University of Otago

The Boltzmann Machine: a Connectionist Model for Supra-Classical Logic

Glenn Blanchette

Supervisors:

Anthony Robins

Willem Labuschagne

Zach Weber & Brendan McCane

Thesis submitted for the degree of

Doctor of Philosophy

at the University of Otago, Dunedin,

New Zealand.

10 June 2017

Abstract

This thesis moves towards reconciliation of two of the major paradigms of artificial intelligence: by exploring the representation of symbolic logic in an artificial neural network. Previous attempts at the machine representation of classical logic are reviewed. We however, consider the requirements of inference in the broader realm of supra-classical, non-monotonic logic. This logic is concerned with the tolerance of exceptions, thought to be associated with common-sense reasoning. Biological plausibility extends these requirements in the context of human cognition.

The thesis identifies the requirements of supra-classical, non-monotonic logic in relation to the properties of candidate neural networks. Previous research has theoretically identified the Boltzmann machine as a potential candidate. We provide experimental evidence supporting a version of the Boltzmann machine as a practical representation of this logic.

The theme is pursued by looking at the benefits of utilising the relationship between the logic and the Boltzmann machine in two areas. We report adaptations to the machine architecture which select for different information distributions. These distributions correspond to state preference in traditional logic versus the concept of atomic typicality in contemporary approaches to logic. We also show that the learning algorithm of the Boltzmann machine can be adapted to implement pseudo-rehearsal during retraining. The results of machine retraining are then utilised to consider the plausibility of some current theories of belief revision in logic. Furthermore, we propose an alternative approach to belief revision based on the experimental results of retraining the Boltzmann machine.

Acknowledgements

I have been very fortunate in finding a primary supervisor, Prof. Anthony Robins, who has been so supportive throughout the research project. He was wholeheartedly involved in every twist and turn of the experimentation: providing insightful direction when I was stuck and encouragement, when he thought I should just get on with it. More than that, I feel that he is a colleague and friend.

Logicians are a scarce resource. Dr Willem Labuschagne has retired from the department at Otago University. His venerable discussions on symbolic logic are irreplaceable. Thank you to Dr Zach Weber, who very kindly agreed to step in at short notice, with the laborious work of editing. I am also grateful to associate Prof. Brendan McCane for his editorial comments.

And not least, theses don't happen in a vacuum, in the 'background' my family life and work has at times been a challenge. I am particularly grateful to my wife Karine, not only for her impatience, in demanding an end to the project; but for her courage. When our child's life was fragile, she was a lion's tooth.[1]

[1] WB Yeats, 1929 - Crazy Jane Grown Old Looks At The Dancers.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Intention

This thesis explores the *connectionist*[1] representation of *supra-classical, non-monotonic* (SCNM) logic and thereby seeks to contribute to some reconciliation between the connectionist and *symbolic* paradigms of *artificial intelligence*. Its aim is to demonstrate that the *Boltzmann machine* is a faithful model of *inference* in supra-classical logic, utilising the *rational consequence* relation. Further, that the association between the network representation and the logic can be utilised for the benefit of both paradigms.

Fodor and Pylyshyn (1988) viewed the two paradigms of artificial intelligence as being in conflict. However many authors, including Gärdenfors (1994), have argued that they are complementary: that *neural networks* can represent the systematic structure of logic because systematicity is intrinsically part of any world that is symbolised by language (Chang, 2002; Frank, Haselager, and van Rooij, 2009).

Logic offers a framework, at an abstract symbolic level, for the understanding of human reasoning (Russell and Norvig, 2003, Chp 8-10). At a lower machine level, neuroscience offers an increasingly sophisticated understanding of the functioning of the brain. Somewhere between the two; connectionist, artificial neural network systems provide a computational framework that is inspired by the brain. A fuller understanding of human cognition should elucidate the relationship between these different levels of description, as captured by Hinton's comments, made originally in 1984, which seem just as relevant today:

> "Ultimately it will be necessary to bridge the gap between hardware-oriented connectionist descriptions and the more abstract symbol manipu-

[1] Terms emphasised at first occurrence, are defined in the glossary.

lation models that have proved to be an extremely powerful and pervasive way of describing human information processing" (Hinton, Sejnowski, and Ackley, 1984).

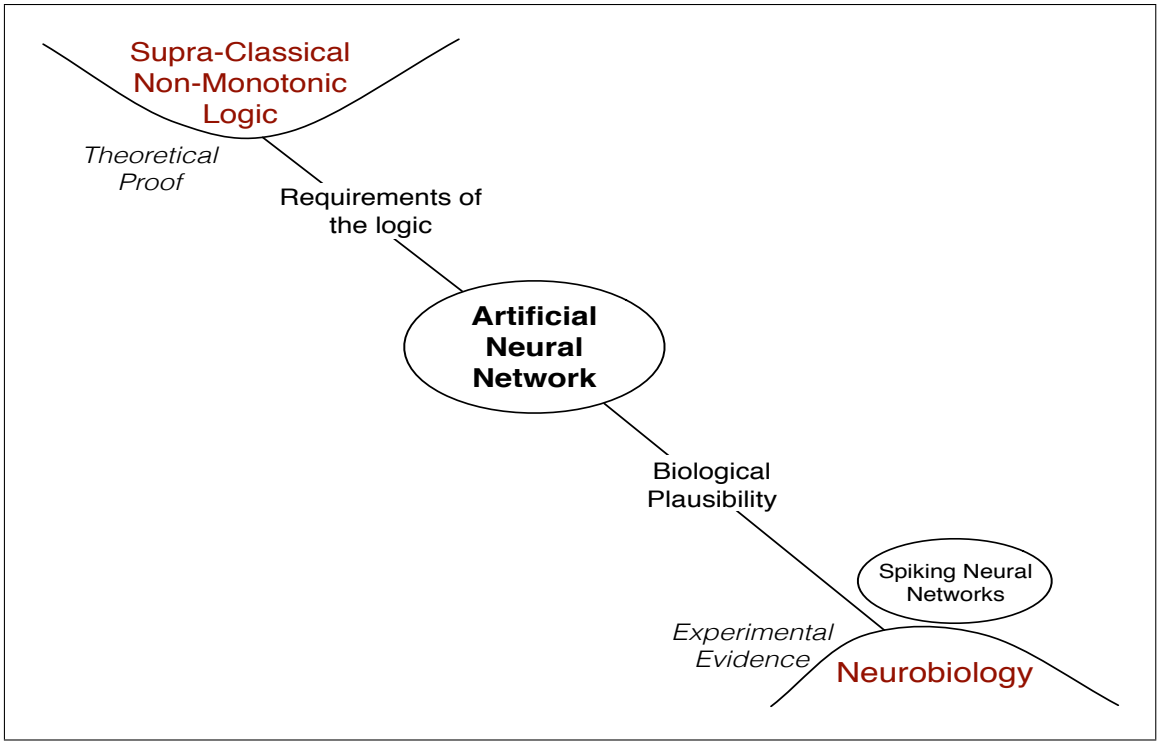


Figure 1.1: An extension of Hinton's bridge reconciling symbolic logic and neurobiology, via an artificial neural network.

We aim towards supporting half of Hinton's bridge: reconciling theoretical formalisations of inference with experimental observations from an artificial neural network (Figure 1.1). We have chosen to use an artificial neural network as a surrogate for the neurobiology. Our goal is to demonstrate that a common neural network, specifically a Boltzmann machine, can be used as a faithful representation of inference in SCNM logic, utilising the rational consequence relation. It is not intended that the thesis examine the details, context or the application of that logic to artificial intelligence: such a goal would be beyond the scope of a single text. Although the choice of this particular logic may be controversial, it is of significance because *common-sense reasoning*, as exemplified by SCNM inference, is felt by many authors (Labuschagne, Heidema, and Britz, 2013; Russell and Norvig, 2003; Lehmann and Magidor, 1992; Kraus, Lehmann, and Magidor, 1990) to be one of the hallmarks of human cognition. Moreover, the connectionist representation of the logic is not intended as an end point, but as an advantage to be utilised in examining common issues in cognition.

1.2 Contribution

The challenge of embodiment, connecting symbolic logic and neural networks, was regarded by many authors in the 1990s (Pinkas, 1995; Gärdenfors, 1994; Jagota, 1994; Hinton *et al.*, 1984) as one of the ultimate challenges of artificial intelligence. It may seem that the importance of the moment has gone by, that it is out of time; but only a partial resolution was previously found in the domain of *classical logic*.

The rational consequence relation in SCNM logic is now a conventional approach to logical inference, utilised in the landmark papers of Kraus, Lehmann and Magidor (Kraus *et al.*, 1990, KLM) and Alchourron, Gärdenfors and Makinson (Alchourron, Gärdenfors, and Makinson, 1985, AGM). It is thought to be strongly associated with common-sense reasoning. A critical analysis of the requirements of the rational consequence relation, from the neural network perspective, is an original contribution of this thesis. Whilst it may seem obvious to propose a Boltzmann machine, which learns a *generative model* of its environment, as a representation for SCNM logic: we believe, no one has previously suggested it. This thesis also confirms by experimentation that the Boltzmann machine is capable of retrieving the one-to-many mappings required as a *discriminative model*.

As yet, direct application of this Boltzmann machine representation in real world situations, for example robotics, would seem unlikely. It was previously considered impractical because of its stochastic nature and complexity (Hinton, Osindero, and Teh, 2000). However, three potential benefits from finding a neural network representation for the logical formalisation of inference are explored in this thesis:

Benefit in the domain of Human Cognition: Most importantly, bridging the gap between the two levels of explanation, symbolic and connectionist, may help in understanding human cognition, particularly common-sense reasoning. Comprehension of human cognition should benefit from the practical application of abstract logical concepts in *biologically plausible* neural networks.

One of the main aims of this thesis is to demonstrate such a connectionist representation of supra-classical inference, as a new approach to reconciling the two paradigms of artificial intelligence. We discuss its correlation to human cognition in Section 4.4.

Benefit in the domain of Neural Networks: Theoretical results in the logic may, via the discovery of a suitable network representation, be applied to or otherwise stimulate neural network research.

A new relationship between the architecture of the Boltzmann machine and the environmental information it learns is described. The discovery, of the complementary selection of different information distributions, was fortuitously made in the process of trying to improve the network's representation of the logic. This architectural selection of information is linked to concepts of *state preference* in traditional logic versus *atomic typicality* in object classification within modern description logics. We utilise the nature of the Boltzmann representation of atomic frequency distribution to design a novel definition of typicality in SCNM logic.

Benefit in the domain of Logic: Embodiment may help to clarify productive directions for practical advancement of the logic.

A slightly modified version of the Boltzmann learning algorithm is utilised to implement *pseudo-rehearsal* and compare the results of re-training in the Boltzmann representation to *belief revision* in the logic. There are currently more than 27 different logical operators for iterative belief change. The machine plausibility of two disparate approaches is considered. We believe, this is the first time belief revision results have been available from an experimental setting.

1.3 Structure

The thesis is organised as follows:

Chapter 2 Literature: Gives an outline of the research related to inference in SCNM logic, together with a discussion of its relationship to common-sense reasoning. In the connectionist domain, it gives a brief account of artificial neural networks particularly the Boltzmann machine and optimisation by simulated annealing. It then provides a detailed review of the previous research relating symbolic logic to connectionist models of cognition.

Chapter 3 Background: Describes the SCNM logic utilised in the thesis and analyses its machine requirements. The design of meta-environments (micro-worlds) for testing candidate network representations, against these requirements, are discussed. An exemplar 4-atom micro-world is presented and used throughout the thesis to provide coherence. The complementary requirements of biological plausibility are considered.

Chapter 4 Representation: Looks at the theoretical reasons for selecting the Boltzmann machine as a candidate representation. It provides details of the software implementation of the machine including; minor modifications to the architecture, learning algorithm and annealing schedules. Experimental evidence from testing in micro-world environments is presented, confirming that the Boltzmann machine is a suitable neural network representation of inference in SCNM logic; concentrating on the main requirements of the logic identified in Chapter 3. A detailed summary of results for single and ensemble machines is presented for the exemplar 4-atom micro-world illustrated in Chapter 3. These results are provided for an *HLR* machine architecture. Further, experimental results for two interesting logic properties are examined to give a flavour of the utilisation of the machine in logical inference. A discussion of the place of the Boltzmann machine in the wider context of human cognition concludes the chapter.

Chapter 5 Incongruence: Considers the two sets of incongruent distributional information present within individual training sets: presenting modifications to the architecture of the Boltzmann machine, which select for one information distribution over the other. In the manner of Chapter 4, detailed error results are presented for a restricted *ILP* architecture for single and ensemble machines, in the context of the exemplar 4-atom micro-world illustrated in Chapter 3. A probabilistic model is used to analyse the internal representation of semantics by the machine. A discussion of the relationship between this information selection and concepts of typicality in modern logic follows. We utilise a counter-example to refute the representation of typicality by preferential semantics. Further, the experimental results from the Boltzmann representation are utilised to characterise an atomic basis for typicality.

Chapter 6 Belief Revision: Provides a brief summary of the literature related to belief revision in logic. It presents a modified learning algorithm for the Boltzmann machine, which implements pseudo-rehearsal as a means of retaining past learning. The relationship between theories of iterative belief revision in the logic and adaptation to new information in a Boltzmann network are explored. The Boltzmann representation is utilised as a benchmark of machine plausibility to evaluate two very different approaches to logical belief revision. A new algorithm for logical belief revision is proposed, utilising the experimental results from the Boltzmann machine.

Chapter 7 Conclusions: Discusses unresolved issues regarding the connectionist representation of logic and indicates areas of further research. It summarises the significant results from the thesis, in confirming a neural network representation of the logic, identifying a machine equivalent of typicality and in offering a new experimental perspective on belief revision.

Appendix A Vocabulary: Abbreviations and a glossary.

Appendix B Implementation on CD: C++ code for the Boltzmann network implementation with some explanatory notes.

Chapter 2

Literature

This chapter is an attempt to broadly cover research in two domains of artificial intelligence. It presents an outline of the literature in supra-classical, non-monotonic logic and its relevance to common-sense reasoning. The chapter then outlines, the properties of artificial neural networks in particular the Boltzmann machine and optimisation by simulated annealing. The end of the chapter focuses in detail on research specifically concerning the connectionist representation of SCNM logic in artificial neural networks.

2.1 Supra-classical, Non-monotonic Logic

Non-monotonic logic, as formalised in the KLM preferential semantics (Kraus *et al.*, 1990), is now a core philosophy of applied logic. It emerged out of the realisation that classical logic was too inflexible to represent common-sense reasoning. Its complex history has been recorded in multi-volume texts (Gabbay, Hogger, and Robinson, 1994; Makinson, 2005). As indicated below, this evolution in logic has been described as "a journey from the absolute to the relative." (Heidema and Labuschagne, 2001).

Classical logic regards truth as absolute, permanent: preserved in the face of change. Non-monotonic logic attempts to capture the concept of defeasible inference. For example, knowing that Tweety is bird and that birds fly, we might reasonably conclude that Tweety flies: a conclusion we might choose to retract, on finding Tweety was a penguin. The common-sense notion is that, agents may tentatively draw conclusions given incomplete (uncertain) information and have the ability to retract them in the light of new evidence (Strasser and Antonelli, 2016; Koons, 2014). Knowledge held in the context of defeasible entailment is alternatively called conditional assertion (Strasser and Antonelli, 2016). In this context, adding premisses available for inference

can lead to loss (as well as gain) of conclusions (Makinson, 2003).

Within the prevailing framework of preferential semantics, non-monotonic logic is implicitly supra-classical. It permits us to infer more from a set of premisses than classical propositional logic would, by generating a preference ranking on model states; including less preferred conclusions. This ability to tolerate counter-examples is a prime characteristic of supra-classicality (Labuschagne *et al.*, 2013) and part of an even wider context of para-consistent logics which specifically support inconsistency (Girard and Tanaka, 2016; Priest, Tanaka, and Weber, 2016). Supra-classicality is emphasised in the thesis because a ranking on preferred states, including the most inconsistent, is the specific property of the logic that we are hoping to emulate in a connectionist representation.

2.1.1 Early Defeasible Reasoning

Huge advances in logic were made at the beginning of the 20$^{\text{th}}$ century. There is not the space to discuss the logic that was inherited from Gottlob Frege (Zalta, 2016) & Bertrand Russell (Irvine, 2015) and passed on to Kurt Gödel (Kennedy, 2016) & Alfred Tarski (Tarski, 1956; Gomez-Torrente, 2015) to be irrevocably changed. Nor is there space to mention the philosophy of Rudolf Carnap (Carnap, 1950; Creath, 2014; Uebel, 2016) and Karl Popper (Popper, 1959; Thornton, 2016; Uebel, 2016). It has been necessary to draw an historical line in the sand. Only the briefest outline of SCNM logic is provided here.

However, a mention of Charles Peirce's philosophy of reasoning cannot be avoided, although his writings are difficult to access (Hartshorne and Weiss, 1958; Peirce-Edition-Project, 2010; Burch, 2014). Peirce argued there were three types of reasoning: deduction, abduction and induction. Deduction is an unconditional inference, in the nature of classical logic. Abduction is about the explanation of observations, generalisation from examples or hypothesising. Induction can be seen as defeasible, an uncertain predictive inference based on prior beliefs. This is often termed 'categorical induction' in the domain of cognitive psychology (Harnad, 1987). SCNM logic is the formalism of 'inductive reasoning'.

In the early years before SCNM logic came of age, McCarthy and Hayes (1969) were among the first authors to define the concept of an intelligent agent, with an internal representation of the world, able to infer a strategy with which to achieve its assigned goal. One problem for such an agent is, how to efficiently determine what things in the world do not change: the 'frame problem' (Minsky, 1974). McCarthy (1959) was one

of the founders of artificial intelligence, who sought a formalisation of common-sense reasoning. However, Minsky (1974) was perhaps the first to realise that monotonicity was a specific problem, that it was too "permissive" in allowing conclusions to be drawn. He also anticipated the tension that was to come, between incorporation of the intuitions of common-sense and the adequacy of their formalisation (Bochman, 2011).

Nineteen eighty was a productive year in the history of SCNM logic with the publication of the special issue Volume 13 of the Journal of Artificial Intelligence. Three important formalisms for defeasible reasoning were proposed:

- McDermott and Doyle (1980) formalised a variety of predicate calculus, contributing to the ideas of auto-epistemic reasoning, which reinforce McCarthy's concept of the agent's representation of the world: the agent's ability to make inferences from reflection on internal beliefs.

- Reiter (1980) proposed his 'default logic'. It has been highly influential and has been incorporated into other formalisms. It deals with the problem of qualifications. Defeasible entailments intrinsically make an infinite number of assumptions: for example "cars normally go at green lights" assuming: their engine does not stall, they haven't run out of fuel, no one is crossing the road, there isn't sugar in the petrol, an earthquake is not occurring, ... etc. Reiter formalises default negation: that "in the absence of information to the contrary" infer the conclusion.

- McCarthy's (1980) idea of circumscription, while initially referencing the problems of qualification and monotonicity, foreshadows the concept of preference. Given the uncertain nature of inference imposed by the qualification problem, an agent may make a number of conclusions, the most 'circumspect' of these is correct given the current information. However, on the receipt of new information a different conclusion may become more 'circumspect'. His proposal implies an ordering on the plausibility of inference.

A number of other approaches followed in more complex (modal) logics, which included the auto-epistemic semantics supported by Sholam (1987). In 1990 the landmark paper of Kraus, Lehmann and Magidor (KLM) was published in the Journal of Artificial Intelligence.

2.1.2 KLM: Preferential Semantics

Let us look at properties related to defeasibility: the intuition we would like to encapsulate in some system of logic, before considering the paper itself.

In the definitions that follow: $\leftrightarrow$ denotes logical equivalence (alternatively in other sources, $\equiv$), $\models$ denotes classical consequence or entailment and $\wedge$ denotes logical 'and'. The symbol $|\sim$ denotes defeasible entailment or entailment by rational consequence. Well-formed formulae in the language are denoted by the Greek letters. For the purpose of this first set of properties related to defeasible entailment, it is intended that α, β represent existing information and that γ represents new information, although this is not determined within the language. In the subsequent two groups of properties (KLM and Classical), α, β and γ could be any well-formed formulae.

Properties Related to Defeasible Entailment, $|\sim$

If $\models \alpha \leftrightarrow \alpha'$ and $\beta \leftrightarrow \beta'$ and $\alpha \mid\sim \beta$, then $\alpha' \mid\sim \beta'$	(Well Behaved Equivalence)
If $\alpha \models \beta$, then $\alpha \wedge \gamma \models \beta$	(Monotonicity - Classical)
If $\alpha \mid\sim \beta$, then for some γ, $\alpha \wedge \gamma \not\mid\sim \beta$	(Non-monotonicity)
If $\alpha \models \beta$, then $\alpha \mid\sim \beta$	(Supra-classicality)

First, defeasible entailment should be a well behaved semantic equivalence, independent of syntactic change. The second property, monotonicity, is stated in the formalism of classical logic: '$\models$'. As previously discussed, this property is inappropriately strong for the context of common-sense reasoning. Surprisingly, the third property, non-monotonicity, is too weak. It results in systems which are irrational. Every time new information is received, the agent must revise all the pre-existing assertions. By default, artificial neural networks are strictly non-monotonic, they irrationally forget past learned assertions: *catastrophic forgetting*[1] (Robins, 1995). In fact common-sense reasoning, as represented by defeasibility, is somewhere between monotonic and non-monotonic. The fourth property states that any information captured by classical entailment is at least defeasible and so defeasibility is part of a broader framework of supra-classical logics.

In their paper, Kraus *et al.* (1990) present a sequence of systems; from weakest (least rational) non-monotonic to strongly monotonic, in order: **C** Cumulative, **CL** Cumulative with Loop, **P** Preferential, **CM** Cumulative Monotonic, and **M** Monotonic. For each system they examine the proof theoretic properties, the semantics and

[1]Terms emphasised at first occurrence, are defined in the glossary.

separately the resulting consequence relations. The weakest system, **C** Cumulative, contains the basic properties of all the systems.

KLM Cumulative Properties

1. $\alpha \mid\sim \alpha$	(Reflexivity)
2. If $\models \alpha \leftrightarrow \beta$ and $\alpha \mid\sim \gamma$, then $\beta \mid\sim \gamma$	(Left Equivalence)
3. If $\alpha \mid\sim \beta$ and $\beta \models \gamma$, then $\alpha \mid\sim \gamma$	(Right Weakening)
4. If $\alpha \wedge \gamma \mid\sim \beta$ and $\alpha \mid\sim \gamma$, then $\alpha \mid\sim \beta$	(Cut)
5. If $\alpha \mid\sim \beta$ and $\alpha \mid\sim \gamma$, then $\alpha \mid\sim \beta \wedge \gamma$	(And, derived)
6. If $\alpha \mid\sim \beta$ and $\alpha \mid\sim \gamma$, then $\alpha \wedge \gamma \mid\sim \beta$	(Cautious Monotonicity)

Reflexivity is a universal requirement. Logical Equivalence can be derived from Left Equivalence and expresses the concept of syntactic independence. Right Weakening states that plausible consequences should include those which are strictly classical. Cut expresses the idea that information which is separately entailed can be removed without loss of assertions. It formalises the concept of foundational information in the knowledge base, as opposed to conjunctions obligated by closure, which do not add to an agent's knowledge. Cautious monotonicity goes some way towards re-establishing the strength of classical entailment within the system.

Classical Properties

1. If $\alpha \models \beta$ and $\beta \models \gamma$, then $\alpha \models \gamma$	(Transitivity)
2. If $\alpha \models \beta$, then $\alpha \wedge \gamma \models \beta$	(Monotonicity)
3. If $\alpha \models \beta$, then $\neg\beta \models \neg\alpha$	(Contraposition)

The **CL** Cumulative Loop system adds a loop property to the base set of cumulative properties. This transitive property is important in preference ranking. However, defeasible entailment itself is not transitive. Both the **CM** & **M** systems are classical in nature, adding respectively monotonicity and finally contraposition, the strongest of the classical attributes. Both of these systems were felt by Kraus, Lehmann and Magidor to be too inflexible to be candidates for defeasible non-monotonic reasoning.

The **P** Preferential system "occupies the central position in the hierarchy of non-monotonic" reasoning. Its semantics were described by Sholam (1987) and it was considered by Adams (1965) & Pearl and Geffner (1988) in the context of conditional assertion and probabilistic logic as the "conservative core of a non-monotonic reasoning system".

Preferential: additional property

1. If $\alpha \mid\sim \gamma$ & $\beta \mid\sim \gamma$, then $\alpha \vee \beta \mid\sim \gamma$ (Or)

Supplementary property

2. If $\alpha \mid\sim \beta$, then Either $\alpha \wedge \gamma \mid\sim \beta$, Or $\alpha \mid\sim \neg\, \gamma$ (Rational Monotonicity)

As above, for these properties it is intended that α, β represent existing information and that γ represents new information. **P** adds the 'Or' property to the base set of cumulative properties and includes the 'And' property, which can be derived via 'Cautious Monotonicity' and 'Cut'.

The original paper (Kraus *et al.*, 1990), discusses three further properties: 'Negation Rationality', 'Disjunctive Rationality' and 'Rational Monotonicity' which potentially might be added to strengthen the **P** system (make it more classical). In a subsequent paper, Lehmann and Magidor (1992) concentrate on 'Rational Monotonicity', which can be envisaged as the upper boundary of defeasibility; above which the logic becomes too strongly, classically monotonic. The formulation of 'Rational Monotonicity' provided above is now traditional, although not the original given by Kraus *et al.* (1990). Alternatively, it could be concisely re-stated (combining both clauses) in the context of common-sense reasoning: when *un*-surprising new information (γ) is received, the agent need not revise previous assertions.

Rather than a theoretical discussion of the preferential semantics and consequence relations, which would involve considerable space and technical detail, we present a simple example in the hope it may be more informative. Let us consider a traffic intersection with a light and a car. In sentences of the language the fixed order of the propositions, the light and the car, will be maintained. Truth valuations on atoms will be denoted as follows: L the light is green, $\neg L$ the light is red, C the car goes through the intersection and $\neg C$ the car stops.

The most plausible, preferred model states in this example world are states with valuations "1 1" where the light is green and the car goes through and "-1 -1" where the light is red and the car stops. The model state with valuations "1 -1" where the light is green but the car stalls happens occasionally, but is less preferred and "-1 1" where the car runs a red light is least preferred. The reflexive and transitive total pre-order on the state models is denoted by: $\prec$, where $\top$ represents truth (tautology) and $\bot$ falsity (contradiction) and $\bot \prec [-1\ 1] \prec [1\ -1] \prec [-1\ -1]$ and $[1\ 1] \prec \top$ is the preference relation, as illustrated in Figure 2.1. The preference ranks are often

indexed with an ordinal. Traditionally the most preferred (plausible or normal) models have been called the minimal models. However, throughout this thesis we choose the natural alternative of calling them the maximal models.

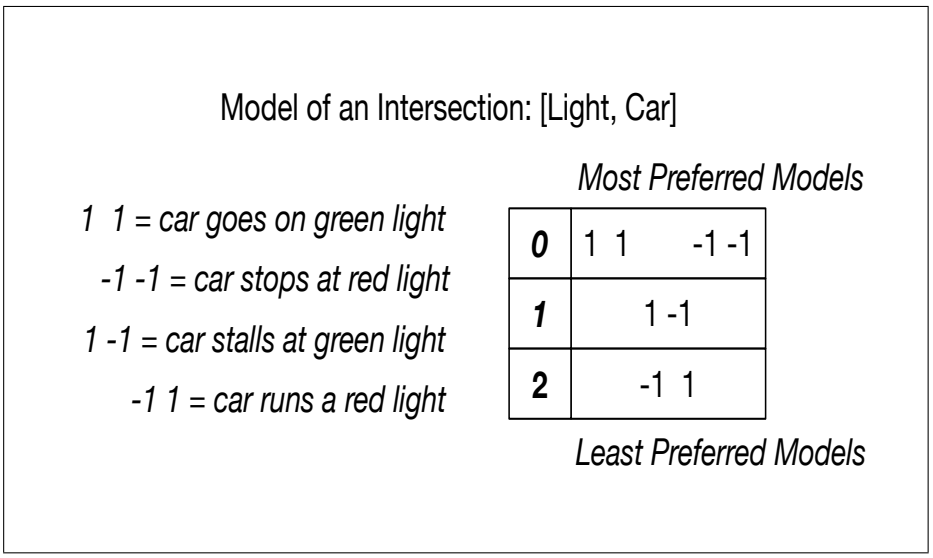


Figure 2.1: An example preference relation, shown with an ordinal index of ranks & most preferred models as maximal; at the top.

If in this example scenario the agent received information that the light was green, it would be plausible to conclude, based on rational consequence, that the car went through the intersection; although it may have stalled. This defeasible conclusion is reached by selecting the most preferred (maximal) models of the premiss from the preference ranking, as discussed above and illustrated in Figure 2.1. In general, defeasible entailment is synonymous with the maximal preference.

$$\alpha \mid\sim \beta \quad \longleftrightarrow \quad Maximal\ Models[\alpha] \ \subseteq \ Models[\beta]$$

However, remembering Peirce's classification of the different types of reasoning, other forms of supra-classical entailment are possible. Considering the set diagram, Figure 2.2, the maximally preferred models ($Models[\alpha \wedge \beta]$) are those selected by defeasible inference (rational consequence). The least preferred models are the highly abnormal counter-examples ($Models[\alpha \wedge \neg\beta]$).

If we consider them, as the least preferred models of $\neg\beta$, this establishes another entailment relation:

$$\alpha \mid\sim^{*} \beta \quad \longleftrightarrow \quad Models[\alpha] \ \subseteq \ S - Maximal\ Models[\neg\beta]$$

where $|\sim^*$ can be thought of as a "partial explanation" or hypothesis: an abductive inference in Peirce's classification of reasoning. In fact, a variety of more esoteric entailments are possible in SCNM logic (Labuschagne and Heidema, 2010). It is unclear which of them may play some part in human reasoning.

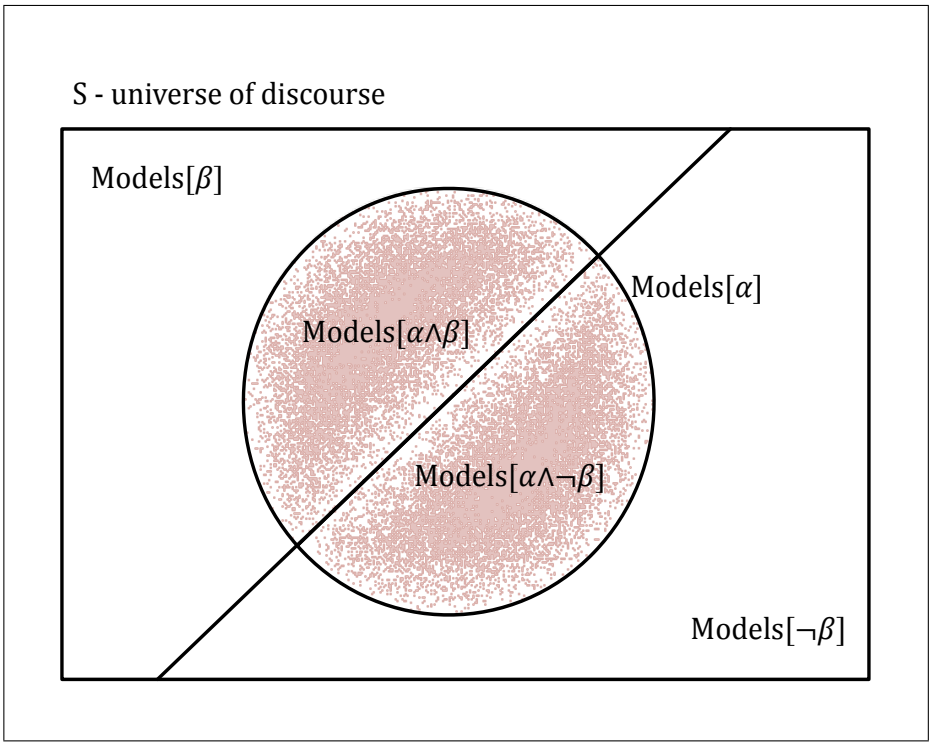


Figure 2.2: A set diagram showing supra-classical relationships (Labuschagne and Heidema, 2010).

2.2 Common-Sense Reasoning

There is no accepted, practical definition of common-sense reasoning. It often applies to the menial and yet extraordinarily complex activities that are common place, such as tying your shoe lace (Davis and Morgenstern, 2004). In contrast, expert opinion in specific domains, such as medical diagnosis, can be effectively represented autonomously by neural networks. Many authors believe that SCNM logic is a credible formalism of common-sense reasoning (Labuschagne *et al.*, 2013; Russell and Norvig, 2003; Lehmann and Magidor, 1992; Kraus *et al.*, 1990). The two domains share many common issues, for example the management of qualifications / exceptions: both fields making the compromise of default negation (Reiter, 1980). However, there are some issues specific to common-sense reasoning related to practical implementation and evidence.

The **first issue** is the division between common-sense knowledge and common-sense reasoning. The tasks to be accomplished in every-day life often require knowledge from multiple domains. Common-knowledge is beyond the scope of rational agents programmed for specific tasks. Attempts have been made to acquire such knowledge online as web based ontologies: however, two major projects have largely failed. The Cyc Project was developed by Lenat, but is now commercialised (Lenat, 2016). It relies on hard-wired heuristic rules, which are regarded as out of date. The other Open Mind Project, developed by Minsky and Singh (Singh, 2002), collapsed after Singh's death in 2008. In contrast, web based ontologies have been successful in specific isolated domains, for example human genetics: the Gene Ontology (Lewis, Blake, Cherry, Sternberg, and Thomas, 2016).

The **second issue** is the nature of the implementation of defeasibility. There is still considerable tension regarding the utilisation of numbers in probabilistic logics. Many logicians including Fodor and Rott (Rott, 2009; Rives, 2010) have an underlying mistrust of representation by probability:

> "All the methods considered are purely qualitative, in the sense that there are no meaningful numbers involved. .... In view of the abundance of qualitative methods at our disposal, we are not likely to subscribe to the view of proponents of numerical methods, according to which purely qualitative methods will always remain too poor to model ..... The problem is rather the reverse: We are facing an embarrassment of (qualitative) riches." (Rott, 2009)

Of the opposing view are Bacchus, Pearl and Darwiche (Bacchus, 1988; Pearl, 1997; Chan and Darwiche, 2002), who feel that numerical exactness is required to avoid inconsistency (see Section 5.3: Typicality):

> "People are notoriously bad number estimators ... (but numbers) no matter how erroneous, prevent us from reaching inconsistent conclusions ... they are the summaries of knowledge left behind when information is abstracted to a higher level." "Probability is not really about numbers; it is about the structure of reasoning" (Pearl, 1997)

They would argue for representation of common-sense reasoning in a probabilistic logic, encapsulating all the qualitative properties of defeasibility in SCNM logic. Support for this view is provided by Chan and Darwiche (2002) demonstrating, using

sensitivity analysis, that small variations in parameter settings for Markov networks (the underlying models for stochastic neural networks) can have a marked effect on predictive outcomes.

The Dempster-Shafer theory in probability (Zadeh, 1986) assigns three separate metrics to: mass, belief and plausibility. This method considers the probability of object combinations from the appropriate power set. Belief and plausibility are denoted as lower and upper bounds, one accumulated by set union the other accumulated by intersection. It seems implausible that common-sense reasoning would invoke such a convoluted scheme for decision making. The research of Pinkas in symmetrical neural networks (discussed in Section 2.4.1) and Spohn in belief revision (discussed in Section 6.1.2) offer an alternative, a half-way house, utilising single natural number indices that might plausibly be part of common-sense reasoning.

The **third issue** is related to what constitutes valid evidence. Logicians and psychologists have tended to shun each other's domains. Further, logicians have been criticised as relying too much on their own intuitions. Pelletier and Elio (1997) argue strongly for empirical evidence as a grounding for all formal logic. They view conjectures about rational agents (Russell and Norvig, 2003) uncharitably, as a means of avoiding the real world of biological human cognition, where experimental evidence is required for scientific enquiry. We strongly agree with the view of Pelletier and Elio, that human reasoning is the only model available for common-sense.

2.2.1 Empirical Evidence from Psychology

For the reasons above, the results of a number of relevant psychological experiments are now reported in some detail. Surprisingly, given the separation between the disciplines, the empirical psychological evidence is very supportive of SCNM logic as a surrogate for human reasoning. One of the earliest and most influential results was reported by Wason (1966, 1969). A short description of the logical basis of the test is necessary, because it is utilised for many of the psychological experiments that follow.

In its original form, Wason's selection test involved human subjects given four cards, their facing and concealed sides potentially supporting some relationship; an inference made in the rules of classical logic (see Figure 2.3). The subjects were able to select independently all the inferences that they felt were correct. There are four potential options to choose from in each situation. The results, in terms of validation for a particular logical property, are able to be read directly from the subject selections. In subsequent research the test was modified, to be used in a verbal setting, where the

options were mutually exclusive and individual logical inferences could not be directly distinguished.

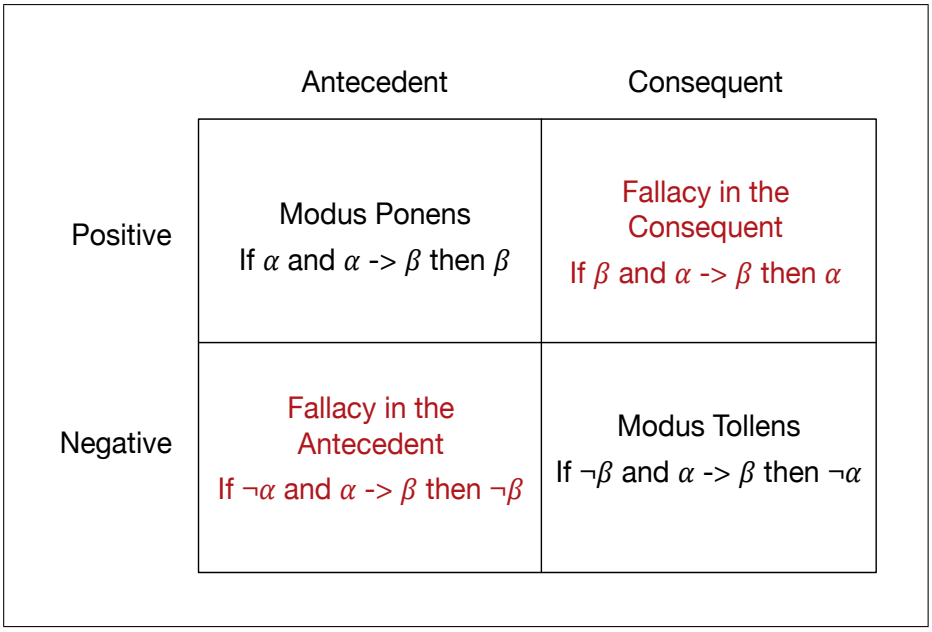


Figure 2.3: A 2 x 2 contingency table for inference in classical logic.

Ninety percent of Wason's original subjects consistently chose the positive inference correctly, in the style of modus ponens: if α and $\alpha \rightarrow \beta$, then β. A further 35% of subjects also mistakenly affirmed the consequent, a positive fallacy: an indication of the degree of possible error. But, only 4% of subjects also chose the negative contradiction correctly. Thus rebutting one of the classical styles of reasoning, modus tollens: if $\neg\beta$ and $\alpha \rightarrow \beta$, then $\neg\alpha$. There has been prolonged discussion about the result. It has been suggested, given that the counter inference is more difficult, that the results simply demonstrate human error. When repeated in a different context, Cheng and Holyoak (1985) showed that placing the test in a familiar setting or prompting the subjects with a concrete rationalisation improved the selection of the negative inference to almost 90%. However, the result can be interpreted in another way: the negative inference is more difficult to make because modus tollens is implicitly based on the classical property of contraposition. Contraposition was specifically removed from SCNM logic because it was felt to be too strongly classical. Wason's test subjects may have validated a significant formalism of defeasible reasoning.

Byrne's suppression test is an extension of Wason's research (Byrne, 1989). Alternative antecedents and consequents were supplied to subjects in a verbal selection test. Prior to adding antecedents, 96% of subjects correctly selected inference

based on modus ponens. After the addition of new antecedents, only 38% of subjects persisted with modus ponens. The study clearly demonstrates the importance of non-monotonicity in common-sense reasoning. Further, the specific nature of the antecedent information demonstrated that the content and context of a language convey implicit information, which is also used for inference. Currently application of such common-knowledge is beyond the scope of formal logic.

1. If $\alpha \models \beta$, then $\alpha \wedge \gamma \models \beta$ (Monotonicity - removed)
2. If $\alpha \models \beta$, then $\neg\beta \models \neg\alpha$ (Contraposition - removed)

Neves, Bonnefon, and Raufaste (2002) have examined the human corroboration of KLM System P (with rational monotonicity) in 88 subjects. The results are difficult to interpret clearly. In the setting of this verbal selection test a contrived statistical scheme was utilised, measuring the proportions of correct inferences compared to the two fallacies, to decide the degree of corroboration of a particular property. If we accept the analysis of the authors, then with respect to the eight properties examined:

- Reflexivity and Logical Equivalence could not be tested. However, the previous research had emphasised the importance of context, which may argue against syntactic independence.
- Cut was not corroborated.
- Right Weakening, And, Or and Cautious Monotonicity were corroborated.
- Rational Monotonicity was partially corroborated.

Further, when considering the overall support for the KLM System **P** (with Rational Monotonicity): 47% of subjects made no violation at all of any property and 84% of subjects only violated one property, where the violation was randomly distributed among properties. Considering that it is doubtful that human reasoning could ever be perfectly efficient: this is a very encouraging result. Their second paper (Benferhat, Bonnefon, and Neves, 2005) is more concerned with results in possibility theory, outside of the realm of traditional SCNM logic.

Pfeifer and Kleiter (2005) have also examined the corroboration of KLM System **P** properties by human subjects. The methodology of the research was strongly probabilistic, requiring estimates of probability intervals by the subjects and reporting lower and upper bound violations. The authors admit that human subjects are notoriously bad at numeric estimations, especially at combining probability estimates.

However, their conclusions supported the foundational properties of KLM, including logical equivalence (syntactic independence).

Ford and Billington (2000) are an exception to this corroborating trend. They report experiments looking at complex patterns of inference related to potential conflicting inheritance structures (box, triangular and diamond), rather than the core preferential semantics of KLM. The objects of these inheritance schemes are purely fictional: "Hittas, Wiflons, Kiglers, Waffs, Jukks, ...". Unsurprisingly, no consistent logical pattern of reasoning was found in any of the 49 test subjects. Moreover, the authors appear to be antagonistic to the ideas of SCNM logic, even in the introduction to their paper: offering multiple quotations regarding the complexity of non-monotonic logic. Schurz (2005) produced research in a very similar area, specificity in inheritance, with 150 test subjects. However, he used common place objects and backgrounds: birds, schools, ethnicity, etc. He states that his results are "just the opposite" of Ford and Billington in supporting non-monotonic inference.

