple set of possible moves. There are points of discontinuity in the actions, as in the region above the right hand arm of the barrier, but they are few.

In his thesis [159], Watkins considers a rather similar problem to this, and solves it using Q-learning (see chapter 4) based on a CMAC representation of the space. CMACs were developed by Albus [2] from his model of the cerebellum, but for Watkins' purposes they act just like a coarse coding of the environment. Since his agent moves in a continuous bounded space, rather than being confined merely to discrete grid points, something of this sort is anyway essential. After the initial learning, Watkins arbitrarily makes the agent move ten times more slowly in a closed section of the space. This has a similar effect to the barrier in inducing a discontinuity in the action space. Despite the CMACs forcing the system to share information across such discontinuities, they were able to learn the task quickly.

The other dimension over which representations may vary involves the extent to which they are sensitive to the direction in which the agent is facing. As mentioned above, it is an unresolved empirical issue whether or not place cells are directionally independent. Certainly other cells responsive more directly to the input stimuli will be directionally sensitive.

Note also that rather than moving North, South, East or West, which are actions registered with the world, the agent should only move Ahead, Left or Right (Behind is disabled as an additional constraint). The effects of these actions are also orientation dependent. This, together with the fact that the representation is likely to be less compact (*ie* having a larger input dimensionality) should make learning slower. Dynamical programming (DP) is notoriously subject to Bellman's curse of dimensionality, which is the engineering equivalent of exponential explosion in search. TD methods, although they provide a natural and incremental way of doing DP, will not be able to stave off degradation in the face of this problem. At best, they may improve its grace.

	Coarseness	
Directionally	Punctate	Distributed
Sensitive	$\mathcal{R}_\alpha$	$\mathcal{R}_A$
Insensitive	$\mathcal{R}_{BSW}$	$\mathcal{R}_{CMAC}$

Table 5.1: Representations.

Table 5.1 shows four possible representations classified along these two dimensions. $\mathcal{R}_{BSW}$ is the representation Barto, Sutton and Watkins used, and is discussed in the previous section. $\mathcal{R}_\alpha$ is punctate and directionally sensitive; just like $\mathcal{R}_{BSW}$ only one unit fires at any one time, but $\mathcal{R}_\alpha$ devotes four units to each grid point, one of which fires for each possible orientation of the agent. $\mathcal{R}_{CMAC}$ was not simulated because in a task with so few discontinuities, its capabilities would be very similar to those of the mapping-based representation developed in the next section.

$\mathcal{R}_A$ is rather different from the other representations. Even apart from the dispute mentioned above about the directional sensitivity of place cells, it seemed important to test a representation which is more closely associated with the sensory information that might be available directly from the cues. Figure 5.7 shows how $\mathcal{R}_A$ works. Various identifiable cues ($C_1 \ldots C_c$) are scattered around the outside of the grid, and the agent has a fictitious 'retina' which rotates with it. This retina is divided into a number of angular buckets (8 in the figure), and each bucket has c units, the i^{th} one of which responds if the cue C_i is visible in that bucket. This representation is clearly directionally sensitive (if the agent is facing a different way, then so is its retina, and so no cue will be visible in the same bucket as it was before), and also distributed, since in general more than one cue will be visible from every location.

There are various things of note about $\mathcal{R}_A$. First, there is no restriction on the number of units that can fire in each bucket at any time – more than one will fire if more than one cue is visible there. Second, it will in general not work if it is ambiguous, and so one can still regard the stimulus as labelling the place directly. This issue is treated in section 5.5. Third, it should be clear that

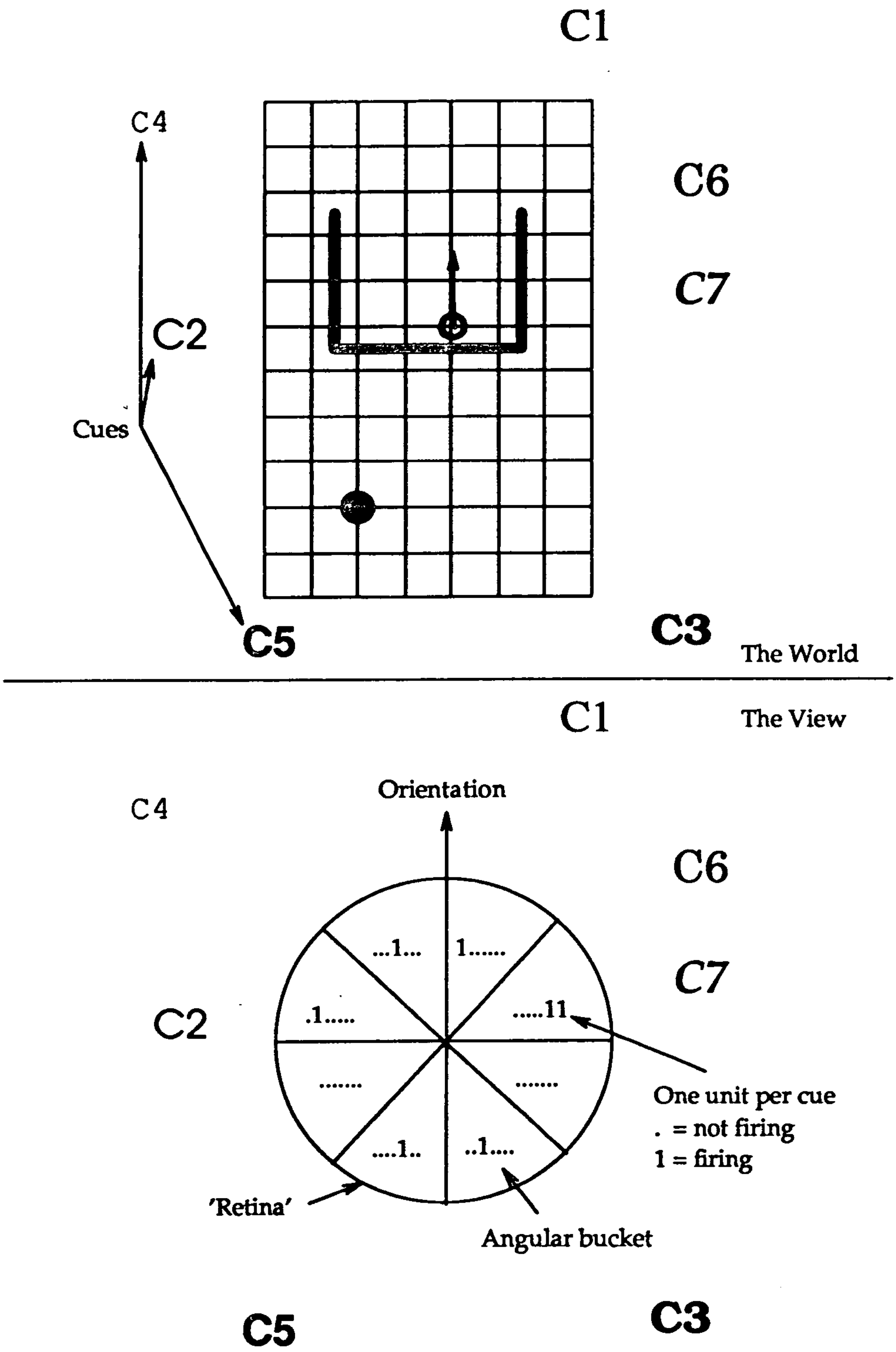


Figure 5.7: The Angular Representation.

$\mathcal{R}_A$ is not remotely biologically plausible, either as a basis for navigation, or in terms of the assumptions it makes about object recognition. Finally, during experiments with $\mathcal{R}_A$ it became apparent that the paths taken early on in learning are qualitatively different from those with the other representations – a form of dervish-like circling/spiraling behaviour is often seen. This arises because the control is crudely of the form:

> ...carry on until cue C_2 changes from being on the right and ahead to being on the right and behind, then turn ...

Only later on in learning will the finer points of the differences in the representations of each point be fully utilised.

Figure 5.8 shows the learning curves for the three representations simulated. It is apparent that $\mathcal{R}_{\alpha x}$ is substantially worse, but, surprisingly, that $\mathcal{R}_A$ is actually slightly better than $\mathcal{R}_{BSW}$.[6] This implies that the added advantage from its distributed nature more than outweighs its disadvantages of having more components and being directionally sensitive.

5.4 Goal-Free Learning

One of the problems with the TD system as described is that it is incapable of learning in the absence of reinforcement or a goal. If the goal is taken away, but the -1 reinforcements are still applied at each step, then the values assigned to each location will tend to $-1/(1-\gamma)$ (or $-\infty$ if $\gamma = 1$). If both are removed, then although the agent will wander about its environment with random gay abandon, it will not pick up anything that could be used to speed subsequent learning. Latent learning experiments in dry-land mazes [16] prove fairly conclusively that rats running in the absence of rewards and punishments learn almost as much as rats that are reinforced.

One way to solve this problem is suggested by Sutton's DYNA architecture,

[6] The differences are statistically significant at at least the 2.5% level until 90 learning iterations.

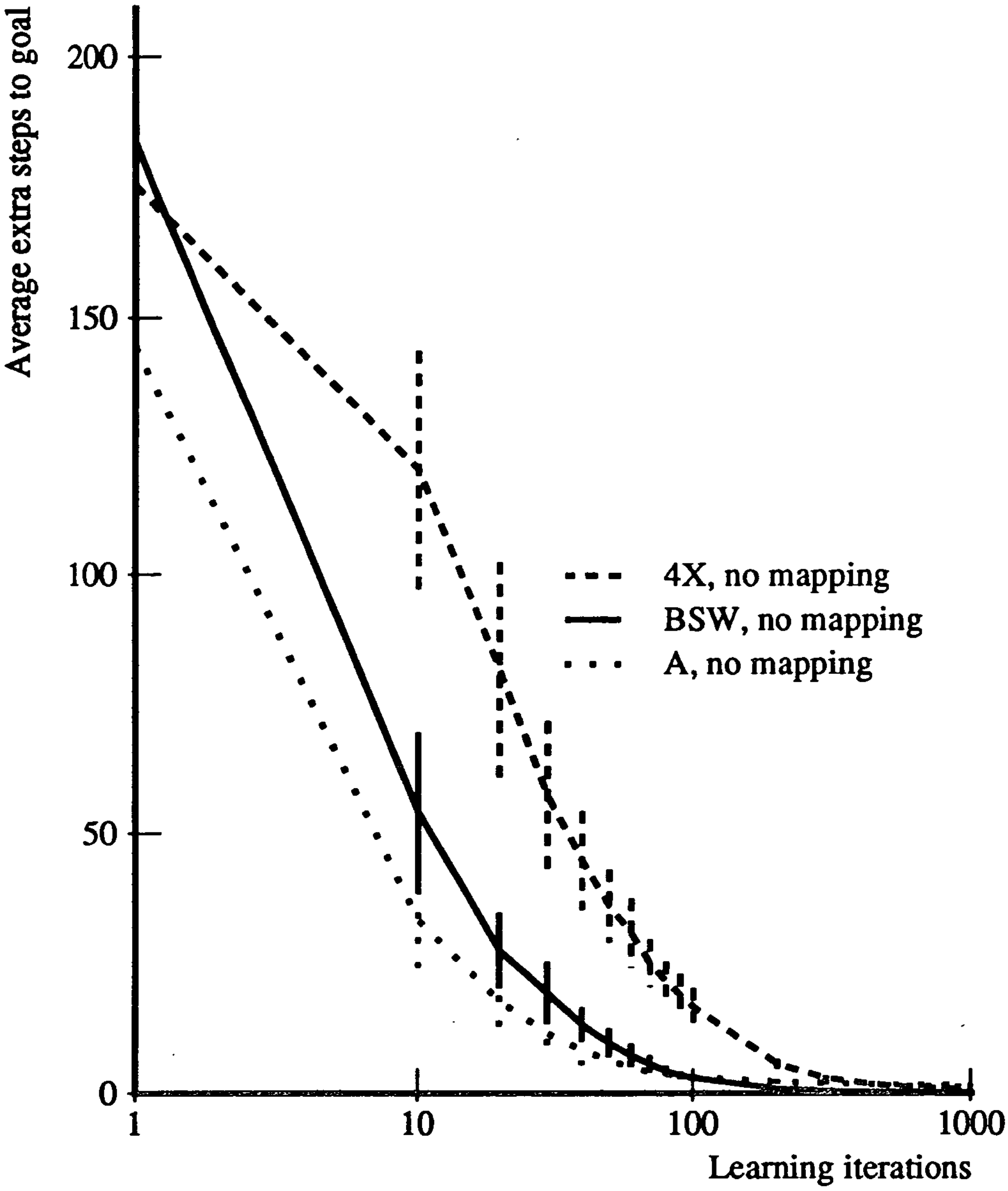


Figure 5.8: Average learning curve for the three representations.

which is described in detail the next section. Briefly, it constructs a map of the McNaughton form, but suggests a way that the 'how' question can be addressed. Unfortunately, its operation relies on a rather unintuitive process of 'imagination' – in between taking real steps in the real world, the agent has to take random imaginary steps in an imaginary world constructed from its map.

If a complete map is not to be constructed, one might seek an intermediate way, not dependent on reinforcement, of incorporating information from the world. The only remaining repository for this is the representation of the environment used for learning the value function and optimal actions. The section on representations concluded that coarse-coded representations are generally better than punctate ones, since information can be shared between neighbouring points. However, not all neighbouring points are amenable to this sharing, because of discontinuities in the value and action functions. If there were a way of generating a coarse coded representation that is sensitive to the structure of the task (generally from a punctate one), rather than being arbitrarily assigned by the environment, it should provide the base for fast learning. In this case, neighbouring points should not be coded together if they are separated by the barrier. The initial exploration would allow the agent to learn this much about the structure of the environment.

It turns out that this can be accomplished using exactly the learning mechanisms already discussed. Although any claim to neurobiological plausibility would be entirely spurious, the method was originally motivated by the connections shown in the block diagram of the hippocampus in figure 5.1 between the entorhinal cortex and CA_3, which augment the well known tri-synaptic loop. In the computational equivalent of this, one could imagine these additional fibres to play a different rôle from the ones feeding in from the dentate gyrus – a rôle more akin to that played by the environmental reinforcement path whose future activation V^λ is intended to predict.