Stenning and Van Lambalgen (2008, Chapter 8: From Logic via Exploration to Controlled Experiment) are generally supportive of a framework of defeasible inference, without specific reference to KLM System **P**. They provide experimental results from 365 subjects, repeating Wason's selection task in a context of improved explanation, making the point that context and familiarity are just as important to human cognition as logical inference. Unfortunately, their motivation seems to be their assumption of Wason's failure to corroborate non-monotonic logic, when in fact Wason's results challenge classical monotonic inference.

2.3 Artificial Neural Networks

We now change to the connectionist domain of artificial neural networks; it is less than a century old. Artificial neural networks are loosely based on the structure and functionality of biological brains. They are a connectionist approach to artificial intelligence characterised by parallel, distributed processing (Rumelhart, Smolensky, McCelland, and Hinton, 1986a,b), involving multiple simple interconnected processing units (nodes). Their ability to learn and represent knowledge lies in the adaptation of their connection weights. Procedural computing originally developed in the context of symbolic language processing in finite automata. However, most practical language processing is now performed in neural networks.

In general the nodes only have two functions:

$$\Delta Enet_i = \sum_j w_{ij} s_j - \theta_i$$

- Input Summation: Here the change in energy at node i is related to the sum of the input from connected nodes j, where w_{ij} is the related weight and s_j the nodal state at j. θ is the bias or threshold of node i.

$$P_{i|T} = \frac{1}{1 + e^{\left(\frac{-\Delta Enet_i}{T}\right)}}$$

- Activation: We have chosen to illustrate a sigmoid activation function as this is of relevance to the Boltzmann machine. Here unusually, the activation is stochastic, the probability of activation of node i given threshold T ($P_{i|T}$), rather than the actual output, is specified by the activation function. The firing is 'all or nothing' (0 or 1). $\Delta Enet_i$ is the input summation for node i as above. Rather than the traditional analogy of $k_B T$, the product of Boltzmann constant and temperature, T should be regarded as the optimisation threshold (see below Section 2.3.2).

There are a huge variety of neural networks, which can be categorised in multiple ways:

- Learning Type: supervised, unsupervised, reinforcement.
- Learning Algorithm: back-propagation, contrastive divergence, Hebbian, ...
- Connections: feed-forward, recurrent
- Nodal Activation: deterministic, stochastic
- Architectures: simple or multi-layer perceptrons, convolutional networks, deep belief networks, stacked networks, self organising maps, Hopfield networks, Boltzmann machines, restricted Boltzmann machines, ...

The most classical variety of feed-forward neural network implements back-propagation as part of supervised learning in a deterministic model; the multi-layer perceptron.

Historically, the first characterisation of a neural model based on threshold units comes from McCulloch and Pitts (1943). The concept of learning and knowledge representation based on the strength of nodal interconnection was first proposed by Hebb

(1949). Early research stalled, following the demonstration by Minsky and Papert (1969) that the simple, deterministic, feed-forward perceptron of Rosenblatt (1958) was unable to represent non-linear classification problems, such as exclusive-OR. Unfortunately, the solution to this problem of credit assignment in multi-layer networks, the back-propagation algorithm first published in the master's thesis of Linnainmaa (1970), lay unrecognised. The algorithm is usually attributed to its later re-invention by Werbos (1975).

2.3.1 The Boltzmann Machine

Whilst this issue delayed the development of multi-layer feed-forward networks for almost a decade, the development of alternative unsupervised recurrent networks continued in the background. It reached its fulfilment in the design of the *symmetric (recurrent) neural networks* (SNNs) derived from the domain of statistical mechanics, the Hopfield network (Hopfield, 1982) and the related Boltzmann machine (Hinton, Sejnowski, and Ackley, 1985; Hinton *et al.*, 1984). By all the criteria mentioned above (Section 2.3) the Boltzmann machine is unusual. It is a variety of SNN without reflexive connections but including stochastic activation functions and requiring simulated annealing for sampling of local cross-firing.

The Boltzmann machine utilises a variety of unsupervised learning and has its own unique learning algorithm (Haykin, 2008). The algorithm has two phases (see below, Figure 2.4 & Figure 2.5 pseudo-code): a clamped phase where external input is applied to the visible nodes and a resting phase where there is no input (the network is run free). The basis of its learning (weight adjustment) is the comparison of cross-firing statistics in the clamped and resting phases. Cross-firing ρ_{ij} is determined by the product of the nodal states, $\tilde{s_i}$ and $\tilde{s_j}$, averaged over a large number of samples.

$$\rho_{ij} = \tilde{s_i} \times \tilde{s_j}$$

$$\Delta w_{ij} = \eta \, (\rho_{ij}^{+} - \rho_{ij}^{-})$$

The change in weight w_{ij} between two nodes i and j is related to the difference in cross-firing between the clamped phase ρ_{ij}^{+} and in the resting phase ρ_{ij}^{-}, multiplied by some learning rate η. In effect, this comparison is a variety of error correction (supervision) utilised to model the network states in the clamped phase. This localised Hebbian learning is biologically plausible. However, the algorithm requires simulated annealing to retrieve the cross-firing statistics (see discussion below Section 2.3.2), particularly in the resting phase.

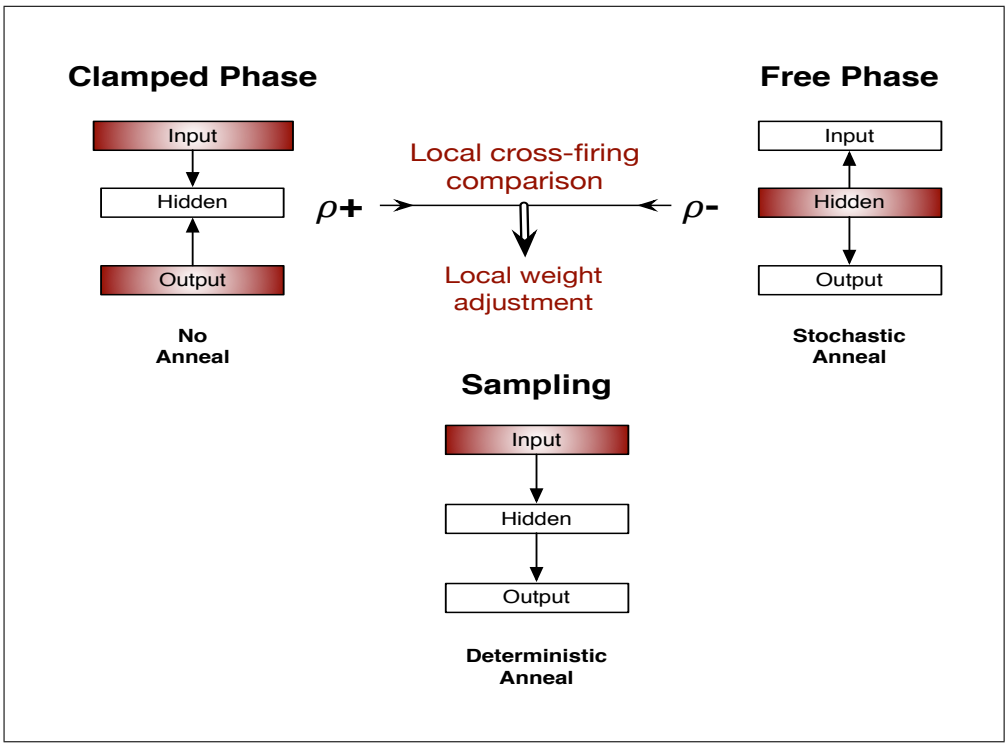


Figure 2.4: The Hebbian nature of the Boltzmann learning algorithm: clamped and free phases. Some minor adjustments were made to the learning algorithm and annealing schedules specific to this thesis (Section 4.2 Implementation). Sampling indicates testing the output of the machine against a clamped input.

There are two major characteristics that equip the Boltzmann machine to be "a device for modelling an underlying probability distribution (... of its training set)" (Haykin, 2008).

- The Boltzmann machine is a symmetrically recurrent network, a characteristic which is both the basis of constraint satisfaction and the ability to characterise the total energy within the network.

$$E_{k|T} \equiv \sum_{i} \sum_{j} w_{ij} s_i s_j \quad where\ i \neq j$$

  The total energy of the network in state k (at equilibrium, for threshold T) is directly related to the sum of all products of the paired nodal states s_i and s_j across each weight w_{ij}.

- The Boltzmann machine is one of very few networks that implement stochastic nodal activation.

```
For Epochs      //until plateau
        {
                Shuffle patterns;

                For each pattern
                        {
                                //active - clamped phase
                                Clamp inputs;
                                Run network;
                                Sample local ρ+ values;

                                //resting - free phase
                                Randomise node activations;
                                Anneal network by layer;
                                Sample local ρ- values;
                        }

                Update weights;
                        ∂Weight(new) = η (ρ+ - ρ-) + μ*∂Weight(previous)
        }
```

Figure 2.5: Pseudo-code for a modified Boltzmann learning algorithm (see Section 4.2 Implementation). Where η is the learning rate, μ is the momentum; $\rho+$ and $\rho-$ are the cross-firing statistics across a specific weight in the clamped and free phases respectively.

In conjunction, these characteristics enable the network to represent high probability states with low energy. These resonant states as discussed by Balkenius and Gärdenfors (1991) or energy minima as discussed by Pinkas (1994) are the basis of the probabilistic representation of logic (see following Section 2.4).

$$P_{k|T} \equiv \frac{e^{\left(\frac{-Enet_k}{T}\right)}}{\sum_l e^{\left(\frac{-Enet_l}{T}\right)}}$$

These properties are encapsulated in the Boltzmann distribution equation, which relates the probability of output states at equilibrium, to the relative entropy (learned preference) of these states. On the left side of the equivalence: $P_{k|T}$ is the equilibrium probability at 'threshold' T of a particular state k and on the right side: $Enet_k$ is the energy of the network in that specific state k, compared to the energy in all states. This sum of energy in all the states is referred to as the partition function.

2.3.2 Simulated Annealing

Simulated annealing might be regarded as the hallmark of Boltzmann learning: both a blessing and a curse. It is an algorithm theoretically able to achieve optimal solutions, given a *sufficiently long* time. The name is derived from the physical process of annealing solids at equilibrium in a thermal bath, by very slowly lowering their temperature. It is the opposite of quenching, which rapidly lowers the temperature of metals.

The method was first published by Metropolis and Rosenbluth (1953) in the context of the evolution of solid states in a thermal bath using Monte Carlo techniques. It was first introduced in optimisation by Kirkpatrick, Gelatt, and Vecchi (1983) but the later reference to stochastic relaxation, Geman and Geman (1984) is more often cited. The best discussion of the optimisation method is in Aarts and Korst (1990, 1997). The algorithm is a generalisation of local search:

$$P_{j|T} \quad = \quad \begin{cases} 1 & \text{if } E_j < E_i, \\ \exp((E_i - E_j)/k_B T & \text{if } E_j \geq E_i. \end{cases}$$

As for other methods of local search, if the new state j has less energy (is more optimal) than the old state i, the new state is always accepted. However initially, if the new state j is locally less optimal than the old, it may still be accepted with a probability ($P_{j|T}$) related to the energy difference $E_i - E_j$, at equilibrium threshold T (where k_B is the Boltzmann constant).

Aarts and Korst regard this algorithm as the parent of all threshold optimisations. Other techniques of descent in local search are a subset of this algorithm, where the threshold (temperature) is set close to zero and the method becomes deterministic. The great benefit of simulated annealing is its up-hill search at initial high thresholds; the method has the ability to overcome local minima. However, the schedule for lowering the threshold (temperature) is critical and has to be performed slowly. This theoretical formulae for the threshold schedule comes form the paper of Geman and Geman (1984).

$$T_k \quad = \quad \frac{c}{log(k+1)}$$

Here the threshold (temperature) for the k^{th} iteration of the schedule is derived from the inverse log of k. The schedule used in the original papers by Hinton *et al.* (1984, 1985) was a stepped approximation. We developed our own 20 point schedule based on the formula above, see Figure 2.6. Our modifications to the generic network, learning algorithm and annealing schedules of a Boltzmann machine are covered in Section 4.2 Implementation.

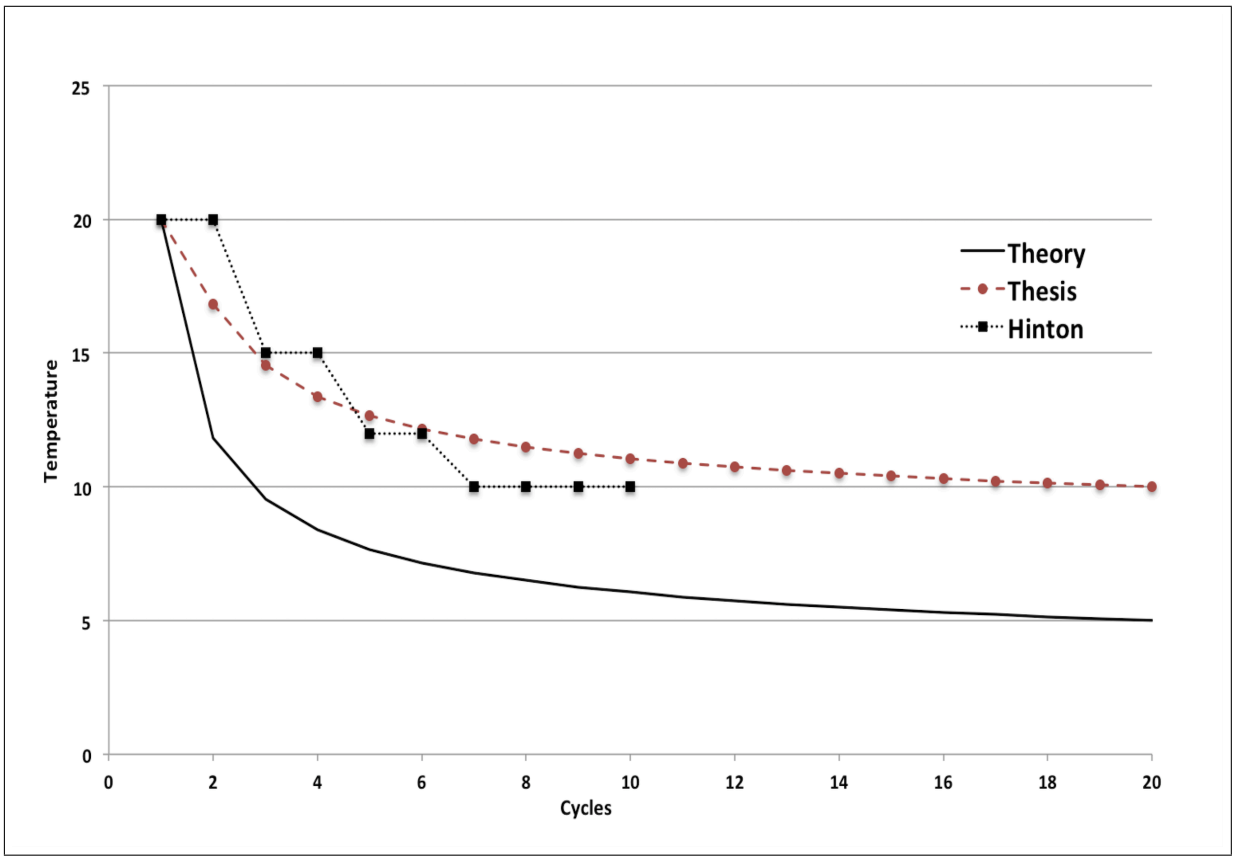


Figure 2.6: Annealing schedules: our approximation and the schedule from Hinton *et al.* (1985) are shown in comparison to the theoretical ideal of inverse log. The theoretical schedule is shown for a wider temperature range ($20 \rightarrow 5$), to avoid overlap with the actual schedules ($20 \rightarrow 10$) and clearly separate the curves. Multiple sampling occurs at each temperature point.

2.4 Representation in an Artificial Neural Network

A number of authors have implicitly approached the topic of systematicity in neural networks related to natural language (Hendler, 1989; Wermter, 1997; Boden and Niklasson, 2000; Hadley and Cardei, 1999; Hadley, Rotaru-Varga, Arnold, and Cardei, 2001; Frank *et al.*, 2009). However, we limit this discussion to literature, which has explicitly explored the implementation of logical inference. McCulloch and Pitts (1943) began this theme by hypothesising a relationship between propositional logic and activity in neural networks.

2.4.1 Early Symmetric Networks

The seminal papers on symmetric neural networks were published in the early 1980s: Hopfield nets (Hopfield, 1982) and Boltzmann machines (Hinton *et al.*, 1984, 1985). Balkenius and Gärdenfors (1991) were the first to recognise the unique characteristics of the newly formulated networks, in relation to the representation of propositional logic. They specifically emphasise the property of constraint satisfaction with regard to the ability of these networks to find single solutions in classical logic. They use the term 'resonance states' to refer to *energy minima* in the network, corresponding to solutions in the logic. They demonstrated theoretically, that simple SNNs could replicate the conclusions of logical schemata. Their detailed description of logical schemata link them to concepts used by Rumelhart *et al.* (1986a, Vol 1, Chp 14: Schemata) and Minsky and Papert (1971); Minsky (1974). Unfortunately there is a confusing collection of related concepts in the literature: Gärdenfors logical schemata, an epistemic state in modern logic, a generative model or frame in computer science and a joint probability distribution in statistics. In this paper we have chosen: 'logical micro-world', a terminology used by Frank *et al.* (2009), see Section 3.2.

Jagota (1994) examined the stable storage of database tuple information in Hopfield style networks serving as associative memory. He related this to Boolean formulae and regular expressions, although not directly to inference. He theorized about the storage capacity of these networks, which are now known to be limited arithmetically by the number of nodes. Many of the authors in this domain reference the work of Rumelhart *et al.* (1986a) and Rumelhart *et al.* (1986b), looking at ideas of micro-structure in parallel distributed processing.

Pinkas (1994, 1995) strengthened the work of Balkenius and Gärdenfors by providing a mathematical foundation, demonstrating an equivalence between two fundamental ideas: a solution in logic and energy minima in a SNN. Pinkas derives the energy formulae (and ultimately the architecture) of SNNs from the logic, with the addition of weights expressing strength of belief or reliability of knowledge: in the negative sense a penalty. His penalty-logic constitutes a means of translation between a specific syntactically correct logical sentence and a unique energy formula for some SNN. He utilises the idea of local and global minima, consistent with Balkenius and Gärdenfors 'resonant states'.

In the first theoretical part of his paper Pinkas indicates that SNNs are capable of learning an ordering on states (a preference ranking) consistent with the concepts of non-monotonic logic (Kraus *et al.*, 1990; Lehmann and Magidor, 1992). Although the

paper deals with many SNNs, Pinkas specifically acknowledges that only the Boltzmann machine is capable of searching multiple energy minima simultaneously. That, as a consequence of its probabilistic learning, only the Boltzmann machine is capable of learning a 'strongly equivalent' or 'magnitude preserving' ranking on model states.

Within the broad theoretical discourse of Pinkas's paper, his experimental results have a very specific focus. Pinkas randomly generates large '3-SAT' problems: sentences with as many as 100 variables in *conjunctive normal form*, with 3 variable clauses. He constructs SNNs based on these sentences using his penalty-logic and runs the derived networks to find a solution. The SAT-solving abilities of his networks are an impressive demonstration of the translational function of his penalty-logic. Further work by Pinkas (1995) emphasises his focus on the global minimum of the networks, at the expense of local minima. However, there are two major issues with these experiments in the context of SCNM logic and common-sense reasoning:

1. These are single problems. Pinkas has engineered each network from a single sentence: each solution found is appropriate only to that specific sentence. To be a useful representation of supra-classical logic, a network should be capable of learning solutions to all the sentences entailed by its logical environment: solutions for all syntactically correct sentences within a micro-world. A specific network constructed from a single sentence 'α' could not be expected to retrieve solutions for a sentence 'β' from which it was not designed. Whereas a generic network, which trains from its environment, should be capable of such learning and adaptation.

2. These are single solutions. They are in the domain of classical logic. There is no evidence, in Pinkas's experimental results, of a preference ranking on states. This would require not just the set of global minima, but also an ordering on all the sets of local minima. Pinkas mistakenly asserts that, "symbolic logic is too rigid to be able to deal with exceptions". Whereas in fact, these multiple and less optimal solutions are the counter-examples, which are a prime characteristic of SCNM logic (see Section 3.1: Logical Preliminaries).

2.4.2 From the Logic

The development of probabilistic logic dates back at least to the introduction of the SNNs. One of the first papers by Nilsson (1986) presents the fundamental mapping of truth in classical logic to probabilities [0 or 1], using binary semantic trees for

sentence analysis. The intuitive extension of this approach to preference relations in non-monotonic logic simply utilises the range of probability values for different preference levels. The work of Bacchus (1988, 1990, 1991, 1996), although not in the main stream of non-monotonic logic, develops probabilistic logics from a consideration of statistical knowledge bases. Bacchus has made the correlation between these logics and Bayesian networks typically used for probabilistic reasoning. However, Bayesian networks are, strictly speaking, propositional acyclic graphs. They are not neural networks in the sense of having some connection with biology and being able to adapt. Myllymaki (1999) has shown that these acyclic graphs can be mapped onto the weights of a Boltzmann machine to avoid the slow, unreliable process of network training. We take a step further, in the context of moving along Hinton's bridge, by relating the symbolic nature of the logic itself directly to the probabilistic learning of a generic Boltzmann machine.

In the first parts of his book, Leitgeb (2004) sets out the properties of logic and inference on the 'low level', considering why embodiment is possible. In discussion he theorises in detail about the structure and behaviour of agents that would have properties compatible with the requirements of the logic. He concludes that 'dynamical agents', such as 'simple inhibition nets', are viable candidates for the representation of logic. He proceeds to theoretically prove the properties of these inhibitory networks. Leitgeb has many valuable ideas (Leitgeb, 2001, 2007), particularly his emphasis on discrete binary states and inhibitory constraints (considered in Section 3.3.2: Neurobiological). However, Leitgeb's 'simple inhibition nets' lack a true distributed representation, there are no connection weights and no discussion of how the networks would learn or adapt. It is unclear whether the networks discussed were ever built, or used experimentally, and there is no experimental evidence provided to support the author's theoretical conclusions.

2.4.3 Neural-Symbolic Integration

The SHRUTI system, as proposed in the field of *neural-symbolic integration* (NSI), has been offered as a model of human cognition and by implication a representation of symbolic logic (Shastri and Ajanagadde, 1993; Shastri and Wendelken, 2000; Shastri, 2007). It primarily attempts to solve the issue of dynamic variable binding in predicate calculus. This system, although said to be motivated by neurobiology, is highly engineered around each specific problem. It postulates positive and negative 'collector nodes', 'enabler nodes' and pairs of 'utility nodes' for each variable or component

of a problem. Further, the network connections are constructed to implement temporal synchronicity to enable learning of object relationships. The authors provide a co-relation in neurobiology for this hypothesis. The system was modified in the later papers to include a degree of probabilistic and Hebbian learning.

However, the system is based on feed-forward networks, which reach a deterministic conclusion with an explicit probability. Without alteration of its nodal activations to some stochastic function, it is difficult to see how the system could retrieve multiple ranked counter-examples to a preferred conclusion. The addition of Hebbian learning moves the SHRUTI system closer to a SNN, but there is limited discussion of how the network obtains and stores the necessary cross-firing statistics to manage such learning (cf. Boltzmann machine).

Authors: d'Avila Garcez, Lamb, and Gabbay (2007, 2009) are prominent in the field of neural-symbolic integration. They have extended the work of Pinkas to non-symmetric neural systems, utilising large ensembles of feed-forward networks: the CLIP system. CLIP can be seen as a hybrid system where the network is first constructed around a specific logical problem using a translation algorithm, analogous to Pinkas for SNNs. d'Avila Garcez *et al.* support the theory that single hidden layer feed-forward networks are universal approximators of any polynomial function. They demonstrate the practical capacity of these networks in a variety of settings including first order logics, temporal and modal logic. Further, they consider the challenge of relational associations and dynamic variable binding using predicates, in the context of a specific problem: Michalski's east-west trains (d'Avila Garcez *et al.*, 2009, Chp 10).

Stenning and Van Lambalgen have published a book on 'Human Reasoning and Cognitive Science' (2008). They consider the implementation of logic in neural networks (Chp 8, p 217-239), where network inputs and outputs represent the atomic nature of the example systems. Their conclusions are relevant to this thesis:

- For implementation a three valued logic is required to represent truth, falsity and uncertainty independently.
- Recurrent neural networks have an advantage in their potential inhibitory interconnections.

Based on the view, that the representation of exceptions makes non-monotonic reasoning difficult to implement, these authors chose to artificially design the wiring of their networks directly from the conditional assertions of the logic and examine only classical inference.

> "Non-monotonicity is highly difficult to manage by artificial intelligence systems because of the necessity of looking for possible exceptions". (Stenning and Van Lambalgen, 2008)

This thesis examines SCNM logic specifically, because knowledge of exceptions is thought to be characteristic of common-sense reasoning. We contend that Shastri's SHRUTI system, Garcez's CLIP system and Stenning's implementation suffer from the same two problems, in regard to the scope of SCNM logic, which also applied to Pinkas's work. Although outlined previously, we now consider these two issues from a slightly different standpoint:

1. Single Problems: These integrated neural-symbolic systems are initially algorithmically engineered from the logic; this makes them very efficient in solving the specific problems for which they were designed. However, this is not sufficient in the broader context of understanding human cognition, where a generic system capable of dealing with multiple premises related within a micro-world must evolve by training alone.

2. Single Solutions: These networks act as function approximators, they are retrieving many-to-one or one-to-one mappings, finding only single optima in the domain of classical logic (albeit in some extended form, e.g. temporal). They are not learning one-to-many relationships, which are the domain of supra-classical logic and common-sense reasoning. This can be seen most clearly in Garcez's example of Michalski's trains. This task requires a highly complex, constructed network to map a set of variables, potentially involving intermediate results as predicates, onto a single outcome which designates a train as either east or west bound. This task is a many-to-one classification: the network is not retrieving one-to-many solutions.

2.4.4 Statistical Relational Learning

Many authors (Khosravi and Bina, 2010; Getoor and Taskar, 2007; Kersting, De Raedt, and Raiko, 2006; Koller and Pfeffer, 1998; Pearl, 1997) have published in the field of *statistical relational learning* (SRL). This domain can be seen as an extension of Bacchus's work on probabilistic logic. Statistical relationships within the data, represented in the joint probability distribution, are viewed from the perspective of database theory (as entity-relationships) and are modelled graphically using Bayesian and Markov

networks. The concept of using both directed and undirected graphs to represent probabilistic data is not new. Markov networks are a more expressive super-set of the Bayesian models (Pearl, 1997). SRL specifically addresses the difficulty of representing one-to-many associations within the data, which are not mathematical functions. These associations between facts in the data form the default rule, which is the basis of the rational consequence relation in SCNM logic. These one-to-many factual relationships are very important for common-sense reasoning and are difficult to capture in any variety of classical logic: as distinct from object relationships. Object relationships form the basis of predicate calculus, in transparent propositional logics or first-order logics with the additional complexity of variables and quantification.

In the context of SRL, these Bayesian and Markov networks are built by a process of 'inductive logic programming', using either algorithmic or manual construction. Weight calculation or parameter learning is then performed directly by calculation from the log likelihood of the data. This calculation is known to be an NP-hard problem, so only an approximation of the data distribution is possible. Finally inference in these networks is achieved by implementing Gibbs sampling where the network nodes are set to the observed inputs or randomised to the un-observed inputs. This process is analogous to 'clamping' in the initial phase of a Boltzmann learning algorithm, where Gibbs sampling is required to retrieve the output of the network at equilibrium.

We believe, that because there are important differences between graphical models and neural networks, a neural network representation is a further step in the reconciliation of SCNM logic and neurobiology. As clarification, we re-visit our earlier concerns, hopefully without repetition:

1. Single Problems: Graphs can be algorithmically altered to represent different logical schemata; there is some biological plausibility, in terms of formation of pre-synaptic connections, related to the concept of growing and pruning the graphical structure. However, weight calculation, even as an optimisation, is not the same as learning in an on going process of adaptation.

2. Single Solutions: The nodes of a probabilistic graph are not neural. They have no activation function; specifically no stochastic activation function. The undirected graph is capable of implicitly representing a joint probability table. However, when clamped with an input (a partially unspecified premiss), it cannot explicitly time-slice through the many outputs, which constitute the relationships in a multi-modal distribution, entailed by this premiss.

2.4.5 Recent and Future Additions

More recent research in computer vision has examined the multi-layer perceptron for generative modelling, utilising modified nodal activation functions, which are stochastic (Tang and Salakhutdinov, 2013; Rosales and Sclaroff, 2006). This fundamental probabilistic alteration in the network's properties moves it towards the behaviour and capacity of the Boltzmann machine. These authors have focused on the learning of non-mathematical functions (one-to-many mappings) required to produce a multi-modal distribution in the output space. They discuss the use of probabilistic knowledge to aid in the process of inference, where observation of the environment is limited or uncertain.

There is a large body of literature on the important topic of dynamic variable binding in predicate calculus (Franca, Zaverucha, and d'Avila Garcez, 2014; Pinkas, Lima, and Cohen, 2012; Hummel, 2011; Bowers, 2009, 2011; Reimers, 2011; van der Velde and de Kamps, 2006), which is directly related to human cognition. We hope to consider this issue in the future. We currently however, limit ourselves to a propositional syntax without predicates because we are attempting to examine a much broader domain of supra-classical logic for the first time. We contend that previous attempts in the connectionist representation of logic have largely been in the domain of extended classical logic. They have looked at single optimal solutions to specific, individual benchmark problems. They have not considered the much boarder issue of the generic learning and the tolerance of counter-examples. Incorporating the requirements of this broader context considerably narrows the field of candidate neural network representations.

Further, we have not approached the coal face of neurobiology: dynamic, spiking neural networks (Adeli and Ghosh-Dastidar, 2009; Mass, 1997). The task of representing logic in such temporally based networks is almost as daunting as directly relating it to the neurobiology itself. In the not too distant future, a virtual human brain will be modelled using detailed structure at a molecular level (Markram, 2014). However, the vast complexity of such an undertaking may not easily advance the understanding of human cognition. We have taken the view that small steps, utilising abstraction, are also valuable.

Chapter 3

Background

This chapter begins with an informal definition of the supra-classical, non-monotonic logic and its connection to inference, as utilised in this thesis. An experimental environment for testing candidate neural-network representations against this logic is described, with a brief description of the mapping from micro-world states to patterns of activation in a neural network. Requirements sufficient for the representation of the logic are identified and an outline of biological plausibility is given in the broader context of human cognition.

3.1 Logical Preliminaries

We give a brief definition of the SCNM logic utilised in this thesis, as first introduced in Section 2.1: Literature Review. A propositional SCNM logic is generated by a finite set of atomic propositions with conventional propositional and set connectives $(\neg, \vee, \wedge, \rightarrow, \leftrightarrow; \cap, \cup, \subseteq, \supseteq)$. Let $\top$ stand for truth, the set of tautologies and $\bot$ stand for falsity, the set of contradictions. The syntax of the language is not transparent, ie. the atoms cannot be decomposed and do not involve predicates: object relationships. However, the data has factual associations or dependencies deliberately included, as discussed previously Section 2.4.4: SRL and below in Section 3.2. The semantics of the logic are based on a finite set of states in a micro-world. For simplicity we identify states with the assignment of truth-values (true 1, false -1) to atomic propositions. A state in which a proposition α is true is a model of α. A proposition α classically entails a proposition β, if and only if every model of α is also a model of β.

$$\alpha \models \beta \quad \longleftrightarrow \quad Models[\alpha] \subseteq Models[\beta]$$

Classical logic is explicitly monotonic (see discussion Section 2.1) and presumes the absolute nature of truth. Classical entailment is very restrictive and fails to capture much of everyday common-sense reasoning. In the example of the traffic light (Section 2.1.2), classical inference from the observation that the traffic light for oncoming traffic is red, to the conclusion that the oncoming car will stop, would result in pedestrians stepping into the path of the oncoming car that has exceptionally 'run a red light'. This is in contrast to most common-sense reasoners understanding that truth is relative. Accordingly, we utilise a logic equipped with a more generous entailment relation known as a rational consequence relation: rational consequence formalises single predictive inferences. In these inferences the limited information supplied by the premiss is amplified by information provided by a default rule. Past experience, observation of the frequency of states in the environment, commonly forms the basis for the default rule. In the example, the default rule is that: 'Cars normally stop for red lights'. These default rules are not universal generalisations; the consequent entailment is defeasible and so supra-classical logic incorporates exceptions to the single conclusions of classical monotonic logic. In the example, cars may exceptionally 'run a red light'.

It is convenient to represent the default rule by means of an ordering on the (valuations on) states of the relevant micro-world. Such an ordering stratifies or ranks the states into layers, as described by Kraus *et al.* (1990); Lehmann and Magidor (1992). These layers may be assigned an ordinal value. It is conventional to refer to the ordering as a preference relation. Traditionally the most preferred model states are denoted as the minimal models. In this thesis, these most preferred models are given the intuitive denotation of the 'maximal' models. Given a default rule represented as such a preference relation, the corresponding rational consequence relation sanctions the defeasible entailment of β by α, if and only if every maximally preferred model of α is also a model of β (Heidema and Labuschagne, 2001).

$$\alpha \mid\sim \beta \quad \longleftrightarrow \quad Maximal\ Models[\alpha] \subseteq Models[\beta]$$

"Arguably, the most important characteristic of non-monotonic logic is not its non-monotonicity, but it supra-classicality" (Labuschagne *et al.*, 2013). Supra-classical logics allow for non-preferred conclusions and can employ a range of consequence relations which tolerate counter-examples to the restrictive view of classical logic. The particular variety of supra-classical logic considered in this thesis is that based on the rational consequence relation. A conclusion in supra-classical logic involves an ordered set of model states entailed by a premiss. In the domain of statistics, these states

are counter-factuals (Pearl, 1997). In the connectionist paradigm, states correspond to stable minima. We suggest that a supra-classical conclusion (a 'problem solution') would be a set of ranked minima, not just the single global minimum.

We have chosen this broader context of the rational consequence relation in supra-classical logic, because many authors regard this ability, to be able to learn exceptions, as a key feature of common-sense reasoning (Kraus *et al.*, 1990; Lehmann and Magidor, 1992; Heidema and Labuschagne, 2001; Labuschagne *et al.*, 2013). Indeed, Pearl (1997, 2015) proposes that these counter-factuals form the basis of reasoning about causality, as opposed to simple statistical association.

3.2 Micro-World Schemata

Logical micro-worlds are the experimental sand-boxes of this thesis. They are simplified, defined environments about which we may logically reason at the symbolic level. In these environments, our candidate neural network representations are trained and tested against the expectations of SCNM logic. For our purposes, considering the rational consequence relation as a proxy for common-sense reasoning, the single solution benchmarking of previous authors: Pinkas (1995), d'Avila Garcez *et al.* (2009) and Shastri and Ajanagadde (1993); was not relevant or appropriate. Fortunately, we have been able to fall back on the ideas of Rumelhart *et al.* (1986a) and Minsky and Papert (1971), who originally coined the term. Micro-worlds are equivalent to the schemata of Balkenius and Gärdenfors (1991) and were elegantly utilised by Frank *et al.* (2009), in examining the connectionist learning of language.

The micro-worlds used to train and test candidate machines were modest incremental extensions of the simple logical 'Light-Fan System' in traditional usage. The logic appropriate for the basic Light-Fan System has just two atomic propositions, L standing for 'the light is on' and F, standing for 'the fan is on'. The states of this micro-world are then the four possible functions assigning true or false to each atom. It is convenient to depict such a function as a sequence of its outputs, which is possible if we take the order of the atomic propositions to be fixed. Thus, the state in which the light is on but the fan is off can be depicted by the binary sequence "1 -1", showing that L is true (value 1) and F is false (value -1). Note that for convenience states are often labelled in an abbreviated decimal form, for example "1 -1" is labelled 2 and "1 1" is 3. Candidate networks were *trained using this binary logic* [1]. However, the

[1]Terms emphasised at first occurrence, are defined in the glossary.

machines were *tested using a ternary logic*, where inputs of zero stand for not observed or unknown. For example an input premiss of "1 0" stands for 'Light on, Fan not observed'; such an input has no binary equivalent.

We considered micro-worlds having 3, 4, 5, and 6 atomic propositions or components. The additional atomic propositions were: H the heater (is on), W the window (is open), A the air-conditioning (is on), and O the open fire (is lit). The motivating analogy for these micro-worlds was that of a temperature-controlled room. Factual associations or dependencies between the data elements arise from this semantic analogy (see Section 2.4.4: Statistical Relational Learning). Active cooling is produced by the fan and air-conditioner, active heating by the heater and the open fire, passive dependent cooling by the window and independent illumination by a light. The analogy and its factual dependencies generate the default rule that is represented by the preference ranking on the states.

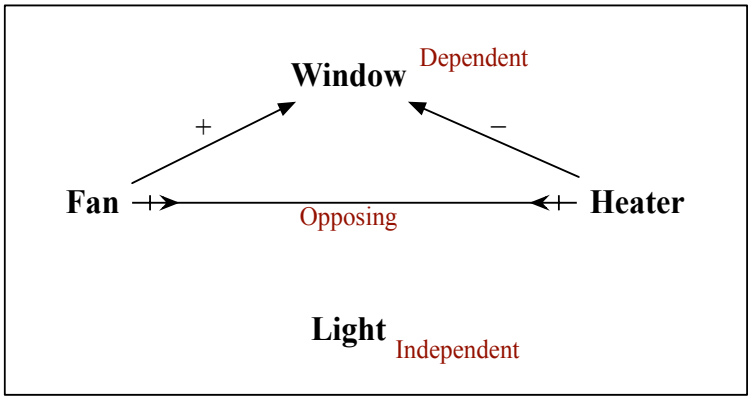
Figure 3.1 illustrates a 4-atom micro-world, the atomic propositions are: Light, Fan, Heater and Window. We revisit this example world many times in the experiments reported in Chapter 4 and Chapter 6. The example default rule represented by the preference relation incorporates these ranked observations:

- Components with a high energy cost, the fan and heater, would typically be off.
- We have specified that the environment is warm and therefore the fan is more likely to be on than the heater.
- When active cooling is being used, the window is more likely to be open whereas during active heating, the window will typically be closed. The window is a dependent component.
- The light may be on or off, independent of the other components.

From the first observation it follows that the most preferred Level 0 consists of all states where atoms F and H are false, "-1". From the second and third observations it follows that Level 1 consists of states where F is true, "1" and H is false, "-1". The dependent component W is true, "1" when the fan is on. From the third observation it follows that Level 2 consists of states where F is false, "-1" and H is true, "1". The dependent component W is false, "-1" when the heater is on. From the fourth observation it follows that Levels 0 - 2 contain all relevant variants of L is true, "1" or L is false, "-1".

4-Atom MicroWorld: Temperature Controlled Room

Light, Fan, Heater, Window



Binary States: example
1 -1 -1 -1 => decimal 8

Default Rule:
Energy Conservation
Warm Environment
Consistency

Lvl	Preference Relation				# Patterns	% Distribution
0	-1-1-1-1 (0)	-1-1-1 1 (1)	1-1-1-1 (8)	1-1-1 1 (9)	16	16.7%
1	-1 1-1 1 (5)	1 1-1 1 (13)			8	8.3%
2	-1-1 1-1 (2)	1-1 1-1 (10)			4	4.2%
3	-1-1 1 1 (3) 1-1 1 1 (11)	-1 1-1-1 (4) 1 1-1-1 (12)	-1 1 1-1 (6) 1 1 1-1 (14)	-1 1 1 1 (7) 1 1 1 1 (15)	1	1.0%

Training Set: Total Patterns = 96

Figure 3.1: An example micro-world consisting of four atoms (Light, Fan, Heater, Window), based on the default rule described in the text. The most preferred states are shown at the top of the ranking, which is based on the default rule. States (patterns) are shown in binary eg. "1 -1 -1 -1" and for convenience labelled with the equivalent decimal value, eg. 8. The number (#) and frequency (%) of each state in the training set is given for each level, for example pattern "1 -1 -1 -1" (8) is seen 16 times - 16.7% of the total. There are $(4 \times 16) + (2 \times 8) + (2 \times 4) + (8 \times 1) = 96$ total patterns in this example training set. This example world is revisited many times in the experiments reported in Chapter 4.

By default, Level 3 consists of all the remaining states, which are inconsistent with the observations that constitute the default rule, particularly those irrational states where the heater and fan are both on: Fan true "1", Heater true "1".

Figure 3.1 illustrates a single example epistemic state with preference relation in a 4-atom micro-world. This example micro-world has been deliberately designed to demonstrate a range of associations between the atomic propositions: independence of the Light, dependence of the Window on the Heater & Fan and opposition of the Heater and Fan. The most preferred states are observed more frequently, such as the state "1 -1 -1 -1" in this example, where only the light is on. Regardless of any semantic analogy used to conceptualise the micro-world, it is only a single example of the trillions of permutations possible on factual associations between 4-atoms.

Testing every micro-world variant would be difficult given that there are more than: 8! 3-atom micro-worlds, 16! 4-atom micro-worlds, 32! 5-atom micro-worlds and 64! 6-atom micro-worlds. Our largest size world with 6-atoms might seem trivial, however, this world has 64 states and the machine is required to learn the preference ranking from the logic. All the relationships the machine 'observes' are one-to-many mappings. An equivalent scenario in real life would be a medical diagnostic sieve with 64 diagnoses. d'Avila Garcez *et al.* (2009) considers Mickalski's trains to be a significant task requiring learning of one-to-many relationships. This problem has only 4 characteristics (atoms) spread over 10 input states with 2 output classifications.

3.2.1 Mapping States to Network Activations

This section is intended to provide some preliminary context describing the way in which the micro-worlds outlined above will be used by the neural networks. As noted above, the semantics of the logic are based on a finite set of states. We identify states with the assignment of truth-values to atomic propositions. A state in which a proposition α is true is a model of α, in the example 4-atom micro-world just discussed one such state is "1 -1 -1 -1", which is a model of L (the Light is on, Fan off, Heater off, Window closed). In our machine networks logical states are represented directly as the patterns of activation on the input or output units of the network, one unit per atom. In other words, the logical state "1 -1 -1 -1" is represented as the pattern of activation "1 -1 -1 -1". While emphasising different aspects of context, the terms state and pattern are effectively interchangeable in our networks.

In order for a machine / network to learn a default rule (a preference order on states), training sets are designed which, in effect, allow the machine to observe a

distribution of states from the environment of the micro-world. This statistical distribution arising from the factual dependencies corresponds to the preference relation in the logic. Although the logic does not require the numerical exactness of probability, it still maintains some notion of magnitude, the distance between preference levels (Pinkas, 1995). This concept of magnitude is particularly important when considering belief revision (Spohn, 1988).

For reasons of space and clarity, we predominantly present samples from 4-atom micro-worlds. Furthermore, all worlds share certain design assumptions: the least preferred states are usually included once in the training set (they thus have a frequency of roughly 1% of the total distribution) and there is usually an exponential change in pattern frequencies between preference levels (a doubling in frequency between levels). Given the very large number of possible permutations on worlds and preference relations, we had concerns that the arbitrary nature of the designed micro-worlds may have resulted in unintended structure impacting on our experimental results. To that end we have also examined many micro-worlds of varying sizes, including random versions of a 5-atom micro-world based on alterations in both the default rule and in the ordering of the atoms. We have also examined worlds where the least preferred states are not included in training and with arithmetic progressions in pattern frequency. These results were generally reassuring, see Chapter 4: Representation and Chapter 5: Incongruence. The full results of examples of all variants are presented in our technical report (Blanchette, McCane, Labuschagne, and Robins, 2015).

3.3 Requirements

The choice of candidate network representation was directed by the requirements of both the logic and the biology, see Figure 1.1. In Section 3.3.1, the relationship between the logic and the network representation will be discussed in detail. In Section 3.3.2, we give a brief outline of the requirements for biological plausibility: the second half of Hinton's bridge.

3.3.1 Logical

Haykin (2008, p38-45) categorises the learning tasks performed by neural networks: pattern recognition, classification, function approximation and control. They all involve mapping multiple or single inputs to single outputs: many-to-one or one-to-one relationships. One-to-many relationships, which map single inputs to multiple outputs,

are not mathematical functions and have been disparagingly called 'ill-posed problems' (Swingler, 1996; Rosales and Sclaroff, 2006). The implication is that these problems need to be re-formulated: re-engineered at some meta-level. This is usually achieved by changing the nature of the data or the nature of the problem: often by re-coding each instance or output permutation of the relationship into a single node, thereby producing a combinatorial explosion in the network size. However, these alterations in the representation of the problem are not a genuine solution for an individual machine training directly from its environment, as they involve a pre-existing knowledge of the structure of the problem, which is not available to the machine during training.

One-to-many relationships are common in the real world, typical examples are: kinematic solutions to the positioning of robot limbs and diagnostic classifications (Freeman, 1994; UCI, 2013), where single symptoms can be associated with many possible diagnoses. In our case, the example is SCNM logic, which also requires a ranking of the outputs within the one-to-many relationship. We want the machine representation to accept a single input (a premiss) and to provide as output the models of the premiss distributed identically to the preference ordering in the logic. It is not sufficient for the machine to learn a generative model of the joint probability distribution; we also require it to retrieve a multi-modal output given uncertain information: a discriminative case.

We considered that the necessary logical requirements of a network representation (the elementary components of the rational consequence relation) were the ability to:

1. Learn a preference ordering on the set of states.

2. Select the appropriate models of a sentence based on limited information (a premiss).

3. To compare the selected model states of two sentences (usually a premiss and a conclusion) involved in inference

4. Revise (re-train) its preference relations in keeping with the theories of iterative belief change in the logic.

These requirements provide a structure for the whole thesis. The first three requirements encompass the process of inference in SCNM logic with rational consequence and are the subject of Chapter 4: Representation. Chapter 4 specifically concentrates on only the first two components of the rational consequence relation. We believe the learning of the preference relation and the selection of appropriate model states

are distinct neural network tasks, which we demonstrate can be satisfied by a single machine. A separate, feed-forward network could compare the outputs of maximally preferred model states: the third requirement.

The fourth requirement, of any network representation of SCNM logic, is not directly associated with inference. The network should be able to revise its learned preference relation in keeping with the theories of iterative belief change in the logic. This ability to adapt to new information is the subject of Chapter 6: Belief Revision. We examine some of the approaches to iterated belief revision in non-monotonic logic (Alchourron and Makinson, 1982; Alchourron *et al.*, 1985; Spohn, 1988; Chopra, Ghose, and Meyer, 2003; Booth and Meyer, 2011) and compare these to adaptation in our artificial neural network representation. The inherent response of any neural network to new information is extreme: catastrophic forgetting (Robins, 1995), whereas the logic emphasises minimal change. This conflict will be examined in Chapter 6: Belief Revision.

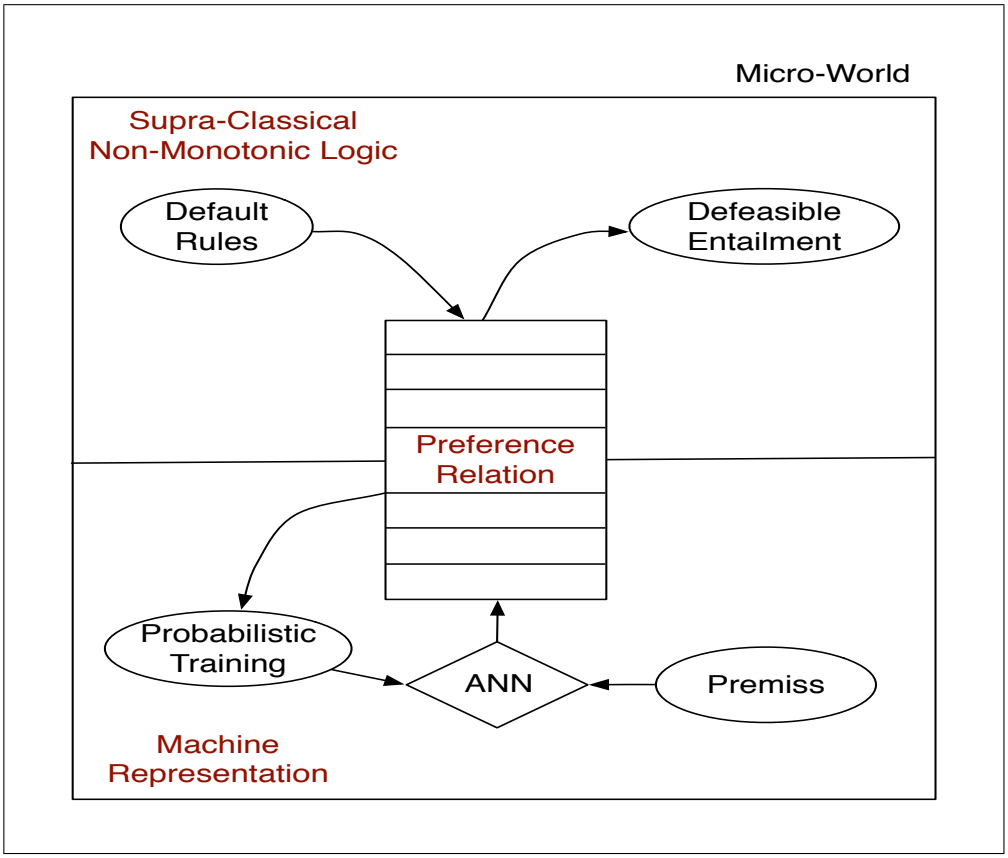


Figure 3.2: The structure of a micro-world: the shared preference relation, provides a connection between the design level logic and the neural network machine. ANN = Artificial Neural Network.

Returning to the concept of a micro-world, it can be seen as divided into two separate levels: one a design meta-level, the other a lower machine level (Figure 3.2). The logic exists throughout the micro-world, but its semantics and properties can only be proven mathematically at the design meta-level. The machine does not 'see' this abstracted level. It is not intended that our machine representations be 'theorem provers', or that they in any way reflect the structure of the logic. Nor do we intend that the contents of their 'black-box' be open to symbolic interpretation (Hinton, 1990). This was the method of previous research in reconciling the two paradigms. We are only interested in the machine outputs: that the experimental evidence they provide fits with the expected conclusions of the logic, within the context of a micro-world.

It is not necessary to show that a suitable candidate network will follow every possible property in the logic. It is sufficient to show that it fulfils the requirements we have identified. The preference relation of each micro-world is the connection between the logic and the machine representation (Figure 3.2). The properties of the logic are a consequence of the preference relation. If the machine representation is able to faithfully learn the preference ordering and select maximally preferred models given a premiss, then we assert it will **always** find solutions that fit the properties provable in the logic: as a consequence of the shared preference relation.

3.3.2 Neurobiological

Biological plausibility is not a principal aim of this research: we intend our network representation largely as an abstraction. However, any network which intends to embody SCNM logic and indirectly account for some aspect of human cognition must have a degree of biological plausibility.

We provide only a brief outline to contrast the Boltzmann machine presented here (a symmetric neural network) and the *multi-layer perceptron* (MLP, a simple feed-forward network) presented in our technical report (Blanchette *et al.*, 2015). A number of authors have published on this topic (Mazzoni, Anderson, and Jordan, 1991; Izhikevich, 2004) and O'Reilly (1998) has identified six principal requirements:

Distributed Representation: which allows for feature detection, robustness and redundancy. Artificial neural networks are distributed by nature. However, the MLP representation required separate coding of model states in the output layer to provide some of the functionality of the Boltzmann machine. As a result, the MLP suffered from the problem of combinatorial explosion as the number of

atoms in each micro-world was incremented.

Inhibitory Competition: by inter-neurones is an important anatomical component in neurobiology, contributing 20% of the cortex, allowing selection of dominant representations. Authors including the logician Leitgeb (2004) have emphasised inhibitory interneurons. Simple feed-forward networks such as the MLP have no intra-layer connectivity. By contrast, the Boltzmann machine has a mathematical complexity related to its intra-layer connections, that was the principal reason for it being abandoned by artificial neural network research (Section 4.1: Network Selection).

Bi-directional Activation: refers to the flow of information through the network. The Boltzmann machine is explicitly bi-directional, a property that is the basis of constraint satisfaction. Simple feed-forward networks (MLP) would need to be given recurrency.

Error Driven Task Learning: relates to supervised learning used in directing weight update. A localised version of the back-propagation algorithm would make it more biologically plausible. This requires a signal difference (cross-firing statistic) to be stored at each synapse, very similar to the Boltzmann machine.

Hebbian Model: the Boltzmann machine uses Hebbian learning. It is worth considering for a moment the remarkable nature of the algorithm. It can be seen as unsupervised associative learning requiring only local information.

Biological Realism: this is a rather vague characteristic, but for the Boltzmann machine we offer three properties:

- The bi-phasic nature of the learning algorithm and its possible co-relation to REM sleep (Sections 4.4: Discussion of Human Cognition & 6.2.1: Machine Re-training),
- Simulated annealing, which is the most biologically plausible variety of threshold optimisation (Sections 4.4 & 4.1: Network Selection),
- And the stochastic nature of the machine itself, which reflects the random nature of biology, compared to the determinism of feed-forward networks (Section 4.1).

In summary: there is a contrast between these two classical networks as candidates for representation of the logic. The MLP requires major modification to fulfil some of the biological requirements whereas, it is possible that the Boltzmann machine satisfies all of them.

Chapter 4

Representation

We considered several varieties of neural network for the representation of supra-classical, non-monotonic logic. In Section 4.1: Network Selection, we summarise the properties of the Boltzmann machine which we believe uniquely qualify it for this role. In Section 4.2, we outline the architecture and training of the Boltzmann machine, including some modifications that benefit performance.

The experimental results that follow in Section 4.3 evaluate the first two logical requirements we identified in Section 3.3.1. The results support the Boltzmann machine as a neural network representation of inference in logic, under the rational consequence relation. Further, detailed statistics are provided for single and ensemble machines, in the context of the exemplar 4-atom micro-world illustrated in Figure 3.1. Experimental results for two important logical properties are also presented, as a practical illustration of inference by the Boltzmann machine.

We conclude the chapter with a brief discussion of the place of the Boltzmann machine in the wider context of human cognition (Section 4.4).

4.1 Network Selection

Experimental testing with a multi-layer perceptron in 4-atom and 5-atom micro-worlds, as demonstrated in our technical report (Blanchette *et al.*, 2015), was unsatisfactory. The multi-layer perceptron is a simple non-recurrent network with at least one hidden layer, trained via back-propagation. Regardless of the issues with biological plausibility (Section 3.3.2), the MLP could not faithfully learn an input distribution; it could not rank the appropriate one-to-many relationships as expected from the logic. Using Pinkas's (1995) classification of equivalence relations, the MLP was at best "weakly

- minima preserving": meaning that the appropriate models were retrieved but their ordering was random. Even this unsatisfactory result required that each output permutation be coded to an individual node, resulting in a combinatorial explosion as the atomic size of the micro-worlds was incremented (Section 3.3.1: Logical Requirements).

We considered Hopfield style networks. Hopfield networks are symmetric recurrent neural networks, with binary threshold nodes, where the energy of the network is guaranteed to converge to a local minimum. Much of the original research on logical representation was done using them: Balkenius and Gärdenfors (1991); Jagota (1994); Pinkas (1995). Pinkas's penalty logic translates between a single syntactically correct sentence in classical logic and a specific SNN. He demonstrates the equivalence of an energy minimum in an individual SNN and a solution satisfying the related logical sentence. Asymmetric Hopfield networks are known to produce multiple outputs, potentially one-to-many mappings, by way of chaotic or cyclical attractors. However, we were not aware of any underlying mathematical principles in an asymmetric network governing the retrieval of these minima (preferred states). Preliminary testing with a symmetrical variety of Hopfield network disappointingly only returned single outputs. This was in contrast to the view of Pinkas (1995), who suggests that Hopfield style networks should be at least "preference preserving": maintaining an ordering on states but not the magnitude of the ordering.

The Boltzmann machine is a variety of SNN, where the nodes have stochastic activation functions. The network is trained in alternating phases, sampling cross-firing statistics, which effectuate Hebbian learning. The Boltzmann distribution, from statistical mechanics, characterises the relationship between information and energy in the network at equilibrium, achieved by simulated annealing (Section 2.3.1: Boltzmann Machine).

The machine inherits all the favourable characteristics of the SNNs, with their logical equivalence demonstrated via Pinkas's penalty-logic. It learns and can simultaneously search multiple local or global minima. The minima can be interpreted as corresponding to solutions or conclusions in the logic. This is not only in the classical realm of SAT-problems but also in the wider context of supra-classical logic, which requires evaluation of counter-examples related to common-sense reasoning. The machine is also the neural network equivalent of the graphical models (Bayesian and Markov) used in statistical relational learning (Section 2.4.4). These models are theoretically designed from statistical associations between factual observations that form the basis of default rules in SCNM logic.

The Boltzmann machine was largely abandoned in the early years of neural network research and its complexity was reduced in the design of the restricted Boltzmann machine, which is utilised in the current research on Deep Belief Networks. The slow learning that resulted from simulated annealing was seen as a disadvantage in the engineering domain. As a consequence of its stochastic nodal activation, the Boltzmann machine is able to cycle through multiple states at equilibrium, searching multiple energy minima simultaneously (Pinkas, 1995): these constitute the one-to-many relationships entailed by a premiss. Further, because of its probabilistic learning, it is theoretically able to represent a complete joint probability table: a generative model. This generative model can be explicitly retrieved from the machine by time-slicing through its learned input distribution, given a completely neutral premiss: clamping with a null input (Hinton *et al.*, 1984). As a consequence, it is the only SNN Pinkas regards as capable of representing a "strong equivalence" relation: not only order preserving but also magnitude preserving.

$$P_{k|T} \equiv \frac{e^{\left(\frac{-Enet_k}{T}\right)}}{\sum_l e^{\left(\frac{-Enet_l}{T}\right)}}$$

As previously illustrated, these favourable properties are encapsulated in the Boltzmann distribution equation, which relates the probability of output states at equilibrium, to the relative entropy (learned preference) of these states.

Many authors have utilised different varieties of Boltzmann machine for the solution of practical problems in the realm of categorisation or function approximation (Chen and Murray, 2003; Egger, 1988; Eslami, Heess, and Win, 2012; Hinton, 2010; Ortega and Parrilla, 1999; Pfleger, 1998; Sathasivam, 2011; Sejnowski, 1986; Tichnor and Barret, 1987). We however, examine the use of the Boltzmann machine in a different context and confirm that the machine can explicitly retrieve the distribution of its training set when presented with a neutral (non-specified, null) input and select appropriately ranked model states given incomplete or partial information (a premiss). This multi-modal output, retrieval of one-to-many relationships, constitutes a ranking of counter-examples, the essence of SCNM logic.

4.2 Implementation and Training

4.2.1 Background

Initially our Boltzmann machine implementation was based on the technical descriptions of the 424-Encoder from Hinton's papers (1984; 1985; 1989) and the work of Aarts & Korst (1990; 1997). We replicated Hinton's results and examined the performance of specific machines / networks on a number of data sets, including a car mechanic analysis (Swingler, 1996) and a medical diagnostic data set (UCI, 2013). The networks performed tolerably well on even the most difficult medical diagnostic task which contained 8 diagnoses spread over 13 symptoms, with a large proportion of one-to-many relationships. The training set consisted of 120 patterns; 40 withheld for testing. When tested on unseen patterns, the networks had a credible accuracy of $\sim$75%.

When we were confident that our implementation was correct, we began exploring its application to the micro-worlds described above. During the process of optimising performance for these tasks some modifications to the Hinton architecture and learning algorithm were made, as described below. These modifications can be regarded as minor and within the natural range of variation when implementing specific versions of a generic Boltzmann machine.

4.2.2 Architecture

The number of input and output units in our networks are dictated by the micro-world being implemented, each atom is represented by one input and one output. The initial choices for number of hidden nodes and interconnection were based largely on a heuristic of biological plausibility. We deliberately intended that the hidden layer have a larger number of nodes which would all be interconnected. The direct connection of input and output layers was not biologically plausible.

A range of numbers of hidden units was explored and was not found to be critical to performance. The final numbers of hidden nodes used were: 4 in the 3-atom worlds, 6 in the 4-atom worlds, 8 in the 5-atom worlds and 10 in the 6-atom worlds. An exception to this was found in the 3-atom world, where learning of the least preferred states or patterns was improved by adding more hidden nodes (see result in Table 4.1 and the discussion following).

Compared to the Hinton networks the intra-layer connections were removed from the input and output layers after experimentation indicated they were not significantly

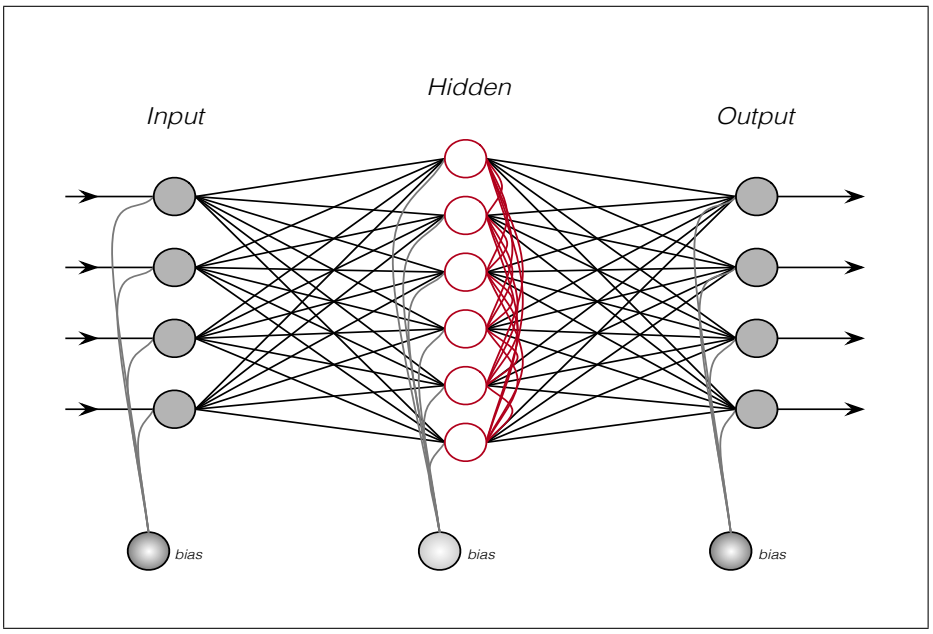


Figure 4.1: HLR Architecture for a modified Boltzmann machine network in a 4-atom micro-world. The network is layered and a standard bias unit is present for each layer. Intra-layer connectivity is maintained only in the hidden layer.

helping performance. However, the hidden layer intra-connections were maintained for biological plausibility, in keeping with the work of Hinton 1985, Balkenius & Gärdenfors 1991 and Leitgeb 2004 who placed importance on inhibitory constraints.

The discrete layers of our networks are similar to an MLP. It was more convenient to divide the visible nodes into input and output. During training, in the clamped phase of each cycle, the atoms of a premiss / state (for example "1 1 -1 1") are clamped on the input and output units (an auto-associative task). During testing the specified atoms of the premiss are clamped on specific input units (+1 true and -1 false). Unspecified units are clamped with zero (an indeterminate value) and we examine the distribution of states created on the output units.

The typical architecture of a network in a 4-atom micro-world is illustrated in Figure 4.1. This Boltzmann machine architecture we have termed *'hidden layer rich' (HLR).*

4.2.3 Learning Algorithm

Learning was carried out in accordance with the standard Boltzmann machine learning algorithm (Hinton *et al.*, 1985, see Section 2.3.1), with the use of a momentum term (Haykin, 2008) and some modifications relating to the use of the annealing schedules.

On our tasks these modifications were found to either improve performance or reduce complexity without damaging performance:

- Annealing was removed from the clamped phase. During this phase the hidden nodal states are largely determined by the clamped nature of the visible nodes.
- Layered annealing rather than pooled was used in the free phase. Nodal states in the hidden layer are randomised and the network is run. This largely determines the states of the nodes in the visible layers. However, because the states of the hidden nodes are determined by chance the whole process has to be annealed to ensure that lowest energy states of the whole network are found.
- Our annealing schedules were designed; based on the inverse log function from thermodynamics (Metropolis and Rosenbluth, 1953; Geman and Geman, 1984).
- Annealing during training and testing were configured at slightly different 'temperature' ranges, using the same inverse log function.

Weight decay (Krogh and Hertz, 1995) and sparcity (Hinton, 2010) were experimentally tested trying to mitigate the Hebbian characteristic of weight saturation, but ultimately only a standard implementation of momentum was retained in our final version of the learning algorithm. A summary of our slightly modified version of the Boltzmann machine learning algorithm was presented earlier in Figure 2.5. Our annealing schedule is an approximation based on the inverse log function, was previously shown in Figure 2.6.

Tuning of the learning process was time consuming: details of the annealing schedules can critically influence results. A wide variety of schemes were examined: from high temperature ranges ($40 \rightarrow 10$) to low temperature ranges ($5 \rightarrow 1$), for varying temperature points and cycles at each temperature point (5 - 30). There was no single correct schedule. The other tuning parameters were on average: training time 2,000 epochs, learning rate 0.3, momentum 0.7, and 20 samples per pattern (for estimating $\rho+$ and ρ-, Figure 2.4).

4.2.4 Implementation

A brief overview of the object-oriented design of the Boltzmann machine implementation is illustrated in Figure 4.2. The coding was performed in C++ and is provided on CD (see Appendix B). Machines are run via an application that accumulates statistical results from parallel threads.

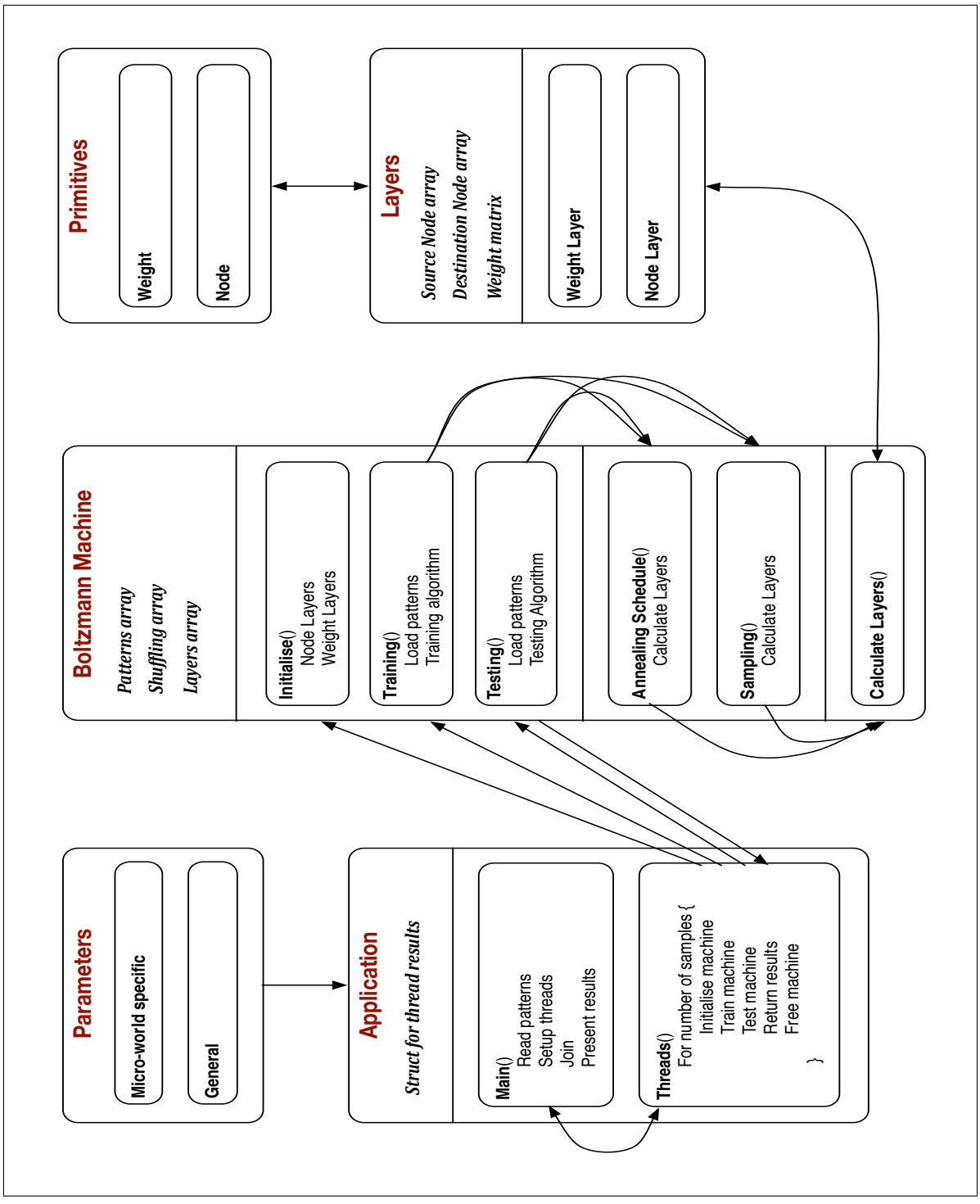


Figure 4.2: Object-oriented implementation of a Boltzmann machine: the machine class has public methods for initialising, training and testing of a Boltzmann machine using private methods for annealing and sampling which utilise layer primitives via a 'calculate' method. Machines are run via an application that collates statistical results from parallel threads.

4.2.5 Data Collection

Our networks were trained using fully specified states (all atoms / input units set to 1 or -1), and were tested with either fully specified or partially specified states. In the partially specified case some atoms / inputs were set to 0 (standing for not observed or unknown), for example an input premiss of "1 0" stands for 'Light on, Fan not observed'.

The performance of individual Boltzmann machines / networks trained in the same way on the same data can vary widely. Given the nature of the optimisation problem and the stochastic nature of the machine, it is likely that several different weight configurations serve as alternative potential solutions. For this reason, our results report the accumulated output sampled from multiple machines.

Characteristically and crucially, the output of a Boltzmann machine is not static or deterministic, it is continually cycling between various states. Thus the representative sample output from a machine is a time-slice at equilibrium of all the output states: the larger the equilibrium sample, the closer to the actual learned distribution of the machine.

When looking at a single premiss, in Sections 4.3.1: Training and Recall and 4.3.3: Model Selection below, we have taken 60,000 output samples over 60 separate machines. When looking at retrieving the whole preference relation, Section 4.3.2: Preference Relation, we have taken 300,000 output samples over 300 separate machines. These raw sample distributions are then converted to a percentage distribution, dividing by the total number of samples.

4.3 Experimental Results

In a technical report (Blanchette *et al.*, 2015), we present an extensive collection of results from testing the Boltzmann machine, together with a comparison with a simple feed-forward network (MLP). Here, we present an overview and samples of our results drawn mostly from a 4-atom micro-world. Section 4.3.1: Training and Recall, presents a basic test of successful training, the ability to recall individual states. Sections 4.3.2: Preference Relation and 4.3.3: Model Selection, describe the representation of core properties of the logic, respectively the representation of the preference relation and the drawing of a conclusion from a premiss. Section 4.3.4: Logical Properties, briefly explores further interesting logical properties.

4.3.1 Training and Recall

Data sets for Boltzmann machine training were derived from the preference relation in the logic, analogous to the machine being able to observe the frequency of states (patterns) within the environment. During training the input and output units are set to the same patterns (an auto-associative task). For each micro-world four training sets were constructed: two with an exponential increase in pattern frequencies between preference levels and two with an arithmetic increase. For each of these pairs: one training set had the least preferred patterns absent and the other had the least preferred patterns present for a single instance. An example of a training set with an exponential pattern distribution and least preferred patterns present, is illustrated in Figure 3.1. As previously stated, the micro-world problems that are the subject of this investigation are not a classification or function approximation task. The distribution of patterns constitutes all of the environmental / training information and, for testing (given a specific input), the behaviour of interest is the distribution of output patterns.

The adequacy of training in each of the micro-worlds can be demonstrated by looking at the results from testing machines against fully specified premises (complete model states). When a complete state is clamped on the input units of a well-trained machine exactly that same state should dominate the output distribution. Remember, these are stochastic machines and are only presented with low preference states for a very small proportion of their training.

Testing Against Fully Specified Premises				
Micro-World	*Most Preferred State Output*		*Least Preferred State Output*	
3-Atom	"-1 -1 1" (1)	95%	"1 1 -1" (6)	54%
4-Atom	"-1 -1 -1 -1" (0)	92%	"-1 1 1 -1" (6)	85%
5-Atom	"1 -1 -1 -1 -1" (16)	97%	"1 1 -1 1 1" (27)	89%
6-Atom	"1 -1 -1 -1 -1 -1" (32)	96%	"-1 1 1 1 1 1" (31)	93%

Table 4.1: A basic test of training and recall. Results for testing various Boltzmann machines against fully specified premises (input states), showing the frequency of the correct state in the output distribution (expressed as a percentage). Examples are given of one most preferred and one least preferred state in each micro-world variant. These results are for 'exponential' training sets with least preferred patterns present.

A brief summary of the adequacy of training across a range of micro-worlds is presented in Table 4.1, by considering one most and one least preferred state in each atomic variety of micro-world. As described in Section 3.2: Micro-World Schemata, the 3-atom micro-world consists of Light (L), Fan (F) & Heater (H) and the additional atoms in the larger worlds are Window (W), Air-conditioning (A) and Open fire (O). Hence the 3-atom premiss "-1 -1 1" with decimal label 1 represents the state where the Light is off, the Fan is off and the Heater is on. In the example shown in Table 4.1 the correct output constituted 95% of the output distribution of states. For the most preferred states the machines trained uniformly well. There was difficulty training on the least preferred states in the smallest 3-atom world (demonstrated in Table 4.1). This was overcome by increasing the number of hidden nodes to 8.

4 Atom Micro-World **Light, Fan, Heater, Window**

Training Set 96-1

Testing with premises:

"-1-1-1-1"

Pref. Relation	*State*	*Expect %*	*Samples*	*Output %*
(0)	0	100%	55240	92%
	1	0%	679	1%
-	2	0%	622	1%
	3	0%	4	0%
-	4	0%	176	0%
	5	0%	2	0%
-	6	0%	7	0%
	7	0%	0	0%
	8	0%	3188	5%
	9	0%	63	0%
	10	0%	9	0%
	11	0%	0	0%
	12	0%	10	0%
	13	0%	0	0%
	14	0%	0	0%
	15	0%	0	0%

"-111-1"

Pref. Relation	*State*	*Expect %*	*Samples*	*Output %*
-	0	0%	42	0%
	1	0%	0	0%
-	2	0%	2502	4%
	3	0%	26	0%
-	4	0%	878	1%
	5	0%	23	0%
(6)	6	100%	51004	85%
	7	0%	803	1%
	8	0%	2	0%
	9	0%	0	0%
	10	0%	155	0%
	11	0%	0	0%
	12	0%	83	0%
	13	0%	1	0%
	14	0%	4429	7%
	15	0%	52	0%

Figure 4.3: Results for testing 60 Boltzmann machines against two fully specified premises in the example 4-atom micro-world. The preference relation has most preferred models at the top. Model states are listed as decimal labels with their expected frequency.

Figure 4.3 shows a specific example in more detail: the accumulated results for a run of 60 machines trained on the example 4-atom micro-world, illustrated in Section 3.2: Micro-World Schemata (Figure 3.1). These machines are tested against two fully specified premises: one at high preference "-1-1-1-1" (0), and one at low preference "-1 1 1-1" (6). Note, that given full input information the conclusion (ordered set of model states entailed by the premiss) rules out all other states. Thus in each case in the figure, the single state shown in the preference relation is the same as the input premiss. The machine output in both cases is almost entirely the expected model state. Recall that in the case of the less preferred model state (6) machines only see it in $\sim$1% of training pattern instances.

4.3.2 Preference Relation: a Generative Model

Generation of a complete joint probability distribution corresponds to the first component we have identified as a requirement of predictive inference, under the rational consequence relation Section 3.3.1: Logical Requirements. This preference relation can be retrieved from the machine by testing it against a neutral or null premiss: an input that contains no observed information about the state of the micro-world: "0 0...0". When clamped with this input and sampled at equilibrium, the machine cycles through all the micro-world states, retrieving the learned distribution of its training set.

When testing machines against the neutral premiss, the output frequencies obtained for each state can be directly compared to the expected training frequencies. Overall accuracy of a machine is simply indicated by the absolute percentage error at each state compared across the whole distribution. We have presented this error estimate averaged per state $\pm$ one standard deviation. The results for testing the Boltzmann machine against the neutral premiss across a range of atomically variant micro-worlds confirm that the machine can learn a preference relation; a generative model of the whole joint probability distribution, *p(x,h)*. Table 4.2 summarises the Boltzmann machine's good performance, using the metric of average error per state. As a generalisation this error per state is less than 2% $\pm$ 0.05% and is a consequence of the stochastic design of the machine.

Initially these results were obtained by running five sets of machines, with 60 machines in each set. A total of 300 separately trained machines. The output samples within each run were accumulated. In effect, each run of machines acted as an ensemble with joint input and output layers; the hidden layer consisting of 60 parallel machines. This architecture implementing physical accumulation of output samples,

Testing Against the Null Premiss			
Micro-World	*Least preferred patterns:*	*Average error per state: %*	*± STDev*
3-Atom	Absent	0.9	0.11
	Present	1.8	0.11
4-Atom	Absent	0.6	0.07
	Present	1.6	0.04
5-Atom	Absent	1.6	0.04
	Present	1.8	0.02
6-Atom	Absent	0.5	0.03
	Present	1.1	0.03

Table 4.2: Results for retrieval of the preference relation, after testing the Boltzmann machine against the neutral premiss ("0 0 0 0") across a range of atomically variant micro-worlds. For exponential training sets, with least preferred patterns either absent or present. The low average error indicates that the preference relation (output distribution) is correctly retrieved.

results in a neutralisation of absolute errors on opposite sides of the mean. Whilst some of this effect occurs by chance, it is greatly facilitated by parallel cohorts of hidden layer machines with mirror image weight matrices which arise during training (see Section 5.2.4: Network Analysis). Weights are initially seeded with small random values, there is an equal chance that for any weight the initial sign of the seed will be negative or positive. During successful training those individual weights train toward the same goal weight (amplitude), but their signs are opposite. These cohorts of machines have weight matrices which are symmetrical, balancing errors on either side of the mean.

Realising this, we present a more detailed account of the errors for both single and accumulated output, ensemble machines. We have chosen for consistency to focus on the 4-atom micro-world illustrated in Figure 3.1. The difference between the training and the output distributions (percentage error across the whole distribution) is illustrated in the detailed statistics presented for 6,000 singles machines and 100 ensemble machines with 60 parallel hidden layers (Table 4.3, Figures 4.4 & 4.5).

There are some important issues identified in these results. We point out that the

total sample size for the single machines is the same as ensemble machines of 60. Given this, the central limit theorem in statistics (Annis, 2014) cannot be an explanation of the improvement in the mean error.

- There are almost ideal single machines, with very low error (4.4%) across the whole distribution. However, they are rare and there is no efficient procedure for generating them.
- There is a huge variation in the error on single machines (total range: 70%). The mean error is moderately acceptable (25.4%) but there is a positive skew.
- The best results for the ensemble machines are not as good as the best single machines. However, the mean results for even a small ensemble are considerably better (10.4%) than for a much larger group of single machines.
- The sampling for ensemble machines approximates a more normal distribution of error, with a very narrow variance (total range: less than 5%).
- The processing time for any ensemble is the same as for a single machine; all the hidden layers can be run in parallel.

In summary, although almost ideal single machines exist, they are rare. Any small ensemble will produce fast, robust results: low mean error and little variance.

Statistics for % Error: across whole distribution		
Statistic	*Single*	*Ensemble 60*
Sample Size	6,000	100 x 60
Range	70.0 : Min 4.4, Max 74.4	4.9 : Min 8.4, Max 13.3
Quartiles	1st 19.8, 3rd 30.4	1st 9.7, 3rd 11.0
Centre	Mean 25.4, Median 24.8	Mean 10.4, Median 10.4
Variation	SE 0.10, StDev 7.8	SE 0.09, StDev 0.95
Shape	Skew 0.46, Kurt 0.34	Skew 0.36, Kurt -0.06

Table 4.3: Statistics for the error between input and output distributions, comparing Single and Ensemble HLR machines in a 4-atom micro-world. Statistics via CRAN (2014).

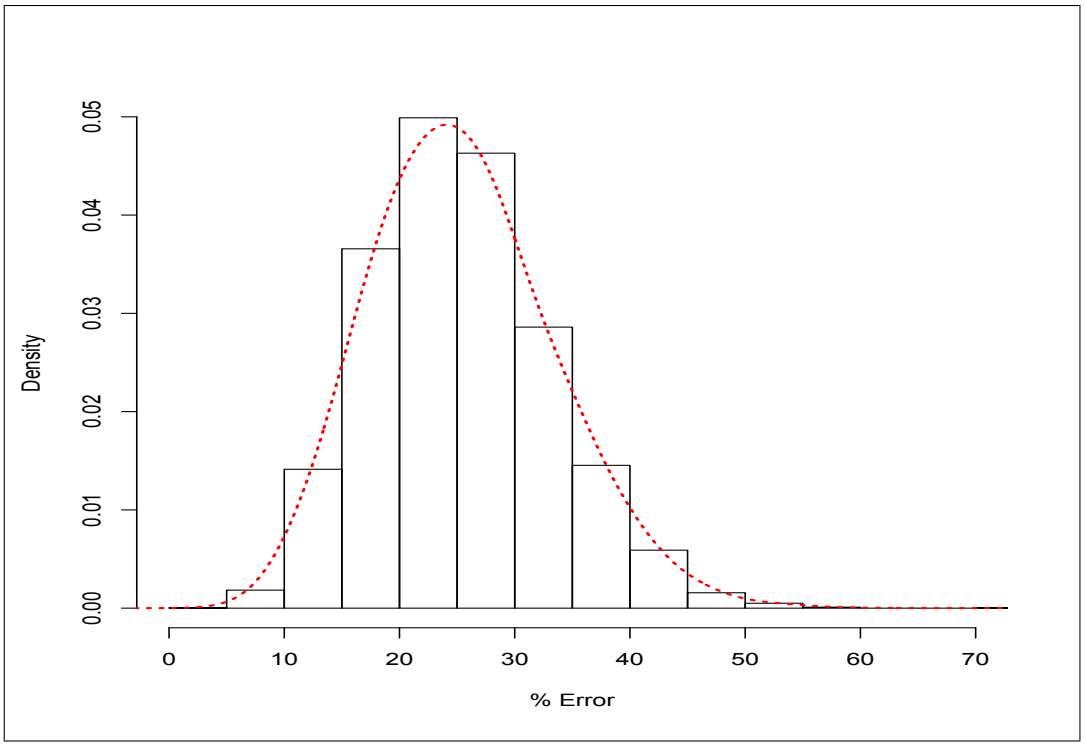


Figure 4.4: Histogram of % Error for Single HLR Machines (across whole distribution in a 4-atom micro-world.)

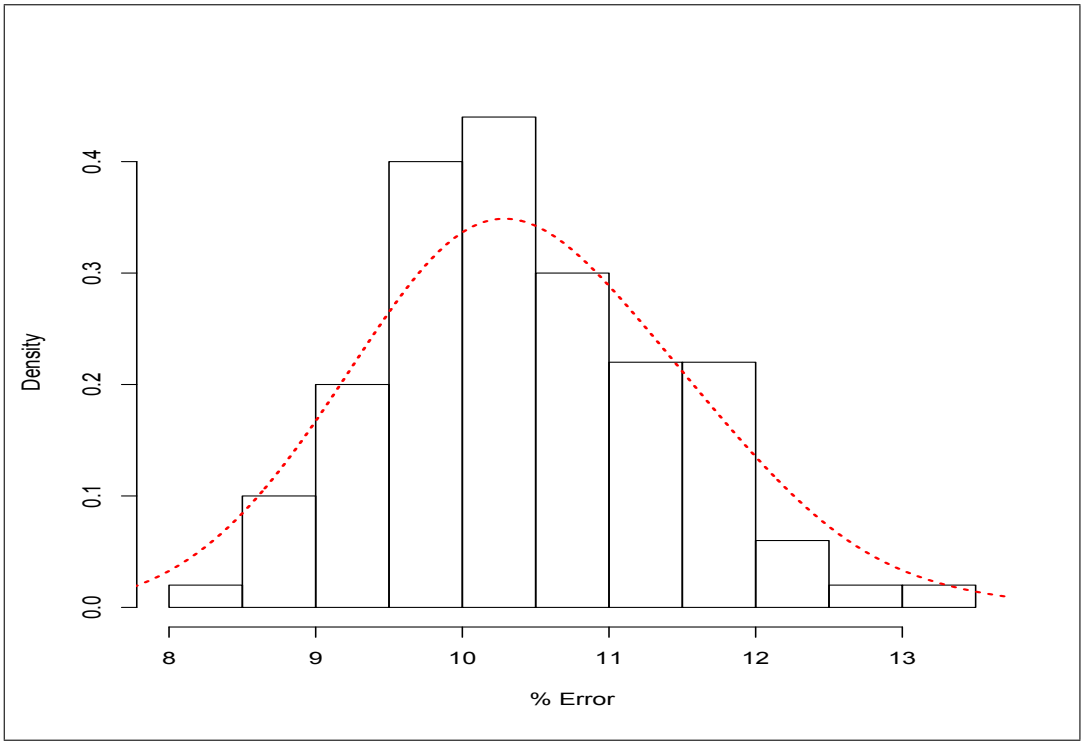


Figure 4.5: Histogram of % Error for Ensemble HLR Machines, with 60 parallel hidden layers.

Further experimentation demonstrates that little benefit is obtained by increasing the ensemble size above 60 parallel hidden layers, Figure 4.6.