If this were so, the units in the modelled version of CA_3 would have the job of predicting the future discounted sum of firings of the raw input lines. This is just what is required. Consider $\mathcal{R}_{BSW}$ and the initial stage of learning when the actions are still random. If the agent is at location $(3,3)$ of the grid, say, then the discounted prediction of how often it will be in $(3,4)$ will be high, since this location is close. However, the prediction for $(7, 11)$ will be low, because it is very unlikely to get there in a hurry. Consider the effect of the barrier; locations on opposite sides of it, *eg* $(1, 6)$ and $(2, 6)$, though close in the Euclidean (or Manhattan) metric on the grid, are far apart in the task. This means that the discounted prediction of how often the agent will be at $(1, 6)$ given that it starts at $(2, 6)$, will be proportionately lower. Overall, the prediction units should act like a coarse code, sensitive to the structure of the task. As required, this information about the environment is entirely independent of whether or not the agent is reinforced during its exploration. In fact, the 'map' will be more accurate if it is not, as its exploration will be more random.

The modifications to the agent's architecture are shown in figure 5.9. Each of the prediction or mapping units has two sources of input from the environment; a 'privileged' one, whose discounted future firing it has to predict, based on the other 'normal' ones, which are just the same as before. Relying for the ultimate control learning solely on the prediction units would be unwise, since they take a number of iterations to produce meaningful outputs. They are used as an additional source of information for the value and action functions. Appendix A gives the learning rules and the values of the parameters used in the experiments.

Since their main aim is to create intelligently distributed representations from punctate ones, it is only appropriate to use these prediction units for $\mathcal{R}_{\text{BSW}}$ and $\mathcal{R}_{\text{\alpha X}}$. Figures 5.10 and 5.11 compare average learning curves for $\mathcal{R}_{\text{BSW}}$ and $\mathcal{R}_{\text{\alpha X}}$ respectively with and without these extra mapping units, and with and without 6000 steps of latent learning in the absence of any reinforcement. A significant improvement is apparent.

Figure 5.12 shows how two sets of predictions for the $\mathcal{R}_{\text{BSW}}$ representation develop,[7] one on each side of the barrier. After 6000 un-reinforced steps, the predictions are fairly well developed and smooth – a predictable exponentially decaying hump. The only deviations from this are at the barrier and along the edges, where the effects of impermeability and immobility are apparent. Introducing the mapping units increases by an order of magnitude the com-

[7] Note that these are normalised to a maximum value of 10, for graphical convenience.

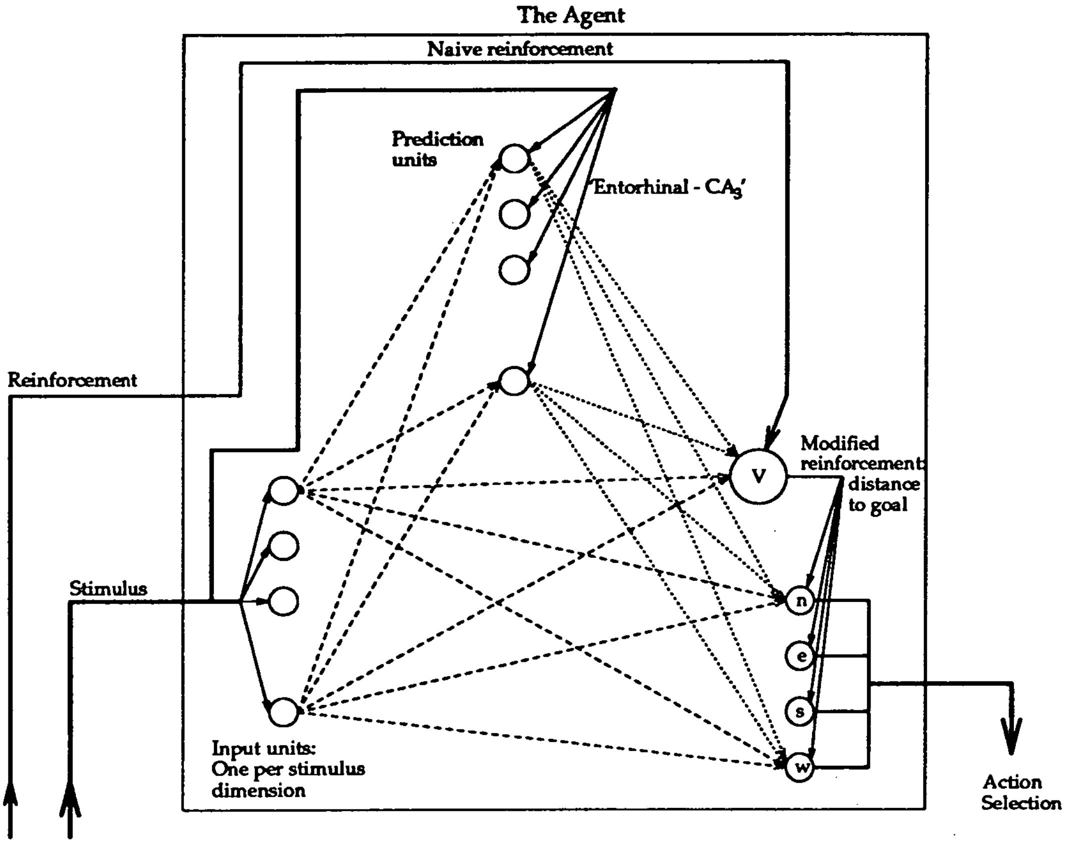


Figure 5.9: Architecture with prediction or mapping units.

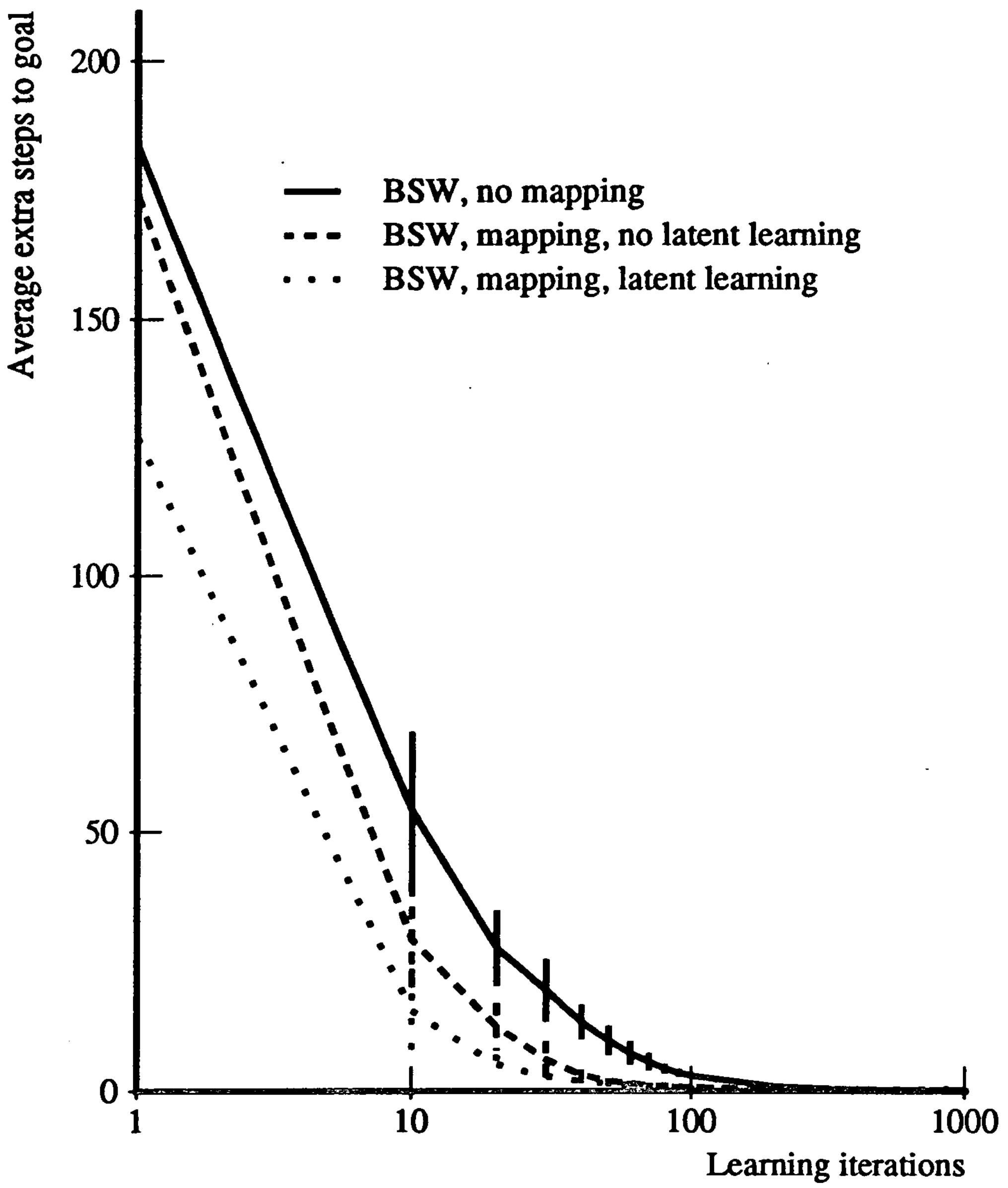


Figure 5.10: Average learning curves for punctate representation with and without mapping and latent learning.

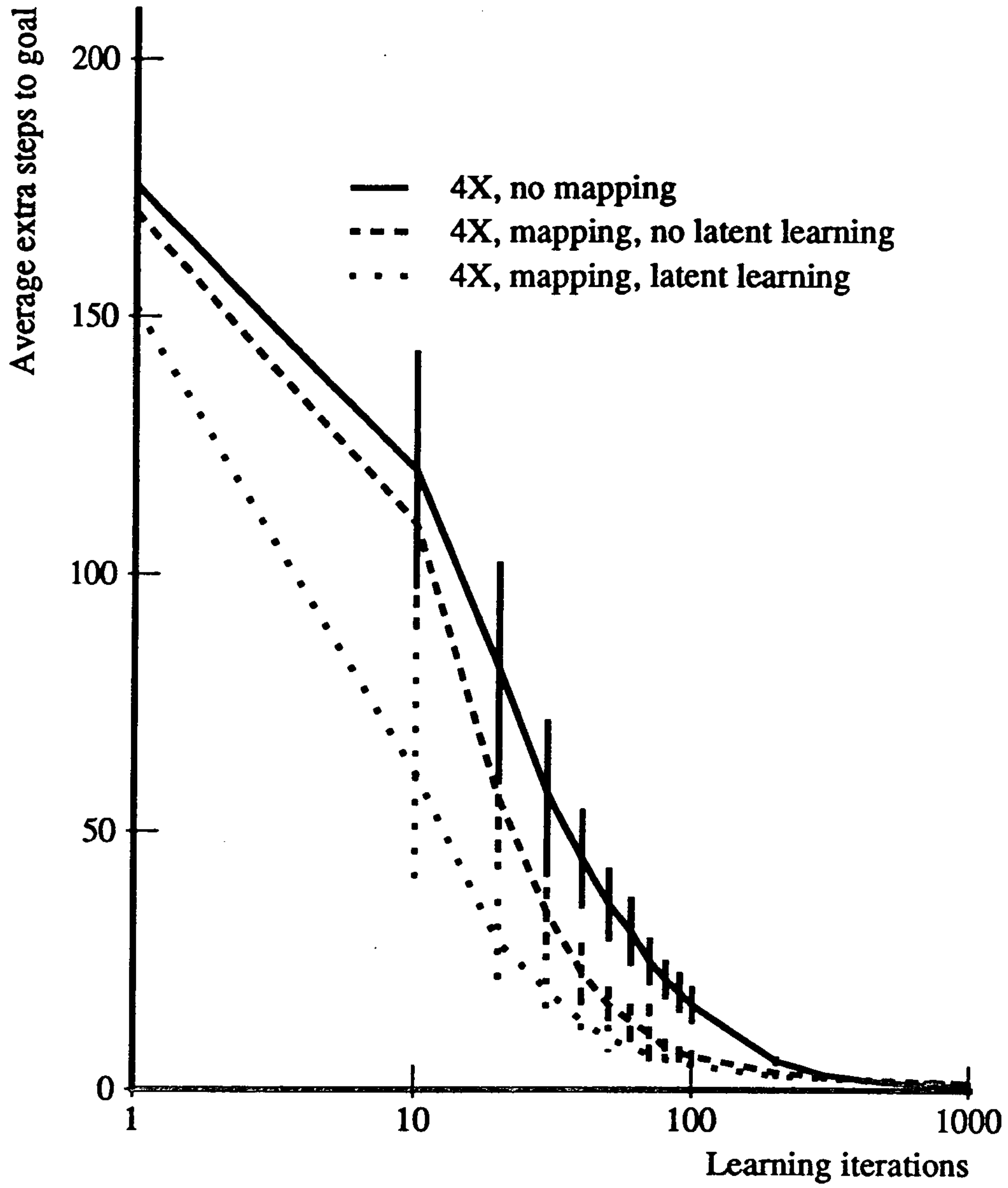


Figure 5.11: Average learning curves for directional punctate representation with and without mapping and latent learning.

putational load of the system based on the number of weights available to it – however, as can be seen from figure 5.12, most of them end up at zero.

The striking feature of figure 5.12 is that as learning continues and the paths of the agent become increasingly optimal, the predictions degenerate from being roughly radially symmetric (bar the barrier) to being highly asymmetric. This is only to be expected given the job they are trying to do. Once the agent has learnt how to get to the goal from some place, the only locations to which it will subsequently go are those on the direct path to the goal. The asymptotic values of the predictions will therefore be 0 for units not on the path, and γ^{τ} for those on the path, where τ is the number of steps since the agent's start point. This is a severe limitation since it implies that the topological information present in the early stages of learning quite evaporates, and with it almost all the benefits of the prediction units. In fact it can be positively harmful; if the goal is moved once this degeneration is complete, the system finds it significantly more difficult to learn its new location than it did to learn its original one.