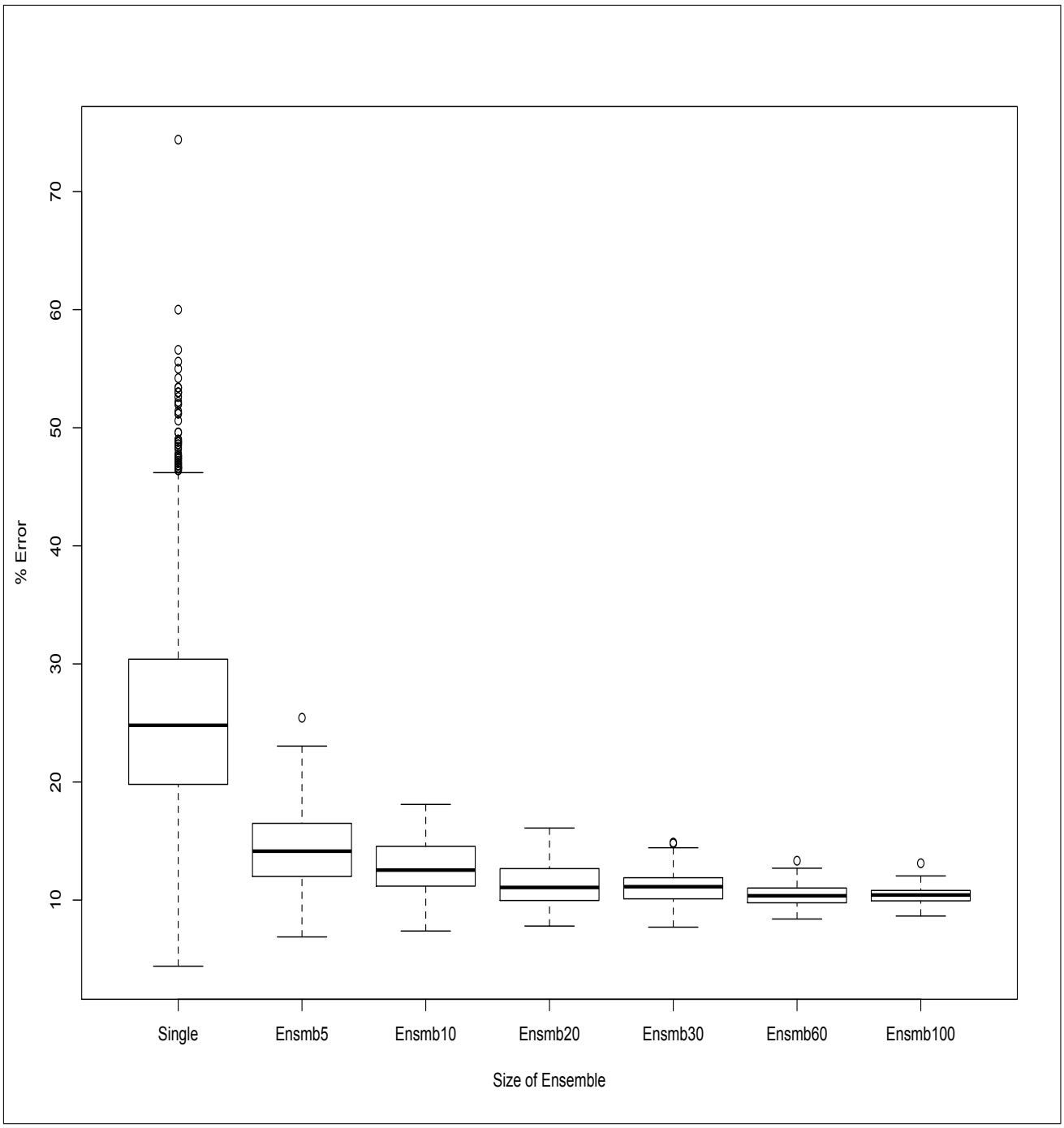


Figure 4.6: Boxplot of % Error across the whole distribution, versus increasing size of HLR ensemble (increasing number of parallel hidden layers).

4 Atom Micro-World **Light, Fan, Heater, Window** **Testing with neutral premiss "0 0 0 0"**

Training Set 88-0

Lvl	Pref Relation	State	Expect %	Samples	Output %	Samples	Output %	Samples	Output %	Samples	Output %	Samples	Output %	Accumulated	
0	(0), (1), (8), (9)	0	18.1%	11466	19.1%	11805	19.7%	11439	19.1%	11241	18.7%	10953	18.3%	56904	19.0%
		1	18.1%	11005	18.3%	10585	17.6%	10698	17.8%	10611	17.7%	11434	19.1%	54333	18.1%
1	(5), (13)	2	4.5%	3168	5.3%	3626	6.0%	3390	5.7%	3057	5.1%	3458	5.8%	16699	5.6%
		3	0.0%	32	0.1%	50	0.1%	26	0.0%	37	0.1%	45	0.1%	190	0.1%
2	(2), (10)	4	0.0%	37	0.1%	28	0.0%	39	0.1%	36	0.1%	36	0.1%	176	0.1%
		5	9.1%	4439	7.4%	4214	7.0%	4197	7.0%	3981	6.6%	4548	7.6%	21379	7.1%
3	(3), (4), (6), (7), (11), (12), (14), (15)	6	0.0%	103	0.2%	86	0.1%	59	0.1%	158	0.3%	70	0.1%	476	0.2%
		7	0.0%	65	0.1%	65	0.1%	109	0.2%	99	0.2%	82	0.1%	420	0.1%
		8	18.1%	11290	18.8%	11373	19.0%	11428	19.0%	11907	19.8%	11520	19.2%	57518	19.2%
		9	18.1%	10843	18.1%	10303	17.2%	10675	17.8%	11655	19.4%	10693	17.8%	54169	18.1%
		10	4.5%	3313	5.5%	3284	5.5%	3263	5.4%	2924	4.9%	3498	5.8%	16282	5.4%
		11	0.0%	49	0.1%	35	0.1%	29	0.0%	37	0.1%	23	0.0%	173	0.1%
		12	0.0%	20	0.0%	21	0.0%	60	0.1%	52	0.1%	38	0.1%	191	0.1%
		13	9.1%	4030	6.7%	4313	7.2%	4419	7.4%	4042	6.7%	3451	5.8%	20255	6.8%
		14	0.0%	87	0.1%	125	0.2%	79	0.1%	92	0.2%	64	0.1%	447	0.1%
		15	0.0%	53	0.1%	87	0.1%	90	0.2%	71	0.1%	87	0.1%	388	0.1%
			Error total =		8.6%		11.2%		9.2%		10.9%		10.7%	9.1%	± 1.12
			Error per state =		0.5%		0.7%		0.6%		0.7%		0.7%	0.6%	± 0.07

Training Set 96-1

Lvl	Pref Relation	State	Expect %	Samples	Output %	Samples	Output %	Samples	Output %	Samples	Output %	Samples	Output %	Accumulated	
0	(0), (1), (8), (9)	0	16.7%	8423	14.0%	7920	13.2%	8709	14.5%	7964	13.3%	8497	14.2%	41513	13.8%
		1	16.7%	8938	14.9%	8305	13.8%	8235	13.7%	8834	14.7%	9064	15.1%	43376	14.5%
1	(5), (13)	2	4.2%	3302	5.5%	3528	5.9%	2769	4.6%	3519	5.9%	3320	5.5%	16438	5.5%
		3	1.0%	1707	2.8%	1682	2.8%	1548	2.6%	1749	2.9%	1838	3.1%	8524	2.8%
2	(2), (10)	4	1.0%	1994	3.3%	1692	2.8%	2219	3.7%	1876	3.1%	1867	3.1%	9648	3.2%
		5	8.3%	3603	6.0%	4122	6.9%	3883	6.5%	4066	6.8%	3672	6.1%	19346	6.4%
3	(3), (4), (6), (7), (11), (12), (14), (15)	6	1.0%	1144	1.9%	1187	2.0%	1278	2.1%	1081	1.8%	1108	1.8%	5798	1.9%
		7	1.0%	1291	2.2%	1158	1.9%	1331	2.2%	1279	2.1%	1018	1.7%	6077	2.0%
		8	16.7%	7944	13.2%	8562	14.3%	8212	13.7%	8258	13.8%	8330	13.9%	41306	13.8%
		9	16.7%	9074	15.1%	8714	14.5%	8138	13.6%	8617	14.4%	8533	14.2%	43076	14.4%
		10	4.2%	2932	4.9%	3107	5.2%	2732	4.6%	3110	5.2%	2916	4.9%	14797	4.9%
		11	1.0%	1982	3.3%	1796	3.0%	1792	3.0%	1696	2.8%	1721	2.9%	8987	3.0%
		12	1.0%	1780	3.0%	1768	2.9%	2222	3.7%	2069	3.4%	1974	3.3%	9813	3.3%
		13	8.3%	3692	6.2%	4221	7.0%	4451	7.4%	3774	6.3%	3771	6.3%	19909	6.6%
		14	1.0%	1064	1.8%	1058	1.8%	1223	2.0%	1024	1.7%	1137	1.9%	5506	1.8%
		15	1.0%	1130	1.9%	1180	2.0%	1258	2.1%	1084	1.8%	1234	2.1%	5886	2.0%
			Error total =		28.1%		27.5%		28.2%		28.6%		27.4%	28.0%	± 0.50
			Error per state =		1.8%		1.7%		1.8%		1.8%		1.7%	1.7%	± 0.03

Figure 4.7: Example results for retrieving the preference relations, on two training sets, against the neutral premiss ("0 0 0 0") in the example 4-atom micro-world. The actual output distribution is a good match for the full preference relation / expected distribution.

Looking at the averaged error, between input and output distributions, only gives an overview of the performance of the machine against the logic. Figure 4.7 presents a complete output for the specific example 4-atom micro-world utilising two different training sets from this micro-world: '88-0' with least preferred states absent and '96-1' with least preferred states present. Five runs of ensemble machines with 60 parallel hidden layers each, are tested against the neutral premiss ("0 0 0 0"). The actual output can be compared with the expected values present in the training set in the left column. Comparing the expected and actual output frequencies state by state illustrates that the network representation is able to separate states correctly according to their preference, across the spectrum of preference levels. These results demonstrate that the Boltzmann machine faithfully represents the shape of the learned input distribution.

However, because of its stochastic error, the machine is only able to usefully separate 4 or 5 levels of preference when there are more than a dozen model states. In the most complex 6-atom micro-world, the machine was just able to maintain a separation between preference levels because the differences in input frequencies were close to its stochastic error. The large number of non-preferred states with frequencies of zero tend to dilute out the error on the few most preferred states. This favourable result would be reduced if the information theoretic divergence (Kullback and Leibler, 1951) had been used for measuring the difference between the distributions.

On the training sets where the least preferred patterns are present for just a single instance the machine has a larger error per state: on average 1.6% when present versus 0.9% where they are absent. Looking at training set '96-1' in Figure 4.7 for example, the reader can appreciate that the machine overestimates the probability of the least preferred input patterns and underestimates the frequency of the most preferred patterns, by 3 - 5%. This is typical of all the situations where the machine trains on the least preferred patterns. This movement of the sample toward a central mean is not an ideal characteristic for representation of the logic, but it may be biologically plausible.

4.3.3 Model Selection: a Discriminative Model

Recall that in Section 3.1: Logical Preliminaries, we suggest that a conclusion in supra-classical logic involves an ordered set of model states entailed by a premiss. Whilst retrieval of the complete probability distribution is statistically important to confirm learning, specific selection of model states based on a premiss is at the heart of the logic. This corresponds to the second component that we have identified as a requirement of predictive inference, under the rational consequence relation, Section 3.3.1: Logical

Requirements. This specific selection of model states provides evidence related to conditional probability *p(h|x)*; a discriminative model. We have proposed that this model selection corresponds to a set of ranked states or energy minima that are output by a Boltzmann machine.

It is difficult to provide a metric of the machine's performance in this context, for individual specific premises numerical analysis can be misleading. Although the expected training distributions are supplied, it is qualitatively more appropriate to compare the output distribution directly to the preference relation expected from the logic. Example results are provided in Figures 4.8 - 4.11: four examples in the 4-atom micro-world instance and two examples in 3, 5 and 6-atom micro-worlds. A much larger collection of examples of model selection based on individual premises in a range of micro-worlds is presented in our technical report (Blanchette *et al.*, 2015). In all the examples presented, the 'exponential' training sets utilised had the least preferred patterns present.

In summary, the results demonstrate that based on a partially specified premiss the Boltzmann machine is able to:

1. Select the appropriate model states entailed by the input premiss
2. Place these states in the correct preference ranking
3. Separate these states with a distance proportional to their preference level.

This type of ranking equivalence is termed "strong or magnitude persevering" (Pinkas, 1995), meaning that the equivalence maintains: the appropriate states themselves, the correct ordering on the states and the correct magnitude of separation between levels of preference. We provide some focused examples from Figures 4.8 - 4.11.

- Figure 4.8, a 3-atom micro-world, premiss "0 0 1"; the machine returns states 1, 5, 3 & 7: nicely separating states over the whole preference ranking.
- Figure 4.9, the 4-atom micro-world instance (the micro-world illustrated in Figure 3.1), premiss "0 -1 1 0"; the machine returns states 2 & 10 versus 3 & 11: correctly separating states at different levels in the low orders of preference.
- Figure 4.10, a 5-atom micro-world, premiss "0 -1 0 0 1"; the machine returns states 17, 1, 5 & 21: correctly separating states at different levels in the high orders of preference.

- Figure 4.11, a 6-atom micro-world, premiss "0 -1 0 -1 0 1"; the machine returns states 9 & 41: correctly placing these states of like ranking at the same level.

3 Atom Micro-World **Fan, Heater, Window**

Training Set 34-1

Testing with premises:

"0 0 1"

Lvl	Pref Relation	State	Expect %	Samples	Output %
0	(1)	0	0%	800	1%
		1	59%	29911	50%
1	(5)	2	0%	156	0%
		3	6%	6897	11%
2	(3), (7)	4	0%	467	1%
		5	29%	15478	26%
		6	0%	130	0%
		7	6%	6161	10%

"1 0 0"

Lvl	Pref Relation	State	Expect %	Samples	Output %
0	-	0	0%	191	0%
		1	0%	189	0%
1	(4), (5)	2	0%	46	0%
		3	0%	53	0%
2	(6), (7)	4	42%	21702	36%
		5	42%	21935	37%
		6	8%	6540	11%
		7	8%	9344	16%

Figure 4.8: Results for selection of model states from two partially specified premises in a 3-atom micro-world. In each case the input premiss entails a different ranked subset of states in the full preference relation, this expected distribution is compared to the actual output distribution.

4 Atom Micro-World **Light, Fan, Heater, Window**

Training Set 98-1

Testing with premises:

"-1 0 0 0"

Lvl	Pref Relation	State	Expect %	Samples	Output %
0	(0),(1)	0	31%	13634	23%
		1	31%	13712	23%
1	(5)	2	10%	5954	10%
		3	2%	3611	6%
2	(2)	4	2%	4142	7%
		5	20%	8534	14%
3	(3)(4),(6),(7)	6	2%	2327	4%
		7	2%	2879	5%
		8	0%	1164	2%
		9	0%	1643	3%
		10	0%	486	1%
		11	0%	352	1%
		12	0%	362	1%
		13	0%	842	1%
		14	0%	140	0%
		15	0%	218	0%

"0 0 0 1"

Lvl	Pref Relation	State	Expect %	Samples	Output %
0	(1),(9)	0	0%	106	0%
		1	28%	13480	22%
1	(5), (13)	2	0%	31	0%
		3	2%	2579	4%
2	-	4	0%	39	0%
		5	19%	10994	18%
3	(3),(7), (11, (15)	6	0%	26	0%
		7	2%	2830	5%
		8	0%	122	0%
		9	28%	13575	23%
		10	0%	29	0%
		11	2%	2204	4%
		12	0%	40	0%
		13	19%	11124	19%
		14	0%	20	0%
		15	2%	2801	5%

"0 -1 0 1"

Lvl	Pref Relation	State	Expect %	Samples	Output %
0	(1),(9)	0	0%	301	1%
		1	47%	25241	42%
1	-	2	0%	88	0%
		3	3%	3770	6%
2	-	4	0%	0	0%
		5	0%	223	0%
3	(3), (11)	6	0%	0	0%
		7	0%	46	0%
		8	0%	343	1%
		9	47%	26200	44%
		10	0%	75	0%
		11	3%	3504	6%
		12	0%	0	0%
		13	0%	196	0%
		14	0%	0	0%
		15	0%	13	0%

"0 -1 1 0""

Lvl	Pref Relation	State	Expect %	Samples	Output %
0	-	0	0%	553	1%
		1	0%	324	1%
1	-	2	42%	22555	38%
		3	8%	6273	10%
2	(2),(10)	4	0%	1	0%
		5	0%	1	0%
3	(3), (11)	6	0%	102	0%
		7	0%	54	0%
		8	0%	571	1%
		9	0%	291	0%
		10	42%	22713	38%
		11	8%	6378	11%
		12	0%	5	0%
		13	0%	1	0%
		14	0%	115	0%
		15	0%	63	0%

Figure 4.9: Results for selection of model states from four partially specified premises in the example 4-atom micro-world.

5 Atom Micro-World **Light, Fan, Air Conditioning, Heater, Window**

Training Set 158-1

Testing with Premises:

"-1-1 0 0 1"

Lvl	Pref Relation	State	Expect %	Samples	Output %
0	-	0	0%	281	0%
		1	62%	29912	50%
1	(1)	2	0%	44	0%
		3	4%	3989	7%
2	(5)	4	0%	141	0%
		5	31%	20485	34%
3	-	6	0%	23	0%
		7	4%	3367	6%
4	(3), (7)	8	0%	1	0%
		9	0%	413	1%
		10	0%	1	0%
		11	0%	79	0%
		12	0%	2	0%
		13	0%	289	0%
		14	0%	2	0%
		15	0%	71	0%
		16	0%	3	0%
		17	0%	468	1%
		18	0%	1	0%
		19	0%	57	0%
		20	0%	1	0%
		21	0%	304	1%
		22	0%	0	0%
		23	0%	48	0%
		24	0%	0	0%
		25	0%	5	0%
		26	0%	0	0%
		27	0%	0	0%
		28	0%	0	0%
		29	0%	4	0%
		30	0%	0	0%
		31	0%	9	0%

"0 -1 0 0 1"

Lvl	Pref Relation	State	Expect %	Samples	Output %
0	(17)	0	0%	188	0%
		1	23%	15086	25%
1	(1)	2	0%	30	0%
		3	1%	1717	3%
2	(5), (21)	4	0%	52	0%
		5	12%	8944	15%
3	-	6	0%	11	0%
		7	1%	1112	2%
4	(3), (7), (19), (23)	8	0%	0	0%
		9	0%	125	0%
		10	0%	0	0%
		11	0%	18	0%
		12	0%	0	0%
		13	0%	101	0%
		14	0%	0	0%
		15	0%	9	0%
		16	0%	205	0%
		17	47%	17909	30%
		18	0%	25	0%
		19	1%	2354	4%
		20	0%	51	0%
		21	12%	10183	17%
		22	0%	17	0%
		23	1%	1563	3%
		24	0%	8	0%
		25	0%	144	0%
		26	0%	0	0%
		27	0%	15	0%
		28	0%	0	0%
		29	0%	114	0%
		30	0%	0	0%
		31	0%	19	0%

Figure 4.10: Results for selection of model states from two partially specified premises in a 5-atom micro-world.

6 Atom Micro-World **Light, Fan, Air Conditioning, Heater, Open Fire, Window**

Training Set 232-1

Testing with premisses:

"0 -1-1-1 0 1"

Lvl	Pref Relation	State	Expect %	Samples	Output %
0	(33)	0	0%	169	0%
		1	43%	22081	37%
1	(1)	2	0%	86	0%
		3	2%	4464	7%
2	-	4	0%	0	0%
		5-8	0%	116	0%
3	-	9	0%	212	0%
		10-16	0%	47	0%
4	-	17	0%	225	0%
		18-31	0%	55	0%
5	(3), (35)	32	0%	183	0%
		33	53%	26967	45%
		34	0%	70	0%
		35	2%	4686	8%
		36	0%	2	0%
		37-40	0%	121	0%
		41	0%	209	0%
		42-48	0%	32	0%
		49	0%	215	0%
		50-63	0%	60	0%

"0 -1 0 -1 0 1"

Lvl	Pref relation	State	Expect %	Samples	Output %
0	(33)	0	0%	82	0%
		1	25%	12904	22%
1	(1)	2	0%	25	0%
		3	1%	2678	4%
2	(9), (41)	4	0%	1	0%
		5-8	0%	83	0%
3	-	9	19%	11684	19%
		10-16	1%	2312	4%
4	-	17	0%	98	0%
		18-31	0%	127	0%
5	(3), (11), (35), (43)	32	0%	88	0%
		33	32%	13676	23%
		34	0%	13	0%
		35	1%	2583	4%
		36	0%	0	0%
		37-40	0%	114	0%
		41	19%	11319	19%
		42-48	1%	1999	3%
		49	0%	93	0%
		50-63	0%	121	0%

Figure 4.11: Results for selection of model states from two partially specified premises in a 6-atom micro-world. The outputs on contiguous groups of least preferred states are accumulated: for example the samples on states 5 - 8, 10 - 16, 18 - 31 etc. are accumulated.

It is important to remember that the logic does not require an exact probability metric (Pinkas, 1995). With this in mind, we have provided the expected state distributions for these results as a means of qualitatively considering their logical correctness. The details of the whole preference ranking in these larger micro-worlds are available in our technical report (Blanchette *et al.*, 2015), but are not necessary considering the

expected distributions provided. The machine often succeeds in model selection when 'asked' to rank limited model states, where its performance against the neutral premiss with the entire ordering may have been marginal.

We have not presented the average error metrics with these results. There were results where the average error was large but the machine faithfully reproduced the preferences of the logic. On the other hand, there were a few results where the numerical error was small but the machine seemed to defy the logic. These last, unexpected results occurred in situations where selection of states at the same high level of preference, also required a secondary selection based on a dependent variable. An example of this can be seen in the next Section 4.3.4: Logical Properties (Figure 4.14, premiss "0 0 -1 0"), where states 0 & 8 have different output frequencies from 1 & 9. We believe, in these circumstances, the machine is performing a tie-break based on the atomic distribution of the dependent variable. This issue will be examined in a Chapter 5: Incongruence.

4.3.4 Logical Properties

We present two examples of properties provable by SCNM logic within the context of these micro-worlds. These specific properties were chosen because of their importance to the logical formalisation of defeasibility. We present them as evidence of our assertion that the Boltzmann machine will **always** provide support for the logical properties that hold in a micro-world: given the two required characteristics that we have identified and demonstrated (Sections 4.3.2: Preference Relation & 4.3.3: Model Selection). Part of the motivation for presenting them is to demonstrate that the third component of inference (Section 3.3.1: Logical Requirements), the comparison of selected model states, could be easily achieved.

Non-Monotonicity

Non-monotonicity is a refutation of the absolute truth of classical logic and can be seen as a foundational property of SCNM logic (Section 2.1). Yet in isolation the property might be regarded as to too weak (irrational). Recalling the definition for defeasible entailment:

$$\alpha \mid\sim \beta \quad \longleftrightarrow \quad Maximal\ Models[\alpha] \subseteq Models[\beta]$$

The principle of non-monotonicity states that:

$$\text{Given } \alpha \mid\sim \beta, \text{ then for some } \gamma,\ (\alpha \wedge \gamma) \not\mid\sim \beta$$

where semantically γ represents new information, a condition that is difficult to denote within the language. An example of this property in our 4-atom micro-world is:

$$F \mid\sim W \text{ but } (F \wedge H) \not\mid\sim W.$$

The Boltzmann machine is able to provide supporting evidence for this property, see Figure 4.12.

Models (F) "0 1 0 0"

State	Expect %	Samples	Output %
0	0%	198	0%
1	0%	267	0%
2	0%	100	0%
3	0%	49	0%
4	4%	5011	8%
5	39%	15643	26%
6	4%	3242	5%
7	4%	5556	9%
8	0%	173	0%
9	0%	222	0%
10	0%	52	0%
11	0%	59	0%
12	4%	5461	9%
13	39%	15787	26%
14	4%	3240	5%
15	4%	4940	8%

>>>

Models (W) "0 0 0 1"

State	Expect %	Samples	Output %
0	0%	106	0%
1	28%	13480	22%
2	0%	31	0%
3	2%	2579	4%
4	0%	39	0%
5	19%	10994	18%
6	0%	26	0%
7	2%	2830	5%
8	0%	122	0%
9	28%	13575	23%
10	0%	29	0%
11	2%	2204	4%
12	0%	40	0%
13	19%	11124	19%
14	0%	20	0%
15	2%	2801	5%

>>>

!!!

!!!

Models (F & H) "0 1 1 0"

State	Expect %	Samples	Output %
0	0%	8	0%
1	0%	8	0%
2	0%	316	1%
3	0%	155	0%
4	0%	245	0%
5	0%	573	1%
6	25%	13859	23%
7	25%	15167	25%
8	0%	8	0%
9	0%	4	0%
10	0%	301	1%
11	0%	129	0%
12	0%	270	0%
13	0%	502	1%
14	25%	13821	23%
15	25%	14634	24%

Figure 4.12: Non-Monotonicity. Output model states for three premises in the example 4-atom world (Light, Fan, Heater, Window). Where '>>>' supports entailment by the most preferred models and '!!!' does not support entailment of W by the most preferred models of F & H.

Consider the two premises F ("0 1 0 0") and W ("0 0 0 1"). The machine returns the maximally preferred models of F as states 5 & 13; these models are clearly a subset of the models of W (as returned by the machine in the middle panel). Whereas if we look at the premiss F $\wedge$ H ("0 1 1 0"), the maximally preferred models returned by the machine include 6 & 14; these model states are not a subset of the models of W (as returned by the machine in the middle panel).

Rational Monotonicity

Rational monotonicity is a more complex retraction and goes some way toward re-establishing classical entailment: providing a rational boundary to the disorder that would ensue from unchecked non-monotonicity. It allows unrestricted update in situations of independence between atoms: doxastic independence (Section 5.3.3: Discussion). Rational monotonicity is required by defeasible reasoning in addition to the six other preferential properties of KLM: Reflexivity, Left Logical Equivalence, Right Weakening, Cut, Cautious Monotonicity & Or (Lehmann and Magidor, 1992). The principal of rational monotonicity has two components:

$$\begin{aligned}&\text{Given } \alpha \mid\sim \beta, \text{ then:}\\ &\qquad \text{Either } (\alpha \wedge \gamma) \mid\sim \beta\\ &\qquad \text{Or if } \alpha \mid\sim \neg\gamma, \text{ then } (\alpha \wedge \gamma) \not\mid\sim \beta\end{aligned}$$

Examples of the two components of this property in our 4-atom micro-world are:

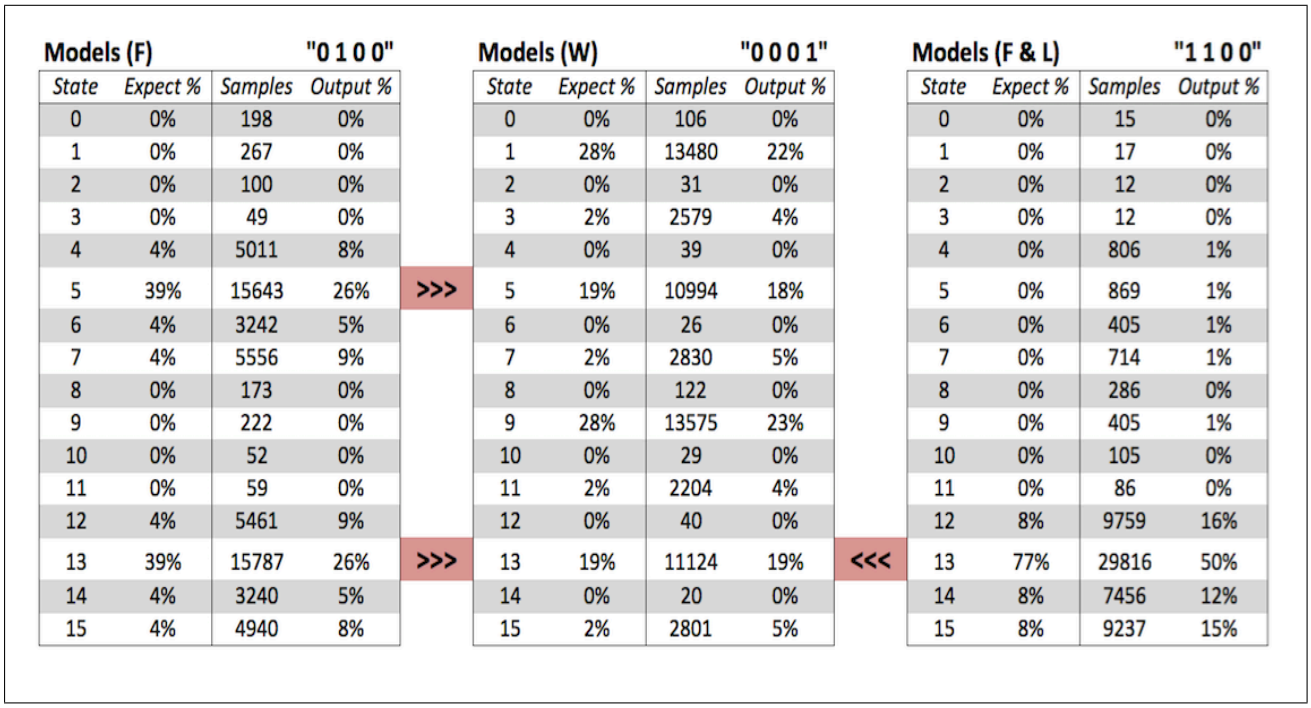
$$\begin{aligned}&F \mid\sim W \text{ and } (F \wedge L) \mid\sim W\\ &F \mid\sim W \text{ but } (F \wedge H) \not\mid\sim W, \text{ as } F \mid\sim \neg H\end{aligned}$$

For the first component, see Figure 4.13. From the machine output for the premises F ("0 1 0 0") and W ("0 0 0 1") we can see, as previously, the machine supports the defeasible entailment F $\mid\sim$ W. Considering the premiss F $\wedge$ L ("1 1 0 0"), we can see that the machine returns 13 as the maximally preferred model state which is a subset of the models of W (as returned by the machine, in the middle panel). The light (L) is a doxasticaly independent atom / proposition.

For the second component, we only need add evidence of F $\mid\sim \neg$H to the outputs already given in Figure 4.12, which already illustrate (F $\wedge$ H) $\not\mid\sim$ W. The outputs from the machine for the premiss $\neg$H ("0 0 -1 0") are shown in Figure 4.14, they include the maximally preferred models of premiss F ("0 1 0 0"), which are model states 5 & 13.

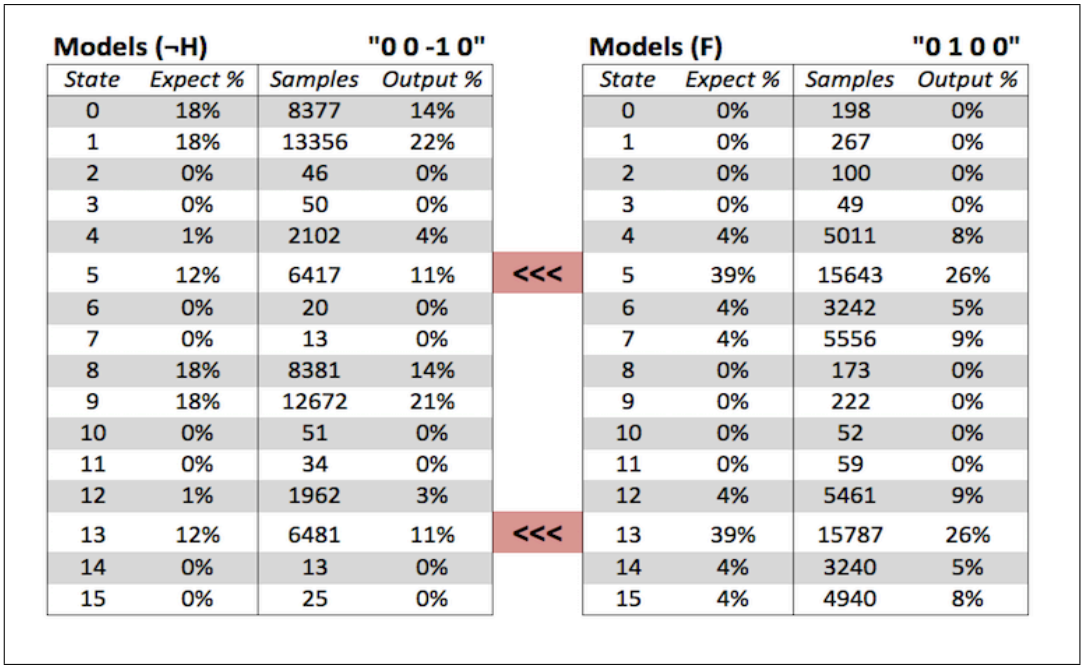
It is not intended that the exposition of these logical properties provide any additional evidence of veracity for the machine as a representation of the logic. However, they illustrate:

- The practical utility of the machine as a representation of the logic.
- That two important logical properties can be emulated by the machine, based on our assertion as a consequence of the shared preference relation.



Models (F)			"0 1 0 0"		Models (W)			"0 0 0 1"		Models (F & L)			"1 1 0 0"
State	Expect %	Samples	Output %		State	Expect %	Samples	Output %		State	Expect %	Samples	Output %
0	0%	198	0%		0	0%	106	0%		0	0%	15	0%
1	0%	267	0%		1	28%	13480	22%		1	0%	17	0%
2	0%	100	0%		2	0%	31	0%		2	0%	12	0%
3	0%	49	0%		3	2%	2579	4%		3	0%	12	0%
4	4%	5011	8%		4	0%	39	0%		4	0%	806	1%
5	39%	15643	26%	>>>	5	19%	10994	18%		5	0%	869	1%
6	4%	3242	5%		6	0%	26	0%		6	0%	405	1%
7	4%	5556	9%		7	2%	2830	5%		7	0%	714	1%
8	0%	173	0%		8	0%	122	0%		8	0%	286	0%
9	0%	222	0%		9	28%	13575	23%		9	0%	405	1%
10	0%	52	0%		10	0%	29	0%		10	0%	105	0%
11	0%	59	0%		11	2%	2204	4%		11	0%	86	0%
12	4%	5461	9%		12	0%	40	0%		12	8%	9759	16%
13	39%	15787	26%	>>>	13	19%	11124	19%	<<<	13	77%	29816	50%
14	4%	3240	5%		14	0%	20	0%		14	8%	7456	12%
15	4%	4940	8%		15	2%	2801	5%		15	8%	9237	15%

Figure 4.13: Rational Monotonicity, Part 1.



Models (¬H)			"0 0 -1 0"		Models (F)			"0 1 0 0"
State	Expect %	Samples	Output %		State	Expect %	Samples	Output %
0	18%	8377	14%		0	0%	198	0%
1	18%	13356	22%		1	0%	267	0%
2	0%	46	0%		2	0%	100	0%
3	0%	50	0%		3	0%	49	0%
4	1%	2102	4%		4	4%	5011	8%
5	12%	6417	11%	<<<	5	39%	15643	26%
6	0%	20	0%		6	4%	3242	5%
7	0%	13	0%		7	4%	5556	9%
8	18%	8381	14%		8	0%	173	0%
9	18%	12672	21%		9	0%	222	0%
10	0%	51	0%		10	0%	52	0%
11	0%	34	0%		11	0%	59	0%
12	1%	1962	3%		12	4%	5461	9%
13	12%	6481	11%	<<<	13	39%	15787	26%
14	0%	13	0%		14	4%	3240	5%
15	0%	25	0%		15	4%	4940	8%

Figure 4.14: Rational Monotonicity, Part 2. Output model states for premises in the example 4-atom world (Light, Fan, Heater, Window). Where '>>>' supports entailment by the most preferred models.

- The ease with which output model states from two separate sentences could be compared by an 'observer machine', to complete the requirements of inference.

4.3.5 Summary

We have identified three requirements necessary for a network to represent inference under the rational consequence relation in SCNM logic (Section 3.3.1). The experimental results do not constitute a mathematical proof but, in the context of the scientific method, they provide compelling evidence that the first two of the identified logical requirements are fulfilled by the Boltzmann machine. These requirements constitute the greatest challenge to such a machine representation. Seen from the standpoint of the machine they are:

1. The ability to learn an input probability distribution constituting a preference relation.

2. The ability to retrieve one-to-many mappings constituting appropriate model selection based on a partially specified premiss.

In the experimental context of these moderate sized logical micro-worlds, the Boltzmann machine is a faithful representation. It is able to learn a preference relation with numerical accuracy: average error per state $\leq 2\%$, and select appropriate model states based on the limited information available in a premiss. It maintains a "strongly magnitude preserving equivalence". This hypothesis is supported theoretically by the work: of Pinkas (1995) in specific SNNs and indirectly by SRL utilising Markov models. We assert that these two requirements are sufficient for the machine to **always** follow the principles of this logic. In a broader context of common-sense reasoning, there are additional biological pre-requites for any network hoping to offer some insight into human cognition.

It should not be surprising that the Boltzmann machine is able to retrieve the results of its probabilistic learning since, at equilibrium, the Boltzmann distribution around which the machine is based, represents the likelihood of the learned states. Yet we suggest that these properties; the ability to retrieve a ranked set of output states characterising a generative model and the retrieval one-to-many relationships in the context of a discriminative model, are rare among neural networks. Further, these properties are fundamental requirements of any network aspiring to embody inference in SCNM logic.

4.4 Discussion on Human Cognition

A comprehensive discussion of human cognition is beyond the scope and space of this thesis. Particularly, there have been advances in neurobiology that are producing results in modelling human cortical anatomy and starting to explore the physical correlates of memory. An example is the Blue Brain Project, which has currently modelled a section of cortical tissue, implementing different cell types and connection topologies and aims towards a whole brain (Markram, 2014). On a different front, new psychological evidence is mounting of social co-evolutionary processes that have shaped human cognition over the 2 million years since the first emergence of Homo sapien ancestors (Heyes, 2012). There is a complete spectrum of research from the highest abstract theoretical level to the most finely grained molecular biology. The discussion that follows concerns a 'general theory' of cognition and the place of the Boltzmann machine. We attempt to avoid philosophical discussion and concentrate on the physical implementation of cognition.

Firstly, we strongly support the perspective of psychology that: "the entire enterprise (of artificial intelligence) has human cognition as its foundation" (Pelletier and Elio, 1997). Is there currently an artificial agent that might be categorised as cognitive? Various arguments can be made regarding the threshold of cognition (Turing, 1950; Saygin, Cicekli, and Akman, 2000). There are machines able to complete specific rational tasks more efficiently than a human, particularly in the competitive context of games (Hsu, 2002). However, the scope of these tasks is limited and the machines do not arise de-novo: they are procedurally programmed by human intelligence. Better candidates for artificial cognitive agents, might be the robotic creatures evolved by artificial processes resembling biology: possessing small artificial neural networks as brains, trained to explore their environment (Zufferey, Floreano, van Leeuwen, and Merenda, 2002). However, these agents lack 'common-knowledge' and could they be said to possess auto-epistemic reasoning?

Secondly, what evidence is available for the basis of human cognition? Microscopically the brains of all living creatures consist of neural networks: massively connected collections of individual nodal cells (Striedter, 2016, Chapter 1: Nervous System Organisation, p2-30). While the exact mechanics of these networks are still in doubt and are likely to be varied, they provide a distributed representation of the information which they learn (Striedter, 2016, Chapter 2: Computing with Neurons, p31-70): localised coding is highly unlikely (Bowers, 2009, 2011) and there is certainly no central

processing unit. Procedural programming in the form of a finite automaton or universal Turing machine does not exist in nature.

Where does that lead us in terms of a theoretical description of cognition? Whilst abstractions can be of value in providing potential representational models, conjectures about cognition should have a possible basis in neural network implementation. Harnad's tripartite level theory of cognition (Harnad, 1987, 1990) is a widely cited example (Gärdenfors, 2004; Labuschagne and Heidema, 2010). It postulates three levels of processing: an iconic level of representation at the sensory boundary with the environment, a categorical level of invariance detection and a higher symbolic level where actual reasoning takes place. Harnad is reluctant to credit any role for a neural network implementation at the symbolic level.

> "a cognitive theory must stand on its own merits ... to constrain a cognitive theory to account for behaviour in a brain-like way is hence premature ... connectionist networks, because they are not symbol systems, do not have the systematic semantic properties that many cognitive phenomena appear to have." Harnad (1990)

We propose that a cognitive theory needs support from the domains of both psychology and neurobiology, because the primary context of cognition is human biology. Harnad's views of neural networks as purely syntactic structures, unable to learn meaning, are contradicted by the seminal experiments of Frank *et al.* (2009). Frank demonstrated that artificial neural networks can represent the physical semantics of an environment including predicate relationships, even when trained on a syntactically incomplete language. Proponents of Harnad's supposition have suggested that "any neural network implementing symbolic manipulation would be irrelevant". Such statements are difficult to comprehend; they amount to a denial that reasoning can take place in a neural network. Certainly neural networks which implicitly implement symbolic manipulation are not irrelevant; they are the only structures currently known to instantiate cognition.

Radermacher describes a scheme for human cognition on four different levels. The first three levels seem very similar to Harnad's: sensory, feature detection and knowledge representation (symbolic), with addition of a forth higher level of theoretic representation. However, Radermacher's research is in the practical context of experimental robotics and he acknowledges that these levels are abstractions: "In humans, the logical and symbolic functions of the brain are realised within a biological neural network." (Radermacher, 1996)

Further, recent research questions even the separation of sensory/motor and symbolic levels, proposing the sensory/motor representation is intimately connected to language and symbolic representation (Knott, 2012). The process of abstraction should not be confused with the actual implementation of cognition. Currently neural networks are the only practical models available for the implementation of cognition. That is not to say that abstracted views of knowledge representation are not important in helping us understand cognition. The entire purpose of this thesis is to provide a link between one abstract representation of cognition, SCNM logic and another slightly less abstract model, a Boltzmann neural network: to demonstrate that these models are complementary.

There are other possible representational models for cognition, which this thesis has not examined. One example is Gärdenfors's conceptual spaces (Gärdenfors, 2004). They constitute more than just the topological connections of the visual and motor cortex. Theoretical concepts can also be represented geometrically: colours, taste etc. Latent semantic analysis in linguistics also relies on a geometric similarity of meaning, as indicated by psychological experiments involving word substitution. Potentially in terms of SCNM logic, conceptual dimensions may represent features within the default rule; the basis of the preference ordering on model states. Such a representation of a linear conceptual space is demonstrated by the hidden node in the Boltzmann machine, Sections 5.2.4: Network Analysis for incongruence and 6.3.5: Discussion of Results for belief revision.

4.4.1 Place of the Boltzmann Machine

The place of the Boltzmann machine representation in this broader context of human cognition is uncertain. We do not propose that a Boltzmann like network is the sole cognitive mechanic of common-sense reasoning. Even within the limited scope of this thesis, the Boltzmann network needs to be extended to capture the third requirement of inference: the comparison of output model states (Section 3.3.1: Logical Requirements). However, the machine's stochastic activation functions and probabilistic learning are likely requirements for distributional representation of the environment, which is the statistical basis for logical preference (Hinton *et al.*, 2000). We have demonstrated, in the earlier sections of this chapter, that these characteristics, learning a generative model and selection of model states ('strong equivalence'), are the prerequisites for representing SCNM logical inference. It is not a biological disadvantage that these specific characteristics make learning slightly slower. So the Boltzmann machine or

equivalent functionality could be seen as a necessary component of a larger network structure, implementing rational consequence.

Looking at the individual characteristics of the Boltzmann machine for biological plausibility (Section 3.3.2) there are other favourable properties, particularly in relation to Hebbian learning. The Boltzmann machine is one of the few networks that relies on cross-firing statistics: it is a truely remarkable algorithm that directs the adaptation of the entire network based solely on local information. It is likely that this adaptation, via cross-firing, has a simple correlate in potentiation across a synapse (Hebb, 1949) and is related to long term memory (Sejnowski and Destexhe, 2000; Stenning and Van Lambalgen, 2008, Chapter 8: Implementing Reasoning in Neural Networks). It may be physically correlated with growth and 'pruning' of the dendritic tree. The division of the Boltzmann learning algorithm into two phases, one requiring no external input into the network, is also extremely biologically plausible. Most complex organisms sleep. Sleep may be the physical correlate of a dual phase learning algorithm required for weight update in the consolidation memory (Sejnowski and Destexhe, 2000).

Simulated annealing would seem to be the most biologically implausible of the machine's properties. However, from an optimisation point of view, simulated annealing is the most comprehensive and adaptable of the threshold optimisation algorithms (Aarts and Korst, 1990). It has the ability to overcome irregularities in the solution space, which are typical of real biological tasks. In fact, it is implausible that a deterministic variant of threshold optimisation, strictly limited to gradient descent, would have evolved as the primary means of representing a biological world. We can only speculate about the neurobiological basis of annealing at a molecular level, because of our incomplete understanding of complex biological mechanisms. However, simulated annealing like cross-firing, has the advantage that it only requires local implementation. Recurrent firing would only need to produce local changes in neuro-modulator chemicals that alter excitability across a synapse. Such rapid (time scale of seconds) synaptic chemical changes have been proposed by other authors (Stenning and Van Lambalgen, 2008; von der Malsburg and Willshaw, 1981) as a basis of fast functional linkage and short term memory.

Closer to the neurobiology, a spiking version of the Boltzmann machine was first proposed by Hinton and Brown (1999). Although, as discussed in Section 4.1: Network Selection, the restricted machine was chosen as a less complex model. The restricted machine has no connectivity within the hidden layer and converges to minima in one-step reconstructions without the need for simulated annealing: *contrastive di-*

vergence. While still Hebbian in nature, its lack of intra-layer inhibition makes the restricted machine less biologically plausible but more computationally efficient. The good performance of this spiking Boltzmann model has been demonstrated in visual recognition tasks (Courville, Bergstra, and Bengio, 2011; Neftci, Das, Pedroni, Kreutz-Delgado, and Cauwenberghs, 2014). Further, the paper by Neftci *et al.* (2014) unveiled an online, asynchronous, event-driven version of the learning algorithm, able to be implemented in neuromorphic systems: very large scale integrated systems (with up to 2 million synapses) with circuits of integrate and fire neurons which utilise spike-time dependent plasticity. These authors make some important statements about knowledge representation:

> "... brains deal with uncertainty in their environments by encoding and combining probabilities optimally: such computations are at the core of cognitive function. ... neural sampling theory postulates that spiking activity of the neurons encodes samples of an underlying probability distribution, ... consistent with the behaviour of biological neurons in implementing Markov Chain Monte Carlo (MCMC) sampling" " ... because restricted Boltzmann machines are generative models, they can act as classifiers ... and carry out probabilistic inference, ... performing MCMC sampling of a Boltzmann distribution." (Neftci *et al.*, 2014).

The authors' opinions support the proposal that: knowledge is represented in stochastic neural networks by the partition function that characterises the network energy at equilibrium.

Chapter 5

Incongruence

Information contained in a training set of vectors, learned by a Boltzmann machine, can be interpreted in two different ways, either as the frequency of whole states or the frequency of the activations of individual atomic units. This contrasting distributional information is illustrated in Section 5.1: Dual Distributions, by reference to simple 2-atom micro-worlds. The two interpretations of the training set may make very different predictions about the output behaviour of a given Boltzmann machine. For some sampled populations the dual distributions will be similar: congruent. However for most sampled populations the interpretations of the dual distributional information will be conflicting: incongruent.

Modifications in the architecture of the Boltzmann machine can select for one type of distributional information over the other; favouring either *state or atomic distributions*[1]. In Section 5.2: Architectural Selection, we define two varieties of Boltzmann machine and provide evidence that: a *'hidden layer rich' (HLR)* machine favours the atomic distribution whilst an *'input layer poor' (ILP)* machine favours the state distribution. An attempt is made to reconcile these two behaviours by analysing the nature of the Boltzmann machine internal representation utilising a probabilistic model.

The traditional view of inference in supra-classical, non-monotonic logic only considers the state distribution. However, there are logics which attempt to utilise typicality. The relationship between typicality and the atomic frequency distribution in a training set of vectors is discussed in Section 5.3. We argue, by counter-example, that the common assumption of typicality as represented by 'most preferred model semantics' is incorrect. Further, we discuss problems for the logical implementation of typicality and offer an alternative definition based on atomic typicality.

[1]Terms emphasised at first occurrence, are defined in the glossary.

5.1 Dual Distributions

The most obvious distributional information available in a training set of vectors is the whole state frequency. This state ranking can be seen as analogous to the preference ranking in SCNM logic. The second, less obvious distribution, is the activation frequency of individual atomic components. The individual frequency of atomic activations across the whole population can be used to derive an atomic distribution, based on the product of these individual frequencies within a state.

$$S_i = \prod_k a_k^i$$

Here the atomic frequency of state i, in a world with k atomic components, is equal to the product of its k individual atomic activations a^i.

The example we will use to illustrate this contrast between the state and atomic distributions is framed in the paradigm of symbolic logic, to reinforce the relationship between the logic and the training environment of the machine. Figure 5.1 shows three training sets (from a larger collection presented in Figure 5.3) related to different preference relations in possible 2-atom micro-worlds:

- Training Set B18: The expected state frequency of state 0 is 44% (8 patterns of the total 18). The high order bit is false (state 0) in 16 patterns out of the total 18, an atomic probability of 0.89. The low order bit is false (state 0) in 9 patterns out of the total 18, an atomic probability of 0.50. Therefore, the expected atomic frequency of state 0 is: 0.89×0.50, also = 44%.

- Training Set C18: The expected state frequency of state 0 is again 44%. The high order bit is false in 9 patterns out of the total 18, an atomic probability of 0.50. The low order bit is false in 9 patterns out of the total 18, an atomic probability of 0.50. Therefore, in this training set, the expected atomic frequency of state 0 is only: $0.50 \times 0.50 = 25\%$, not 44%.

- Training Set J41: The expected state frequency of state 0 is 39% (16 patterns out of the total 41). The high order bit is false in 32 patterns out of the total 41, an atomic probability of 0.78. The low order bit is false in 24 patterns out of the total 41, an atomic probability of 0.59. Therefore, in this training set, the expected atomic frequency of state 0 is: $0.78 \times 0.59 = 46\%$, not 39%.

Three Different Preference Relations in 2-Atom Micro-Worlds

Training Set B18: Congurent

Lvl	Preference Relation	State	Patterns #	State Distrb.	Atomic Distrb.
0	(0), (1)	0	8	44%	44%
		1	8	44%	44%
1	(2), (3)	2	1	6%	6%
		3	1	6%	6%
	Total Patterns =		18		

Atomic Probablities	
"1 0"	0.11
"-1 0"	0.89
"0 1"	0.50
"0 -1"	0.50

Training Set C18: Incongurent

Lvl	Preference Relation	State	Patterns #	State Distrb.	Atomic Distrb.
0	(0), (3)	0	8	44%	25%
		1	1	6%	25%
1	(1), (2)	2	1	6%	25%
		3	8	44%	25%
	Total Patterns =		18		

Atomic Probablities	
"1 0"	0.50
"-1 0"	0.50
"0 1"	0.50
"0 -1"	0.50

Training Set J41: Incongurent

Lvl	Preference Relation	State	Patterns #	State Distrb.	Atomic Distrb.
0	(0), (1)	0	16	39%	46%
1	(2)	1	16	39%	32%
2	(3)	2	8	20%	13%
		3	1	2%	9%
	Total Patterns =		41		

Atomic Probablities	
"1 0"	0.22
"-1 0"	0.78
"0 1"	0.41
"0 -1"	0.59

Figure 5.1: An example of incongruent dual distributions. Three different preference relations in 2-atom micro-worlds. The atomic distribution of each state (pattern) is calculated from the product of the appropriate atomic probabilities.

Similar calculations can be made for each state in the three training sets to produce the atomic distributions given in Figure 5.1. It can be seen that, although two of the preference relations in sets B18 and C18 appear to be similar, one preference relation, training set B18, is congruent in its state and atomic distributions; whereas the other, C18 is not. Likewise for training set J41, the order of the most preferred states is very different between the state and atomic distributions. We believe that this specific concept of incongruence: the calculation of an expected state distribution based on global atomic frequencies and its potential inconsistency with observed state frequencies, has not been explicitly reported before.

A survey of 2-atom micro-worlds shows that congruent environments with consistent distributional information are rare. Only 15 out of the total possible 65 2-atom micro-worlds are congruent. There are more than 50,000 permutations on 3-atom micro-worlds and trillions of possible permutations on 4-atom micro-worlds. We speculate that, because of increasing complexity, congruency should become progressively less likely in the larger systems.

5.2 Architectural Selection

The experimental results presented in this section provide evidence that the Boltzmann machine is capable of representing both types of distributional information: that by variation in architecture, the machine can favour one type of distribution over another. The results have been collected from two separate lines of experimentation:

- Architectural modification of the Boltzmann machine to favour the state distribution was completed in two incongruent 4-atom micro-worlds. These two worlds are illustrated in Figure 5.2.
- Two different extremes of Boltzmann machine architecture, hidden layer rich (HLR) and input layer poor (ILP) were defined and subsequently tested against a selection of 26 possible 2-atom micro-worlds, with different preference relations. These are illustrated in Figure 5.3.

5.2.1 HLR: a Mixed-Atomic Distribution

The modified Boltzmann machine utilised for the representation of SCNM logic in Chapter 4 was fully inter-layer (between layers) connected, although direct connections between input and output were removed. It had intra-layer (within layer) connections only in the hidden layer, with an ample number of hidden nodes. Figure 4.1 illustrates this architecture in a 4-atom micro-world, which we have called an HLR Boltzmann machine: **H**idden **L**ayer connected and hidden layer **R**ich.

Such a machine performed well in most of the example worlds we tested (Blanchette *et al.*, 2015). However, in hindsight, the reservation which was expressed concerning the large number of possible worlds was pertinent. By chance, or unconscious selection, many of the worlds we previously chose to examine were congruent, in terms of their distributional information.

Example Incongurent 4-Atom Micro-Worlds

Training Set K80

Lvl	Preference Relation	State	Patterns #	State Distrb.	Atomic Distrb.
0	(2), (10)	0	8	10.0%	14.1%
		1	8	10.0%	7.6%
1	(0), (1), (8), (9)	2	16	20.0%	12.7%
		3	1	1.3%	6.9%
2	(5), (13)	4	1	1.3%	3.0%
		5	4	5.0%	1.6%
3	(3), (4), (6), (7)	6	1	1.3%	2.7%
	(11), (12), (14), (15)	7	1	1.3%	1.5%
		8	8	10.0%	14.1%
		9	8	10.0%	7.6%
		10	16	20.0%	12.7%
		11	1	1.3%	6.9%
		12	1	1.3%	3.0%
		13	4	5.0%	1.6%
		14	1	1.3%	2.7%
		15	1	1.3%	1.5%
Total Patterns =			80		

Atomic Probablities	
"1 0 0 0"	0.50
"-1 0 0 0"	0.50
"0 1 0 0"	0.18
"0 -1 0 0"	0.83
"0 0 1 0"	0.48
"0 0 -1 0"	0.53
"0 0 0 1"	0.35
"0 0 0 -1"	0.65

Training Set L88

Lvl	Preference Relation	State	Patterns #	State Distrb.	Atomic Distrb.
0	(1), (2), (10)	0	4	4.5%	11.6%
		1	16	18.2%	11.6%
1	(5), (9), (13)	2	16	18.2%	8.8%
		3	1	1.1%	8.8%
2	(0), (8)	4	1	1.1%	3.9%
		5	8	9.1%	3.9%
3	(3), (4), (6), (7)	6	1	1.1%	2.9%
	(11), (12), (14), (15)	7	1	1.1%	2.9%
		8	4	4.5%	9.7%
		9	8	9.1%	9.7%
		10	16	18.2%	7.4%
		11	1	1.1%	7.4%
		12	1	1.1%	3.2%
		13	8	9.1%	3.2%
		14	1	1.1%	2.5%
		15	1	1.1%	2.5%
Total Patterns =			88		

Atomic Probablities	
"1 0 0 0"	0.46
"-1 0 0 0"	0.55
"0 1 0 0"	0.25
"0 -1 0 0"	0.75
"0 0 1 0"	0.43
"0 0 -1 0"	0.57
"0 0 0 1"	0.50
"0 0 0 -1"	0.50

Figure 5.2: Examples of two different preference relations in incongruent 4-atom micro-worlds. The atomic distribution of each pattern is calculated from the product of the appropriate atomic probabilities.

Figure 5.3: A survey of 26 preference relations in 2-atom micro-worlds. The results of architectural selection in the most incongruent of these worlds are examined in the discussion that follows, Figures 5.5 & 5.8.

There were some indications, even within these congruent worlds, that the HLR machine favoured a mixed distribution. Where the selection of model states based on a partially specified premiss involved a dependent variable (atom), the machine would select the most preferred states based on the conditional probability of the dependent atom.

For example in Figure 5.4 given the premiss "0 0 -1 0", the probability of the low order bit (dependent variable) being on is 0.61. This makes states 1 & 9 conditionally more probable than 0 & 8, even though the observed frequencies of these whole state patterns were equal. The machine selects these states more frequently. It can be appreciated from this example that information contained in a partially specified premiss may produce a discrepancy between the state and atomic distributions of the selected model states, despite the two distributions in the full training set being congruent.

Conditional Probability Based on a Partially Specified Premiss

Tested with Premiss "0 0 -1 0"

Lvl	Preference Relation	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(0), (1), (8), (9)	0	18.3%	14.4%	8377	14.0%
		1	18.3%	22.6%	13356	22.3%
1	(5), (13)	2	0.0%	0.0%	46	0.1%
		3	0.0%	0.0%	50	0.1%
2	-	4	1.2%	5.1%	2102	3.5%
		5	12.2%	7.9%	6417	10.7%
3	(4), (12)	6	0.0%	0.0%	20	0.0%
		7	0.0%	0.0%	13	0.0%
		8	18.3%	14.4%	8381	14.0%
		9	18.3%	22.6%	12672	21.1%
		10	0.0%	0.0%	51	0.1%
		11	0.0%	0.0%	34	0.1%
		12	1.2%	5.1%	1962	3.3%
		13	12.2%	7.9%	6481	10.8%
		14	0.0%	0.0%	13	0.0%
		15	0.0%	0.0%	25	0.0%

Atomic Probablities	
"1 0 0 0"	0.50
"-1 0 0 0"	0.50
"0 1 0 0"	0.26
"0 -1 0 0"	0.74
"0 0 1 0"	0.00
"0 0 -1 0"	1.00
"0 0 0 1"	0.61
"0 0 0 -1"	0.39

Figure 5.4: The observed distribution of the output of an HLR Boltzmann machine in a 4-atom micro-world given a partially specified premiss, demonstrating conditional probability.

Figure 5.5: The output from an HLR Boltzmann machine in a selection of incongruent 2-atom micro-worlds. Errors are calculated relative to the state distribution only.

When tested on our survey of 2-atom micro-worlds the HLR modified Boltzmann machine favoured a mixed or atomic distribution. This is illustrated in Figure 5.5, which shows a subset of 2-atom systems with preference relations that result in incongruent distributional information. For example in Figure 5.5, training set F25 has three states: (decimal) 0, 1 & 2 at equal state frequency but both the high and low order bits are more likely to be off, so from the atomic probabilities state 0 is more likely than state 1 or 2. By contrast, training set G25 has states: 1, 2 & 3 at equal state frequency but both the high and low order bits are more likely to be on, so from the atomic probabilities state 3 is preferred over states 1 & 2. In both these cases the HLR machine utilises the additional information learned from the atomic distribution to adjust preference levels, particularly for the most preferred states.

These machine results are not in keeping with the traditional view of inductive inference. SCNM logic traditionally only considers whole state frequencies and places no value on the individual atomic probabilities. In this respect, the HLR machine is a faithful representation of the logic **only** in congruent worlds. This failure of our initial Boltzmann machine to fully represent the logic in incongruent worlds prompted us to look for other modifications, which might better suit the machine to representing the traditional view of logical inference.

5.2.2 ILP: the State Distribution

Hinton's original Boltzmann machine (Hinton *et al.*, 1984, 1985) was fully connected; intra-layer and inter-layer. We had previously preserved the inter-layer connections (except for direct input to output), but retained only the hidden intra-layer connections for the sake of biological similarity. The effect of changing the intra-layer connectivity was investigated in relation to the two incongruent 4-atom micro-worlds introduced in Figure 5.2.

Table 5.1 shows the accumulated errors across 4 different training sets in the two incongruent 4-atom micro-worlds. The results indicate a subtle improvement in learning of the state distribution when only the input layer is intra-connected. Intra-layer connection of any other combination of layers (particularly the hidden layer) moved the representation away from the state distribution toward the atomic distribution.

This input modification of a Boltzmann machine was then re-tuned across all of its other parameters: number of hidden nodes, training epochs, learning rate, momentum, annealing temperature range, cycles per temperature point, sampling cycles, changes in activation function and methods of weight initialisation. Only changing the number

Intra-Layer Connections in a Boltzmann Machine			
Single Layer	*Error*	*Multiple Layer*	*Error*
Restricted BM	38%	Input + Hidden	37%
Input Only	30%	Input + Output	40%
Hidden Only	47%	Hidden + Output	48%
Output Only	42%	Fully	40%

Table 5.1: The result of changing the intra-layer connectivity of a Boltzmann machine with respect to the error relative to the state distribution: in 4-atom micro-worlds with incongruence.

of hidden layer nodes produced significant improvement. As illustrated in Figure 5.6 the accumulated error in the representation of the state distribution was improved by reducing the hidden layer to a single node. What the figure does not show is the corresponding deterioration in the representation of the atomic distribution. Figure 5.7 illustrates the final architecture in a 4-atom micro-world, for an ILP Boltzmann machine: **I**nput **L**ayer connected and hidden layer **P**oor.

Tuning an 'Input' Boltzmann Machine in Incongurent 4-Atom Micro-Worlds

Hidden Nodes \ *Epochs*	*500*	*1000*	*2000*	*5000*	*10000*	*20000*
0	75.3%	76.1%	77.6%	82.0%	87.4%	102.3%
1	15.2%	15.4%	11.4%	**8.9%**	10.3%	13.4%
2	19.1%	20.6%	17.7%	17.2%	15.3%	17.3%
3	21.8%	23.1%	20.1%	15.9%	14.2%	16.6%
4	25.7%	27.7%	25.6%	20.2%	15.9%	18.0%
6	30.4%	33.3%	31.2%	23.6%	26.2%	31.8%
8	33.9%	35.6%	32.9%	28.9%	37.3%	51.1%
10	36.6%	38.8%	35.4%	33.5%	46.8%	63.8%

Figure 5.6: Partial tuning of an input layer intra-connected Boltzmann machine in 4-atom micro-worlds with incongruence. The results are accumulated errors across 4 training sets relative to the state distribution only.

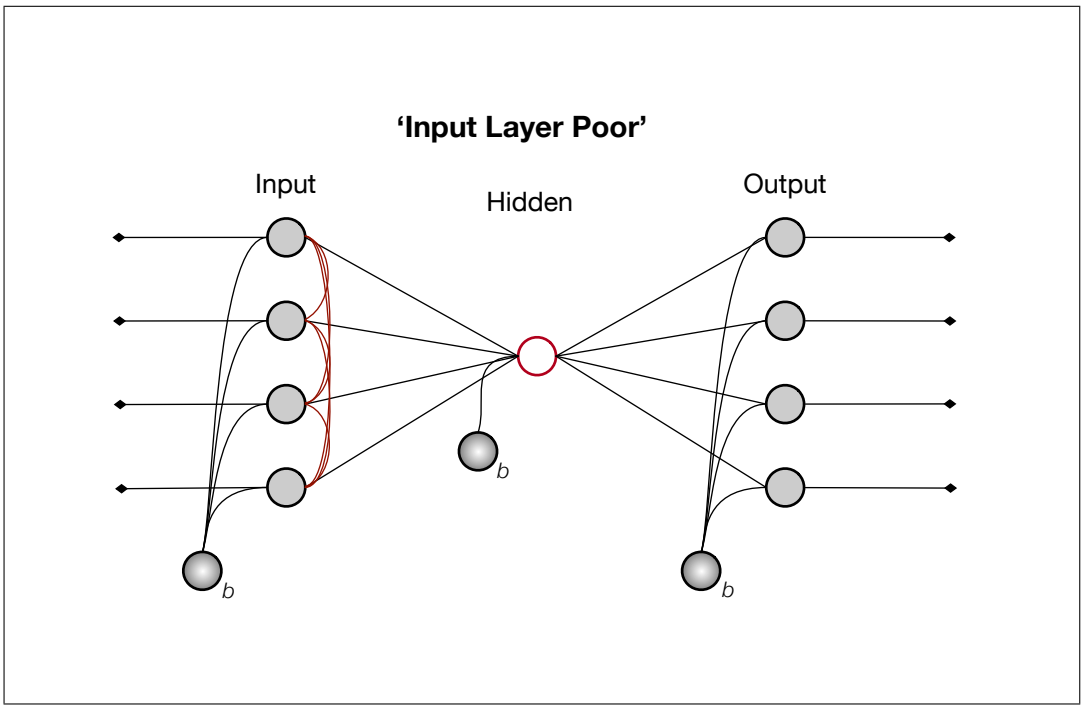


Figure 5.7: An example architecture of an ILP modified Boltzmann machine; in a 4-atom micro-world.

The 2-atom version of this machine was then tested against our survey of 2-atom micro-worlds. The results for the subset with incongruence are shown in Figure 5.8. The performance of the ILP machine was dramatically better than the HLR machine, with respect to representation of the state distribution. However, it was correspondingly worse with respect to the atomic distribution. The ILP machine did not seem to retrieve the complex mixture of state and atomic distributions that were favoured by the HLR modified Boltzmann machine.

The selection of either state or atomic distribution by these different machines (HLR & ILP) can be seen by comparing the output from the corresponding preference relations in the incongruent 2-atom micro-worlds. This contrast in representations is illustrated in the examples below, Figures 5.9 – 5.11. The HLR machines represent a mixed distribution close to the atomic and the ILP machines represent the state distribution.

Example 2-Atom Micro-Worlds with Incongurence

Training Set: C18

Lvl	Pref. Relation	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(0), (3)	0	44%	25%	12411	41%
		1	6%	25%	1719	6%
1	(1), (2)	2	6%	25%	1659	6%
		3	44%	25%	14211	47%
			Accumulated error =		30000	**6.2%**
			Avg. error per state =			**1.6%**

Training Set: D18

Lvl	Pref. Relation	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(1), (2)	0	6%	25%	1686	6%
		1	44%	25%	13122	44%
1	(0), (3)	2	44%	25%	13436	45%
		3	6%	25%	1756	6%
			Accumulated error =		30000	**1.4%**
			Avg. error per state =			**0.4%**

Training Set: F25

Lvl	Pref. Relation	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(0), (1), (2)	0	32%	41%	9612	32%
		1	32%	23%	9513	32%
1	(3)	2	32%	23%	9557	32%
		3	4%	13%	1318	4%
			Accumulated error =		30000	**0.9%**
			Avg. error per state =			**0.2%**

Training Set: G25

Lvl	Pref. Relation	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(1), (2), (3)	0	4%	13%	1227	4%
		1	32%	23%	9915	33%
1	(0)	2	32%	23%	9373	31%
		3	32%	41%	9485	32%
			Accumulated error =		30000	**2.3%**
			Avg. error per state =			**0.6%**

Training Set: H25

Lvl	Pref. Relation	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(0), (1), (3)	0	32%	23%	9534	32%
		1	32%	41%	9484	32%
1	(2)	2	4%	13%	1234	4%
		3	32%	23%	9748	32%
			Accumulated error =		30000	**1.2%**
			Avg. error per state =			**0.3%**

Training Set: L41

Lvl	Pref. Relation	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(0), (1)	0	39%	46%	11375	38%
1	(2)	1	39%	32%	11914	40%
2	(3)	2	20%	13%	5914	20%
		3	2%	9%	797	3%
			Accumulated error =		30000	**2.2%**
			Avg. error per state =			**0.6%**

Training Set: M41

Lvl	Pref. Relation	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(2), (3)	0	2%	9%	747	2%
1	(1)	1	20%	13%	6342	21%
2	(0)	2	39%	32%	11130	37%
		3	39%	44%	11781	39%
			Accumulated error =		30000	**3.8%**
			Avg. error per state =			**1.0%**

Training Set: N41

Lvl	Pref. Relation	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(0), (3)	0	39%	24%	11121	37%
1	(1)	1	20%	34%	6077	20%
2	(2)	2	2%	17%	732	2%
		3	39%	24%	12070	40%
			Accumulated error =		30000	**3.9%**
			Avg. error per state =			**1.0%**

Training Set: W57

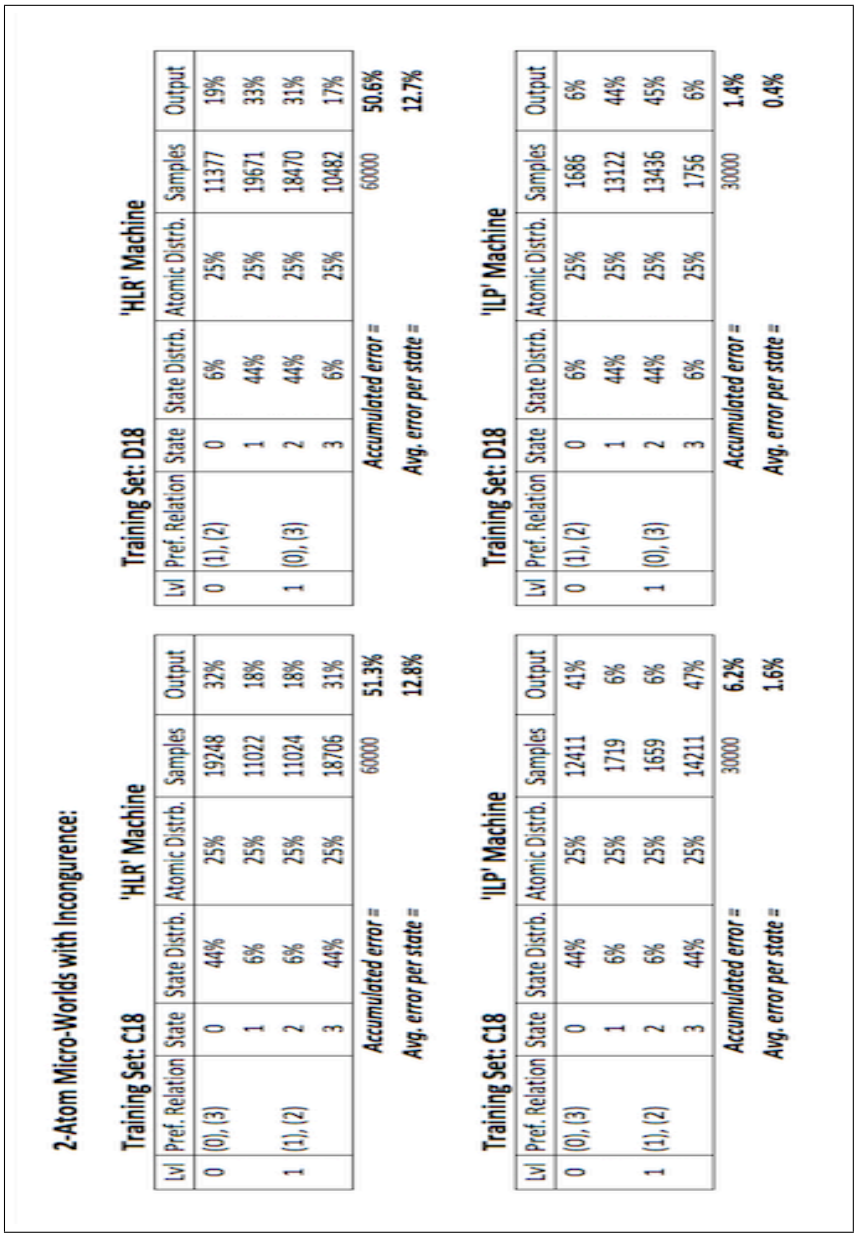
Lvl	Pref. Relation	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(2)	0	2%	17%	557	2%
1	(1)	1	28%	13%	8678	29%
2	(3)	2	56%	41%	16291	54%
3	(0)	3	14%	30%	4474	15%
			Accumulated error =		30000	**3.7%**
			Avg. error per state =			**0.9%**

Training Set: Y57

Lvl	Pref. Relation	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(1)	0	14%	30%	4379	15%
1	(2)	1	56%	41%	16499	55%
2	(0)	2	28%	13%	8534	28%
3	(3)	3	2%	17%	588	2%
			Accumulated error =		30000	**2.3%**
			Avg. error per state =			**0.6%**

Figure 5.8: The output from an ILP Boltzmann machine in a selection of incongruent 2-atom micro-worlds. Errors are calculated relative to the state distribution only.

- Figure 5.9 shows training sets C18 & D18. The HLR machine outputs a mixed distribution, state levels 32% versus 18%, based partially on the atomic probabilities, whereas the ILP machine closely matches the whole state distribution 44% versus 6%.



2-Atom Micro-Worlds with Incongurence:

Training Set: C18 'HLR' Machine

Lvl	Pref. Relation	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(0), (3)	0	44%	25%	19248	32%
		1	6%	25%	11022	18%
1	(1), (2)	2	6%	25%	11024	18%
		3	44%	25%	18706	31%
			Accumulated error =		60000	**51.3%**
			Avg. error per state =			**12.8%**

Training Set: D18 'HLR' Machine

Lvl	Pref. Relation	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(1), (2)	0	6%	25%	11377	19%
		1	44%	25%	19671	33%
1	(0), (3)	2	44%	25%	18470	31%
		3	6%	25%	10482	17%
			Accumulated error =		60000	**50.6%**
			Avg. error per state =			**12.7%**

Training Set: C18 'ILP' Machine

Lvl	Pref. Relation	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(0), (3)	0	44%	25%	12411	41%
		1	6%	25%	1719	6%
1	(1), (2)	2	6%	25%	1659	6%
		3	44%	25%	14211	47%
			Accumulated error =		30000	**6.2%**
			Avg. error per state =			**1.6%**

Training Set: D18 'ILP' Machine

Lvl	Pref. Relation	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(1), (2)	0	6%	25%	1686	6%
		1	44%	25%	13122	44%
1	(0), (3)	2	44%	25%	13436	45%
		3	6%	25%	1756	6%
			Accumulated error =		30000	**1.4%**
			Avg. error per state =			**0.4%**

Figure 5.9: Output distributions for the HLR vs ILP machines from training sets C18 & D18. Errors are calculated relative to the state distribution only.

- Figure 5.10 shows training sets F25 & G25. The HLR machine outputs a single state at highest frequency ∼39%, based on the atomic probabilities, whereas the ILP machine ranks the three most likely states equally at 32%, appropriate to the whole state distribution.

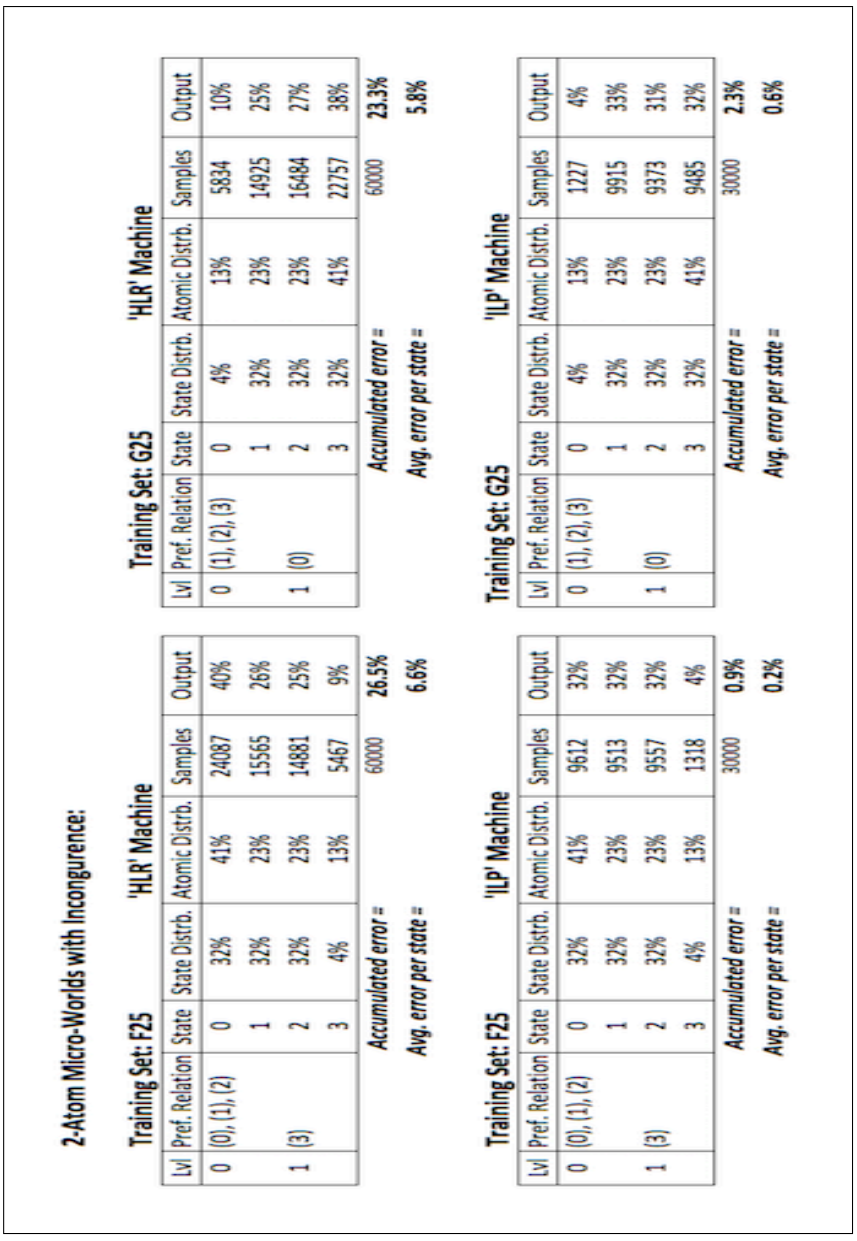
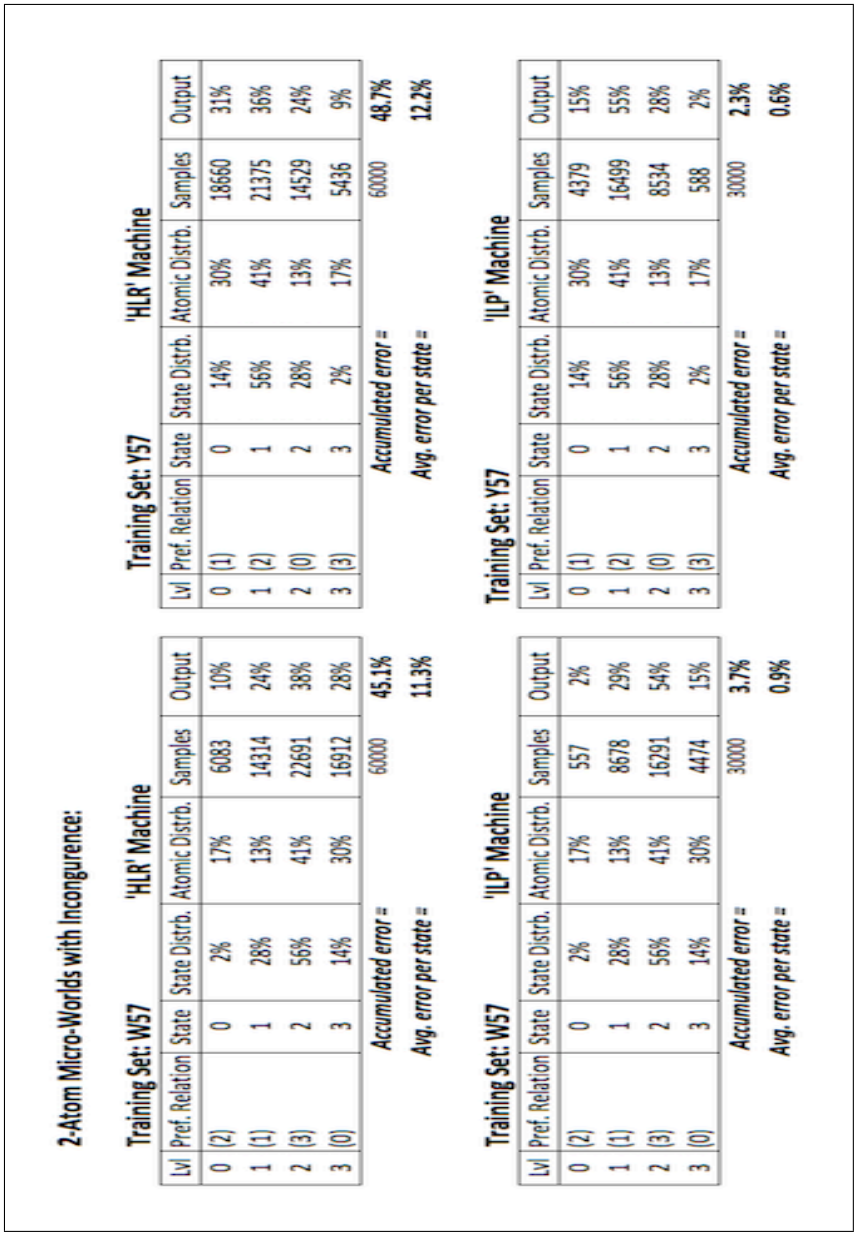


Figure 5.10: Output distributions for the HLR vs ILP machines from training sets F25 & G25. Errors are calculated relative to the state distribution only.

- Figure 5.11 shows training sets W57 & Y57. The HLR machine ranks outputs across four levels of preference appropriate to the atomic distribution, its accumulated error relative to the whole state distribution is high >45%, whereas the ILP machine ranks almost exactly to the whole state distribution with an accumulated error for that distribution of less than 4%.



2-Atom Micro-Worlds with Incongurence:

Training Set: W57 — **'HLR' Machine**

Lvl	Pref. Relation	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(2)	0	2%	17%	6083	10%
1	(1)	1	28%	13%	14314	24%
2	(3)	2	56%	41%	22691	38%
3	(0)	3	14%	30%	16912	28%
			Accumulated error =		60000	**45.1%**
			Avg. error per state =			**11.3%**

Training Set: Y57 — **'HLR' Machine**

Lvl	Pref. Relation	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(1)	0	14%	30%	18660	31%
1	(2)	1	56%	41%	21375	36%
2	(0)	2	28%	13%	14529	24%
3	(3)	3	2%	17%	5436	9%
			Accumulated error =		60000	**48.7%**
			Avg. error per state =			**12.2%**

Training Set: W57 — **'ILP' Machine**

Lvl	Pref. Relation	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(2)	0	2%	17%	557	2%
1	(1)	1	28%	13%	8678	29%
2	(3)	2	56%	41%	16291	54%
3	(0)	3	14%	30%	4474	15%
			Accumulated error =		30000	**3.7%**
			Avg. error per state =			**0.9%**

Training Set: Y57 — **'ILP' Machine**

Lvl	Pref. Relation	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(1)	0	14%	30%	4379	15%
1	(2)	1	56%	41%	16499	55%
2	(0)	2	28%	13%	8534	28%
3	(3)	3	2%	17%	588	2%
			Accumulated error =		30000	**2.3%**
			Avg. error per state =			**0.6%**

Figure 5.11: Output distributions for the HLR vs ILP machines from training sets W57 & Y57. Errors are calculated relative to the state distribution only.

5.2.3 ILP: Statistics

In the same way that the detailed results for single and ensemble machines were presented for the HLR architecture in Section 4.3.2: Preference Relation, we now present detailed results for the ILP architecture; 6,000 single machines and 100 ensemble machines with 60 parallel hidden layers (Table 5.2, Figures 5.12 & 5.13).

The ILP architecture is a closer approximation of the state exemplar philosophy of the logic than the HLR architecture, therefore the ILP error results relative to the whole state distribution, are in general better than those for the HLR architecture.

However, the same conclusions apply in terms of the benefits of accumulating output samples in an ensemble network. We re-iterate that:

- The sample sizes of the 6,000 single machines and 100 ensembles of 60 parallel hidden layers are the same.
- Although almost ideal single machines exist, they are rare. There is no efficient procedure for generating them during training.
- Any small ensemble will produce fast, robust results with low mean error and little variance.

Statistics for % Error: across whole distribution		
Statistic	*Single*	*Ensemble 60*
Sample Size	6,000	100 x 60
Range	52.4 : Min 4.0, Max 56.4	7.5 : Min 5.3, Max 12.8
Quartiles	1st 10.8, 3rd 26.0	1st 7.9, 3rd 9.9
Centre	Mean 19.2, Median 16.6	Mean 9.0, Median 8.9
Variation	SE 0.13, StDev 10.8	SE 0.10, StDev 0.96
Shape	Skew 0.74, Kurt -0.41	Skew 0.08, Kurt -1.10

Table 5.2: Statistics for the error between input and output distributions, comparing Single and Ensemble ILP machines in a 4-atom micro-world. Statistics via CRAN (2014).