There is a close relationship between these predictions and those learnt by Sutton and Pinette's [152] world modelling scheme. Their system is based on a recurrent network whose connections are essentially the Markov matrix for a task. When fed with a persistent input, the punctate representation of the current state, the network calculates the discounted sum of future expected occupancy of all the other states. The network can also be augmented so as to predict the discounted sum of future reinforcement, if some states and actions lead to reinforcement from the environment. Just like the systems described here, the world model is learnt using the discrepancies between the predictions at adjacent time steps as errors for adjusting the weights. The formula for the weight change is based on an analogy; Sutton and Pinette give no guarantees that it will learn correctly. It should be pointed out that CA_3 has recurrent connections.

Under normal circumstances, Sutton and Pinette's predictions are identical to the ones learnt by the system described in this section; however those here are easier to calculate, since only feedforward connections are involved, and they also can, in principle at least, work with non-punctate representations of the state. Sutton and Pinette's system suffers from the same handicap that possible

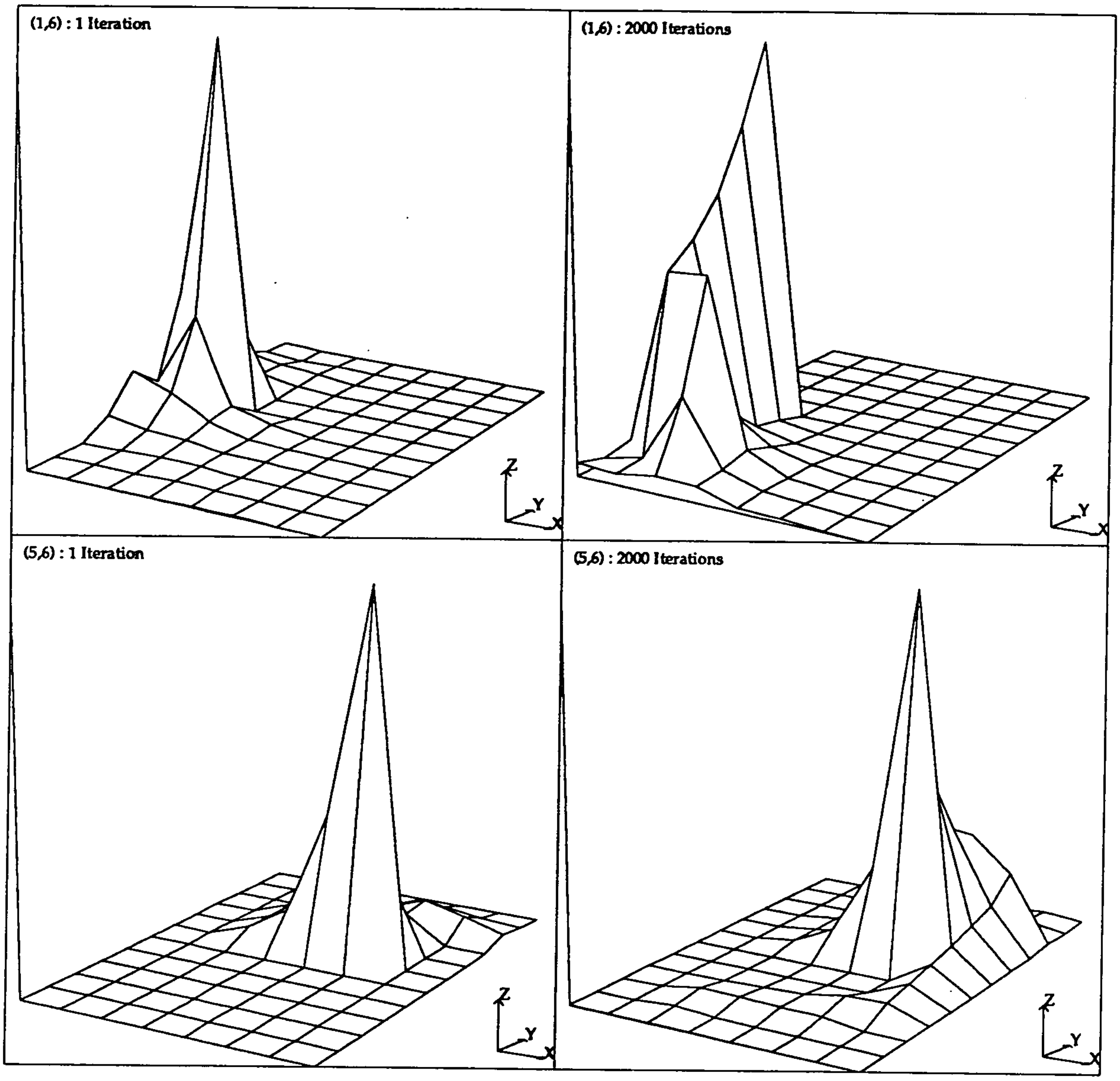


Figure 5.12: Predictions starting from (1, 6) and (5, 6) after 1 and 2000 learning iterations.

action choices are only known implicitly, and so, for instance, would show the same degeneration in its map as is evident in figure 5.12. The advantage the recurrence should give is that a small change in the model (*eg* a slight extension to the barrier) can instantaneously lead to dramatic changes in the predictions. The system here would have to relearn all the affected predictions explicitly. This capacity was not tested in the original paper.

Certain interesting parallels between the prediction units and place cells present themselves. Foremost, the predictions will often be far more place-like than the inputs on which they are based. Imagine an input unit that fires at some location when the agent is at a given orientation. The predictor of the firing of this input should be somewhat active over a complete neighbourhood of it, formed of the places which are relatively close in terms of the task. The activation of the prediction unit will therefore be more distributed and less orientation sensitive than the original input.

Secondly, the outputs of the prediction units will be sensitive to the introduction of barriers in the environment that actually force the agent to change its course. They will not, in general, respond to barriers that merely alter the stimuli without changing the agent's course. Muller and Kubie [104] found precisely this effect when they tried adding clear plastic barriers to an environment which rats had previously experienced.

Figure 5.13 compares the best learning curve for each representation. It is apparent that the extra units and the latent learning benefit both representations, making $\mathcal{R}_{\text{BSW}}$ preferable to $\mathcal{R}_{\text{A}}$. All the visible differences are statistically significant at better than the 5% level, however the parameters (given in Appendix A) are not necessarily optimal.

5.5 Further Developments

The problems of mapping and navigation using connectionist systems have recently come to the fore. Three foci of the work are ambiguous stimulus representations (essentially a 'where' issue), more effective planning (a 'how'

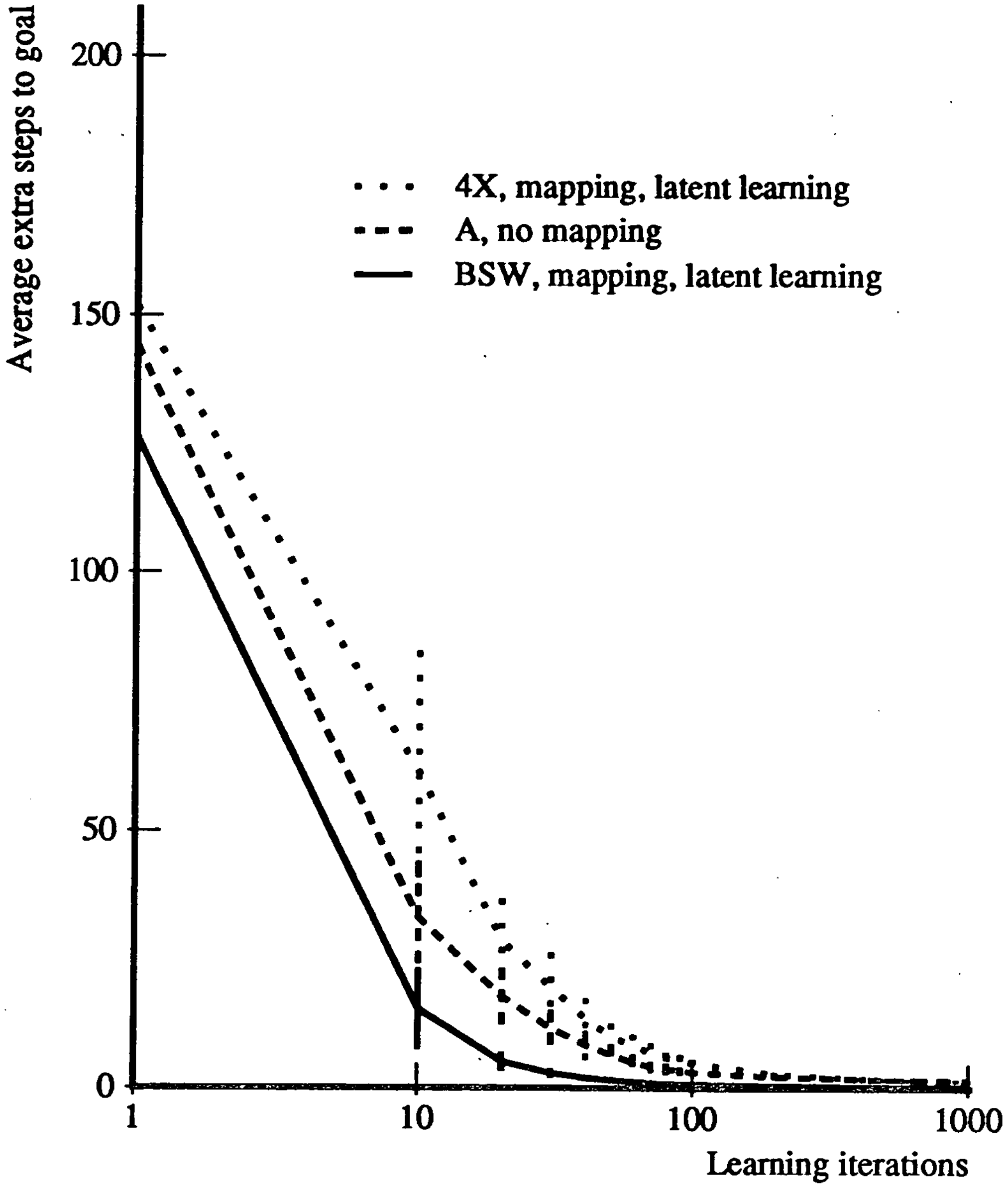


Figure 5.13: Overall best learning curve for each representation.

one), and computationally better representations (a 'what' one).

5.5.1 'Where'

On the ambiguity of input, there are a number of recent proposals (*eg* [24, 137]) for solving the problem of constructing unique maps from non-unique stimuli, when many locations in the world look the same. All of them are based on the use of recurrent networks. These develop a 'context' that can disambiguate such locations based on the sequence of places the agent visits. Consider what might happen if the environment delivers stimulus $\mathbf{b}$ in two locations, but in one it happens never to follow stimulus $\mathbf{a}$, whereas in the other it does. Then, observing the sequence $\mathbf{a}, \mathbf{b}$ would serve to distinguish the two. Recurrence is necessary for the classification system to be sensitive to such sequences. This method is rather similar to a traditional one in the theory of Markov chains. If a particular process is non-Markovian – its state transitions depend partially on its history – then it can be made Markovian by wrapping up into a state the entire history of the process.

Chrisley [24] makes one such proposal for resolving the ambiguity, using the simple recurrent architecture due to Jordan [67] and Elman [37]. His system can use dead-reckoning – the agent's estimation of the distance and directions it thinks it has travelled – as one of its sources of data. There is empirical evidence that this information may be available to the navigation system. For instance O'Keefe [111] found that the appropriate place cells fired when some rats were moving through a previously learnt environment even if all the visual cues were absent as they ran. They were given an initial glimpse of the environment so that they could orient themselves. In general, and particularly in environments like the water maze, artificial dead reckoning systems are very prone to error. They need continual correction on the basis of externally registered information such as cues. It is also very hard to simulate the errors of dead reckoning systems convincingly, since they may be apparently systematic in some dimensions, yet apparently random in others.

Building on Rivest and Schapire's [129, 130] update graphs, Mozer and Bachrach

[102, 103] provide another method of using stochastic exploration to learn the structure of an environment whose state representations are ambiguous. An update graph is a particular representation of a finite environment, which is normally compact compared with the equivalent description as a finite state machine. It works by representing only the equivalence classes of the results of executing sequences of actions; for example in the grid, moving `North` is almost always the same as moving `East`, `West`, then `North`, except at the boundaries and the barrier. Such equivalences lead to the parsimony of the update graphs. Mozer and Bachrach show how a recurrent connectionist system, with an architecture very similar to Chrisley's can be encouraged to learn the update graph, and discuss the circumstances under which it might be useful. Unlike Chrisley's, this method explicitly decouples the actions, by providing a different set of weights for each one.

It is unclear that ambiguity is much of a problem for animals in realistic environments. Facets of the stimulus array such as the distance of the agent from the cues (which representation $\mathcal{R}_A$ ignores), or the different appearance of the same cue from a different part of the environment, will help create the desired uniqueness. In fact, the problem is more likely to be of the opposite character – places look different each time the animal visits. This would make the task for the recurrent networks radically more difficult, as they would be trying to satisfy impossible demands. By contrast, static networks do not make the attempt, and so would not find it so difficult.

Whitehead and Ballard [163] deliberately create an ambiguous representation to permit fast processing, and then provide a method for disambiguation. Inspired by Agre and Chapman's system Pengi [1], described in chapter 6, they adopt indexical representations as the stimulus input to a reinforcement learning system. Such representations are based on only part of the available state information, and are hence compact. However, omitting this information tends to make them ambiguous. Whitehead and Ballard define a set of 'internal' actions the agent can do (such as turning its head) to collect more and different information that is available at its current location in the environment so as to resolve the ambiguity, *if* this is causing the agent to fail. If its actions are successful in accordance with its predictions, then any ambiguity is not harming the agent and can be tolerated. This is an attractive way of resolving the computational concerns about DP, but depends on there being appropriate indexical representations. The alternative approach is to try to learn which of the stimulus inputs available at a location are significant – *ie* to learn an appropriate input transformation.

5.5.2 'How'

The field of planning is in constant foment – impossible demands unsatisfied by inadequate programs based on unreal theories. The residue left after the evaporation of AI's optimism suggests that no general method will be able to tackle anything other than toy-sized problems, and that bespoke tailoring of methods will always be necessary. DP and TD are both general methods, and so can be criticised along these lines.

Sutton has further developed the architecture described in section 5.2 into the DYNA range of systems, which was briefly mentioned above. These continue the tradition of fast incremental learning (which he calls relaxation planning) and compiled plans, but additionally consider possible ways of using a world model. Sutton's premise is that it costs little extra to learn a McNaughton-style world model of:

`place × action → next place`

as the agent explores its environment. This is more true from an engineering perspective, given no restrictions on the memory architecture, than from a connectionist one, where such conjunctive maps are difficult to learn. The criticism above of McNaughton is that he ignores the lack of compelling methods of inference over such one step maps. Sutton gets his agent repeatedly to 'day-dream' starting at random locations in the map, to 'imagine' making a random move, and to evaluate and learn about that move according to the current value and action systems.

Initially, the map is empty, so the imaginary moves are useless. As the agent's model of the world improves, though, the imaginary moves can increase the speed of learning. To see this, consider that at any intermediate stage of the TD learning, there are two sources of error. One is that not every action has been tried in every location, so their consequences are uncertain – this is unchanged by the map. The other is that there can be inconsistencies in the value and action systems. As a contingent fact of its history, the agent might evaluate two adjacent locations dramatically differently, or its value function might evaluate one as being better than the other whilst its action selection system favours a move in the opposite direction. The imaginary moves can iron out these computational bumps.

Paradoxically, for a purely simulated system, doing the extra imaginary steps actually slows learning down. For in the early stages, when the map is still incomplete, the extra relaxation is practically useless, and throughout, the imaginary actions cost just the same as the real ones. Were the agent moving in the real world, though, it is reasonable to assume that the cost of an imagined move would be much less than that for a real one, and DYNA would prove its worth.

Having constructed the map, DYNA turns it to the task of handling changing environments. Two types of changes can be significant; existing paths may be blocked, and new paths may be opened. Sutton concentrates on the use of Q-learning rather than the value function learning system described above, and he also introduces an extra measure of how long it is since the agent tried each action in each location. Q-learning helps with the first concern, since it is possible neatly to try out the effects of actions which the agent actually believes to be sub-optimal. The extra measure helps with the second concern, because, during its day-dreaming, it can plan to try actions whose consequences it has not observed for a long time. This will work even if the actions are available at states that are some way off the current path.

In practice, the system described in section 5.2 does not seem to have difficulty with the first problem in a domain as simple as the grid. If a path is blocked, then the incomplete annealing in the system, which was discussed above, typically allows it to learn an alternative route without great difficulty. Sutton does present other domains which are substantially more difficult. The troubles with the DYNA solution to the second problem are that the parameters governing the relative importance of exploration (driving it to try out 'stale' actions) and exploitation (driving it to get to the goal) are very sensitive to the size of the problem, and that the extra data structure may need to contain very large numbers.

Chrisley [24] also discusses an alternative to relaxation planning. His system, which was mentioned above, learns a forward map, from current state and action to next state. Using Jordan's techniques [70] for inverse modelling discussed in chapter 3, the system could learn an inverse map from this forward map, relating current state and desired next state to action. There being no buckshee lunches, this method will not be capable of automatically handling the exponential intractabilities in action planning. A further, incremental, alternative Chrisley suggests involves clamping the current state and the desired next state (or at least its perceptual properties, given the mechanics of his net), choosing a random action, and then backpropagating the resulting error to criticise the choice of action, rather than the choice of weights. This is an application of Linden and Kindermann's method of inverting a network [85]. Unfortunately, not only is it subject to the usual exponential explosion, but also the resulting choice will only make sense under strong conditions on the metric on the space of possible actions.

Although an update graph may be a compact representation of an environment, this does not guarantee efficient planning. In practice, Bachrach (personal communication), in the work following [102], uses TD methods for navigation, possibly augmented by a dead-reckoning heuristic method which would not appear to be fruitful in the grid task with the barrier.

5.5.3 'What'

The third strand of work in this area covers the nature of the function representation used. Earlier in the thesis the two different rôles that connectionism was playing were discussed – the methodological one of encouraging the search for alternative algorithms, and the practical one of offering old methods of function approximation in new and easy-to-use guises. The linear method that Barto, Sutton and Watkins used, has simplicity to recommend it – other schemes might be expected to perform better.

Andrew Moore [100] has used kd-trees (see Omohundro [115] for a description of these and their application to connectionist function approximation) to precisely this end. They combine many of the desirable properties of connectionist systems (particularly local access and generalisation, at least under particular norms) with speed of use and economy of storage. Moore applied them successfully to a variety of control problems, including ones far more difficult than this.

In terms of section 5.3, which looked at alternative representations, kd-trees are essentially distributed representations, allowing neighbouring grid points to share information. They are not, however, sensitive to the nature of the task, and so would happily mix together points on either side of the barrier.

A further approach which is quite similar to the prediction units is to feed the raw stimulus vectors into a Kohonen-style mapping device [76]. The outputs of this would then be 'topographically' organised in the task rather than the world (so coping appropriately with the barrier), and might make for faster learning. In principle, the Kohonen map would suffer from the same problem as the prediction units in that once the agent starts running regularly on good paths to the goal, the topography will lose almost all connection to the spatial layout of the grid. This is again because proximity along a path is being used as a surrogate for proximity in the world. The dimension, size, and speed of annealing of the Kohonen map will all be problem dependent.

5.6 Conclusions

Map building and navigation are very complex interrelated problems. There has been a tendency to ignore the latter in favour of the former, although the planning involved in navigation seems to be computationally tractable only if it is based on a known model. Notably, the existence of place cells, while in itself a remarkable finding, does not establish a complete theory of cognitive mapping.

Temporal difference methods provide a general way of addressing such planning problems. In particular they can be used neatly to solve Barto, Sutton and Watkins' didactic grid task, which bears certain similarities with the open field water maze. TD is able to work with whatever representation it is fed, and the different effects of the two dimensions on which representations can vary (distributedness and orientation dependence) have been teased apart. Distributed representations prove better – especially in a task like this for which the optimal choice of actions is continuous almost everywhere. Representations that are sensitive to the orientation of the agent are generally worse; however one based on scattering recognisable cues around the environment is adequate to the task, and indeed its distributed nature makes it perform rather well.

Intermediate between having a complete map and having no map at all, it is possible to use a representation which is sensitive to the structure of the task. The representation need not necessarily be isomorphic with the task (in the cases where this would even make sense). Parsimoniously, the same mechanisms that enable the agent to learn how far away any location is from the goal, enable it also to learn such a representation, based on so-called prediction units. In this case, their use does indeed hasten the overall solving of the problem.

Despite ostensible advances in this and related areas, modeling the current state of neurobiological and psychological knowledge is still far too taxing. Although the evidence is quite compelling that rats really do have some form of a map, it is far less clear under what circumstances they use it, what methods they have for solving the planning problem, and indeed how much navigation their more general cognitive systems can cope with. Also, essentially arbitrary decisions were made for the model described here in the experimentally important issues of the method of exploration and the reward structure.

The use of recurrent networks to overcome potential ambiguity of stimuli seems to carry certain risks. Particularly worrying are the question of stability of the environment, and the uniqueness problem in navigating based on inverse models. Apart from this, though, the recent extensions to relaxation planning and the use of more appropriate representations augur well for development, at least in pure engineering terms.

Chapter 6

Context in Connectionism

"Æpilogia": The end of the academic beginning.

6.0 Summary

Although chapter 1 discusses the relationship between learning and levels-based accounts of computational systems, it does not justify why learning is important. This chapter considers the rôle context might play in making inference tractable, briefly reviews the development of knowledge representation and inference techniques as trying to take advantage of ever larger 'chunks' of context, and suggests that contextual dependency can never be fully captured through introspection and calculation – learning will be necessary. Finally, in an attempt to motivate alternatives for context sensitive inference, it paints a caricature of a von Neumann computer as a finite state machine.

6.1 Introduction

Knowing, from chapter 1, how learning can be incorporated into a levels-based description of connectionist and classical systems, and seeing in the rest of the thesis theoretical and empirical analysis undertaken in this framework, may not have sufficiently justified the importance of learning. In addition, no notion of the link between symbolic and connectionist systems has been presented.

Two arguments support the importance of learning; one extolling sloth in investigating the regularities in the world, and the other denying our sleuthing capacities to do this at all. Neither is tied to connectionism. A conclusion of the discussion in chapter 1 was that it is partly the statistical regularities in the world that make computation worthwhile. The argument for sloth says that it is wasteful to try to calculate ahead of time all the regularities a system should be sensitive to. Rather, it should be equipped to learn them directly from its environment, perhaps starting from some prior knowledge garnered from previous learning experiences by it or other systems.

The considerations telling against human sleuthing abilities are rather more speculative. The only alternative proposed to some form of explicit learning of the regularities is some form of indirect learning mediated by introspection and calculation. This is normal practice in one section of traditional AI, in which knowledge engineers elicit expertise from experts, typically using verbal protocols. This learning is indirect because the experts and knowledge engineers had to learn about and/or seek out the regularities in the first place before regurgitating it.[1] Wherein lies the guaranteed power of introspection though? Repeated attempts at making an analogy between scientific theories in such areas as quantum physics and folk psychological theories in such areas as restaurants have failed (see [109] for a discussion of this and [107] for an even stronger conclusion). It would be to make a very strong and unsupportable claim about the organisation of human cognition to require experts in a domain to have introspectable or at least calculable access to all the details of their expertise. Anything less, and the sleuthing will fail.

The issue of *effective* calculability is where this all bites. For instance, I can report how I perform a particular analytic integration, but my account is radically enthymematic. At every stage I am faced with a multitude of options, many of which may appear promising. As an expert, I can report why I chose option A over option B, but not why I never even considered choosing options C-Z. The devil lies in the details of the choice.

[1] Ignoring the stronger (philo)sophist claims [40] that everything is innate in any case.

It seems that this is exactly where attempts to increase the size of knowledge bases founder. More knowledge means more possibilities for every inference, and the expertise lies as much in the heuristics of the choice as to which should even be contemplated as in the facts themselves. I can find no evidence that these heuristics are introspectable or calculable. Acquisition of them is one way of looking at how experts transcend novice behaviour with its rigid, step-by-step consideration of all possibilities. In fact, domains such as integration in which there is even any semblance of decision between choices are not ubiquitous. Our abilities in abduction (inference to the best explanation) are quite mysterious – the whole task lies in the choice.

This chapter takes a rather sideways look at these concerns through context. Its answer to the question of how expert choices are made is that prior experience in the particular inferential context determines which options are best, and that the all-pervasive and delicate nature of context-sensitivity provides a strong constraint on the mechanisms that can represent it adequately. Attempts in traditional AI to capture this have lead to the more interesting species of knowledge representational methods, but ultimately to little avail.

The next section looks at various notions of context, from the all-embracing views of Dreyfus to the rather more limited ones that have influenced the development of AI, section 6.3 looks at how one aspect of context, defeasible inheritance and inference, has been successively (but hardly successfully) incorporated into traditional AI knowledge representation techniques such as semantic networks, frames and scripts, and section 6.4 develops a rather fanciful account that ties together a connectionist method of handling context together with combined connectionist and classical processing. The latter is not intended to be a definitive proposal, merely a hint for the far future.

A slight scent of empiricism, a rather discredited doctrine, might be thought to pervade this and some of the previous chapters. Although the mechanisms might seem so simple as to exclude prior organisation, this is not in fact the case. Consider the mapping system in chapter 5; it learns about arbitrary environments only given its very particular structure. Generally there must be a balance between 'nature' and 'nurture', but nothing is said beyond this platitude. This chapter is intended to be unconstrained by the mechanisms discussed above; rather just to roam fancy-free.

6.2 Context

> *"The passages before and after a sentence that determine its meaning."*
> Oxford English Dictionary

Context is an issue that has long taxed traditional AI, because it seems at the heart of three crucial computational and psychological issues – 'common-sensible' reasoning, tractability and learning. Despite the rather narrow dictionary definition above, Putnam [124], amongst innumerable others, has pointed out just how deeply context affects our entire interaction with the world, in a way that cannot be captured purely linguistically. Here, it is taken as defining the cognitive *milieu* in which observations, utterances, and behaviour, are to be understood. Context affects both conscious and sub-conscious processing; examples of the latter coming from priming effects in syntax and memory tasks.

At its most basic, context can determine expectations for resolving ambiguities, predicting properties or behaviour, and divining what in a situation is unusual, deserving of comment, or worth remembering. However, there are more all-embracing versions. In the rest of this chapter, it is these expectations that are important, since they offer a glimmer of hope of making defeasible inference tractable. Three notions will be barely more than sketched; grounded context in the sense of Heidegger, Lakoff's more restricted view of how metaphor based on our prepositions governs our cognitive economy, and finally 'typicality', the grin left after the departure of the contextual cat.

6.2.1 Grounded Context

Hubert Dreyfus' initially almost solitary banshee wail against AI [32, 33] contains a claim that it is impossible to encode explicitly enough of the essential facts about the universe, and human beings' place in it, for a machine actually to understand, in our sense, the same entities as us – we are fundamentally *grounded* in the world, whereas computers are not. In [32], an attack on frame theory (which itself is discussed in the next section), Dreyfus discusses pre-computational attempts by such figures as Husserl [65] to categorise objects rigorously, and cites Husserl's pessimistic conclusion that phenomenology is an 'infinite' task, since any enquiry into the nature of an object always leads to ever deeper questions (an adult version of a child's relentless "Why?"). Dreyfus paraphrases Heidegger's conclusion on the matter as:

> '...since the outer horizon or background of cultural practices was the condition of the possibility of determining relevant facts and features and thus prerequisite for structuring the inner horizon, as long as the cultural context has not been clarified the proposed analysis of the inner horizon of the *nœma*[2] [can]not even claim progress.'
> [32]-p182

Dreyfus illustrates this view by considering some of the ingredients for a computationally useful definition of a chair:

> 'What makes an object a *chair* is its function, and what makes possible its role as equipment for sitting is its place in a total practical context. This presupposes certain facts about human beings (fatigue, the way the body bends) and a network of other culturally determined equipment (tables, floors, lamps) and skills (eating, writing, going to conferences, giving lectures, etc). Chairs would not be equipment for sitting if our knees bent backwards like those of flamingos, or if we had no tables, as in traditional Japan, or the Australian bush.'
> [32]-p183

He thus delineates a (literal) view of common sense, and claims that only those things that experience the world in the way we do, can conceive it in the way we do – there is a very close relationship between our experience, conception and perception of the world. This objection has particular force against those knowledge representations that are hand-crafted on the basis of introspection and conscious reports, since understanding them presupposes this vast shared cultural background – the reports experts might make are

[2] Mental representation.

merely the accessible tip of the iceberg of their meaning. This criticism is less damning to those systems, such as Brooks' insects [19] that do at least operate in the real world, even if they cannot currently learn from it, and those others, such as connectionist systems, that can learn from their experience, even if it is not like ours. This is only very weak reassurance for the programme in this chapter. Dreyfus' conclusion is essentially that although humans have the competence to recognise and categorise chairs, no computational theory that can describe this competence in isolation from a computational theory of almost everything – a strong holism.