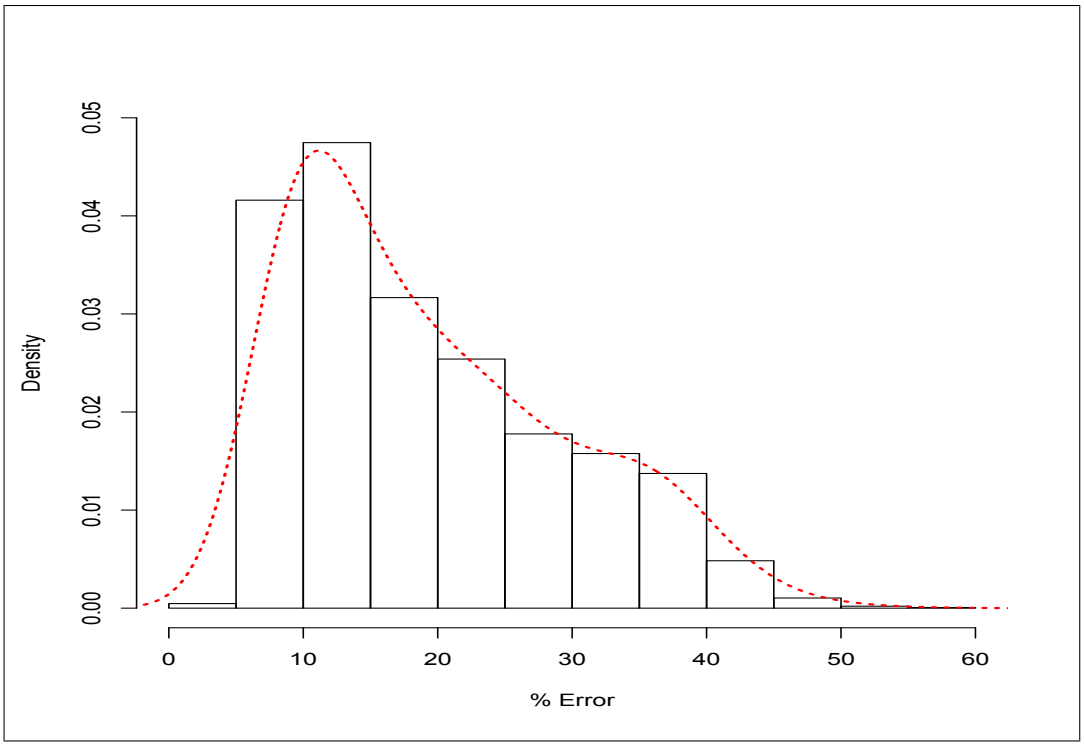


Figure 5.12: Histogram of % Error for Single ILP Machines (across whole distribution in a 4-atom micro-world.)

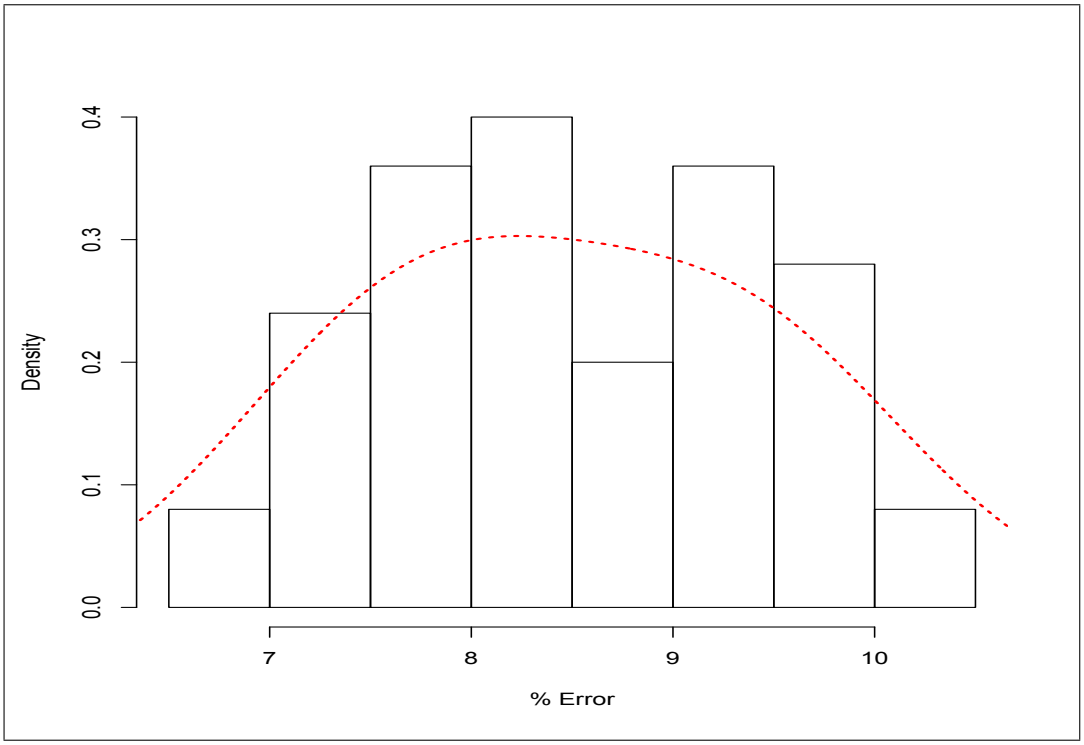


Figure 5.13: Histogram of % Error for Ensemble ILP Machines, with 60 parallel hidden layers.

As for Section 4.3.2, experimentation demonstrates there is little benefit increasing the ensemble size for the ILP architecture above 60 parallel hidden layers, Figure 5.14.

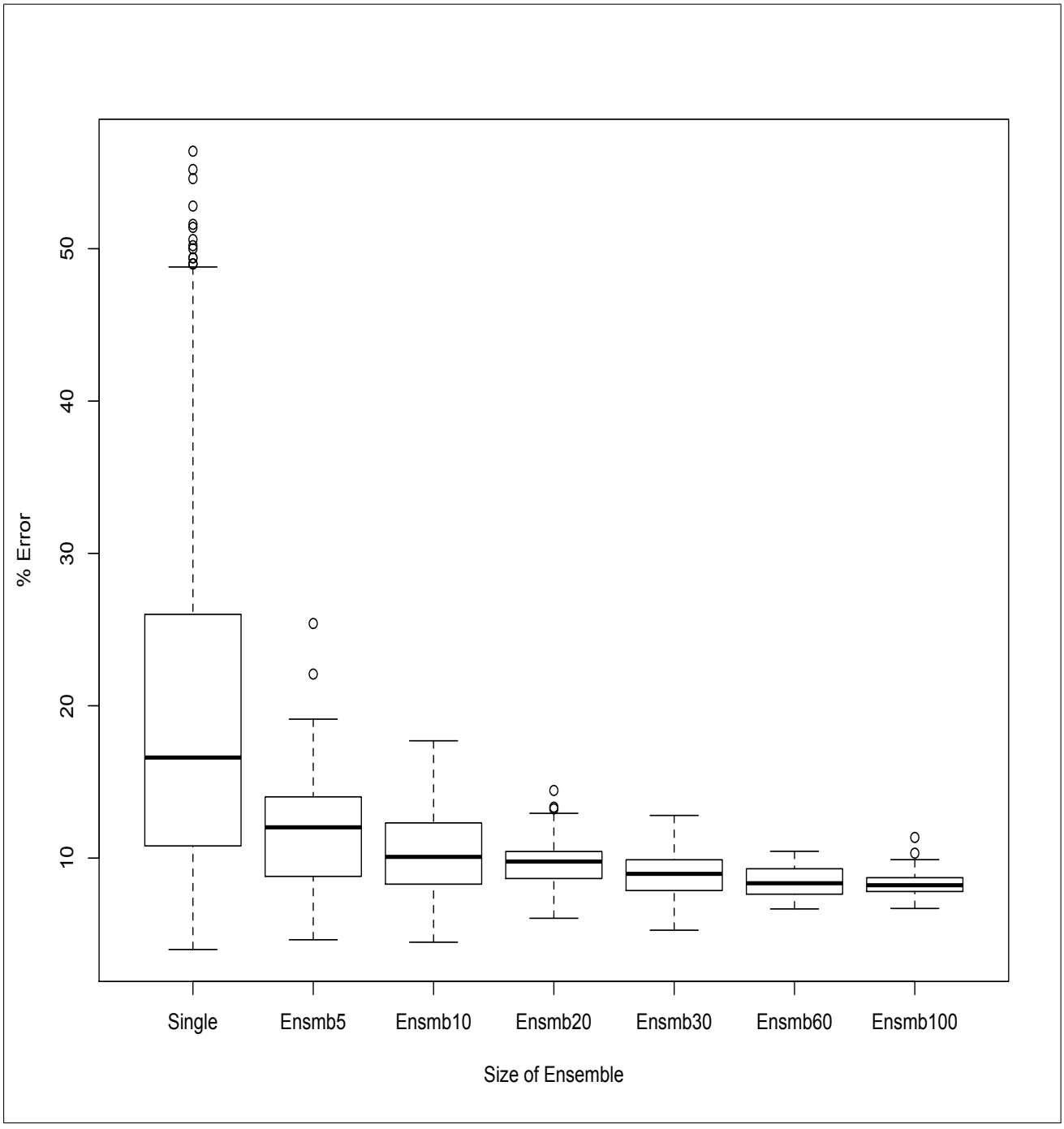


Figure 5.14: Boxplot of % Error across the whole distribution, versus increasing size of ILP Ensemble (increasing number of parallel hidden layers).

5.2.4 Network Analysis

Before completing this section, it would be interesting to consider the behaviour of the Boltzmann machine with respect to its selection of information distributions: to understand the underlying dynamics of the network by looking inside the 'black box'. How is the machine able to represent both the whole state distribution and the atomic distribution? The following analysis of the limited ILP modified Boltzmann machine is provided, in our example 4-atom micro-world (Figure 3.1), utilising a framework similar to a Markov model.

The HLR machine has no constraints on the permutation of the input and a generous number of hidden nodes. The ILP machine has constraints placed on input permutations and has a single hidden node. Why should these modifications be effective in moving the machine to a representation of the state distribution? Originally, we mistakenly surmised that the single hidden node could only act as a label for the whole state and that there should be less freedom available in this 'restricted' architecture for the representation of individual atoms. The analysis, however, shows that the ILP machine also learns a representation of the individual atomic activation frequencies.

The method utilised by the analysis was to examine the weight matrices of well trained ILP machines in this particular micro-world. This method enabled the construction of a probabilistic model of the network. When an ensemble of 60 machines was examined, it was found that there were three different patterns of weight matrices. Two of these patterns (pA.1 & pA.2, $\sim$25 machines each) were closely related. The biases and intra-layer weights were the same, the inter-layer weights had the same magnitude but their signs were reversed: the networks were like reflections of each other. The third category (pB, 10 machines) had a very different pattern of weights. The accuracy of each pattern of machine, relative to the state distribution, was separately checked: pA.1 accumulated error $\sim$10%, pA.2 accumulated error $\sim$4%, and pB accumulated error $\sim$24%.

An idealised weight matrix for the most accurate pA.2 machines is presented in Figure 5.15. First, we draw the reader's attention to the bias weights in the output layer, Table 5.3. As demonstrated in the table, for these almost ideal machines (pA.1 as well as pA.2), the output biases are the machine's representation of the atomic distribution. The availability of these individual atomic biases ensures that the machine will always be able to model the atomic distribution.

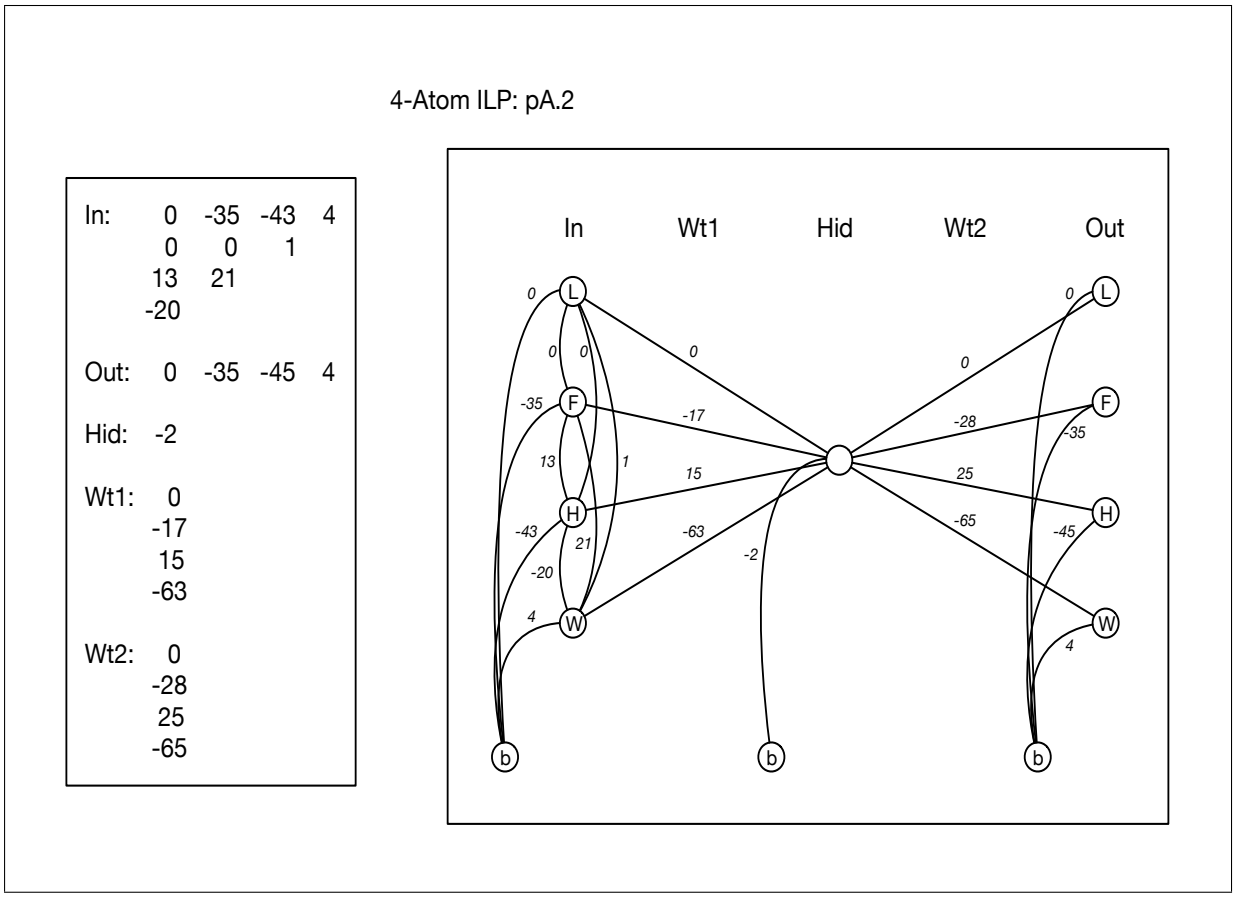


Figure 5.15: The weight matrix and network diagram for an ILP Boltzmann machine trained in a 4-atom micro-world. The most accurate A.2 pattern.

Atomic Distribution and Output Bias Weights			
Node	*Atomic Probability*		*Output Bias Wt*
Light	50 : 50	Neutral	0
Fan	18 : 82	Highly negative	-35
Heater	9 : 91	Highly negative	-45
Window	55 : 45	Slightly positive	+4

Table 5.3: Comparison of the probability of output node activation to the bias weights in the output layer, as a representation of the atomic distribution, in the example 4-Atom micro-world.

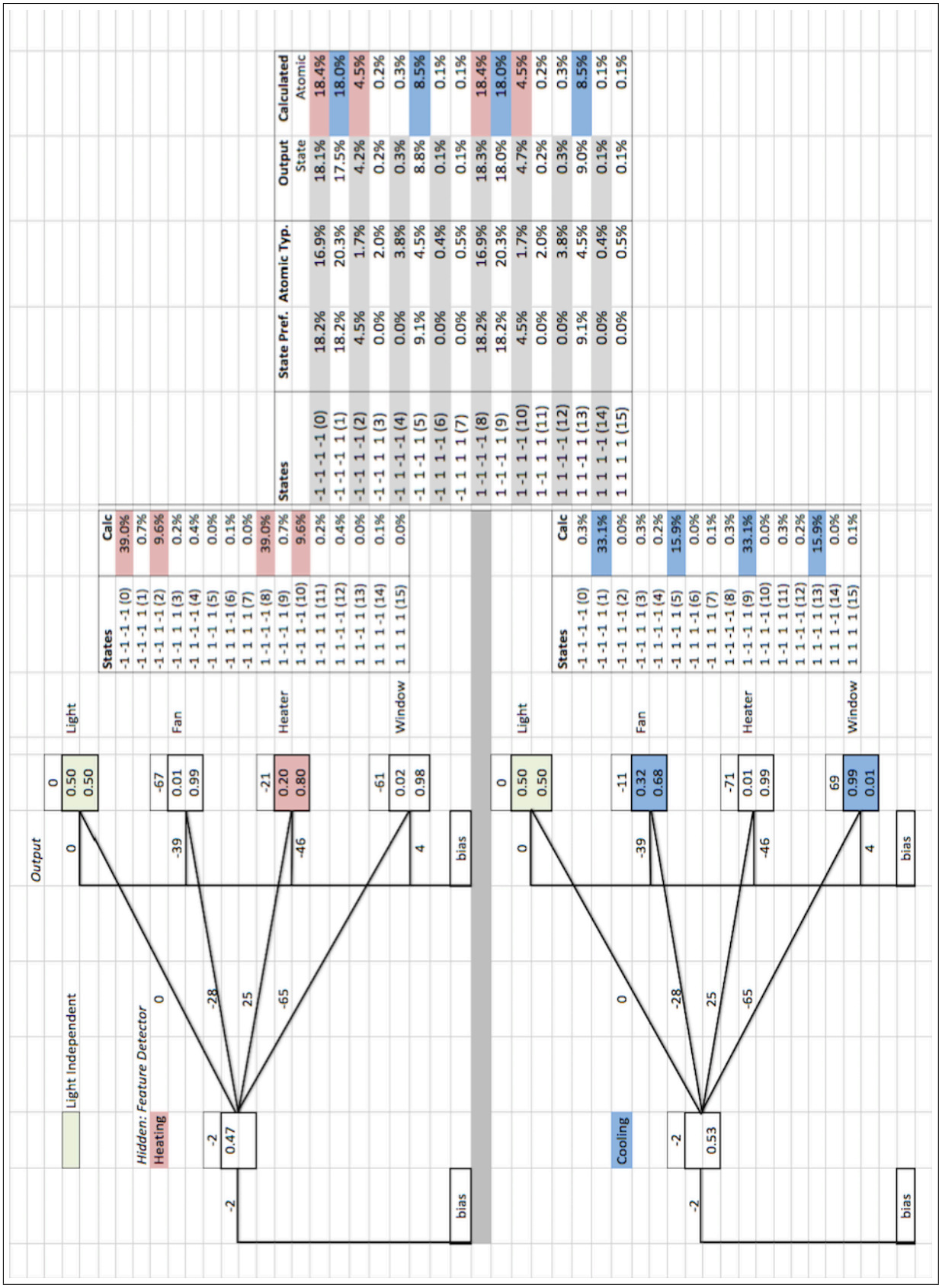


Figure 5.16: Network Analysis: see text for explanation.

Second, Figure 5.16 presents an analysis of the dynamics of the network when clamped with the neutral premiss, "0 0 0 0". The input half of the network is not shown, as in this circumstance it makes no contribution to the output. In the top half of Figure 5.16, the activation of the hidden node is +1: in the bottom half it is -1. There is a slight negative bias on the hidden node, so the relative firing proportions are 47 : 53.

- When positively firing, the hidden node (and output bias) activations combine to select out those states which represent 'heating': passive states with the window closed (states 0 & 8), as well as active heating states with the heater on (states 2 & 10).

- When negatively firing, the hidden node (and output bias) activations combine to select out those states which represent 'cooling': passive states with the window open (states 1 & 9), as well as active cooling states with the fan on (states 5 & 13).

The light is independent of this feature selection. In the circumstances depicted, where the whole distribution is being retrieved on testing with the neutral premiss, the cases of 'heating' and 'cooling' are combined to produce the total final output: representing the complete state distribution.

The hidden node, which can be seen as a feature detector, constitutes a semantic model of the micro-world. It is consistent with an intermediate level of geometric abstraction in the representation of knowledge, described by Gärdenfors (2004), as a conceptual space. The hidden node represents a linear conceptual space. The proportion of 'heating' and 'cooling' in the model can be adjusted by altering the bias on the hidden node. This adaptation allows the network to revise its learning when new information is presented (regarding the degree of environmental heating), which is partially consistent with prior features in the micro-world. This result is considered in Section 6.3.5 in relation to belief revision.

5.3 Typicality

The conventional view of rational consequence, in the preferential semantics of non-monotonic logic, only takes account of the whole state distribution: that is the preference relation, a consequence of the default rule, ranks whole states only. The in-

formation available in the distribution of individual component atoms, or prototypical features, is not normally considered (Lehmann and Magidor, 1992).

This conflict, between the exemplar and atomic views, relates back to the representation of a concept in logic (Frixione and Lieto, 2011). The traditional logical approach is to consider only the external opaque nature of the unitary exemplar. This approach is strongly supported by logicians such as Fodor (Rives, 2010; Fodor and Pylyshyn, 1988) who argues, from the perspective of a linguist, that lexical concepts (words) are whole states; they have no components. Although shared by most logicians, this view seems counter-intuitive; for example, to argue that a 'dog' is just a unitary symbol, requires that the symbol itself be divorced from its common meaning: a furry mammal, that runs on all fours, barks and wags its tail. Can a lexical symbol without semantic attachment be a concept?

The contrary view comes from psychology, where it is believed that humans have the ability to re-formulate and recognise new concepts because of their component nature (Barsalou, 1985, 1999): that our world is implicitly structured by classification based on prototypical component characteristics. There is evidence from neurobiological imaging that a generic object recognition system may be present in new-born babies and innate across species; adapted by evolution rather than learning (Spekle and Kinzer, 2007). This discovery adds weight to a view of the environment as a hierarchical structure constructed from inheritance and compositionality.

The divergent views, in philosophy and psychology, probably relate to the heterogenous requirements in these different domains. To clarify discussion, the following definitions are used within this thesis and the use of the ambiguous term 'concept' will be avoided as much as possible.

Whole State Preference: a total pre-order based on the preference of whole state models: each state model is considered as a unitary exemplar. It is the basis of preferential semantics in SCNM logic, equivalent to the distribution of whole states: determined by simple proportion in a training set of vectors.

Atomic Typicality: an ordering of state models, based on their individual component, atomic, prototypical characteristics. It is equivalent to the atomic distribution identified in Section 5.1: determined by the product of the atomic activation frequencies in a training set of vectors.

There remains ambiguity around the term 'prototypical', it is often used to refer to the most typical whole state exemplar of a class. However, in this thesis, proto-

typically will be used in the context of the specificity (to a class) of individual atomic components.

The modified HLR Boltzmann machine, used in Chapter 4: Representation, seemed to fulfil the two criteria identified as the basis of inductive inference in the logic: the ability to learn a distribution representing the preference relation in logic and the ability to retrieve one-to-many mappings constituting appropriate model selection based on a premiss. In hindsight, this conclusion was **only** true in congruent worlds. In fact, the HLR machine learns a complex mixture of distributional information much closer to the atomic probabilities (typicality), than to the whole state frequencies (preference). This does not mean that its representation of inference in general is wrong, it is just different from the view held in traditional symbolic logic.

Logic attempts to formalise the process of inference, whether in an abstract setting or in the broader context of human cognition. Is logic's customary approach of only considering whole state information too restrictive and inappropriate for biological cognition? The ILP modification of the Boltzmann machine is faithful to the conventional representation of inference in SCNM logic and will be used for experimentation in Chapter 6: Belief Revision. However, the ILP modification of the Boltzmann machine seems highly biologically implausible. Its very effectiveness as a representation in this context, casts doubt on the principle of using whole state preference in logical inference and on the opaque exemplar view of a concept (object).

5.3.1 Literature

Alternative approaches to representation, utilising typicality in semantic ontologies, have been proposed by Giugino and Lukasiewciz (2002) and are based on the early work of Lukasiewciz (1999). Considering these richer relationships and roles between concepts leads naturally to the formulation of the description logics (Baader, 2003). Description logics (DLs) are based around an object-oriented paradigm, which involves a set of concepts (classes) and a set of instances (objects). However, traditional DLs still fail to capture the essential feature of the object-oriented paradigm: inheritance via the implementation of prototypical component characteristics (Frixione and Lieto, 2011).

Possibly the earliest consideration of typicality in first-order logic, including the idea of conditionality, was published by Delgrande (1987). Contemporary approaches in logic, both in DL (Giordano, Gliozzi, Olivetti, and Pozzato, 2007, 2013) and propositional logic (Booth, Meyer, and Varzinczak, 2012, 2013) specifically incorporate a

typicality operator. The important concept of typicality, looks at the contributions individual atoms (component characteristics) make to the preference relations of whole states. It allows for the elaboration of the default rule by the application of conditional probability and independence ('irrelevance'). Giordano's conclusions regarding typicality were:

"(the logics), which are the basis of our semantics, are related to probabilistic reasoning ... the notion of conditional constraint allows typicality assertions to be expressed".

We believe that conditional constraint is a preliminary requirement for instituting entailment based on typicality but, in itself, it is not sufficient. The frequency of these typical characteristics must be globally assessed (Giordano *et al.*, 2013), in a manner similar to the derivation of our atomic distributions from global atomic activations and a separate ordering based on typicality is required, analogous to the ordering within the atomic distribution of our training sets (Section 5.1: Dual Distribution).

The implementation of typicality in Giordano *et al.* (2013) appears to be based on 'most preferred model semantics'; subsuming typicality within the whole state preference ranking. As explained below this is a mistaken assumption that: the preferred models are able to represent the most typical models or specific concepts. In this context, the word 'concept' is used to mean a prototypical characteristic:

"(most preferred models) these are the models that minimise the number of atypical concepts. ... Indeed, there is no (most preferred) model of a knowledge base that contains a non-typical instance of some concept." (Giordano *et al.*, 2013)

Booth & Meyer (2012) utilise a novel definition of modular ranking based on a Cartesian product, contrasting it with the standard ranking of Lehmann and Magidor (1992). It is unclear whether this definition is intended to incorporate prototypical characteristics, there is no discussion of such characteristics, their influence on typicality and the requirement for their global evaluation elsewhere in their paper. They explicitly declare typicality to be equivalent to the selection of the most preferred models in the initial definition of their typicality logic:

Definition 4. Let $\alpha \in \bar{L}$ and let $R = (V, \prec)$, then $[[\bar{\alpha}]] := min_{\prec}[[\alpha]]$

In this definition α is a state in the typicality language $\bar{L}$. R = (V, $\prec$) is a ranked model based on an ordering $\prec$, we must assume, a modular preference on whole states. Typicality is defined as: $[[\bar{\alpha}]]$, the set of typical models (concepts) equivalent to $min_{\prec}[[\alpha]]$. These are the most preferred whole state models of α.

We believe that this assumption: the representation of typicality by 'most preferred model semantics' is incorrect. The assumption can be refuted by counter-example: showing that the most typical models cannot always be found amongst the most preferred models and that the most preferred models can contain non-typical instances of prototypical characteristics. At the heart of this counter-example is a demonstration of incongruence: the divergence of preference and typicality.

5.3.2 Counter-Example

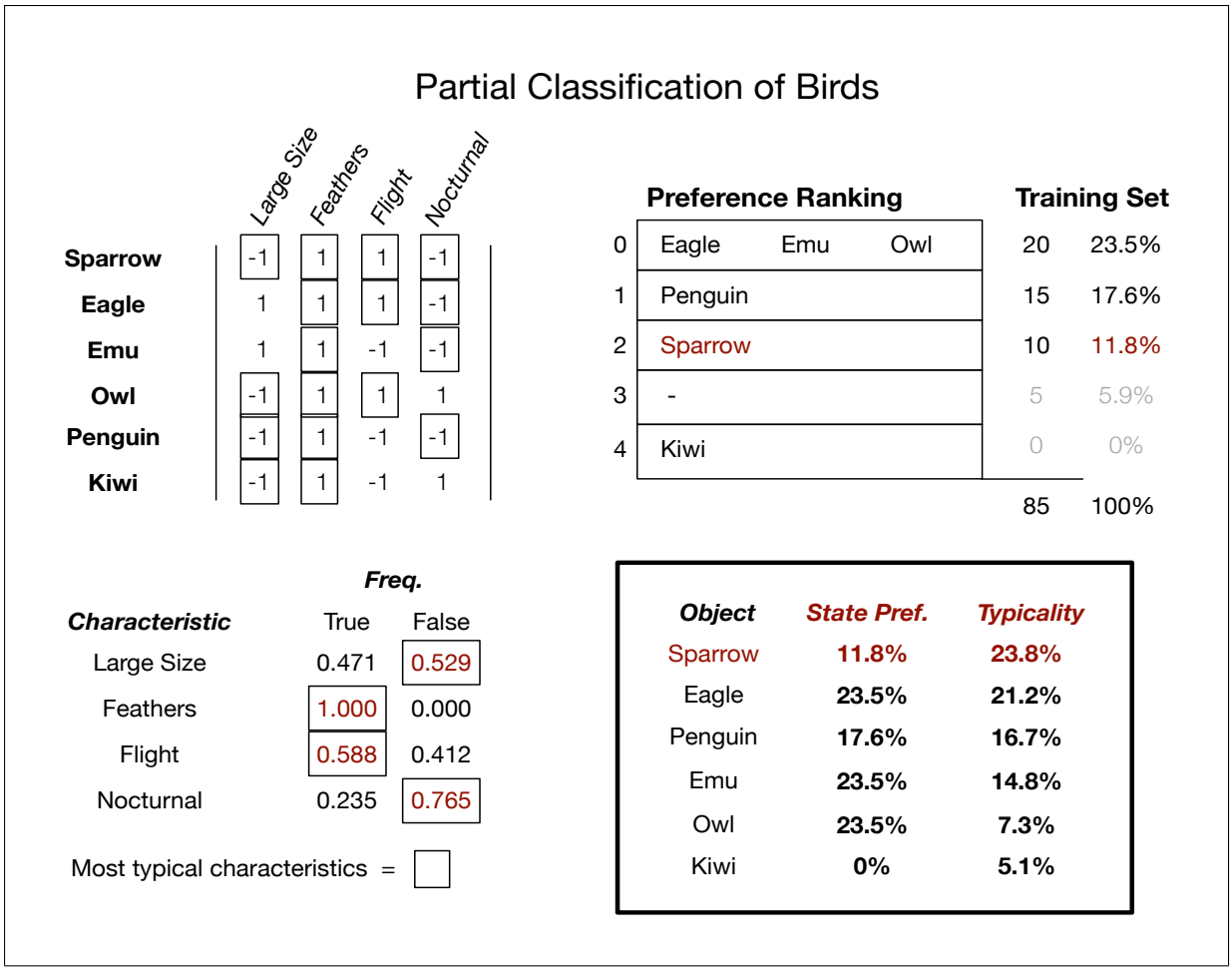


Figure 5.17: A partial classification system for birds, demonstrating incongruence between preference and typicality. Total patterns = $(3 \times 20) + (1 \times 15) + (1 \times 10) = 85$. Calculation of character frequencies for the derivation of typicality are explained in the text.

Let us consider a partial classification of birds, Figure 5.17, based on the selection of a few prototypical (atomic) characteristics: size, feathers, ability to fly and nocturnal

disposition. Suppose, for the sake of providing a scenario, that we were to travel 200 years into the future, to a world ravaged by climate change, to find that only a small number of birds survive in aviaries.

We observe, in the aviaries, the numbers of each type of bird, 85 in total: 20 eagles, 20 emus, 20 owls, 15 penguins, 10 sparrows and no kiwis. This information constitutes the training set, at the top right of the Figure 5.17. The percentages are calculated by simple proportion ($3 \times 23.5\% + 1 \times 17.6\% + 1 \times 11.8\% = 100\%$). Unusual birds are obviously more popular and have been preferred over their less exciting fellows: the sparrows. Other uncommon birds may be discovered later, so a blank preference level 3 is included. There are many birds that have become extinct, these are illustrated by a single example species, the kiwi. From our observations we are able to form a default rule and a preference ranking, top right of Figure 5.17.

We are also able to calculate the frequencies related to each prototypical (atomic) characteristic. Consider large size for example: there are two birds the eagle and emu with this atomic characteristic, they occur in 40 patterns out of total 85, which gives a character frequency of 0.471. The other character frequencies are similarly calculated, they are presented in the table at the bottom left of Figure 5.17. The reader should be able to appreciate, without reference to the numbers, that the most typical characteristics are: small size, feathers, flight and non-nocturnal disposition. The typicality of each model (bird) is calculated from the product of individual characteristic frequencies. Consider for example the penguin: it has small size (0.529), feathers (1.000), doesn't fly (0.412) and is non-nocturnal (0.765): its typicality is $0.529 \times 1.000 \times 0.412 \times 0.765 = 0.167$ (16.7%). The other model typicalities are similarly calculated.

Again, without considering the numbers, it is easy to see that the sparrow is the most typical bird: the only bird with all four typical characteristics. However, contrary to the assumption of 'most preferred model semantics', it is not among the most preferred models. Moreover, there are most preferred models (eagle, emu, owl) which contain atypical instances of prototypical characteristics (large size, flightless, nocturnal disposition).

It could be argued that the arbitrary nature of the setting compromises the generality of this counter-example, in particular the partial nature of the classification could be regarded as a problem. Given a complete classification system for birds, a large number of additional atomic characteristics could result in the most preferred models (eagle, emu and owl) also being the most typical. Such a complete classification system might contain an infinite number of atomic characteristics and as a consequence no

definitive counter-example would ever occur. A refutation of this argument is available when considering the micro-world schemata presented in Chapter 4 Representation. As the complexity of the micro-worlds are increased (by increasing the number of atoms) the proportion of permutations where the dual information distributions are incongruent increases. More atomic prototypical characteristics result in a greater chance of discrepancy between the pre-orders based on whole state frequency (preference) and atomic character frequency (typicality).

5.3.3 Atomic Description of Typicality

Such counter-examples to the assumption of 'most preferred model semantics' are common; other examples can be seen in Figure 5.8, training sets C18, D18 and N41. It is not possible to manage typicality by subsuming it into the standard preference ordering. Typicality constitutes a semantically different ranking, from the traditional preference ordering, on the same set of model states. Observations from Figure 5.17 provide other important insights into the general properties of typicality.

1. Varying Typicality. Not all typicality characteristics are equally typical. In the Figure 5.17 for example, feathers and non-nocturnal disposition are more strongly typical than small size and flight. This results in the emu being more typical than the owl on a probabilistic basis, even though the owl has three typical characteristics and the emu has only two. This is a problem for the logical representation of typicality because, without the combinative power of numbers, even separate assessment of each characteristic with individual typicality operators would be insufficient. This notion of the usefulness of numbers was proposed by Pearl: "probability as a faithful guardian of common-sense" (1997).

2. Abstract Classes. Extinct birds, such as the kiwi (in the example), play no part in the training set or in the formation of the default rule or preference relation. Yet, they are able to be ranked in terms of typicality. It is possible that an abstract class could represent the most typical model. Such an abstract class, without an instance, would be least preferred in the traditional pre-order on whole states. In fact, the implementation of an abstract class is a common scheme for inheritance in object-oriented programming. Even the potential existence an abstract class implies that the most preferred state models are an inadequate representation of typicality.

The following description is offered as an attempt to represent typicality in a propositional logic, given that prototypical compositionality is unconventional. Falsity is denoted as -1, not 0, in keeping with the use of a ternary logic within the micro-world schemata (Section 3.2).

Definition: Let A be the set of atomic propositions: $A = \{a_i : i = 1, 2, ..., k\}$

– symbolising the atomic composition of states in a world with atomic cardinality k.

Then: $\overrightarrow{t_a} = a_1^L \prec a_2^L \prec .... \prec a_{k-1}^L \prec a_k^L \prec a_k^M \prec a_{k-1}^M \ .... \ \prec a_2^M \prec a_1^M$

– is a total pre-order on the typicality of atomic valuations

– where a_i^L and a_i^M represent complementary truth valuations of an atom

– $a_i^M,\ a_i^L \in [false,\ true]$

– where if $i < j$, then a_i^L is less typical than a_j^L and a_i^M is more typical than a_j^M.

To clarify this definition of atomic typicality lets us consider the previous example of a classification of birds, Figure 5.17. There are four atoms, which instead of numbering, we will label a_{size}, $a_{feathers}$, a_{flight} and $a_{nocturnal}$. We know that having feathers is the most typical characteristic valuation and not having feathers is the most atypical characteristic valuation therefore: $a^M_{feathers}$ has a truth valuation of 1 (true) and $a^L_{feathers}$ has a truth valuation of -1 (false). Nocturnal disposition is the next most specific typicality characteristic, in this case the valuations are reversed, $a^M_{nocturnal}$ has a truth valuation of -1. Flight is the next most specific typicality characteristic, a^M_{flight} has a truth valuation of 1. Large size is the least specific, a^M_{size} has a truth valuation of -1. The corresponding values for a^L_x are complementary. Then a total pre-order for the atomic typicality in this example is given by:

$$\overrightarrow{t_a} = a^{-1}_{feathers} \prec a^{+1}_{nocturnal} \prec a^{-1}_{flight} \prec a^{+1}_{size} \prec a^{-1}_{size} \prec a^{+1}_{flight} \prec a^{-1}_{nocturnal} \prec a^{+1}_{feathers}$$

Then continuing with the description, from the general definition of atomic typicality, a total pre-order on state typicality $\overrightarrow{T_S}$, can be characterised as:

$$\begin{aligned} \overrightarrow{T_S} \quad = \quad & a_k^L \wedge a_{k-1}^L .... \wedge a_2^L \wedge a_1^L \quad \prec \quad a_k^M \wedge a_{k-1}^L .... \wedge a_2^L \wedge a_1^L \quad \prec \\ & a_k^M \wedge a_{k-1}^M .... \wedge a_2^L \wedge a_1^L \quad ..... \quad \prec \quad a_k^M \wedge a_{k-1}^M .... \wedge a_2^M \wedge a_1^L \\ & \prec \quad a_k^M \wedge a_{k-1}^M .... \wedge a_2^M \wedge a_1^M \end{aligned}$$

Persisting with the example of the bird classification for the sake of illustration. The least typical bird would be characterised by: $a^{+1}_{size} \wedge a^{-1}_{flight} \wedge a^{+1}_{nocturnal} \wedge a^{-1}_{feathers}$, such a bird is large, does not fly, is nocturnal and has no feathers. No such bird exists in

the classification. The most typical bird would be characterised by: $a^{-1}_{size} \wedge a^{+1}_{flight} \wedge a^{-1}_{nocturnal} \wedge a^{+1}_{feathers}$, such a bird is small, flies, is not nocturnal and has feathers. In the example classification this most typical bird is the sparrow.

Whilst this proposal, for describing typicality in a propositional logic, is not developed to the level of proof required of a theorem, it fits the experimental results from the Boltzmann representation and captures many properties expected in the complementary paradigm of probability. Looking ahead to the next chapter (Section 6.1.2: Iterative Revision) such a description resolves one of the challenges posed by probability. The robust declaration of individual atomic compositionality allows a satisfactory explanation of conditional constraint: as illustrated in Figure 5.4 and more generally in the design of the example 4-atom micro-world (Figure 3.1). The experimental results from the Boltzmann machine in Section 4.3.4: Rational Monotonicity demonstrate both the dependence and independence of atoms in a propositional SCNM logic.

This characterisation of atomic typicality still remains problematic, because the logical implication of $a^*_i \wedge a^*_j$ does not convey all of the numeric properties of $a^*_i \times a^*_j$ available in the context of probability (see item 1 above).

5.3.4 Typicality versus Preference

How is the conflicting information of typicality versus preference managed by human cognition? Currently an answer is uncertain, however in general, there are two obvious possible approaches.

- The information about atomic and state frequencies could be combined, as we see in the artificial Boltzmann machine learning. This allows the machine to operate in a common-sense way in most circumstances, even allowing it to adjust for conditional probabilities. However, it is not ideal, because some information from both (typicality & preference) orderings is inevitably lost.

- The alternative approach is to have two parallel systems of information processing: a "dual process approach". The older system (typicality) may be evolved by physically engineered change and be generic within the animal kingdom. Whilst the phylogenically younger system (preference) capable of adaptation by conscious learning, may have only recently emerged.

> "...systems of the first type are phylogenically older, unconscious, automatic, fast, reasoning about prototypically and exceptions, (typi-

> cality) ... the other phylogenically younger second system: conscious, slow, sequential, reasoning using explicit rules (preference)." (Frixione and Lieto, 2011, 2014)

Considering the experimental results in Section 5.2: Architectural Selection, the Boltzmann machine provides a practical, generic neural network for the implementation of this theory. It would be possible to have two parallel Boltzmann networks with dual architectures: one selecting for typicality based on atomic characteristics and the other selecting for preference based on whole state information. Following the proposal of Frixione and Lieto (2014) the outputs of the dual systems could then be compared in a manner analogous to a 'system of experts'. In congruent worlds there would be no conflict. However further questions arise, in incongruent worlds with outputs conflicting between the systems: Which system takes precedence? And in what context? Can a conscious choice be made between representations?

Chapter 6

Belief Revision

Adaptation to new information was identified as a requirement of any network representation of supra-classical, non-monotonic logic, Section 3.3.1. This chapter considers belief revision as it relates to adaptation to new learning in a Boltzmann machine.

Section 6.1 reviews the literature of belief revision in SCNM logic, starting with its foundations in the AGM postulates for single revision and moving onto the differences in theory for iterated revision. It attempts to highlight some of the on-going issues and open questions related to the theory of revision. Currently there are a plethora of twenty-seven different belief change operators. The section ends with an attempt to summarise the major distinctions between the operators.

Section 6.2 gives an overview of belief revision from the machine perspective: where the proliferation of different approaches in the logic are condensed to a single spectrum. An implementation of pseudo-rehearsal in the Boltzmann machine learning algorithm is designed to allow re-training. We propose a plan for investigating the mechanics of belief revision utilising the machine representation as a measure of plausibility. We give an overview for this investigation in Figure 6.6.

Section 6.3 compares the experimental results from re-training the Boltzmann machine with two contrasting approaches to revision in logic: Booth & Meyer and Spohn. Three levels of revision are considered: simple, inconsistent and differential: experimental results are provided for each of these levels. Finally an overview of the results describes the patterns of revision demonstrated by the Boltzmann machine.

Section 6.4 discusses a potential alternative approach to logical revision, based on the patterns in the machine results. This novel approach utilises the information on atomic typicality, already contained within the initial training set and also implicit within the new information.

6.1 Review of the Logic

The principles underlying belief revision in logic attempt to model rationality, often using faithfulness to human cognition as a guide. The reader is asked to revisit the preliminary descriptions and definitions of the logic from Sections 2.1: Supra-classical, Non-monotonic Logic & 3.1: Logical Preliminaries.

6.1.1 The AGM Postulates

The well known work of Alchourron, Gärdenfors & Makinson (AGM) is still relatively young (1985). It describes the properties of belief change operators in the limited context of single revisions. There are some important assumptions contained in its initial formulation:

- Minimal change, a universal key assumption.
- The unquestioned reliability of any new information. Modified by authors in non-prioritised revision.
- A uniform degree of commitment to any existing belief. Modified later by the authors themselves.
- A God-like level of knowledge, instituted by closure in the belief set and consequence relation. Modified by work on Knowledge Bases.

These limitations, as well as supporting evidence from a number of external sources, are considered in the discussion that follows. AGM is regarded by many authors as a core philosophy of belief revision in logic (Peppas, 2008; Hansson, 2011).