6.2.2 Metaphorical Context

Lakoff states a view accordant with this [78]. He argues that our concepts are at the centre of our understanding and are crucially determined by our uniquely human experience. For instance, the prepositions *'up'* and *'down'* have expressive power outside their narrow (classical) dictionary definitions, *eg* in describing feelings and status, precisely because we have notions of them situated with respect to our bodies. This common background again tells against the usefulness of only the tip of the expert's iceberg above. Lakoff battles strongly against the classical symbolic approach to understanding language, and concludes, like Dreyfus, that since machines do not situate their concepts like us, they cannot 'understand' like us either.

In a later paper entitled 'A suggestion for a linguistics with connectionist foundations' [79], Lakoff discusses the possibility of understanding metaphor and metonymy and the construction of concepts in terms of topographic maps between the sensory systems and the brain. He argues that we can only learn from what we understand, that we understand by structuring novel experience in terms of our existing cognitive framework, and that at the base of this inductive process is our physical constitution, and the way it is represented in the brain. This is obviously closer to a mechanisable proposal; once again it is the learning that is key.

6.2.3 Typicality

Agre and Chapman, in their work on Pengi [1], have recently made an attempt to grasp Dreyfus' nettle and mechanise a small part of the extravagantly unmechanisable proposals in Heidegger's *Weltanschauung*. They built a finite state machine for playing a video game in (almost) real time. It controls a penguin which moves around a simple environment, avoiding attacks from malicious bumble-bees whilst trying to squash them. Agre and Chapman's solution to the computational intractabilities inherent in handling large numbers of bees; the 'bee-31a', 'bee-31b' *etc* of a traditional system, is to adopt indexical representations, of the character of 'the-bee-that-is-currently-attacking-me' or 'the-bee-I-am-trying-to-squash'. This takes advantage of background conditions (such as only one bee can attack at once) to save substantial unnecessary inference (such as asking and answering questions like 'Is bee-31a attacking?', 'Is bee-31b attacking?'). Of course, such a representation hinders some tasks, as for instance if arbitrary numbers of bees might be squashed simultaneously. Their wider claim must therefore be that our environments also only require a restricted competence, which can be serviced by restricted representations.

Similar benefits might be available from an equally impoverished notion of context, essentially capturing *typicalities* of various sorts – *eg* the 'normal' properties of objects in complex circumstances, and the habitual behaviour appropriate to people with various goals or motives. The crux of the problem below is that the full complexity of a context can be relevant to these typicalities – for example a typical restaurant would have waiters, a paradigmatic hamburger restaurant would not, but a normal *posh* hamburger restaurant would. Again, expert knowledge in some domain also involves sensitivity to the complete context – expert drivers tend not to change gear when turning corners at high speed, even if this contravenes usual practice based on the engine note. Traditional AI systems are led to their intractability by the difficulties of representing and manipulating these intricacies, which are essential for flexibility rather than brittleness. Having to introspect them makes matters even worse.

Psychological experiments on prototypicality (see [5, 6] and references therein for some examples) reveal a plethora of related effects such as the time taken for us to judge whether some object is a member of a class and the degree to which we consider it to be representative of that class. Contextual effects are seen in the biasing of these results that occurs if appropriate prior information is provided. Other experiments such as those on priming also show how even very limited contexts can affect our memory retrieval.

6.3 Knowledge Representation and Context

These typicalities are genuine regularities in the world, albeit ones which have resisted folk psychological formalisation. As with Agre and Chapman's indexical representations, their potential value seems to lie in restricting the amount of complete and correct inference it is necessary to do on the fly. Even if our sensitivity to them simply arises from a failing in our computational substrate, they are still psychologically interesting, and they may actually helpfully constrain explanations of our methods of knowledge representation and inference (KRI).

Those involved in trying to build AI systems are faced with a similar problem – how to produce usefully correct answers in adequately short times from limited computational resources. It is possible to view the development of KRI techniques in AI, from classical and then non-classical logics through semantic networks to frames, scripts, and production systems, in terms of the increasing sensitivity to larger 'chunks' of context. As will become apparent, each of these developments has a common pattern – increasing the size of the chunks increases the speed of inference in cases that are regular with respect to the encoded context. However, computational intractability irrepressibly pops up again in the irregular cases, and yet more fiercely, the larger the chunks.

There is an ongoing debate about the use of logics of various sorts for KRI. Part of the confusion arises from infelicities over levels of analysis; as discussed in chapter 1 the 'neat' logicians (*eg* [58]) pieced together a computational level theory of semantic networks, which showed the nature of its partial equivalence to first order logic. Also the advent of logic programming languages such as PROLOG makes computational level theories appear algorithmic. However, particularly in the light of the Pauline conversion of one of its strongest proponents [87], and as persuasively argued by Oaksford and Chater [108] amongst others, the main criticism of the direct use of logics is the radical intractability of their theorem provers. Inference over formal logics with reasonable representational power is unacceptably difficult, being computationally NP. Classical logic, even though forming part of our repertoire in restricted circumstances, fails to capture aspects of our KRI, and more exotic forms better equipped in this direction, such as non-monotonic and fuzzy logics, are either more or equally intractable, or have associated undesirable computational properties, such as awarding some likelihood to the proposition $p \wedge \bar{p}$. 'Pure' logics generally use no context, and defeasible logics typically only license unacceptably weak conclusions [87].

Semantic networks were the first attempt to use context to speed inference. They are designed to represent hierarchical knowledge, for which all information should be located at the most appropriate place. So, for instance, the fact that restaurants tend to have waiters is attached at a position close to restaurants in the hierarchy, whereas that fast-food emporia tend to serve fries is attached further down, as it is not true of all restaurants. This is effectively wrapping up a very impoverished set of typicalities (*ie* regularities in the world), in such a way that inference respecting them can be speeded up – it is not necessary to search exhaustively through a complete database of facts to determine one set of employees of *Maxims*. Indeed, in suitably restricted domains, and depending on the way the inheritance hierarchies are constructed, Quillian [126] demonstrated that semantic networks can reproduce certain results about how long it takes humans to decide whether an object has a particular property, or is a member of a particular class.

Unfortunately, as the size of the domain is increased, it rapidly becomes apparent that handling knowledge in such small chunks – items are individually positioned in hierarchies – is untenable. Worse, when atypicalities plague the prior expectations from the hierarchy, as in the example of the posh hamburger restaurant which *does* have waiters, intractability re-emerges. Once again it becomes necessary for the system to manipulate each piece of knowledge explicitly.

Expanding the small size of the unit of representation in semantic networks was the spur to developing frames and scripts. Minsky first posited frame theory [99] more as a methodology than a mechanism. As Marr [96] points out, and indeed as happened with semantic networks, Minsky also introduced frames more at an algorithmic than a computational level of concreteness. Minsky wrote:

> 'It seems to me that the ingredients of most theories both in Artificial Intelligence and Psychology have been on the whole too minute, local, and unstructured to account – either practically or phenomenologically – for the effectiveness of common-sense thought. The 'chunks' of reasoning, language, memory and perception ought to be larger and more structured; their factual and procedural contents must be more intimately connected in order to explain the apparent power and speed of mental activities. ...
>
> ... Here is the essence of the theory: When one encounters a new situation (or makes a substantial change in one's view of the present problem), one selects from memory a structure called a *frame*. This is a remembered framework to be adapted to reality by changing details as necessary.'

Interestingly, Minsky does make a computational level proposal in the appendix to this paper, albeit a purely negative one. He attacks logic-based approaches to KRI, apparently the only computational straw accessible to the traditionalists, on various counts, including monotonicity, the impossibility of formalising common-sense (or typicality-based) knowledge, and the resulting combinatorial explosion and intractability. He concludes that formal logics are not appropriate for KRI, and, since logical *completeness* is essentially trivial, that logical *consistency* ought to be sacrificed.

In practice, this last suggestion has proved too difficult to follow up. There is almost a slippery slope to logical consistency. For an initially logical framework, there is no obvious way to decide *exactly* where consistency can be abandoned. In the absence of a computational alternative, therefore, implementors produced systems that the 'neats' [59] could once again attack. They demonstrated the same mixture as for semantic networks of essential computational equivalence to some formal logic, coupled with delinquency in permitting absurd conclusions. Note that logical completeness is far from trivial in connectionist systems, and that statistical methods can eliminate at least some forms of absurd conclusions (although possibly introducing some new ones of their own).

Minsky's frames are algorithmically useful because they improve tractability in the same two ways as semantic networks, although acting on larger chunks:

- They contain default information, *ie* expectations of the values (*fillers*) of attributes (*slots*).
- The slots are attached at 'appropriate levels ' in the knowledge representation hierarchy, so, for instance, the information that birds breathe could be stored as a fact about animals, rather than cluttering up the frame dealing specifically with birds.

Unfortunately, also just like semantic networks, they can only represent an impoverished form of context (consider once more the posh fast-food restaurant), and the combination of atypicality with default expectations is intractability.

Schank and Abelson [136], who were thinking along similar lines to Minsky, developed a slightly different notion called scripts. The similarity comes in the use of larger chunks, but scripts wrap up knowledge about expected trains of events in particular circumstances rather than frame theory's object attributes. The paradigmatic example is the restaurant script, which contains the default expectation that customers enter, may be shown to their tables by a hostess, get given menus by a waiter, read the menu, order, get served, *etc.* Schank and Abelson explicitly recognised the problem of richer context dependencies, allowing different *tracks* as adaptations of the basic script to particular circumstances, such as fast-food restaurants. However, just as for frames, the tracks can only cope with impoverished contexts, and their combination can lead to intractability. Script theory is also essentially algorithmic rather than computational, but it has proved harder to translate into formal logics than frame theory, given its emphasis on temporally ordered trains of events.

Schank subsequently moved away from scripts towards dynamic memory and Memory Organisation Packets (MOPs) in [135], and it is possible to interpret his dissatisfaction with scripts in the light of inadequate representation of context.

At least, this book is mainly concerned with learning. However, it focuses particularly on episodic rather than semantic memory, and the notion of context underlying episodic memory exhibits much more this 'one-shot' nature. The problem Schank addresses is how our vast memory for particular events can be indexed so that we can be reminded appropriately at just the right moment, and he concludes that the way an event differs from the expectations held in semantic structures like scripts is the key. Unfortunately, the scheme seems to require underpinning with something akin to the old semantic memory scripts, and the new theory does not extend to show how the old context problems can be solved.

In conclusion, although it is possible to view the development of classical mechanisms of knowledge representation as attempts to represent and use more sophisticated notions of context, none of them is particularly successful. To borrow a phrase from Chater and Oaksford (personal communication), context sensitivity is fractal – it re-emerges at every grain of representation. This is what destroys these valiant attempts.

6.4 Epilogue

Bluntly, no connectionist mechanisms fully solve the problems or grasp the opportunities of context either. One advantage they have is that their notion of parallel processing fits more naturally with context sensitivity, another is that they may be better equipped to learn the regularities for themselves, avoiding the problems Dreyfus and Lakoff pointed out about the cultural background inherent in understanding experts' reports. The classical systems described above had difficulty in grasping the joint significance of the two adjectives 'fast-food' and 'posh'; giving all such facts their due effect led to intractability. Are there alternatives?

One rather fanciful account was developed in [110]. It starts from the trivial observation that von Neumann machines are finite, sketches an alternative picture of them as finite state machines, and then relaxes the von Neumann constraints on transitions between states. This is a simple route to allowing holistic context effects, and rather radical revision of the contents of memory. It does not solve any of the problems raised – it is merely a thought experiment to show that the range of possibilities is not exhausted by traditional systems, and that a simple redescription of a familiar system suggests how very different properties might readily be available.

There is always a slight prestidigitation in regarding the finite von Neumann machines (vNMs), our work-a-day computers, as infinite Turing Machines (TMs). The obvious evolutionary path from Turing to von Neumann machines is to leave unchanged the nature and rôle of the central processor (the reader) and its states, and to regard the memory of the vNM as a finite tape. Arbitrary locations on this virtual tape can be accessed in O(1) rather than O(N) time, where N is its virtual length. The ability to write to or read from arbitrary locations simply makes a vNM considerably easier to program than a TM, and the (almost) size-independent read/write times, make a large class of algorithms computationally feasible. However, the recent debate pitting Reduced Instruction Set (RISC) against Complex Instruction Set (CISC) microprocessors points to an obvious trade off between the complexity of the set of states and state transitions of the central processor, and the complexity and length of the programs that perform a task. This is rather hidden in the details of Turing universality.

Since the memory of a vNM is finite, one can take this view on complexity even further and draw the 'state *versus* tape' boundary around not only the central processor, but also some or all of the memory. Under the new description, states represent not only the contents of the internal flags and registers of the microprocessor, but also the entire contents of memory – roughly, all the transistors that comprise the operation of the machine. This new description is just that of the entire computer as a finite state machine (FSM), which of course is all its finite memory permits it to be. State transitions are instigated as before by the operation of the central processor, but now the state transition diagram is extremely rich and complex. The way that vNM machines are designed has the effect of enforcing what might be described as locality in this state transition diagram, since were some form of metric assigned to the states in an appropriate way (based the 'invisible' values of the program counter, the registers and flags, the possible values of the memory location pointed to by that program counter and then all the other possible values of all the other memory locations), most of the next states would be local.

The complexity and richness of the set of states and the state transition diagram in this alternative viewpoint (which account for the infrequency of its adoption) are precisely its attractions here. The distinctions between program and storage appear considerably eroded, and can be eroded further. Note that at this stage this is only an alternative perspective of exactly the same machine, which is condemned to traverse exactly the same states and execute in exactly the same way exactly the same programs.

Various key properties of computation look very different on this alternative picture of the vNM, particularly symbolic representation and the potentially holistic influence of memory. What could previously be identified as separate contents of memory can only be found, if at all, in the **labels** of a whole set of states, and the labels of course have no causal rôle in engendering the behaviour of the system. It is therefore no longer even sensible to try to isolate out particular symbol structures stored at particular places in memory or to determine the nature and provenance of their logical effects. The symbol processing properties of the entire machine are entirely emergent from the dynamics of the simple low level entities in this picture.

In addition, the whole of memory is wrapped up in a state, so moving from one state to another could correspond to a radical change in its entire contents. Of course, the vNM architecture prevents such transitions from happening – memory is like tape, inviolate except serially. One justification for this severe constraint is that such radical changes are potentially dangerous in the absence of any well-founded method of policing them. However, different mechanisms for state transition need not necessarily labour under such constraints. Removing another such bar could allow the whole contents of memory not only to be changed by such a state transition, but also to determine which state transition occurs. This new picture naturally encompasses a much richer theory of the interaction between memory and inference, mediated in a complex fashion by the hardware of the machine. It would also appear to be more commensurable with the views on cognition of Maturana and Varela [97]. They consider the nervous system less as a machine responding via some program with a set of outputs to a set of inputs, and more as a system whose states and dynamical state transitions are affected by all its components, including ones whose characteristics are determined by and determine events in the external world.

The hardware of the machine yokes the permissible state transitions, and hence its entire behaviour. One possible change is to consider the states as the local minima of some function (an *energy function*, for want of a better term), and the state transitions as movements from one local minimum to another (*cf* [63, 61]). The hardware can be considered as moving the system around the energy surface, according to its principles, landing up in states at the local minima. Learning changes the energy surface so as to alter either the minima, or the traversable paths from one minimum to another, and consequently the states actually entered (∼ the behaviour produced) by the machine under given conditions. This, as advertised, licenses state transitions other than those allowed by the vNM, in which the whole contents of memory are causally implicated in every move. In [110], partly influenced by Anderson's ACT* [3], we considered a hybrid of a traditional structure-sensitive inference engine coupled with a non-standard memory which stores the database of rules and facts.

Thus, contextual flexibility enters the picture. In the old, von Neumann, machine, the problems arose in explaining how whole sets of modifiers ('fast-food', 'posh', *etc*) could influence the conclusions about the presence or absence of waiters, without each one and its consequences being painfully considered in turn. Under the new account, these facts are all wrapped up together in the energy function, allowing them a holistic effect. Of course, since the energy function is unspecified, and currently unspecifiable, this whole description is no more than just an intuition pump about how altering the mechanism might alter the behaviour. The real solution will lie in the hidden details.

Returning to the introduction to this chapter, how might learning work on this picture – which laws should govern the shaping of the energy function? There is ample evidence that associative principles in the traditional sense will be important, with items retrieved on the basis of their content. Reinforcement learning will also be necessary to criticise particular retrievals in particular contexts, modifying the associative links on the basis of performance.

More significantly, much of the inference has the same character as the maze task in chapter 5 – the system will retrieve many rules and facts on the way to solutions, some of which will ultimately be irrelevant or positively harmful. It was suggested above that one part of the transition from novice to expert behaviour lies in moving from considering and/or retrieving explicitly every piece of information and every rule to which it might be relevant, to considering or retrieving only the appropriate facts and rules for a particular context. Experts find this process of selection very difficult to explain, since it is almost literally wired in as part of the expertise, and is never explicitly performed. Becoming an expert may be such a painful process precisely because these context sensitivities have to be built into the retrieval system in such a deep and holistic fashion.

Since now the problem is like that in chapter 5, where the information from the environment is impoverished in terms of its criticism of rule or fact retrieval (equivalent to action selection), perhaps the solution may not be too dissimilar either. The agent in the grid task learns to associate particular contexts with particular actions, based on the quality of its performance over whole sequences. In the same way, experts might be learning to associate particular contexts with the particular rules or facts appropriate there, based on slower and faster inferential performance (*ie* more and fewer retrievals respectively). Of course, this says little to the issue of the representation of the rules and facts – some structure sensitive method will definitely still be required, as indeed will the structure sensitive inference engine. However it does suggest a way that semi-supervised learning can be used for these symbolic tasks.

Even though it includes principles rooted in behaviourism – the associative memories from chapter 2, and the reinforcement and secondary reinforcement principles from chapters 3 - 5 – this proposal is clearly not behaviourist itself. Qualitatively, the system can combine compositionality and systematicity, Fodor and Pylyshyn's [42] *sine qua non* for structure-dependent symbolic processing, together with the capacity to handle radical context sensitivities. It has the power to learn these sensitivities by observing the regularities in its environment and to use them to reduce the exponential intractabilities of inference. Fantastic?

Coda

Much Ado About Not a LoT

Dramatis Personæ

The House of Arragon	**The Symbolic Paradigm**
Don Pedro:	
Benedick	Symbolic computations
Claudio	Symbolic mechanisms
Don John	Jerry Fodor
Borachio	Biological nirvana
The House of Messina	**The Connectionist Paradigm**
Duke Leonato:	
Beatrice	Connectionist computations
Hero	Connectionist mechanisms
Dogberry Verges	Shallow fools

The Plot[1]

The Prince of Arragon, with Claudio and Benedick in his suite, visits Leonato, Duke of Messina, father of Hero and uncle of Beatrice.

The sprightly Beatrice has a teasing relationship with the sworn bachelor Benedick.

The teasing tradition between connectionism and traditional AI is well established. The former is often decried as an amorphous collection of half-baked ideas stolen, possibly unknowingly, from the creative ovens of more serious sciences, and applied indiscriminately either with malice aforethought, or, and more likelily, without aforethought at all. Additionally, it is accused of being a technique whose contribution is irrelevant to the heartland mechanisms of cognitive science. At best it provides a way of implementing existing insights in passably efficient ways. Connectionism's prolix enthusiasms to explain everything are seen as repeating the errors of the past.

Conversely, symbolic AI is pilloried for its evident failure to produce adaptable systems at a large scale. Introspection and traditional knowledge engineering are not adequate to the task of divining delicate context sensitivities, and traditional methods are incapable of representing them and using them for inference. Secession from the doctrines is becoming more frequent, in areas such as behaviour-based robotics.

Beatrice and Benedick are each tricked into believing the other in love, and this brings about a genuine sympathy between them.

A few people, particularly in the machine learning and robotics/control communities are trying to trick connectionism and traditional AI into working together – as described above, they should not be in theoretical combat at all. Each provides a set of computational techniques and mechanisms that has its own qualities and they ought to be brought together for best effect. Implementation is an irrelevant issue, since either can be implemented unperspicuously

[1] As taken from Drabble, M, editor (1985). *The Oxford Companion to English Literature.* Oxford, England: Oxford University Press.

in the other.

Meanwhile Don John, the malcontented brother of the prince, thwarts Claudio's marriage by arranging for him to see Hero apparently wooed by his friend Borachio on her balcony – it is really her maidservant Margaret in disguise.

Although Fodor is often associated with symbolic AI, he is notoriously discontented with it [41]. His recent article with Pylyshyn [42] spread confusion through the ranks by taking insufficient note of the relationship between levels of explanation and replication, and by associating connectionism directly with biological neural networks. Their arguments for structured representations do not rule out connectionist mechanisms, which seem to be different from those available traditionally. In addition, these mechanisms must stand or fall on their own computational merits; basking in reflected microscope envy is hardly safe given the very weak relationship between most connectionist and most biological systems.

Hero is publicly denounced by Claudio on her wedding day, falls into a swoon, and apparently dies.

Benedick proves his love for Beatrice by challenging Claudio to a duel.

Confusions over the levels abound even without malign influences. Within the traditional community, the 'neats', coming from a logical perspective, attack the 'scruffies', who have a more algorithmic bent. Indeed, there are many similarities of content and intent between traditional and connectionist computational levels. As an example, there are just starting to be suggestions in the literature that connectionism's computational contrivances can be divorced from its function approximators, and so be applied more easily in a non-connectionist fashion. This is highly appropriate given the historical lack of emphasis on choosing high quality function approximators in connectionism, and the lack of suitable hardware.

The plot by Don John and Borachio is unmasked by the 'shallow fools' Dogberry and

Verges, the local constables.

Statistical computational theories play the rôle of discovering the true hetero- and homo-geneous links between the connectionist and symbolic computational and mechanistic accounts. Learning from the example of biology is different from aping it in every respect.

Claudio promises to make Leonato amends for his daughter's death, and is asked to marry a cousin of Hero's; the veiled lady turns out to be Hero herself. Benedick asks to be married at the same time;

Beatrice [agrees], 'upon great persuasion; and partly to save your life, for I was told you were in a consumption.'

Appendix A

Parameters for Chapter 5

Where:

$x_t \in \{0, 1\}^c$ is the representation of the agent's location at time t,

$r_{t+1} \in \Re$ is the reinforcement received for the move at time t,

$m_{t+1} \in \{n, s, e, w\}$ is the move at time t,

n_t, s_t, e_t, w_t are the weight vectors determining the choice of action,

η is a typical random variable for the action choice,

v_t^λ is the weight vector generating the evaluation function at time t according to $V_t^\lambda(x) = v_t^\lambda.x$,

define

$$\bar{x}_{t+1} = (1 - \lambda)\bar{x}_t + \lambda x_t$$

as the *eligibility* (equivalently trace decay) for learning the evaluation function, and

$$\bar{x}^e_{t+1} = \begin{cases} (1 - \lambda')\bar{x}^e_t + \lambda' x_t & \text{if East was the chosen action at time t,} \\ (1 - \lambda')\bar{x}^e_t & \text{otherwise} \end{cases}$$

as the eligibility for action vector *e*, and similarly for the other actions.

Also, define

$$\epsilon_{t+1} = r_{t+1} + \gamma V_t^\lambda(x_{t+1}) - V_t^\lambda(x_t)$$

where γ is the discount factor, as simultaneously the error in the prediction and the amount by which executing the chosen action was unexpectedly good. Then

$$\begin{aligned} \Delta v^{\lambda}_{t+1} &= \beta \epsilon_{t+1} \bar{x}_{t+1}, \\ \Delta e_{t+1} &= \beta' \epsilon_{t+1} \bar{x}^{e}_{t+1} \end{aligned}$$

and similarly for the other actions.

For mapping, define:

$p^{i}_{t} \in \Re^{c}$ as the mapping weights,

$[p_t]_i = p^{i}_{t}.x_t$ as the vector of activations of the prediction units,

$v^{\lambda_p}_{t}$ as the weights from the prediction units to the value function,

e^{p}_{t} as the weights from the prediction units to the East action function (and similarly).

Then

$$V^{\lambda}_{t}(x_t) = v^{\lambda}_{t}.x_t + v^{\lambda_p}_{t}.p_t$$

and action choice operates on the basis of

$$e_t.x_t + e^{p}_{t}.p_t + \eta^{e}_{t}$$

and similarly for the other action vectors.

Also define

$$\epsilon^{i}_{t+1} = [x_t]_i + \gamma_p(p^{i}_{t}.x_{t+1}) - p^{i}_{t}.x_t$$

and

$$\bar{x}^{p}_{t+1} = (1 - \xi)\bar{x}^{p}_{t} + \xi x_t$$

as the trace decay for the weights from input units to prediction units, and

$$\bar{p}_{t+1} = (1 - \lambda_p)\bar{p}_t + \lambda_p p_t$$

as the trace decay from the prediction units to the value function, and

$$\bar{p}^{e}_{t+1} = \begin{cases} (1 - \lambda'_p)\bar{p}^{e}_{t} + \lambda'_p p_t & \text{if East was the chosen action at time t,} \\ (1 - \lambda'_p)\bar{p}^{e}_{t} & \text{otherwise} \end{cases}$$

as the eligibility from the prediction units to the East action function, and similarly for the others.

Then, update $\mathbf{v}^{\lambda}{}_{t}$ and $\boldsymbol{e}_t$ *etc* using the same formulæ as before, and

$$\begin{aligned}
\Delta \mathbf{p}^{i}_{t+1} &= \rho \epsilon^{i}_{t+1} \bar{\mathbf{x}}^{p}_{t+1} \\
\Delta \boldsymbol{v}^{\lambda_p}_{t+1} &= \beta_p \epsilon_{t+1} \bar{\mathbf{p}}_{t+1} \\
\Delta \boldsymbol{e}^{p}_{t+1} &= \beta'_p \epsilon_{t+1} \bar{\mathbf{p}}^{e}_{t+1}
\end{aligned}$$

and similarly.

In sum, the twelve parameters are:

β	learning rate input – value
β'	learning rate input – actions
β_p	learning rate predictions – value
β'_p	learning rate predictions – actions
γ	discount rate for the value function
γ_p	discount rate for the predictions
λ	trace decay rate for input – value
λ'	trace decay rate for input – actions
λ_p	trace decay rate for predictions – value
λ'_p	trace decay rate for predictions – actions
ξ	trace decay rate for input – prediction
ρ	learning rate for input – predictions

In addition, the random variables η are exponentially distributed, with unit mean. All the results from the simulations are based on averages over 200 runs, and the graphs in chapter 5 also show standard error bars. The values of the parameters are:

Parameter	Experiment						
	5.4	5.8	5.8	5.10	5.10	5.11	5.11
	$\mathcal{R}_{BSW}$	$\mathcal{R}_{\alpha x}$	$\mathcal{R}_{A}$	No LL	LL	No LL	LL
β	0.5	0.5	0.1	0.5	0.5	0.5	0.5
β'	0.5	0.5	0.1	0.5	0.5	0.5	0.5
β_p				0.25	0.25	0.25	0.25
β'_p				0.5	0.5	0.5	0.5
γ	0.95	0.95	0.95	0.95	0.95	0.95	0.95
γ_p				0.65	0.65	0.8	0.8
λ	0.8	0.8	0.2	0.8	0.8	0.8	0.8
λ'	0.2	0.2	0.2	0.2	0.2	0.2	0.2
λ_p				0.9	0.9	0.9	0.9
λ'_p				0.2	0.2	0.2	0.2
ξ				0.8	0.8	0.8	0.8
ρ				0.25	0.25	0.25	0.25

Bibliography

[1] Agre, P & Chapman, D (1987). Pengi: An implementation of a theory of activity. *Proceedings of the Sixth National Conference on Artificial Intelligence, AAAI-87*. Los Altos, CA: Morgan Kaufmann.