The theory, formulated under classical consequence (Cn), applies to a closed belief set (S) which contains all of the sentences that represent the agent's classical knowledge. This closed belief set is equivalent in the broader perspective to the maximally preferred models of a SCNM preference relation. However, as a result of closure, the belief set also contains all of the sentences that are a consequence of these explicit beliefs. Most of these additional sentences involve trivial conjunctions that do not contribute to the agent's knowledge.

Belief change results from the acquisition of new information (denoted as α, β). A variety of individual belief change operators have been described and some detail of these is given in the text that follows up until the end of Section 6.1.3: Survey of

Current Approaches. However, most authors consider there are three basic types of operation: contraction ($\dot{-}$), expansion (+) and revision (*). Pure expansion can be seen as a limited form of revision, not requiring contraction. This trivial expansion is accomplished directly and not generally defined by a set of properties. The other overarching operations of contraction and revision are inter-related. Revision can be seen as requiring both contraction and expansion: via the Levi identity, (Levi, 1977) and (Tamminga, 2003):

$$S * \alpha = (S \dot{-} \neg\alpha) + \alpha \qquad \text{(Levi Identity)}$$

and contraction by α can be constructed from the intersection with the revision by $\neg\alpha$: via the Harper identity (Meyer, Labuschagne, and Heidema, 2000):

$$S \dot{-} \alpha = S \cap (S * \neg\alpha) \qquad \text{(Harper Identity)}$$

The purpose of these two identities is to demonstrate the inter-relationship between these complex operations and define the process involved in the individual operators. Only the revision postulates are described in detail below. However, before presenting these postulates, two of the contraction properties will be considered.

$(S\dot{-}2)$ $\quad if\ \alpha \notin Cn(\emptyset),\ then\ \alpha \notin (S \dot{-} \alpha)$ $\qquad$ (Success)

$(S\dot{-}6)$ $\quad S \subseteq (S \dot{-} \alpha) + \alpha$ $\qquad$ (Recovery Axiom)

The success postulate for contraction (S$\dot{-}$2) states that: if the new information is not a tautology it should be removed from the contracted belief set. This classical view however, is not supported in SCNM logic; an agent may choose not to relinquish some strongly held beliefs. The recovery axiom (S$\dot{-}$6) states that: the original belief set should be recoverable after expansion with the removed information: it has also been problematic. Whilst it captures the intuition of rational, human cognition, two well known counter-examples have been described: "Cleopatra's children" and "George the criminal" (Hansson, 1991, 1993). The following is a brief description of the "Cleopatra's children" counter-example:

> "I believe that 'Cleopatra had a son' (ϕ) and that 'Cleopatra had a daughter' (ψ), and thus also that 'Cleopatra had a child' ($\phi \vee \psi$). Then I receive information that 'Cleopatra had no children', which makes me give up my belief in $\phi \vee \psi$. But then I am told that 'Cleopatra did have children', and so I add $\phi \vee \psi$. But I should not regain my belief in either ϕ or ψ as a result."

The validity of the recovery axiom remains an open question.

AGM Revision Postulates

Basic:

$(S*1)$	$S*\alpha = Cn(S*\alpha)$	(Closure)
$(S*2)$	$\alpha \in S*\alpha$	(Success)
$(S*3)$	$S*\alpha \subseteq S+\alpha$	(Inclusion)
$(S*4)$	$if\ \neg\alpha \notin S,\ then\ S*\alpha = S+\alpha$	(Vacuity)
$(S*5)$	$if\ \alpha\ is\ consistent,\ then\ S*\alpha\ is\ consistent$	(Consistency)
$(S*6)$	$if\ \models \alpha \leftrightarrow \beta,\ then\ S*\alpha = S*\beta$	(Extensionality)

Supplementary:

$(S*7)$	$S*(\alpha \wedge \beta) \subseteq (S*\alpha)+\beta$	(Super-expansion)
$(S*8)$	$if\ \neg\beta \notin S*\alpha,\ then\ (S*\alpha)+\beta \subseteq S*(\alpha \wedge \beta)$	(Sub-expansion)

Postulate 1 declares closure under conditions of ideal, omniscient reasoning. Postulate 2 states that the new information should always be included in the revised belief set. Other authors have pointed out, however, that reliability of the new information is not guaranteed (Spohn, 1988). Postulates 3 & 4, taken together, indicate it is not necessary to remove any information if the new information is consistent with the original belief set. Postulate 5 states that consistency is maintained during revision. Postulate 6 states the revision is independent of syntactical changes. The supplementary postulates 7 & 8, taken together, state that successive revisions with α and β are equivalent to revision with $\alpha \wedge \beta$.

The postulates identify the properties of rational contraction or revision functions but they do not specify a model for those functions. If for simplicity, we consider a contraction function: it should remove from the belief set only that minimal information entailed by α. To put this another way, the α-remainder of contraction should include the maximal subset of S that does not entail α. Unfortunately, there is more than one such maximal remainder subset. Partial meet functions are the models for contraction or revision which select for the intersection of these maximal subsets.

The postulates rationally follow human cognition; they are consistent with the universally accepted assumption of minimal change. However, they are not complete: they are insufficient to uniquely specify / encode the new belief set following revision. This insufficiency arises from the assumption of uniform commitment to the existing beliefs. Knowledge is held with varying degrees of belief and is revised according to

the agents degree of commitment to the new information. This concept of the 'level of belief' is encapsulated by the preference relation in SCNM logic.

Extensions of AGM

A number of alterations have been made to the original theory of AGM, which broaden its application and support its conclusions. The first of these was epistemic entrenchment developed by the original authors (Gärdenfors and Makinson, 1988). Epistemic entrenchment is a way of placing a logical value on the commitment that an agent has to a particular belief (Meyer *et al.*, 2000). It requires a well behaved ordering on the agent's beliefs, $\preceq$:

- Reflexive: $a \preceq a$
- Transitive: if $a \preceq b$ and $b \preceq c$, then $a \preceq c$
- With Minimal Bounds: $\exists\ a : \forall\ b \neq a,\ a \prec b$
- And Maximal Bounds: $\exists\ c : \forall\ b \neq c,\ b \prec c$

As stated previously, such a total pre-order is equivalent to a non-monotonic preference relation. An extension of this approach was modelled by Grove (1988) utilising concentric spheres ($) to represent an ordering on possible worlds (a total pre-order on models), as illustrated in Figure 6.1. This system of spheres successively concatenates (accumulates) sentences across levels of belief ranking: progressively limiting the satisfying models. It reverses the numbering of the belief set: whereas H_n is the most reliable belief held by the agent, $\$_1$ is the sphere containing the most plausible models of those beliefs. The system is therefore, not as intuitive as it might seem. As more complex concepts such as comparison and bounds are added, the scheme becomes almost unusable.

Let H_i be the set of sentences of rank i in a prioritised belief set, such that the agent's knowledge could be represented as the total pre-order on beliefs:

$\overrightarrow{H} = \bot_{(contradictions)} \prec H_1 \prec H_2 \prec .... \prec H_n \prec \top_{(tautologies)}$

– where for $i < j$, beliefs in H_j are more reliable than beliefs in H_i,

If $H_{\geq i} = H_i \cup H_{i+1} .... \cup H_n$ then a set of spheres is defined by:

$\$_i = \{Models[H_{\geq i}] : i = 1, ..., n\}$

– where for $i < j$, worlds in $\$_i$ are more plausible than worlds in $\$_j$.

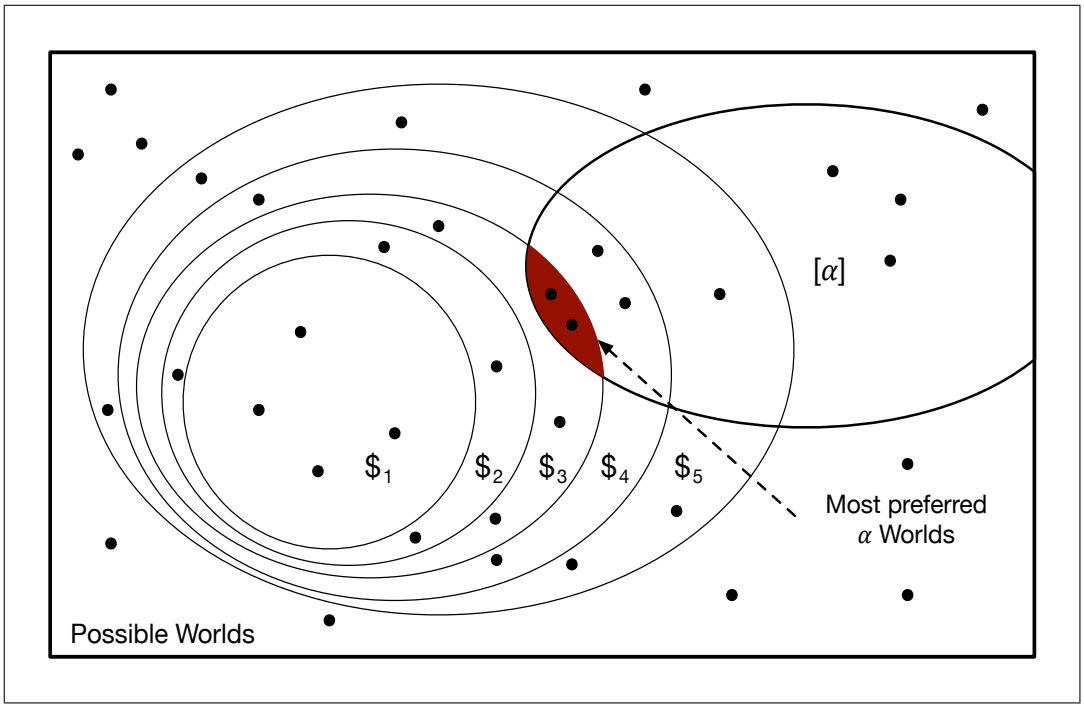


Figure 6.1: Revision modelled as a system of belief spheres (Grove, 1988).

A further refinement of the original theory was made (Rott, 1993; Nayak, 1994) to remove the limiting requirement of closure under classical consequence. It relinquishes all the trivial conjunctions from the belief set and no longer requires omniscient (God-like) reasoning. This finite representation of the agent's explicit, foundational, independent beliefs has subsequently been called a 'prioritised knowledge base' (K).

6.1.2 Iterative Revision

Iterated revisions can be defined as separate sequential revisions. There are a number of problems in applying the initial work of AGM to this iterated process. A major issue is in uniquely specifying / encoding the ordering on the agent's final beliefs after a single revision, as this is required for subsequent revisions. From the standpoint of probability, Darwiche and Pearl (DP, 1997) realised that "the AGM postulates were too weak to ensure the rational preservation of conditional beliefs". The information contained within the knowledge base (equivalent to the maximal models) was broadened to include the epistemic state. An epistemic state (Φ) encompasses an agent's entire knowledge: the foundational beliefs in the knowledge base and also the total

pre-order on all the less committed beliefs constituting the preference relation.

Darwiche & Pearl modified the original AGM postulates to include this additional knowledge from the epistemic state. The properties are re-formulated with respect to the knowledge base which results from the revision of the epistemic state by α: $K(\Phi * \alpha)$.

DP Revision Postulates

$(\Phi * 1)$	$K(\Phi * \alpha) = Cn(K(\Phi * \alpha))$	(Closure)
$(\Phi * 2)$	$\alpha \in K(\Phi * \alpha)$	(Success)
$(\Phi * 3)$	$K(\Phi * \alpha) \subseteq K(\Phi) + \alpha$	(Inclusion)
$(\Phi * 4)$	$if\ \neg\alpha \notin K(\Phi),\ then\ K(\Phi * \alpha) = K(\Phi) + \alpha$	(Vacuity)
$(\Phi * 5)$	$\bot \in K(\Phi * \alpha) \leftrightarrow \models \neg\alpha$	(Consistency)
$(\Phi * 6)$	$if\ \models \alpha \leftrightarrow \beta,\ then\ \Phi * \alpha = \Phi * \beta$	(Extensionality)
$(\Phi * 7)$	$K(\Phi * (\alpha \wedge \beta)) \subseteq K(\Phi * \alpha) + \beta$	(Super-expansion)
$(\Phi * 8)$	$if\ \neg\beta \notin K(\Phi * \alpha),\ then\ K(\Phi * \alpha) + \beta \subseteq K(\Phi * (\alpha \wedge \beta))$	(Sub-expansion)

The postulates have the same intuitions as those for AGM. The closure postulate $(\Phi*1)$: the knowledge base formed by revision is closed under rational consequence. The success postulate $(\Phi * 2)$: the new information is part of the revised knowledge base. The inclusion postulate $(\Phi * 3)$: the revised knowledge base is part of the knowledge base expanded by α. The vacuity postulate $(\Phi * 4)$: if the original knowledge base does not contain any 'non-model' of α, then revision and expansion of the knowledge base by α are the same. The consistency postulate $(\Phi * 5)$ is stated slightly differently: that inconsistent knowledge only arises from revision with contradictions. The extensionality postulate $(\Phi * 6)$: states the changes in the epistemic state are syntactically independent. Similarly, super & sub-expansion $(\Phi * 7\ \&\ \Phi * 8)$ are concerned with successive revisions. In addition, Darwiche & Pearl proposed four further postulates that consider the order of the applied new information and strengthen subsequent revisions:

Additional Postulates

$(C * 1)$	$if\ \beta \models \alpha,\ then\ K(\Phi * \alpha * \beta) = K(\Phi * \beta)$	(Succession)
$(C * 2)$	$if\ \beta \models \neg\alpha,\ then\ K(\Phi * \alpha * \beta) = K(\Phi * \beta)$	(Contradiction)
$(C * 3)$	$if\ K(\Phi * \beta) \models \alpha,\ then\ K(\Phi * \alpha * \beta) \models \alpha$	(Retention)
$(C * 4)$	$if\ K(\Phi * \beta) \not\models \neg\alpha,\ then\ K(\Phi * \alpha * \beta) \models \alpha$	(Self-Contradiction)

These postulates are conventionally indexed as illustrated ('C1 – 4'). Postulate 1 is a stronger version of AGM (S*7 & S*8) and states: that when new information is applied to the epistemic state, the most recent, more specific information takes precedence. Postulate 2 has been particularly problematic and states: that when contradictory information arrives, the most recent evidence prevails. There is a basic equivalence to the recovery axiom; in effect contract by α and then expand by α. Many authors have argued that the C2 postulate is too strong (Booth and Meyer, 2011; Nayak, Pagnucco, and Peppas, 2003). Postulate 3 states: that previous non-contradictory information should be retained. Postulate 4 states: information should not act as its own defeater.

Whilst not in the mainstream of traditional belief revision theory, Wolfgang Spohn's (1988; 1999; 2009) pioneering philosophy establishes a connection between ranking functions in logic and probability. He makes the following important arguments in relation to belief revision in general:

1. All belief revision is iterated. No theory of single revision is sufficient.

2. No logical theory of belief revision has a satisfactory explanation for doxastic independence: α is independent of β relative to epistemic state Φ if and only if revision of Φ by α or $\neg\alpha$ does not affect the belief in β.

   More precisely, it is not conditional independence but rather the reciprocal circumstance of conditional dependence which is the problem. As well as potentially affecting the results of belief revision, conditional dependence also complicates inference, as illustrated in Figure 5.4. The incorporation of atomic typicality into a propositional logic makes available an explanation of conditional constraint (Section 5.3.3: Atomic Description of Typicality).

3. Current theories of iterative revision, utilising a qualitative single belief ranking, are insufficient to implement the recovery axiom: the operators cannot uniquely specify the final, revised epistemic state.

4. Under the current theories iterated revision is not commutative: $K((\Phi * \alpha) * \beta) \neq K((\Phi * \beta) * \alpha)$. Intuitively belief revision should be commutative; an agent's knowledge should accumulate.

5. Strict prioritisation of the new information is an unreasonable assumption. After revision of Φ by α it is possible that the most preferred models of $\neg\alpha$ might prevail over the least preferred models of α.

Spohn provides an efficient solution to the problems in 3 – 5 by implementing a simple numerical index of belief; a half-way house between traditional logic and probability. He proposes that the levels of the epistemic ranking function (the preference relation of SCNM logic) be indexed from zero: the strongest belief (most preferred models). These ranking functions he calls 'Ordinal Conditional Functions'. He proposes that even empty 'levels of belief' should be retained following revision, in contrast to the prevailing theory which removes them: purification (Rott, 2009). Spohn's approach is in accord with the Pinkas classification of strong equivalence (Pinkas, 1995). Further, Spohn proposes that an agent should have some measurable commitment to any new information that is received: [α, β – index of belief]. This index of commitment is utilised to appropriately relinquish belief in the non-satisfying models ($Models$ [$\neg\alpha$]).

Nayak *et al.* (2003) criticise the work of Darwiche & Pearl for excessively strengthening the specification of AGM. However, they indicate no acceptance of AGM's foundational limitations. They propose dynamic revision operators, which implement changes in the epistemic entrenchment or provide a lexicographic ordering. They offer a new model of belief change without any reference to the preceding work of Spohn, demonstrating the inadequacy of such a qualitative pre-order in uniquely specifying the outcome of iterative change.

In the mainstream of belief revision theory, are a number of papers by Booth, Chopra, Ghose & Meyer (Chopra, Ghose, and Meyer, 2002; Booth, Chopra, Meyer, and Ghose, 2004; Booth, Meyer, and Wong, 2006; Booth and Meyer, 2011). They acknowledge the limitations of the traditional AGM approach whilst attempting to incorporate previous work. Their revision has been described by Rott as a dual operator. They strengthen the specification of restrained revision by partitioning each rank in the pre-order of the agents beliefs: a refinement. The additional meta-information for partitioning is based on the satisfying models of the new information, as described in their paper:

> "Our idea is to associate with each world, two abstract objects x^+ and x^- with the intuition that x^+ represents x 'on a good day, in positive circumstances; while x^- represents x 'on a bad day, in negative circumstances'." (Booth and Meyer, 2011)

The ranking structure in Figure 6.2(a) relies on a monotonically increasing index, both horizontally and vertically. The source of the meta-information is not so much of an issue since it is partially provided by the new information itself. However, the

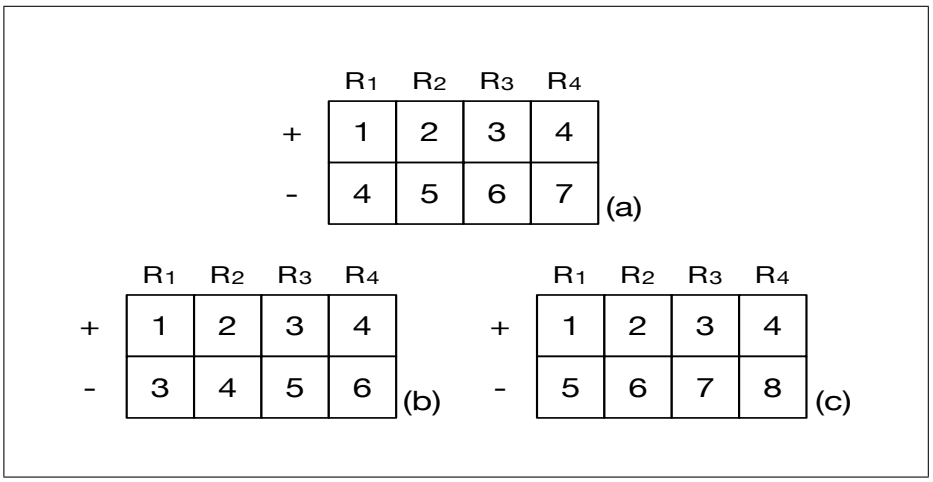


Figure 6.2: Refinement, utilising a two-dimensional ranking: after the scheme of Booth & Meyer (a), with variations on the commitment to belief of the non-satisfying models (b) & (c).

scheme suffers from the problem expressed by Spohn, in item 5 above. Whilst the scheme modifies the agent's existing epistemic ranking, it does not provide sufficient information about the level of commitment to the new information. Therefore, alternative outcomes for a revision are possible, Figure 6.2(b) & (c). Such alternatives disrupt linkage in the process of iteration.

6.1.3 Survey of Current Approaches

Rott (2009) has published a concise summary of 27 current iterative belief change operators using equivalent categorisations as: prioritised knowledge bases and systems of spheres diagrams. A representative sample of operators is reproduced here to illustrate some of their key characteristics. The concept of a prioritised base is utilised for the discussion in the text that follows. As the system of spheres diagrams are inclined to become disordered, a survey of these is presented in (Figure 6.3), at the end of the section.

However, before proceeding, two areas of potential confusion will be addressed. There is a subtle difference between a prioritised knowledge base and an epistemic state. The prioritised knowledge base contains a ranking of beliefs held by an agent. An epistemic state contains a ranking on *all possible* beliefs, including the truely inconsistent counter-examples at least preference. As an aside, machine learning can be understood as a knowledge base. The probability of the least preferred models is close to zero since a machine may never be exposed to them during training. The second

issue is the confusing use of the term 'prioritised'. In this context prioritisation relates to the ranking of the knowledge base, rather than commitment of the agent to the new information.

Recalling the definitions from Section 6.1.1: Extensions of AGM, the sentences contained within each rank i of a prioritised knowledge base are denoted H_i, derived from a total pre-order on the agent's beliefs:

$$\overrightarrow{H} = \bot_{(contradictions)} \prec H_1 \prec H_2 \prec .... \prec H_n \prec \top_{(tautologies)}$$
$$H_{\geq i} = H_i \cup H_{i+1} .... \cup H_n \quad \text{and spheres: } \$_i = \{Models[H_{\geq i}] : i = 1, ..., n\}$$

If we define h_i as a concatenation of the sentences within each rank: $h_i = \wedge H_i$. Then we can extend the interpretation of a pre-order on the agent's beliefs in various ways, for example:

$$\begin{array}{rclcrcl} h_{\geq i} & = & h_i \wedge h_{i+1} .... \wedge h_n & & h_{\leq i} & = & h_1 \wedge h_2 .... \wedge h_i \\ \overrightarrow{h \vee \alpha} & = & h_i \vee \alpha .... \prec h_n \vee \alpha & & \overrightarrow{h \vee^+ \alpha} & = & h_1 \prec h_1 \vee \alpha .... \prec h_n \prec h_n \vee \alpha \\ \overrightarrow{h} \prec . \, \alpha & = & h_1 .... \prec h_n \prec \alpha & & \overrightarrow{h} \prec . \, \overrightarrow{g} & = & h_1 .... \prec h_n \prec g_1 .... \prec g_n \end{array}$$

Where '$\prec .$' represents a concatenation of ordered sequences and $\overrightarrow{g}$ is a further distinct pre-ordering.

Variants of Expansion

Isolated expansion of an epistemic state is technically impossible, except following update with a new proposition: as all the existing non-satisfying models that are already part of the epistemic state would also need to be revised.

Conservative expansion:	$\overrightarrow{h}$	$\longmapsto$	$\alpha \prec . \, \overrightarrow{h}$	Fig. 6.3(a)
Plain expansion:	$\overrightarrow{h}$	$\longmapsto$	$h_1 \wedge \alpha \prec . \, \overrightarrow{h_{>1}}$	
Moderate expansion:	$\overrightarrow{h}$	$\longmapsto$	$\overrightarrow{h} \prec . \, \alpha \prec . \, \overrightarrow{h \vee \alpha}$	Fig. 6.3(b)
Radical expansion:	$\overrightarrow{h}$	$\longmapsto$	$\overrightarrow{h_{<n}} \prec . \, h_n \wedge \alpha$	
Very Radical expansion:	$\overrightarrow{h}$	$\longmapsto$	$\overrightarrow{h} \prec . \, \alpha$	Fig. 6.3(c)

However, we present these variations on expansion to illustrate the potential assumed levels of commitment to any new information (α). Conservative expansion can be seen as the complement of very radical expansion, adding the new models to a separate level beyond the pre-existing ordering. Plain expansion is the complement of radical

expansion adding the new models to the lowest or highest level of belief respectively. Moderate expansion adds the new models at intermediate levels. Both radical and very radical expansion are forms of prioritised change whereas: conservative, plain and moderate expansion are variations on non-prioritised change.

Variants of Revision

The previously considered approaches to expansion have matching revision processes. Conservative revision was examined by Boutlier (1996) and is traditionally called 'natural revision'. Restrained revision was examined by Booth *et al.* (2006); Booth and Meyer (2011). Moderate revision was examined by Nayak (1994) and is called 'lexicographic revision', and radical revisions are traditionally called 'irrevocable'.

Conservative revision:	$\overrightarrow{h}$	$\longmapsto$	$\alpha \prec . \overrightarrow{h_{\leq \neg\alpha}} \prec . \overrightarrow{h_{>\neg\alpha}}$	Fig. 6.3(d)
Restrained revision:	$\overrightarrow{h}$	$\longmapsto$	$\alpha \prec . \overrightarrow{h_{\leq \neg\alpha} \wedge \alpha} \prec . \overrightarrow{h_{>\neg\alpha} \vee^{+} \alpha}$	
Moderate revision:	$\overrightarrow{h}$	$\longmapsto$	$\overrightarrow{h} \prec . \alpha \prec . \overrightarrow{h \vee \alpha}$	Fig. 6.3(e)
Radical revision:	$\overrightarrow{h}$	$\longmapsto$	$\overrightarrow{h_{<n}} \prec . h_n \wedge \alpha$	
Very Radical revision:	$\overrightarrow{h}$	$\longmapsto$	$\overrightarrow{h} \prec . \alpha$	

None of these revisions are sufficient, as described, to uniquely encode the resultant epistemic state following revision. None of them can be considered, in themselves, a sufficient theory of iterative revision. The reader will notice the descriptions omit to give any indication of what happens to the non-satisfying models ($Models\ [\neg\alpha]$). Further the degree of withdrawal of these non-models is often assumed within the framework of the individual approaches.

Variants of Contraction

Severe withdrawal:	$\overrightarrow{h}$	$\longmapsto$	$\overrightarrow{h_{>\alpha}}$
Conservative withdrawal:	$\overrightarrow{h}$	$\longmapsto$	$\alpha \prec . \overrightarrow{h_{>\neg\alpha}}$
Refinement:	$\overrightarrow{h}$	$\longmapsto$	$\overrightarrow{h_{<\neg\alpha}} \prec . \overrightarrow{h_{\geq\neg\alpha} \vee^{+} \alpha}$

Variants of contraction can also be informative: severe withdrawal, complete removal of models from an epistemic state, is not possible. However, conservative withdrawal of models to the lowest level of belief could be possible. The last variant, refinement partitions each belief rank implicitly separating the satisfying and non-satisfying models. This operator is part of the approach of Booth *et al.* (2006).

Two-Dimensional Operators

The two-dimensional operators can be considered the only realistic candidates for theories of iterative revision. The favoured traditional approach of Booth *et al.* (2006); Booth and Meyer (2011) should be included among them.

Simple:	$\overrightarrow{h}$	$\longmapsto$	$\overrightarrow{h_{<\beta}} \prec . \, h_{=\beta} \wedge \alpha \prec . \, \overrightarrow{h_{>\beta}}$	Fig. 6.3(f)
Strict:	$\overrightarrow{h}$	$\longmapsto$	$\overrightarrow{h_{\leq(\alpha\to\beta)}} \prec . \, h_{=(\alpha\to\beta)+1} \wedge \alpha \prec . \, \overrightarrow{h_{>(\alpha\to\beta)+1}}$	
Bounded:	$\overrightarrow{h}$	$\longmapsto$	$\overrightarrow{h} \prec . \, \alpha \prec . \, \overrightarrow{h_{<(\alpha\to\beta)} \vee \alpha} \prec . \, \overrightarrow{h_{\geq(\alpha\to\beta)}}$	Fig. 6.3(g)
Booth & Meyer:	$\overrightarrow{h}$	$\longmapsto$	$\alpha \prec . \, \overrightarrow{h_{\leq\neg\alpha} \wedge \alpha} \prec . \, \overrightarrow{h_{>\neg\alpha} \vee^{+} \alpha}$	
		$\longmapsto$	$\overrightarrow{h_{<\neg\alpha}} \prec . \, \overrightarrow{h_{\geq\neg\alpha} \vee^{+} \alpha}$	Fig. 6.3(h)
Spohnian:	$\overrightarrow{h}$	$\longmapsto$	$\overrightarrow{h_{\leq\beta} \vee \neg\alpha} \prec . \, \overrightarrow{h_{(>\beta,<n)}} \prec . \, h_{\geq n} \vee \alpha$	Fig. 6.3(i)

These two dimensional operators demonstrate an increased complexity as they involve a dual pre-order. The second component provides additional information for promotion of satisfying models. The 2nd dimension for the simple comparison operator is provided by a specified reference equality '$\beta \leq \alpha$'. This 'simple revision by comparison' can result in either raising or lowering of the satisfying $Models[\alpha]$ and was investigated by Cantwell (1997). The second strict comparison operator is related to the simple revision and involves a reference inequality '$\beta < \alpha$'. The third bounded operator relies on the relationship '$\alpha \to \beta$' as investigated by Rott (2007); it is a compromise between conservative and moderate revision.

The Booth & Meyer approach can be visualised as a two step process of restrained revision and then refinement, as discussed above, Section 6.1.2: Iterative Revision. We have derived the prioritised knowledge base description of the Spohnian revision from the original paper (Spohn, 1988). The 2nd dimension for the revision is provided by the index of commitment to the new information proposed by Spohn, [α, β – index of belief]. This revision was not included in the summary published by Rott (2009) because of Spohn's alternative approach to numerically indexing belief: both in the epistemic pre-order held by the agent and in the commitment to the new information. Spohn's approach explicitly deals with the fate of the non-satisfying models. His approach should not be seen as abandoning the qualitative flavour of logic, but rather as a clarification of its assumptions.

In summary, the reader should appreciate from this discussion of the literature: that the domain of belief revision is relatively young, it has many open questions and represents a branch of philosophical logic in evolution.

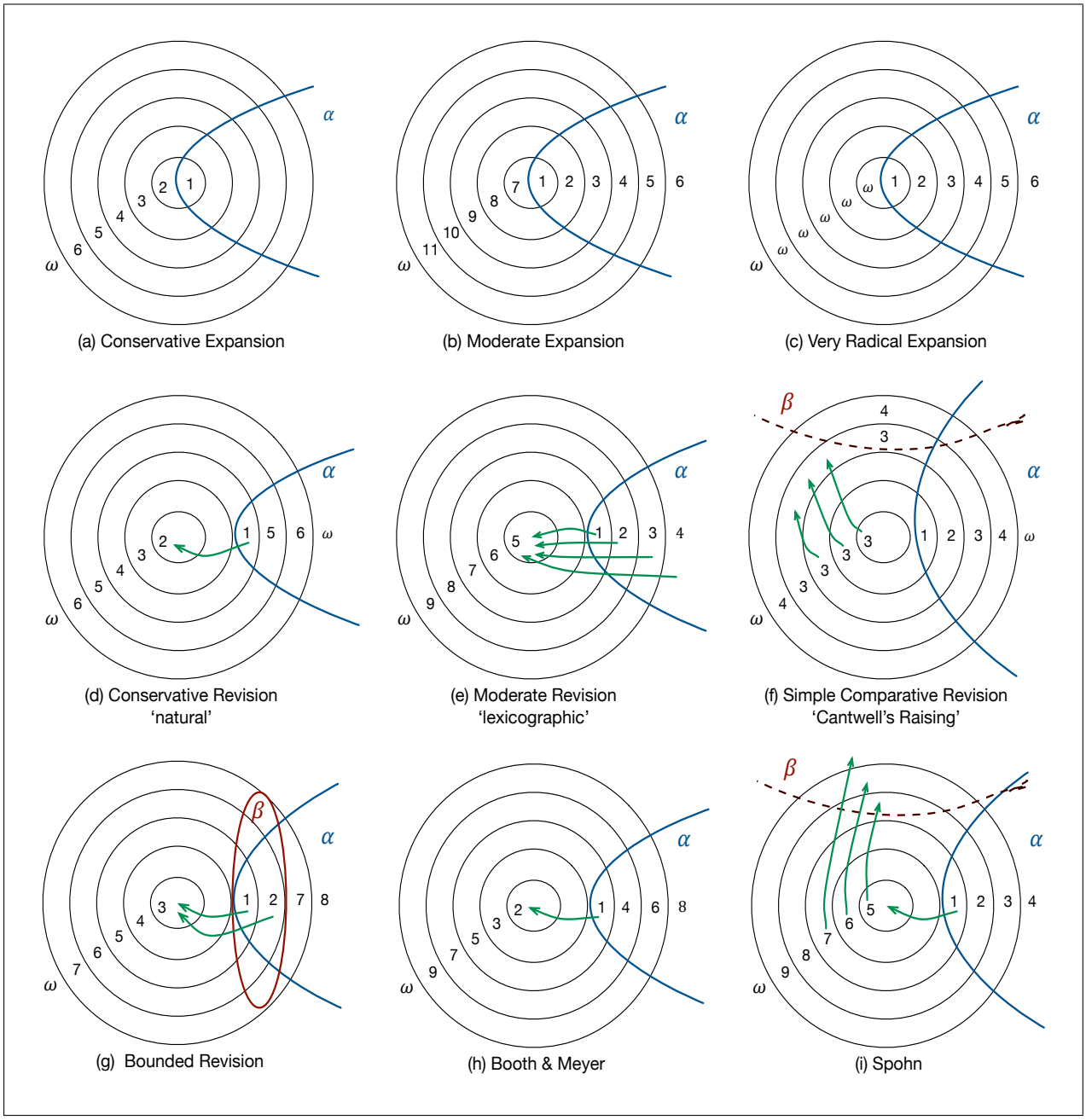


Figure 6.3: A survey of some belief change operators, illustrated by sphere diagrams (Rott, 2009). The three expansions illustrate varieties of prioritisation. The three revisions cover common non-prioritised schemes. Two 2nd dimensional operators are illustrated. The approaches of Booth & Meyer and Spohn are individually discussed in the text. The diagrams are inclined to become counter-intuitive and cluttered.

6.2 The Machine Perspective

When trying to analyse the results obtained from re-training of the Boltzmann machine; the proliferation of individual revision approaches, found in the logic, can be simplified to a spectrum. In general, for any type of logical revision, the models satisfying the new information (premiss) are moved up the preference ranking, usually to maximal preference. The non-models, those not satisfying the new information, are moved down the preference ranking towards least preference. The degree of separation of the models and non-models varies between the individual approaches to revision. A generalised scheme for revision from the machine perspective is illustrated in Figure 6.4.

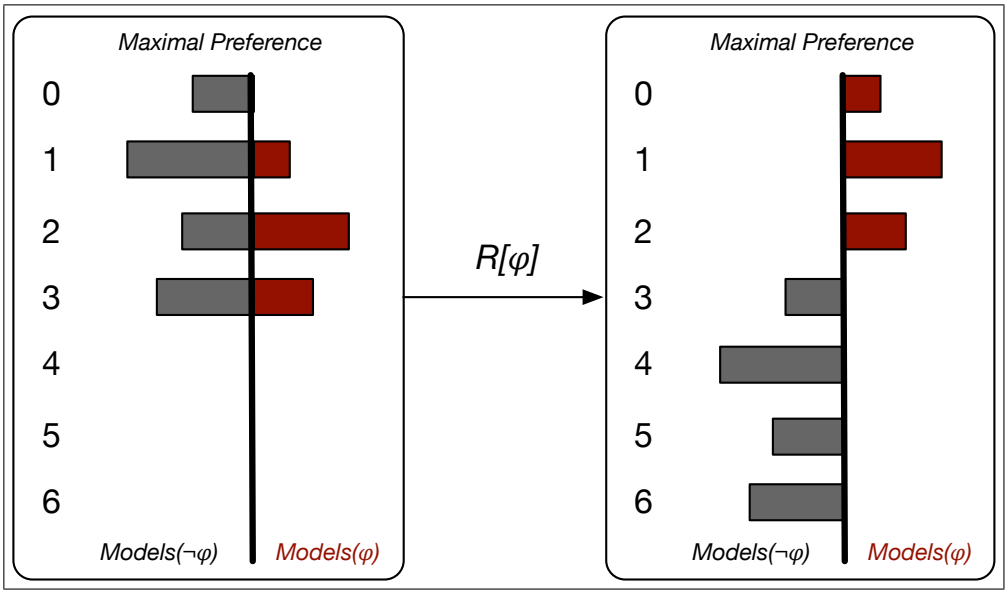
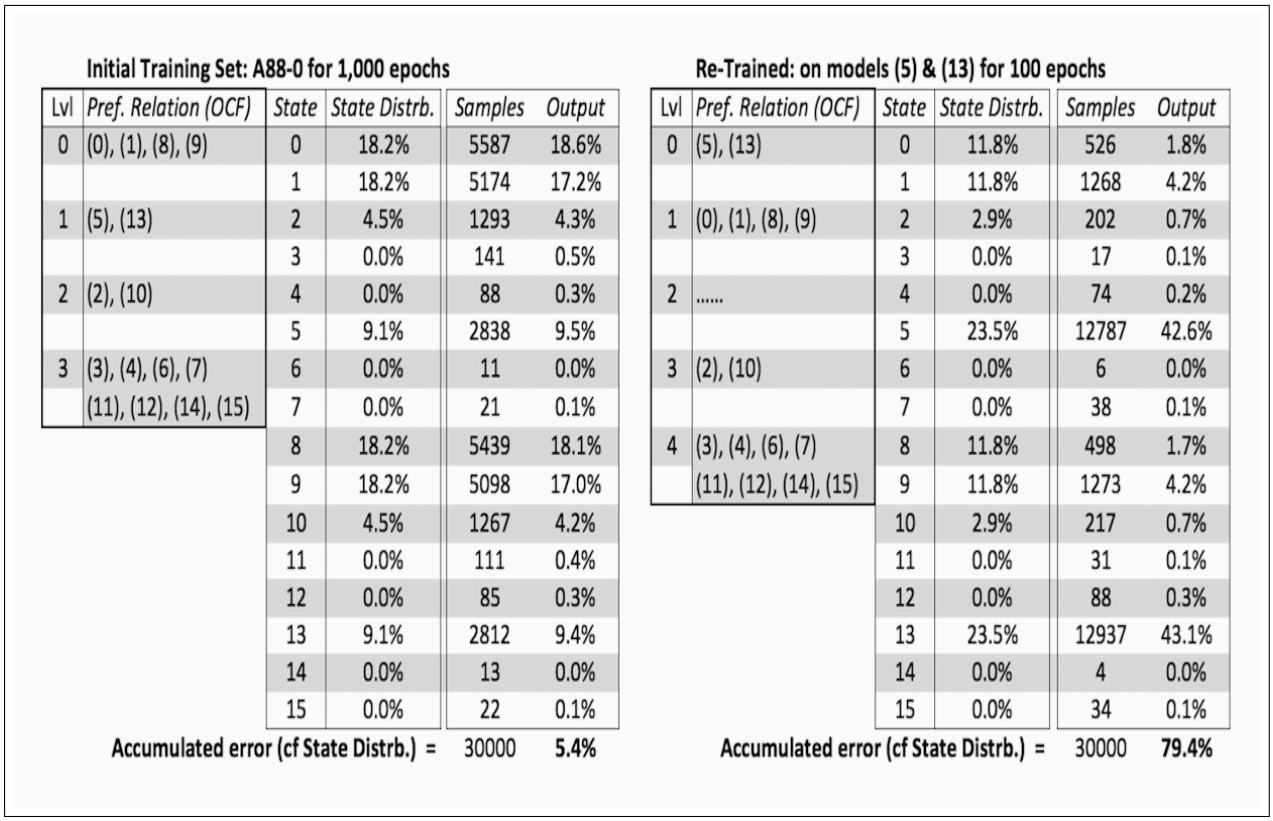


Figure 6.4: A generalisation of revision $R[\varphi]$, from the machine perspective, where the models and non-models are separated from one another along a spectrum by some degree of belief.

The traditional prioritised approach to revision assumes that the new information is reliable and should be believed regardless of any inconsistency with the previously learned default rule. In consequence, all of the satisfying models of the premiss are promoted to the highest level of preference. However this assumption, of strong prioritisation, has been subsequently modified, even to the extent of rejecting any inconsistent new information.

In the context of re-training a neural network, the idea of inserting a revised model at a specific level in the preference ranking is impractical, as the amount (number of

epochs) of re-training cannot be directly equated to the level of preference. Further, re-arranging the order within the satisfying or non-satisfying models would be impossible without separately re-training individual models. So, in general, for a neural network there is a simple continuum of re-training. At one extreme is no re-training; any new information is rejected, the models remain at their original preference levels. At the other extreme is catastrophic forgetting (Robins, 1995; McCloskey and Cohen, 1989; Ratcliff, 1990); the satisfying models are given maximal preference and the non-models are driven down the preference ranking towards zero.



Initial Training Set: A88-0 for 1,000 epochs

Lvl	*Pref. Relation (OCF)*	*State*	*State Distrb.*	*Samples*	*Output*
0	(0), (1), (8), (9)	0	18.2%	5587	18.6%
		1	18.2%	5174	17.2%
1	(5), (13)	2	4.5%	1293	4.3%
		3	0.0%	141	0.5%
2	(2), (10)	4	0.0%	88	0.3%
		5	9.1%	2838	9.5%
3	(3), (4), (6), (7)	6	0.0%	11	0.0%
	(11), (12), (14), (15)	7	0.0%	21	0.1%
		8	18.2%	5439	18.1%
		9	18.2%	5098	17.0%
		10	4.5%	1267	4.2%
		11	0.0%	111	0.4%
		12	0.0%	85	0.3%
		13	9.1%	2812	9.4%
		14	0.0%	13	0.0%
		15	0.0%	22	0.1%
Accumulated error (cf State Distrb.) =				30000	**5.4%**

Re-Trained: on models (5) & (13) for 100 epochs

Lvl	*Pref. Relation (OCF)*	*State*	*State Distrb.*	*Samples*	*Output*
0	(5), (13)	0	11.8%	526	1.8%
		1	11.8%	1268	4.2%
1	(0), (1), (8), (9)	2	2.9%	202	0.7%
		3	0.0%	17	0.1%
2	.....	4	0.0%	74	0.2%
		5	23.5%	12787	42.6%
3	(2), (10)	6	0.0%	6	0.0%
		7	0.0%	38	0.1%
4	(3), (4), (6), (7)	8	11.8%	498	1.7%
	(11), (12), (14), (15)	9	11.8%	1273	4.2%
		10	2.9%	217	0.7%
		11	0.0%	31	0.1%
		12	0.0%	88	0.3%
		13	23.5%	12937	43.1%
		14	0.0%	4	0.0%
		15	0.0%	34	0.1%
Accumulated error (cf State Distrb.) =				30000	**79.4%**

Figure 6.5: Re-training with catastrophic forgetting.

Consider the example in Figure 6.5, based on our previously specified 4-atom micro-world (Figure 3.1). On the left-hand side of the figure, a collection of 60 ILP-modified Boltzmann machines have been trained with data set 'A88-0' for 1,000 epochs. The representation of the original whole state preference relation is accurate, accumulated error ∼5%. The machines are then presented with new information, premiss "0 1 -1 1". This premiss has satisfying models "-1 1 -1 1" (decimal 5) and "1 1 -1 1" (decimal 13), the remainder of the models in the micro-world are non-satisfying models. On the right-hand side of Figure 6.5 the reader can observe the results of re-training for

only 100 epochs. The machines are almost entirely re-trained to the new information; satisfying models 5 & 13 both at more than 40% of the output and the non-satisfying models moved down towards zero. The accumulated error on the expected revised state distribution is high: ∼80%. This demonstrates that memory of the original distribution is destroyed after only a short period of re-training: catastrophic forgetting (Robins, 1995).