[2] Albus, JS (1975). A new approach to manipulator control: The Cerebellar Model Articulation Controller (CMAC). *Transactions of the ASME: Journal of Dynamical Systems, Measurement and Control*, **97**, pp 220-227.

[3] Anderson, JR (1983). *The Architecture of Cognition*. Cambridge, MA: Harvard University Press.

[4] Artola, A, Bröcher, S & Singer, W (1990). Different voltage-dependent thresholds for the induction of long-term depression and long-term potentiation in slices of the rat visual cortex. *Nature*, **347**, pp 69-72.

[5] Barsalou, LW (1982). Context-independent and context-dependent information in concepts. *Memory and Cognition*, **10**, pp 82-93.

[6] Barsalou, LW (1989). The instability of graded structure: Implications for the nature of concepts. In U Neisser, editor, *Concepts and Conceptual Development*. Cambridge, England: Cambridge University Press.

[7] Barto, AG (1985). Learning by statistical cooperation of self-interested neuron-like computing elements. *Human Neurobiology*, **4**, pp 229-256.

[8] Barto, AG & Singh, SP (1990). On the computational economics of reinforcement learning. In DS Touretzky, J Elman, TJ Sejnowski & GE Hinton, editors, *Proceedings of the 1990 Connectionist Models Summer School*. San Mateo, CA: Morgan Kaufmann.

[9] Barto, AG & Sutton, RS (1981). *Goal Seeking Components for Adaptive Intelligence: An Initial Assessment*. Technical Report AFWAL-TR-81-1070, Air Force Wright Aeronautical Laboratories/Avionics Laboratory, Wright-Patterson AFB, Ohio.

[10] Barto, AG, Sutton, RS & Anderson, CW (1983). Neuronlike elements that can solve difficult learning problems. *IEEE Transactions on Systems, Man, and Cybernetics*, **13**, pp 834-846.

[11] Barto, AG, Sutton, RS & Watkins, CJCH (1989). *Learning and Sequential Decision Making*. Technical Report 89-95, Computer and Information Science, University of Massachusetts, Amherst, MA.

[12] Baum, EB & Haussler, D (1989). What size net gives valid generalisation? *Neural Computation*, **1**, pp 151-160.

[13] Bellman, RE & Dreyfus, SE (1962). *Applied Dynamic Programming*. RAND Corporation.

[14] Bienenstock, E, Cooper, LN & Munro, P (1982). Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, **2**, pp 32-48.

[15] Bliss, TVP & Lømo, T (1973). Long-lasting potentiation of synaptic transmission in the dentate area of the anæsthetized rabbit following stimulation of the perforant path. *Journal of Physiology (London)*, **232**, pp 331-356.

[16] Blodgett, HC (1929). The effect of the introduction of reward upon the maze performance of rats. *University of California Publications in Psychology*, **4**, pp 113-134.

[17] Brachman, R (1983). What IS-A is and isn't: An analysis of taxonomic links in semantic networks. *IEEE Computer*, **16**(10), pp 30-36.

[18] Brachman, R (1985). 'I lied about the trees' (or, defaults and definitions in knowledge representation). *The AI Magazine*, **6**(3).

[19] Brooks, RA (1986). *Achieving Artificial Intelligence through Building Robots*. Memo 899, AI Laboratory, MIT, Cambridge, MA.

[20] Buckingham, JT (1991). *Computation in the Archicortex*. PhD Thesis, Department of Artificial Intelligence, University of Edinburgh, Edinburgh.

[21] Buhmann, J, Divko, R & Schulten, K (1989). On sparsely coded associative memories. In L Personnaz & G Dreyfus, editors, *Neural Networks: From Models to Applications*. nEURO88, Paris.

[22] Chater, N & Oaksford, M (1990). Autonomy, implementation and cognitive architecture: A reply to Fodor and Pylyshyn. *Cognition*, **34**, pp 93-107.

[23] Cherniak, C (1986). *Minimal Rationality*. Cambridge, MA: MIT Press, Bradford Books.

[24] Chrisley, RL (1990). Cognitive map construction and use: A parallel distributed approach. In DS Touretzky, J Elman, TJ Sejnowski & GE Hinton, editors, *Proceedings of the 1990 Connectionist Models Summer School*. San Mateo, CA: Morgan Kaufmann.

[25] Churchland, PS (1986). *Neurophilosophy*. Cambridge, MA: MIT Press, Bradford Books.

[26] Clark, A (1990). Connectionism, competence, and explanation. *British Jounal of the Philosophy of Science*, **41**, pp 195-222.

[27] Cohen, M & Grossberg, S (1983). Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **9**, pp 195-219.

[28] Denker, JD, Schwartz, D, Wittner, B, Solla, S, Hopfield, JJ, Howard, R & Jackel, L (1987). Automatic learning, rule extraction, and generalisation. *Complex Systems*, **1**, pp 877-922.

[29] Dennett, DC (1987). *The Intentional Stance*. Cambridge, MA: MIT Press, Bradford Books.

[30] Derthick, M (1988). *Mundane Reasoning by Parallel Constraint Satisfaction*. PhD Thesis, Carnegie Mellon University, Pittsburgh, PA.

[31] Dickinson, A (1980). *Contemporary Animal Learning Theory*. Cambridge, England: Cambridge University Press.

[32] Dreyfus, H (1981). From micro-worlds to knowledge representation: AI at an impasse. In J Haugeland, editor, *Mind Design*. Cambridge, MA: MIT Press.

[33] Dreyfus, H & Dreyfus, SE (1986). *Mind over Machine: The Power of Human Intuition and Expertise in the Era of the Computer*. New York, NY: Free Press, MacMillan.

[34] Durbin, R, Szeliski, R & Yuille, A (1989). An analysis of the elastic net approach to the travelling salesman problem. *Neural Computation* **1**, pp 348-358.

[35] Durbin, R & Willshaw, DJ (1987). An analogue approach to the travelling salesman problem using an elastic net method. *Nature* **326**(6114), pp 689-691.

[36] Duda, RO & Hart, PE (1973). *Pattern Classification and Scene Analysis*. New York, NY: Wiley.

[37] Elman, J (1988). *Finding Structure in Time*. Technical Report 8801, Centre for Research in Language, UCSD, La Jolla, CA.

[38] Fodor, JA (1975). *The Language of Thought*. New York, NY: Crowell.

[39] Fodor, JA (1980). Methodological solipsism considered as a research strategy in cognitive psychology. *The Behavioural and Brain Sciences*, **3**, pp 63-109.

[40] Fodor, JA (1981). The current status of the innateness controversy. In *Representations*. Cambridge, MA: MIT Press.

[41] Fodor, JA (1981). Tom Swift and his procedural grandmother. In *Representations*. Cambridge, MA: MIT Press.

[42] Fodor, JA & Pylyshyn, ZW (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition* **28**, pp 3-71.

[43] Foster, CL (1990). *Algorithms, Abstraction and Implementation: A Massively Multilevel Theory of Strong Equivalence of Complex Systems*. PhD Thesis, Centre for Cognitive Science, University of Edinburgh, Edinburgh.

[44] Gaffan, D (1974). Recognition impaired and association intact in the memory of monkeys after transection of the fornix. *Journal of Comparative and Physiological Psychology*, **86**, pp 1100-1109.

[45] Gallistel, CR (1990). *The Organisation of Learning*. Cambridge, MA: MIT Press, Bradford Books.

[46] Gardner, E (1989). The space of interactions in neural network models. *Journal of Physics A: Mathematics General*, **21**, p 257.

[47] Golden, R (1987). *A Probabilistic Framework for Neural Network Models*. Unpublished manuscript.

[48] Golden, R (1988). A unified framework for connectionist systems. *Biological Cybernetics*, **59**, pp 109-120.

[49] Gullapalli, V (1990). Modelling cortical area 7a using stochastic real-valued (SRV) units. In DS Touretzky, J Elman, TJ Sejnowski & GE Hinton, editors, *Proceedings of the 1990 Connectionist Models Summer School*. San Mateo, CA: Morgan Kaufmann.

[50] Gullapalli, V (1990). A stochastic reinforcement learning algorithm for learning real-valued functions. *Neural Networks*, 3(6), pp 671-692.

[51] Gullapalli, V (1991). *Associative Reinforcement Learning of Real-valued Functions*. In preparation.

[52] Hampson, SE (1983). *A Neural Model of Adaptive Behavior*. PhD Thesis, University of California, Irvine.

[53] Hampson, SE (1990). *Connectionistic Problem Solving: Computational Aspects of Biological Learning*. Boston, MA: Birkhäuser Boston.

[54] Hancock, PJB, Smith, LS & Phillips, WA (1991). A biologically supported error-correcting learning rule. *Neural Computation*, in press.

[55] Harnad, S (1989). The symbol grounding problem. *CNLS Conference on Emergent Computation*, Los Alamos.

[56] Haugeland, J (1978). The nature and plausibility of cognitivism. *The Behavioral and Brain Sciences*, 3, pp 215-226.

[57] Haussler, D (1988). Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. *Artificial Intelligence*, 36, pp 177-221.

[58] Hayes, P (1977). In defence of logic. *Proceedings of the International Joint Conference on Artificial Intelligence*, pp 559-565.

[59] Hayes, P (1979). The logic of frames. In D Metzing, editor, *Frame Conceptions and Text Understanding*, pp 46-61. Berlin, Germany: Walter de Gruyter.

[60] Hebb, DO (1949). *The Organization of Behavior: A Neuropsychological Theory*. New York, NY: Wiley.

[61] Hinton, GE, Sejnowski, TJ & Ackley, D (1984) *Boltzmann Machines: Constraint Satisfaction Networks that Learn*. Technical Report CMU–CS–84–119. Department of Computer Science, Carnegie–Mellon University, Pittsburgh, PA.

[62] Holland, JH (1986). Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems. In RS Michalski, JG Carbonell & TM Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, 2. Los Altos, CA: Morgan Kaufmann.

[63] Hopfield, JJ (1982). Neural networks and physical systems with emergent computational abilities. *Proceedings of the National Academy of Science*, 79, pp 2554-2558.

[64] Hopfield, JJ & Tank, DW (1985). Neural computation of decisions in optimization problems. *Biological Cybernetics*, 52, pp 141-152.

[65] Husserl, E (1960). *Cartesian Meditations*. The Hague, Holland: Martinus Nijhoff.

[66] Ito, M, Sakurai, M & Tongroach, P (1982). Climbing fibre induced depression of both mossy fibre responsiveness and glutamate sensitivity of cerebellar Purkinje cells. *Journal of Physiology*, 324, pp 113-134.

[67] Jordan, MI (1986). *Serial Order: A Parallel Distributed Approach*. Technical Report 8604, Institute for Cognitive Science, UCSD, La Jolla, CA.

[68] Jordan, MI & Jacobs, RA (1990). Learning to control an unstable system with forward modelling. In DS Touretzky, editor, *Advances in Neural Information Processing Systems, 2*. San Mateo, CA: Morgan Kaufmann.

[69] Jordan, MI & Rosenbaum, DA (1989). Action. In MI Posner, editor, *Foundations of Cognitive Science*. Cambridge, MA: MIT Press.

[70] Jordan, MI & Rumelhart, DE (1990). Forward models: Supervised learning with a distal teacher. Submitted to *Cognitive Science*.

[71] Kaelbling, LP (1990). *Learning in Embedded Systems*. PhD Thesis, Stanford University, Stanford, CA.

[72] Kanerva, P (1988). *Sparse Distributed Memory*. Cambridge, MA: MIT Press, Bradford Books.

[73] Kirkpatrick, S, Gelatt, CD Jr & Vecchi, MP (1983). Optimisation by simulated annealing. *Science*, **220**, pp 671-680.

[74] Klopf, AH (1972). *Brain Function and Adaptive Systems – A Heterostatic Theory*. Air Force Research Laboratories Research Report AFCRL-72-0164. Bedford, MA.

[75] Klopf, AH (1982). *The Hedonistic Neuron: A Theory of Memory, Learning, and Intelligence*. Washington, DC: Hemisphere.

[76] Kohonen, T (1982). Self-organized formation of topographically correct feature maps. *Biological Cybernetics*, **43**, pp 59-69.

[77] Kushner, H & Clark, D (1978). *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Berlin, Germany: Springer-Verlag.

[78] Lakoff, G (1987). *Women, Fire and Dangerous Things*. Chicago, IL: University of Chicago Press.

[79] Lakoff, G (1988). A suggestion for a linguistics with connectionist foundations. In DS Touretzky, TJ Sejnowski & GE Hinton, editors, *Proceedings of the Second Connectionist Models Summer School*. San Mateo, CA: Morgan Kaufmann.

[80] Land, E & McCann, J (1971). Lightness and retinex theory. *Journal of the Optical Society of America*, **61**, pp 1-11.

[81] Lawler, EL, Lenstra, JK, Rinnooy Kan AHG & Shmoys DB, editors (1985). *The Travelling Salesman Problem*. New York, NY: Wiley.

[82] Le Cun, Y (1987). *Modèles Connexionnistes de L'apprentissage*. PhD Thesis, Université Pierre et Marie Curie, Paris, France.

[83] Legendre, G, Miyata, Y & Smolensky, P (1990). *Harmonic Grammar – A Formal Multi-level Connectionist Theory of Linguistic Well-formedness: An Application*. ICS Technical Report 90-4. Institute for Cognitive Science, University of Colorado at Boulder, Boulder, CO.

[84] Legendre, G, Miyata, Y & Smolensky, P (1990). *Harmonic Grammar – A Formal Multi-level Connectionist Theory of Linguistic Well-formedness: Theoretical Foundations*. ICS Technical Report 90-5. Institute for Cognitive Science, University of Colorado at Boulder, Boulder, CO.

[85] Linden, A & Kindermann, J (1989). Inversion of multilayer nets. *Proceedings of the International Joint Conference on Neural Networks, Volume II*, pp 425-430.

[86] McCulloch, WS & Pitts, W (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, **5**, pp 115-133.

[87] McDermott, D (1987). A critique of pure reason. *Computational Intelligence*, **3**, pp 151-160.

[88] Mackintosh, NJ (1983). *Conditioning and Associative Learning*. Oxford, England: Oxford University Press.

[89] McNaughton, BL, Barnes, CA & O'Keefe, J (1983). The contributions of position, direction and velocity to single unit activity in the hippocampus of freely moving rats. *Experimental Brain Research*, **52**, pp 41-49.

[90] McNaughton, BL (1989). Neuronal mechanisms for spatial computation and information storage. In L Nadel, LA Cooper, P Culicover & RM Harnish, editors, *Neural Connections, Mental Computation*. Cambridge, MA: MIT Press, Bradford Books.

[91] Marr, D (1969). A theory of cerebellar cortex. *Journal of Physiology*, **202**, pp 437-470.

[92] Marr, D (1970). A theory for cerebral neocortex. *Proceedings of the Royal Society of London, B*, **238**, pp 137-154.

[93] Marr, D (1971). Simple memory: A theory for archicortex. *Philosophical Transactions of the Royal Society of London, B*, **262**, pp 23-81.

[94] Marr, D (1974). The computation of lightness by the primate retina. *Vision Research*, **14**, pp 1377-1388.

[95] Marr, D (1977). Artificial intelligence: A personal view. *Artificial Intelligence*, **9**, pp 37-48.

[96] Marr, D (1982). *Vision*. New York, NY: Freeman.

[97] Maturana, H & Varela, F (1972). *De Maquinas y Seres Vivos*. Chile: Editorial Universitaria. Published in English in 1980 as *Autopoiesis and Cognition: The Realization of the Living*. Dordrecht: Reidel.

[98] Michie, D & Chambers, RA (1968). BOXES: An experiment in adaptive control. *Machine Intelligence* 2, pp 137-152.

[99] Minsky, M (1974). *A Framework for Representing Knowledge*. Memo 305, AI Laboratory. MIT, Cambridge, MA.

[100] Moore, AW (1990). *Efficient Memory-based Learning for Robot Control*. PhD Thesis, University of Cambridge Computer Laboratory, Cambridge, England.

[101] Morris, RGM (1981). Spatial localisation does not require the presence of local cues. *Learning and Motivation*, 12, 239-260.

[102] Mozer, MC & Bachrach, J (1990). Discovering the structure of a reactive environment by exploration. In D Touretzky, editor, *Advances in Neural Information Processing Systems*, *2*, pp 439-446. San Mateo, CA: Morgan Kaufmann.

[103] Mozer, MC & Bachrach, J (1990). Discovering the structure of a reactive environment by exploration. *Neural Computation*, **2**, pp 447-457.

[104] Muller, RU & Kubie, JL (1987). The effects of changes in the environment on the spatial firing of hippocampal complex-spiking cells. *Journal of Neuroscience*, 7(7), pp 1951-1968.

[105] Nadal, JP & Toulouse, G (1990). Information storage in sparsely coded memory nets. *Network*, **1**(1), pp 61-74.

[106] Newell, A (1980). Physical symbol systems. *Cognitive Science*, **4**, pp 135-183.

[107] Oaksford, M & Chater, N (1990). Logicist cognitive science and the falsity of common-sense theories. Submitted to *The Behavioral and Brain Sciences*.

[108] Oaksford, M & Chater, N (1991). Against logicist cognitive science. *Mind and Language*, in press.

[109] Oberlander, J (1989). *How the Laws of Thought Lie*. Research Paper EUCCS/RP-35, Centre for Cognitive Science, University of Edinburgh.

[110] Oberlander, J & Dayan, P (1990). *Altered States and Virtual Beliefs*. Presented at Turing 1990. Sussex.

[111] O'Keefe, J (1983). Spatial memory within and without the hippocampal system. In W Seifert, editor, *Neurobiology of the Hippocampus*. New York, NY: Academic Press.

[112] O'Keefe, J & Dostrovsky, J (1971). The hippocampus as a spatial map: Preliminary evidence from unit activity in the freely moving rat. *Brain Research,* **34,** pp 171-175.

[113] O'Keefe, J & Nadel, L (1978). *The Hippocampus as a Cognitive Map.* Oxford, England: Oxford University Press.

[114] O'Keefe, J & Speakman, A (1987). Single unit activity in the rat hippocampus during a spatial memory task. *Experimental Brain Research,* **68,** pp 1-27.

[115] Omohundro, S (1987). Efficient algorithms with neural network behaviour. *Complex Systems,* **1,** pp 273-347.

[116] Palm, G (1988). On the asymptotic information storage capacity of neural networks. In R Eckmiller & C von der Malsburg, editors, *Neural Computers. NATO ASI Series,* **F41,** pp 271-280. Berlin, Germany: Springer Verlag.

[117] Palm, G (1988). Local synaptic rules with maximal information storage capacity. In H Haken, editor, *Neural & Synergetic Computers, Springer Series in Synergetics,* **42,** pp 100-110. Berlin, Germany: Springer Verlag.

[118] Parker, D (1985). *Learning Logic.* Technical Report TR-47. Sloan School of Management, Cambridge, MA.

[119] Pearl, J (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* San Mateo, CA: Morgan Kaufmann.

[120] Perez-Vincente, CJ & Amit, DJ (1989). Optimised network for sparsely coded patterns, *Journal of Physics A: Mathematics General,* **22,** pp 559-569.

[121] Pollack, JB (1987). *On Connectionist Models of Natural Language Processing.* PhD Thesis, University of Illinois, Urbana, Il.

[122] Pollack, JB (1988). Recursive auto-associative memory: Devising compositional distributed representations. *Proceedings of the Tenth Annual Conference of the Cognitive Science Society,* pp 33-39.

[123] Putnam, H (1975). The meaning of meaning. In K Gunderson, editor, *Minnesota Studies in the Philosophy of Science. 7. Language, Mind and Knowledge.* Minneapolis: University of Minnesota Press.

[124] Putnam, H (1988). *Representation and Reality.* Cambridge, MA: MIT Press, Bradford Books.

[125] Pylyshyn, ZW (1984). *Computation and Cognition: Toward a Foundation for Cognitive Science.* Cambridge, MA: MIT Press, Bradford Books.

[126] Quillian, MR (1966). *Semantic Memory.* Technical Report AFCRL-66-189. Cambridge, MA: Bolt, Beranek, and Newman.

[127] Racine, RJ, Milgram, NW & Hafner, S (1983). Long-term potentiation phenomena in the rat limbic forebrain. *Brain Research,* **260**, pp 217-231.

[128] Riley, MS & Smolensky, P (1984). A parallel model of (sequential) problem solving. *Proceedings of the Sixth Annual Conference of the Cognitive Science Society.*

[129] Rivest, RL & Schapire, RE (1987). Diversity-based inference of finite automata. *Proceedings of the Twenty-Eighth Annual Symposium on the Foundations of Computer Science,* pp 78-87.

[130] Rivest, RL & Schapire, RE (1987). A new approach to unsupervised learning in deterministic environments. *Proceedings of the Fourth International Workshop on Machine Learning,* pp 364-375.

[131] Rumelhart, DE, Hinton, GE & Williams, RJ (1986). Learning internal representations by error propagation. In DE Rumelhart & JL McClelland, editors, *Parallel Distributed Processing: Volume 1,* pp 318-363. Cambridge, MA: MIT Press, Bradford Books.

[132] Samuel, AL (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development,* **3**, pp 211-229.

[133] Samuel, AL (1967). Some studies in machine learning using the game of checkers II: Recent Progress. *IBM Journal of Research and Development,* **11**, pp 601-617.

[134] Sato, M, Abe, K & Takeda, H (1988). Learning control of finite Markov chains with explicit trade-off between estimation and control. *IEEE Transactions on Systems, Man and Cybernetics,* **18**, pp 677-684.

[135] Schank, R (1982). *Dynamic Memory.* Cambridge, England: Cambridge University Press.

[136] Schank, R & Abelson, R (1977). *Scripts, Plans, Goals, and Understanding.* Hillsdale, NJ: Lawrence Erlbaum.

[137] Schmidhuber, JH (1990). An on-line algorithm for dynamic reinforcement learning and planning in reactive environments. *Proceedings of the International Joint Conference on Neural Networks, Volume 2,* pp 253-258. Piscataway: IEEE Press.

[138] Sejnowski, TJ (1977). Storing covariance with nonlinearly interacting neurons. *Journal of Mathematical Biology,* **4**, pp 303-321.

[139] Sejnowski, TJ (1977). Statistical constraints on synaptic plasticity. *Journal of Theoretical Biology,* **69**, pp 385-389.

[140] Sejnowski, TJ & Rosenberg, CR (1987). Parallel networks that learn to pronounce English text. *Complex Systems,* **1**, pp 145-168.

[141] Singer, W (1985). Activity-dependent self-organization of synaptic connections as a substrate of learning. In JP Changeux & M Konishi, editors, *The Neural and Molecular Bases of Learning*, pp 301-335 New York, NY: Wiley.

[142] Smith, B (1982). *Reflection and Semantics in a Procedural Language*. PhD Thesis, MIT, Cambridge, MA.

[143] Smolensky, P (1988). On the Proper Treatment of Connectionism. *The Behavioral and Brain Sciences*, **11**, pp 1-74.

[144] Smolensky, P & Riley, MS (1984). *Harmony Theory: Problem Solving, Parallel Cognitive Models, and Thermal Physics*. Technical Report 8404, Institute for Cognitive Science, UCSD, La Jolla, CA.

[145] Stanton, P & Sejnowski, TJ (1989). Associative long-term depression in the hippocampus: Induction of synaptic plasticity by Hebbian covariance. *Nature*, **339**, pp 215-218.

[146] Stent, GS (1973). A physiological mechanism for Hebb's postulate of learning. *Proceedings of the National Academy of Science*, **70**, pp 997-1001.

[147] Sutton, RS (1984). *Temporal Credit Assignment in Reinforcement Learning*. PhD Thesis, University of Massachusetts, Amherst, MA.

[148] Sutton, RS (1988). Learning to predict by the methods of temporal difference. *Machine Learning*, **3**, pp 9-44.

[149] Sutton, RS (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. *Proceedings of the Seventh International Conference on Machine Learning*. San Mateo, CA: Morgan Kaufmann.

[150] Sutton, RS & Barto, AG (1987). A temporal-difference model of classical conditioning. *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*. Seattle, WA.

[151] Sutton, RS & Barto, AG (1991). Time-derivative models of Pavlovian conditioning. In M Gabriel & JW Moore, editors, *Learning and Computational Neuroscience*. Cambridge, MA: MIT Press.

[152] Sutton, RS & Pinette, B (1985). The learning of world models by connectionist networks. *Proceedings of the Seventh Annual Conference of the Cognitive Science Society*, pp 54-64. Irvine, CA: Lawrence Erlbaum.

[153] Touretzky, DS & Hinton, GE (1985). Symbols among the neurons: Details of a connectionist inference architecture. *Proceedings of the International Joint Conference on Artificial Intelligence*, pp 238-243.

[154] Touretzky, DS & Geva, S (1987). A distributed connectionist representation for concept structures. *Proceedings of the Ninth Annual Conference of the Cognitive Science Society,* pp 155-164. Seattle, WA: Lawrence Erlbaum.

[155] Tsodyks, MV & Feigel'man, MV (1988). The enhanced storage capacity in neural networks with low activity level. *Europhysics Letters,* **6,** pp 101-105.

[156] Valiant, LG (1984). A theory of the learnable. *Communications of the ACM,* **27,** pp 1134-1142.

[157] Vapnik, VN & Chervonenkis, AY (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications,* **16,** pp 264-280.

[158] Varga, RS (1962). *Matrix Iterative Analysis.* Englewood Cliffs, NJ: Prentice-Hall.

[159] Watkins, CJCH (1989). *Learning from Delayed Rewards.* PhD Thesis, University of Cambridge, England.

[160] Werbos, PJ (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioural Sciences.* PhD Thesis, Harvard University, Cambridge, MA.

[161] Werbos, PJ (1990). Consistency of HDP applied to a simple reinforcement learning problem. *Neural Networks,* **3,** pp 179-189.

[162] White, H (1989). Some asymptotic results for learning in single hidden layer feedforward networks. *Journal of the American Statistical Association,* **84**(408), pp 1003-1013.

[163] Whitehead, SD & Ballard, DH (1990). Active perception and reinforcement learning. *Neural Computation,* **2,** pp 409-419.

[164] Widrow, B & Stearns, SD (1985). *Adaptive Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall.

[165] Williams, RJ (1988). *Toward a Theory of Reinforcement-learning Connectionist Systems.* Technical Report NU-CCS-88-3, College of Computer Science, Northeastern University, 360 Huntingdon Avenue, Boston, MA.

[166] Williams, RJ & Baird, LC III (1990). A mathematical analysis of actor-critic architectures for learning optimal controls through incremental dynamic programming. *Proceedings of the Sixth Yale Workshop on Adaptive and Learning Systems,* August 15-17, 1990. New Haven, CT.

[167] Willshaw, DJ (1971). *Models of Distributed Associative Memory.* PhD Thesis, University of Edinburgh, Edinburgh.

[168] Willshaw, DJ (1981). Holography, associative memory, and inductive generalisation. In GE Hinton & JA Anderson, editors, *Parallel Models of Associative Memory.* Hillsdale, NJ: Lawrence Erlbaum.

[169] Willshaw, DJ & Buckingham, JT (1990). An assessment of Marr's theory of the hippocampus as a temporary memory store. *Philosophical Transactions of the Royal Society of London, B,* **329,** pp 205-215.

[170] Willshaw, DJ, Buneman, OP & Longuet-Higgins, HC (1969). Nonholographic associative memory. *Nature,* **222,** pp 960-962.

[171] Willshaw, DJ & Dayan, P (1990). Optimal plasticity in matrix memories: What goes up MUST come down. *Neural Computation,* **2,** pp 85-93.

[172] Witten, IH (1977). An adaptive optimal controller for discrete-time Markov environments. *Information and Control,* **34,** pp 286-295.

[173] Woods, W (1975). What's in a link: Foundations for semantic networks. In D Bobrow & A Collins, editors, *Representation and Understanding: Studies in Cognitive Science,* pp 35-82. New York, NY: Academic Press.

[174] Zipser, D & Andersen, RA (1988). A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature,* **331,** pp 679-684.