However, there is a point of balance, along this re-training continuum where a machine might be considered appropriately re-trained. Whilst difficult to achieve practically, it is identified in the logical approaches to revision. It is the moment when the models ($M[\alpha]$) and non-models ($M[\neg\alpha]$) are just separated. The satisfying models are at maximal preference and the non-satisfying models are pushed down to at least the lowest level of the satisfying models; maintaining their previous preference ordering. It can be appreciated, given the previous example of catastrophic forgetting after only 100 epochs of re-training, that this point of balance is critically narrow.

Reassuringly, two logical approaches to revision that recognise this balance point are quite disparate. These approaches were selected for investigation. One is the conventional, conservative, qualitative, non-prioritised revision of Booth and Meyer (2011) and the other is the contrasting non-traditional, radical, quantified, prioritised revision of Spohn (1988). With regard to the Spohnian revision we only intend considering a quantifier of one. That is, the non-models are pushed down by one 'level of belief', allowing the preference of satisfying model states to be maximised.

6.2.1 Machine Re-Training

For the experimental results which follow (Section 6.3: Experimental Results), the Boltzmann representation was trained on a preference relation from Figure 3.1, which for future reference will be abbreviated to $O[0, 1, 8, 9]$. New information was then introduced. The preference relation in logic was revised according to some of the current belief revision operators. The machine was re-trained using pseudo-rehearsal. The separately achieved preference relations: one from the logic and the other from the machine were then compared as a means of considering the machine plausibility of the different logical revision algorithms. This methodology is summarised in Figure 6.6.

Neural networks have difficulty maintaining old information when faced with new learning, whereas logic is conservative and favours minimal change. Initially, we attempted to avoid catastrophic forgetting during machine re-training by simply reducing

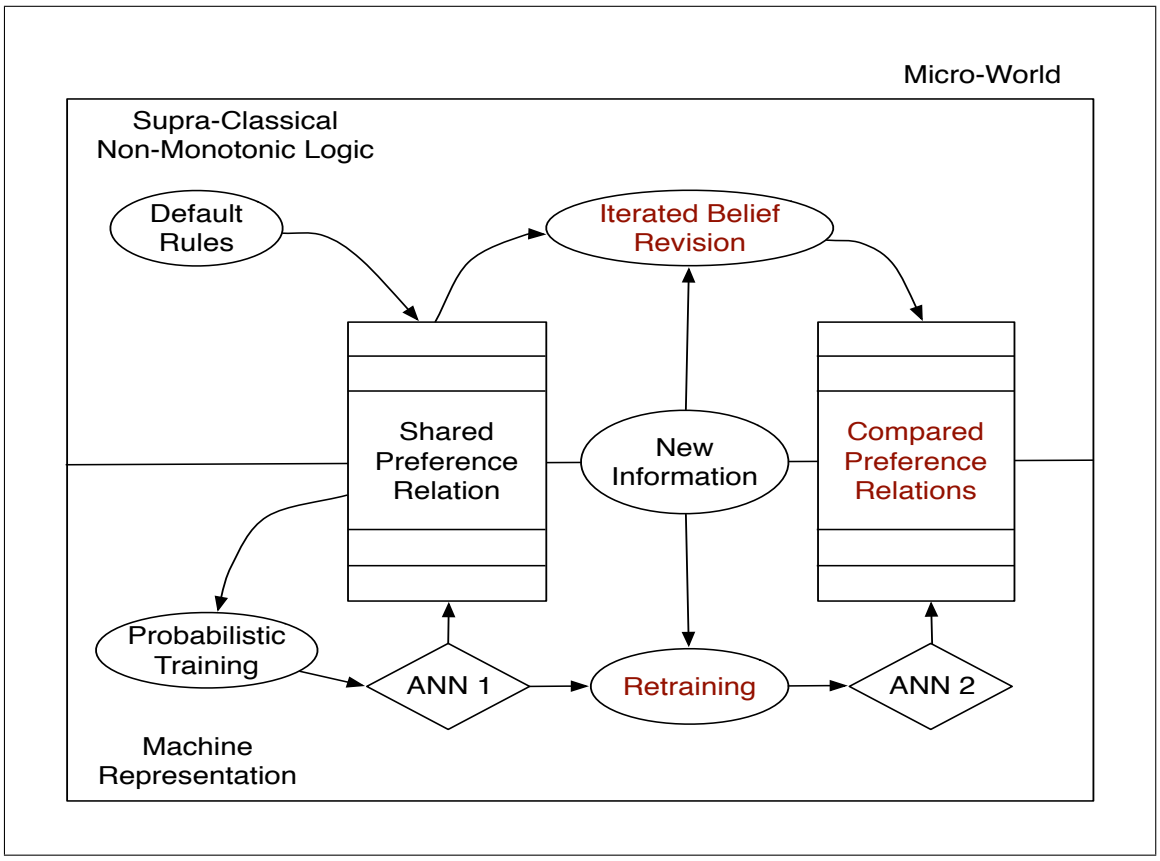


Figure 6.6: The comparison of logical belief revision vs re-training in a Boltzmann machine, within the context of a micro-world.

the number of training epochs with the new data set. This was partially effective but the training times were very short (10-30 epochs) and the results were not robust; they varied significantly between experiments.

The Boltzmann machine has many characteristics that are biologically attractive. The bi-phasic nature of the Boltzmann learning algorithm makes it inherently suitable for pseudo-rehearsal (Robins, 1995). In this context, pseudo-rehearsal can be achieved by double cycling of the machine's unclamped-free phase. Normally, the Boltzmann machine learning algorithm alternates between cycles of clamping, where external inputs are applied and 'positive' cross-firing statistics are collected and free phases, where there is no input, the machine is annealed and 'negative' cross-firing statistics are collected. By placing two free phases back-to-back the machine can be made to retrieve its previously learned patterns. When this re-cycling (free & free phase) is combined, in a suitable proportion, with normal cycling (active & free phase), the machine in-effect integrates the new patterns into the old training set. In the experimental results, which follow, the proportion of re-cycling is estimated empirically.

Initial Training Set: A88-0

Lvl	Pref. Relation (OCF)	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(0), (1), (8), (9)	0	18.2%	16.9%	5587	18.6%
		1	18.2%	20.3%	5174	17.2%
1	(5), (13)	2	4.5%	1.7%	1293	4.3%
		3	0.0%	2.0%	141	0.5%
2	(2), (10)	4	0.0%	3.8%	88	0.3%
		5	9.1%	4.5%	2838	9.5%
3	(3), (4), (6), (7)	6	0.0%	0.4%	11	0.0%
	(11), (12), (14), (15)	7	0.0%	0.5%	21	0.1%
		8	18.2%	16.9%	5439	18.1%
		9	18.2%	20.3%	5098	17.0%
		10	4.5%	1.7%	1267	4.2%
		11	0.0%	2.0%	111	0.4%
		12	0.0%	3.8%	85	0.3%
		13	9.1%	4.5%	2812	9.4%
		14	0.0%	0.4%	13	0.0%
		15	0.0%	0.5%	22	0.1%

Accumulated error (cf State Distrb.) = 30000 **5.4%**

Accumulated error (cf Atomic Distrb.) = **35.8%**

Re-Cycled: 100%

Lvl	Pref. Relation (OCF)	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(0), (1), (8), (9)	0	18.2%	16.9%	5452	18.2%
		1	18.2%	20.3%	5323	17.7%
1	(5), (13)	2	4.5%	1.7%	1432	4.8%
		3	0.0%	2.0%	134	0.4%
2	(2), (10)	4	0.0%	3.8%	168	0.6%
		5	9.1%	4.5%	2565	8.6%
3	(3), (4), (6), (7)	6	0.0%	0.4%	23	0.1%
	(11), (12), (14), (15)	7	0.0%	0.5%	30	0.1%
		8	18.2%	16.9%	5415	18.1%
		9	18.2%	20.3%	5065	16.9%
		10	4.5%	1.7%	1434	4.8%
		11	0.0%	2.0%	136	0.5%
		12	0.0%	3.8%	157	0.5%
		13	9.1%	4.5%	2604	8.7%
		14	0.0%	0.4%	21	0.1%
		15	0.0%	0.5%	41	0.1%

Accumulated error (cf State Distrb.) = 30000 **5.8%**

Accumulated error (cf Atomic Distrb.) = **33.5%**

Figure 6.7: Initial training followed by re-cycling. These results, from 60 retrained ILP Boltzmann machines, indicate that there is very little 'drift' in the initial learned distribution when re-cycling over 200 epochs.

This implementation permitted us to lengthen the training train times and achieve more stable results. Figure 6.7 shows the effect of re-cycling over 200 epochs in previously trained machines. After the initial training, on the left hand-side of the figure, the machine accurately retrieves the $O[0, 1, 8, 9]$ preference relation. Following re-cycling for 200 epochs, on the right hand-side of the figure, there is a small drift from the original learned distribution; as a consequence of the stochastic nature of the annealing process.

6.3 Experimental Results

6.3.1 Method

Practically, there are only three ways of selecting satisfying models to take part in any revision, i.e. only three varieties of new information are available for revision. This arises because the satisfying models to be maximised can only be selected from a lower intermediate level of preference or the lowest level of preference or a combination of both. This categorisation relates back to the principle of rational monotonicity (Section 4.3.4: Rational Monotonicity):

Given α defeasibly entails $\beta : \alpha \mid\sim \beta$, and new information γ is added to α, the following can occur:

Simple Revisions: promote models whose information was already entailed within the knowledge base where entailment of β holds: $(\alpha \wedge \gamma) \mid\sim \beta$. The promoted models are less preferred models, partially consistent with the default rule.

Inconsistent Revisions: promote least preferred models whose surprising information was not entailed within the knowledge base, where entailment of β does not hold: $(\alpha \wedge \gamma) \not\mid\sim \beta$, as α defeasibly entails not $\gamma : \alpha \mid\sim \neg\, \gamma$. The promoted models are inconsistent with the default rule.

Differential Revision: promotes satisfying models from disparate levels of preference, both intermediate & lowest levels. Some promoted models will be partially consistent with the knowledge base and others will be entirely inconsistent.

The rationalisations above allowed us to restrict machine re-training to a small representative selection of logical revisions. For these comparisons we have considered the two contrasting approaches of Booth & Meyer versus Spohn. Spohnian revision preserves empty ordinal levels of preference.

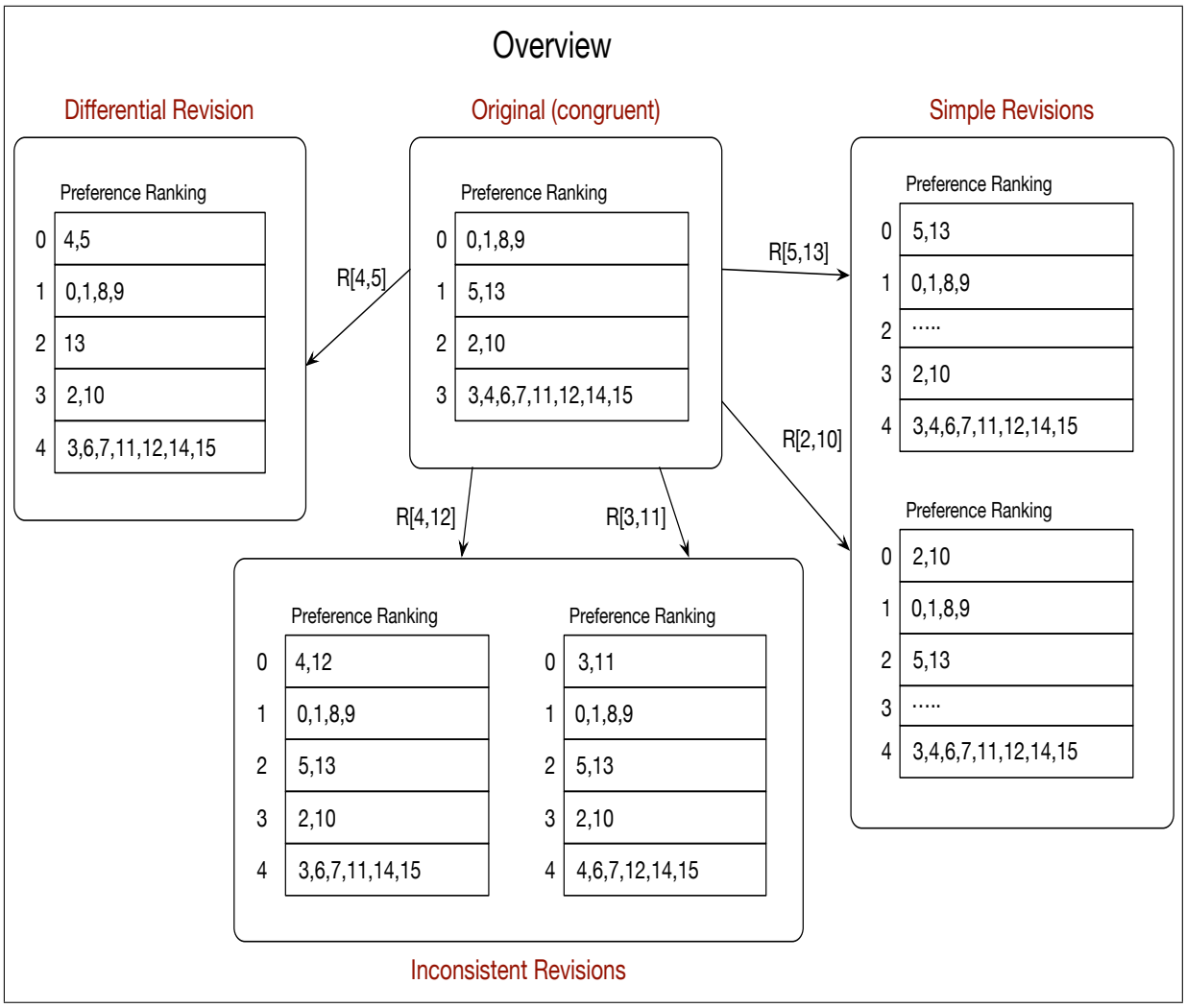


Figure 6.8: An overview for our experimental investigation, utilising three different categories of revision: where $R[x, y]$ identifies a revision by the set of satisfying models.

A framework for the experimental results is provided in Figure 6.8. Each individual revision is identified by an abbreviation specifying the satisfying models, rather than the premiss containing the new information. All the revisions examined begin from the congruent world abbreviated to $O[0, 1, 8, 9]$. Assumptions had to be made regarding the relative expected numerical preferences to be assigned to the satisfying models following revision. In the original world $O[0, 1, 8, 9]$, the relative numerical preferences in the training set were assigned on an exponential scale (Figure 3.1). For the experimental results that follow, an expected exponential scale was assumed in the revision preferences, based on the exponential scale in the original training set.

During the retraining process: the length (epochs) of initial training, the proportion of re-cycling (pseudo-rehearsal) and the length (epochs) of re-training were varied. Re-cycling was implemented by randomly interleaving, with a probability equal to the required proportion. So, for example, if the desired re-cycling proportion was 80%, then before each machine cycle a random number was generated between 0.00 and 1.00; if the number was ≤ 0.80 the machine was re-cycled. In Table 6.1 and Table 6.2 below, the correct proportion of re-cycling had to be judged empirically, based on the fit to the whole state distribution (accumulated error 's') or the atomic distribution (accumulated error 'a').

It was thought that shorter initial training times in the original $O[0, 1, 8, 9]$ world, might allow the machine to generalise better. Experiments varying the initial training lengths from 500 through to 2000 epochs confirmed that halting the initial training early at $\sim$500 epochs, prior to plateauing of the weights, achieved the best results. As a consequence, a two dimensional search space had to be partially explored, to find the machine result closest to the expected outcome of each logical revision. Table 6.1 & Table 6.2 illustrate these search spaces. In the case of the simple revision $R[5, 13]$, the machine achieves a close approximation of the state distribution (accumulated error 's'). However, in the case of the inconsistent revision $R[3, 11]$ the machine does not approximate the state distribution, but achieves a close approximation of the atomic distribution (accumulated error 'a').

In Sections 6.3.2 – 6.3.4 that follow, we present examples of the best results for each revision rather than overload the reader with un-helpful data. For each example result, we first demonstrate that the machine representation is capable of learning the expected revised distribution on initial training, based on the complete final preference relation. Finally, the patterns related to each level of revision are summarised and analysed in Section 6.3.5.

Simple Revision R[5, 13]						
Re-Cycling	*(s) 100 Epochs (a)*		*(s) 150 Epochs (a)*		*(s) 200 Epochs (a)*	
60%	49%	70%				
65%	40%	62%				
70%	33%	54%	52%	74%		
75%	24%	47%	42%	63%	55%	76%
80%	18%	44%	34%	55%	44%	65%
85%	22%	47%	16%*	43%	29%	53%
90%	28%	31%	23%	37%	22%	38%

Table 6.1: The search space for a simple revision: R[5, 13]. Accumulated errors are shown relative to the state distribution (s) and the atomic distribution (a). The best example (16%*) is illustrated in Section 6.3.2: example Simple Revisions.

Inconsistent Revision R[3, 11]						
Re-Cycling	*(s) 100 Epochs (a)*		*(s) 150 Epochs (a)*		*(s) 200 Epochs (a)*	
55%	39%	32%				
60%	34%	19%				
65%	40%	13%	41%	46%		
70%	48%	16%	34%	30%	47%	55%
75%	52%	21%	38%	17%	37%	36%
80%	72%	59%	48%	13%*	32%	24%
85%			51%	36%	50%	20%
90%			65%	54%	56%	41%

Table 6.2: The search space for inconsistent revision: R[3, 11]. Accumulated errors are shown relative to the state distribution (s) and the atomic distribution (a). The best example (13%*) is illustrated in Section 6.3.3: example Inconsistent Revisions.

6.3.2 Simple Revisions

We report two example simple revisions from intermediate levels of preference, $R[5, 13]$ and $R[2, 10]$. Both the Booth & Meyer and Sphonian approaches produce similar results for these logical revisions. The expected Spohnian preference relations, illustrated in Figures 6.9 & 6.10, have empty ordinal levels at 2 and 3 respectively. These simple revisions introduce new information, which is partially consistent with the conditional constraints previously learned by the machine.

Considering revision $R[5, 13]$, Figure 6.9. Firstly, when given the whole data set 'E144-0a', the machine can accurately learn the preference relation: accumulated error $\sim$5%. When re-trained (revised) from the original preference relation of $O[0, 1, 8, 9]$, the satisfying models (decimal) 5 & 13 are correctly moved to maximal preference at $\sim$23% of the output distribution. The non-satisfying models at least preference stay at almost 0% and models (decimal) 2 & 10 are both at the same low level, at $\sim$2% of the output distribution. However, models (decimal) 0, 1, 8 & 9 have output proportions: 0 & 8 at $\sim$9% and 1 & 9 at $\sim$12%, divergent from the expected state preferences. Similarly, model states (decimal) 4 & 12 have outputs $\sim$2%, divergent from the state distribution. In this revision, the third highest and lowest order bits are likely to be set (Fan on, Window open), making states 1 & 9 and 4 & 12 atomically more probable. The output of these isolated, conditionally constrained states is closer to their atomic distribution. Therefore, the overall fit of the revised representation to the state distribution is worse than expected from basic training on the whole data set: accumulated error of $\sim$16%.

Similarly, considering revision $R[2, 10]$, Figure 6.10. Firstly, when given the whole data set 'E144-0b', the machine can accurately learn the preference relation: accumulated error $\sim$9%. When re-trained (revised) the satisfying models 2 & 10 are correctly moved to maximal preference at $\sim$20% of the output distribution. The non-satisfying models stay at least preference, $\sim$0% and models 5 & 13 are both at the same low level, at $\sim$5%. However, models 0, 1, 8 & 9 have output proportions: 0 & 8 at $\sim$14% and 1 & 9 at $\sim$8%, divergent from the expected state preferences. Similarly, model states 3 & 11 have outputs slightly above their expected state distribution. In this revision, the second highest order bit is likely to be set and the lowest order bit is likely to be cleared (Heater on, Window closed). Here too, the output of isolated, conditionally constrained states is closer to their atomic distribution. Therefore, the overall fit of the revised representation to the state distribution is worse than expected from basic training on the whole data set: accumulated error of $\sim$21%.

Initial Training Set: E144-0a

Lvl	Pref. Relation (OCF)	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(5), (13)	0	11.8%	7.3%	3515	11.7%
		1	11.8%	17.6%	3396	11.3%
1	(0), (1), (8), (9)	2	2.9%	0.5%	744	2.5%
		3	0.0%	1.1%	61	0.2%
2	.....	4	0.0%	6.5%	75	0.3%
		5	23.5%	15.6%	7292	24.3%
3	(2), (10)	6	0.0%	0.4%	5	0.0%
		7	0.0%	1.0%	23	0.1%
4	(3), (4), (6), (7)	8	11.8%	7.3%	3450	11.5%
	(11), (12), (14), (15)	9	11.8%	17.6%	3241	10.8%
		10	2.9%	0.5%	827	2.8%
		11	0.0%	1.1%	71	0.2%
		12	0.0%	6.5%	60	0.2%
		13	23.5%	15.6%	7208	24.0%
		14	0.0%	0.4%	4	0.0%
		15	0.0%	1.0%	28	0.1%

Accumulated error (cf State Distrb.) = 30000 4.8%

Accumulated error (cf Atomic Distrb.) = 59.9%

Retraining: 85% re-cycled for 150Ep

Lvl	Pref. Relation (OCF)	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(5), (13)	0	11.8%	7.3%	2702	9.0%
		1	11.8%	17.6%	3576	11.9%
1	(0), (1), (8), (9)	2	2.9%	0.5%	639	2.1%
		3	0.0%	1.1%	269	0.9%
2	.....	4	0.0%	6.5%	606	2.0%
		5	23.5%	15.6%	7033	23.4%
3	(2), (10)	6	0.0%	0.4%	65	0.2%
		7	0.0%	1.0%	287	1.0%
4	(3), (4), (6), (7)	8	11.8%	7.3%	2770	9.2%
	(11), (12), (14), (15)	9	11.8%	17.6%	3527	11.8%
		10	2.9%	0.5%	633	2.1%
		11	0.0%	1.1%	274	0.9%
		12	0.0%	6.5%	573	1.9%
		13	23.5%	15.6%	6716	22.4%
		14	0.0%	0.4%	62	0.2%
		15	0.0%	1.0%	268	0.9%

Accumulated error (cf State Distrb.) = 30000 16.3%

Accumulated error (cf Atomic Distrb.) = 42.9%

Figure 6.9: Revision [5, 13]. These results show accurate representation of the preference relation after basic training with the whole data set. Compared to an altered conditional representation after retraining.

Initial Training Set: E144-0b

Lvl	Pref. Relation (OCF)	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(2), (10)	0	11.1%	16.5%	3348	11.2%
		1	11.1%	8.2%	3563	11.9%
1	(0), (1), (8), (9)	2	22.2%	13.2%	6020	20.1%
		3	0.0%	6.6%	120	0.4%
2	(5), (13)	4	0.0%	2.1%	66	0.2%
		5	5.6%	1.0%	1827	6.1%
3	.....	6	0.0%	1.6%	47	0.2%
		7	0.0%	0.8%	24	0.1%
4	(3), (4), (6), (7)	8	11.1%	16.5%	3214	10.7%
	(11), (12), (14), (15)	9	11.1%	8.2%	3641	12.1%
		10	22.2%	13.2%	6050	20.2%
		11	0.0%	6.6%	174	0.6%
		12	0.0%	2.1%	56	0.2%
		13	5.6%	1.0%	1778	5.9%
		14	0.0%	1.6%	37	0.1%
		15	0.0%	0.8%	35	0.1%
					30000	**9.1%**

Accumulated error (cf State Distrb.) =

Accumulated error (cf Atomic Distrb.) = **62.8%**

Retraining: 75% re-cycled for 150Ep

Lvl	Pref. Relation (OCF)	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(2), (10)	0	11.1%	16.5%	4075	13.6%
		1	11.1%	8.2%	2318	7.7%
1	(0), (1), (8), (9)	2	22.2%	13.2%	5765	19.2%
		3	0.0%	6.6%	284	0.9%
2	(5), (13)	4	0.0%	2.1%	122	0.4%
		5	5.6%	1.0%	1497	5.0%
3	.....	6	0.0%	1.6%	64	0.2%
		7	0.0%	0.8%	59	0.2%
4	(3), (4), (6), (7)	8	11.1%	16.5%	4586	15.3%
	(11), (12), (14), (15)	9	11.1%	8.2%	2531	8.4%
		10	22.2%	13.2%	6437	21.5%
		11	0.0%	6.6%	381	1.3%
		12	0.0%	2.1%	140	0.5%
		13	5.6%	1.0%	1603	5.3%
		14	0.0%	1.6%	78	0.3%
		15	0.0%	0.8%	60	0.2%
					30000	**21.3%**

Accumulated error (cf State Distrb.) =

Accumulated error (cf Atomic Distrb.) = **45.7%**

Figure 6.10: Revision [2, 10]. These results show accurate representation of the preference relation after basic training with the whole data set. Compared to an altered conditional representation after retraining.

6.3.3 Inconsistent Revisions

We report two example inconsistent revisions from lowest levels of preference $R[3, 11]$ and $R[4, 12]$. Both the Booth & Meyer and Sphonian approaches produce similar results for these revisions; in this case there are no empty ordinal levels for the Spohnian revision, Figures 6.11 & 6.12. These difficult revisions promote satisfying models, which are inconsistent with the conditional constraints previously learned by the machine, from the original congruent $O[0, 1, 8, 9]$ world.

Considering revision $R[3, 11]$, Figure 6.11. Firstly, when given the whole data set 'E152-0a', the machine is able to learn the preference relation with only moderate accuracy: accumulated error ∼24%, average error per state ∼1.5%. The maximally preferred models 3 & 11 are correctly placed with an output proportion at ∼18%. All the models at lower levels of preference: 5 & 13 and 2 & 10 are suitably represented including those at least preference. However, the second level of maximal models 0, 1, 8 & 9 have output proportions that are (already) close to their atomic probabilities: 1 & 9 at ∼14% and 0 & 8 at ∼9%. The low order bit is likely to be set (Window open). The atomic, conditional constraints vary within these models.

When re-trained (revised) from the original preference relation of $O[0, 1, 8, 9]$, both satisfying and non-satisfying models are further adjusted to match their atomic probabilities. Maximal models 3 & 11 are adjusted down to sit just below models 1 & 9, which are preferred, based on the atomic distribution. Similarly even models in the lower orders of preference, for example 2 & 10 vs 5 & 13, are adjusted to match their atomic probabilities. Overall the accumulated error relative to the state distribution is very high: ∼44%, but the machine results are very close to the atomic distribution: with an accumulated error, relative to this distribution, of ∼12% and average error per state of only ∼0.8%.

Similarly, considering revision $R[4, 12]$, Figure 6.12, when initially trained on the whole data set the machine has only a moderate accumulated error ∼14%. However, the second level of maximal models 0, 1, 8 & 9 have output proportions close to their atomic probabilities: 0 & 8 at ∼12% and 1 & 9 at ∼9%. The low order bit is likely to be cleared (Window closed). Again, the atomic, conditional constraints vary within these models.

When re-trained (revised), both satisfying and non-satisfying models are further adjusted to match their atomic probabilities. Overall, the accumulated error relative to the state distribution is very high: ∼44%, but relative to the atomic distribution is very small: accumulated error ∼9%: average error per state of only ∼0.6%.

Initial Training Set: E152-0a

Lvl	Pref. Relation (OCF)	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(3), (11)	0	10.5%	6.2%	2752	9.2%
		1	10.5%	17.3%	4306	14.4%
1	(0), (1), (8), (9)	2	2.6%	5.6%	761	2.5%
		3	21.1%	15.6%	5474	18.2%
2	(5), (13)	4	0.0%	0.7%	656	2.2%
		5	5.3%	2.0%	991	3.3%
3	(2), (10)	6	0.0%	0.7%	12	0.0%
		7	0.0%	1.8%	82	0.3%
4	(4), (6), (7)	8	10.5%	6.2%	2833	9.4%
	(12), (14), (15)	9	10.5%	17.3%	4155	13.9%
		10	2.6%	5.6%	777	2.6%
		11	21.1%	15.6%	5411	18.0%
		12	0.0%	0.7%	664	2.2%
		13	5.3%	2.0%	1036	3.5%
		14	0.0%	0.7%	14	0.0%
		15	0.0%	1.8%	76	0.3%
					30000	**24.3%**

Accumulated error (cf State Distrb.) =

Accumulated error (cf Atomic Distrb.) = **33.8%**

Retraining: 65% re-cycled for 150Ep

Lvl	Pref. Relation (OCF)	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(3), (11)	0	10.5%	6.2%	1981	6.6%
		1	10.5%	17.3%	5593	18.6%
1	(0), (1), (8), (9)	2	2.6%	5.6%	1078	3.6%
		3	21.1%	15.6%	5062	16.9%
2	(5), (13)	4	0.0%	0.7%	81	0.3%
		5	5.3%	2.0%	816	2.7%
3	(2), (10)	6	0.0%	0.7%	34	0.1%
		7	0.0%	1.8%	443	1.5%
4	(4), (6), (7)	8	10.5%	6.2%	2019	6.7%
	(12), (14), (15)	9	10.5%	17.3%	5420	18.1%
		10	2.6%	5.6%	1173	3.9%
		11	21.1%	15.6%	4910	16.4%
		12	0.0%	0.7%	102	0.3%
		13	5.3%	2.0%	788	2.6%
		14	0.0%	0.7%	48	0.2%
		15	0.0%	1.8%	452	1.5%
					30000	**43.5%**

Accumulated error (cf State Distrb.) =

Accumulated error (cf Atomic Distrb.) = **12.5%**

Figure 6.11: Revision [3, 11]. These results show a difficulty representing the preference relation after basic training with the whole data set. After retraining, the machine accurately represents the atomic distribution.

Initial Training Set: E152-0b

Lvl	Pref. Relation (OCF)	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(4), (12)	0	10.5%	15.4%	3733	12.4%
		1	10.5%	7.1%	2649	8.8%
1	(0), (1), (8), (9)	2	2.6%	0.9%	428	1.4%
		3	0.0%	0.4%	286	1.0%
2	(5), (13)	4	21.1%	17.1%	6225	20.8%
		5	5.3%	7.9%	1593	5.3%
3	(2), (10)	6	0.0%	0.9%	102	0.3%
		7	0.0%	0.4%	20	0.1%
4	(3), (6), (7)	8	10.5%	15.4%	3788	12.6%
	(11), (14), (15)	9	10.5%	7.1%	2696	9.0%
		10	2.6%	0.9%	428	1.4%
		11	0.0%	0.4%	299	1.0%
		12	21.1%	17.1%	6125	20.4%
		13	5.3%	7.9%	1521	5.1%
		14	0.0%	0.9%	91	0.3%
		15	0.0%	0.4%	16	0.1%
				Accumulated error (cf State Distrb.) =	30000	13.6%
				Accumulated error (cf Atomic Distrb.) =		26.0%

Retraining: 65% re-cycled for 100Ep

Lvl	Pref. Relation (OCF)	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(4), (12)	0	10.5%	15.4%	4293	14.3%
		1	10.5%	7.1%	1937	6.5%
1	(0), (1), (8), (9)	2	2.6%	0.9%	475	1.6%
		3	0.0%	0.4%	186	0.6%
2	(5), (13)	4	21.1%	17.1%	5284	17.6%
		5	5.3%	7.9%	2301	7.7%
3	(2), (10)	6	0.0%	0.9%	419	1.4%
		7	0.0%	0.4%	127	0.4%
4	(3), (6), (7)	8	10.5%	15.4%	4283	14.3%
	(11), (14), (15)	9	10.5%	7.1%	1840	6.1%
		10	2.6%	0.9%	516	1.7%
		11	0.0%	0.4%	198	0.7%
		12	21.1%	17.1%	5405	18.0%
		13	5.3%	7.9%	2247	7.5%
		14	0.0%	0.9%	356	1.2%
		15	0.0%	0.4%	133	0.4%
				Accumulated error (cf State Distrb.) =	30000	33.8%
				Accumulated error (cf Atomic Distrb.) =		8.6%

Figure 6.12: Revision [4, 12]. These results show a difficulty representing the preference relation after basic training with the whole data set. After retraining, the machine accurately represents the atomic distribution.

6.3.4 Differential Revision

The premiss for a differential revision entails models which are both partially consistent and those which are entirely inconsistent with the pre-existing knowledge. The procedure for managing these different types of models can promote both to the same level of preference or promote them while maintaining their relative order (and separation) of preference.

The traditional strictly prioritised approach is to move all the models satisfying the premiss to the maximal preference. An example preference relation utilising a Spohnian revision is presented in Figure 6.13. On the left-hand side of the figure, the results demonstrate adequate learning on the whole data set. The output frequencies of models 4 & 5 are similar (19 – 21%) and the output frequencies of models 0, 1, 8 & 9 are similar (11 – 13%). However, when the machine reaches this final distribution by re-training, on the right-hand side of the figure, it is unable to represent the expected distribution. The output frequencies of the maximally preferred models 4 & 5 are now 15% & 31% respectively.

The non-prioritised approach is to preserve the relative order (and separation) of the different models: partially consistent and entirely inconsistent. An example of this approach utilising a revision of Booth & Meyer is presented in Figure 6.14. On the left-hand side of the figure, the results demonstrate adequate learning on the whole data set. The output frequencies of models 4 & 5 are 17% & 37% respectively, close to their expected frequencies. The output frequencies of models 0, 1, 8 & 9 are similar (8 – 11%), as expected. When the machine reaches this final distribution by re-training, on the right-hand side of the figure, it still provides a reasonable representation of the expected distribution. The output frequencies of the maximally preferred models 4 & 5 are still acceptable, 18% & 34% respectively. However, the output frequencies of model states at second preference, 0 & 1 (9 – 14%) vs 8 & 9 (3 – 5%) are adjusted to reflect the expected atomic distribution. The conditional constraints vary within these models, where it is likely that the high order bit (Light) is cleared.

To represent prioritised revision in this context, where models at different levels of preference are being promoted, a learning machine would have to be differentially trained on individual models. This would defeat the concept of entailment by a single premiss. During re-training, the Boltzmann machine is only able to maintain the relative ordering and separation of the satisfying models and similarly for the non-satisfying models.

Initial Training Set: E144-0c

Lvl	Pref. Relation (OCF)	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(4), (5)	0	11.1%	16.4%	3935	13.1%
		1	11.1%	16.4%	3273	10.9%
1	(0), (1), (8), (9)	2	2.8%	1.0%	445	1.5%
		3	0.0%	1.0%	298	1.0%
2	(13)	4	22.2%	16.4%	5628	18.8%
		5	22.2%	16.4%	6421	21.4%
3	(2), (10)	6	0.0%	1.0%	122	0.4%
		7	0.0%	1.0%	88	0.3%
4	(3), (6), (7)	8	11.1%	7.2%	3741	12.5%
	(11), (12), (14), (15)	9	11.1%	7.2%	3277	10.9%
		10	2.8%	0.4%	450	1.5%
		11	0.0%	0.4%	364	1.2%
		12	0.0%	7.2%	811	2.7%
		13	5.6%	7.2%	1089	3.6%
		14	0.0%	0.4%	33	0.1%
		15	0.0%	0.4%	25	0.1%
			Accumulated error (cf State Distrb.) =		30000	18.3%
			Accumulated error (cf Atomic Distrb.) =			37.5%

Retraining: 70% re-cycled for 150Ep

Lvl	Pref. Relation (OCF)	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(4), (5)	0	11.1%	16.4%	4886	16.3%
		1	11.1%	16.4%	3168	10.6%
1	(0), (1), (8), (9)	2	2.8%	1.0%	704	2.3%
		3	0.0%	1.0%	213	0.7%
2	(13)	4	22.2%	16.4%	4394	14.6%
		5	22.2%	16.4%	9206	30.7%
3	(2), (10)	6	0.0%	1.0%	307	1.0%
		7	0.0%	1.0%	310	1.0%
4	(3), (6), (7)	8	11.1%	7.2%	1974	6.6%
	(11), (12), (14), (15)	9	11.1%	7.2%	1159	3.9%
		10	2.8%	0.4%	289	1.0%
		11	0.0%	0.4%	113	0.4%
		12	0.0%	7.2%	972	3.2%
		13	5.6%	7.2%	2129	7.1%
		14	0.0%	0.4%	93	0.3%
		15	0.0%	0.4%	83	0.3%
			Accumulated error (cf State Distrb.) =		30000	44.3%
			Accumulated error (cf Atomic Distrb.) =			32.7%

Figure 6.13: Revision [4, 5] strictly prioritised. These results show a failure to adequately represent the prioritised revision; as demonstrated in the most preferred states by the discrepancy between the output frequencies of states 4 and 5 (15% & 31%) and their shared expected frequency (22%).

Initial Training Set: E176-0

Lvl	Pref. Relation (OCF)	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(5)	0	9.1%	12.0%	3174	10.6%
		1	9.1%	17.3%	2390	8.0%
1	(4)	2	2.3%	0.6%	369	1.2%
		3	0.0%	0.8%	316	1.1%
2	(0), (1), (8), (9)	4	18.2%	17.3%	4965	16.6%
		5	36.4%	25.0%	11096	37.0%
3	(13)	6	0.0%	0.8%	15	0.1%
		7	0.0%	1.2%	25	0.1%
4	(2), (10)	8	9.1%	4.0%	3034	10.1%
		9	9.1%	5.8%	2485	8.3%
5	(3), (6), (7)	10	2.3%	0.2%	395	1.3%
	(11), (12), (14), (15)	11	0.0%	0.3%	318	1.1%
		12	0.0%	5.8%	460	1.5%
		13	4.5%	8.3%	955	3.2%
		14	0.0%	0.3%	1	0.0%
		15	0.0%	0.4%	2	0.0%

Accumulated error (cf State Distrb.) = 30000 **13.8%**

Accumulated error (cf Atomic Distrb.) = **46.8%**

Retraining: 65% re-cycled for 150Ep

Lvl	Pref. Relation (OCF)	State	State Distrb.	Atomic Distrb.	Samples	Output
0	(5)	0	9.1%	12.0%	4137	13.8%
		1	9.1%	17.3%	2770	9.2%
1	(4)	2	2.3%	0.6%	579	1.9%
		3	0.0%	0.8%	166	0.6%
2	(0), (1), (8), (9)	4	18.2%	17.3%	5520	18.4%
		5	36.4%	25.0%	10242	34.1%
3	(13)	6	0.0%	0.8%	396	1.3%
		7	0.0%	1.2%	360	1.2%
4	(2), (10)	8	9.1%	4.0%	1435	4.8%
		9	9.1%	5.8%	957	3.2%
5	(3), (6), (7)	10	2.3%	0.2%	203	0.7%
	(11), (12), (14), (15)	11	0.0%	0.3%	96	0.3%
		12	0.0%	5.8%	983	3.3%
		13	4.5%	8.3%	1991	6.6%
		14	0.0%	0.3%	87	0.3%
		15	0.0%	0.4%	78	0.3%

Accumulated error (cf State Distrb.) = 30000 **28.7%**

Accumulated error (cf Atomic Distrb.) = **30.5%**

Figure 6.14: Revision [4, 5] relatively non-prioritised. These results show a moderately adequate representation of the non-prioritised revision; as demonstrated in the most preferred states by the similarity between the output frequencies of states 4 and 5 (18% & 34%) and their expected frequencies (18% & 36%).

6.3.5 Discussion of Results

For simple revisions, the Boltzmann machine is able to accurately learn the whole data set on basic training. On revision, the new information is partially consistent with the previous default rule. Small changes contained in the new conditional constraints result in variations in the atomic distribution between models occupying the same level of state preference. Only these isolated models are adjusted to their atomic distributions. Therefore, only minor differences from the expected whole state distribution occur: whole state preference prevails.

An inside the box analysis, looking at the distribution of atomic probabilities in the training set compared to the network weights, was performed for the simple revision $R[2, 10]$. By a method identical to the analysis done for the original $O[0, 1, 8, 9]$ world in Section 5.2.4. There were four different patterns of machines. The majority of these machines had two patterns which were mirror images of each other; 48 of the total 60 machines tested. The analysis of these machine patterns indicated that the bias weights on the output layer remained representative of the new individual atomic frequencies. Remember, that these bias weights are revised from their original learned values: so changes in individual atomic probabilities are compared to changes in the bias weights, see Table 6.3. The bias weight on the hidden node likewise moved from the original -2 (Section 5.2.4) to +7, in keeping with the revised proportion of heating in the micro-world. These simple revisions include adjustments in constraints, which the machine was able to incorporate into its previously learnt model / representation. The single hidden node continued to act as a feature detector, or linear conceptual space, following revision.

Atomic Distribution vs Output Bias following Revision			
Node	*Atomic Probability Change*		*Output Bias Change*
Light	0%	None	0
Fan	-7%	Slightly negative	-3
Heater	+35%	Highly positive	+20
Window	-22%	Moderate negative	-11

Table 6.3: Comparison of the change in probability of output node activation to the change in bias weights in the output layer, as a representation of the atomic distribution, following the revision $R[2, 10]$.

For inconsistent revisions, minor differences from the expected whole state distribution are already present on initial training with the complete data set. On revision, major inconsistencies between the previous default rule and the new conditional constraints result in the machine being unable to represent the whole state preference relation. In the examples, there is inconsistency between the state preference ordering and atomic typicality on every level, even involving the maximally preferred models. Under these conditions of inconsistent revision, the network accurately represents the atomic distribution: atomic typicality prevails.

Atomic Distribution vs Output Bias following Revision			
Node	*Atomic Probability Change*		*Output Bias Change*
Light	0%	None	0
Fan	+35%	Highly positive	+32
Heater	-4%	Slightly negative	+7
Window	-23%	Moderate negative	-14

Table 6.4: Comparison of the change in probability of output node activation to the change in bias weights in the output layer, as a representation of the atomic distribution, following the revision $R[4, 12]$.

A probabilistic analysis was also performed for an inconsistent revision $R[4, 12]$. The changes in bias weights on the output layer remained relatively representative of the revision to the new individual atomic frequencies, see Table 6.4. However, no patterns were able to be identified in the inter-layer connection weights and, particularly, the hidden layer bias weight varied considerably from machine to machine, taking a bimodal distribution, see Figure 6.15. These inconsistent revisions include information which is entirely at odds with the previously learnt model of the world. It is not clear if or how the machine can incorporate this surprising information. However, it is clear that the single hidden node representation of a linear conceptual space, is lost following these revisions.

For differential revisions, it is evident that no learning machine could adequately represent strict prioritisation. To do so would require individual training of each model state. The Boltzmann machine can only maintain the relative ordering and separation of the differential, satisfying models that existed on initial training. In non-prioritised revisions (Booth & Meyer, Figure 6.14), the Boltzmann machine is able to represent the whole state preference of the maximal models. However, these revisions still include

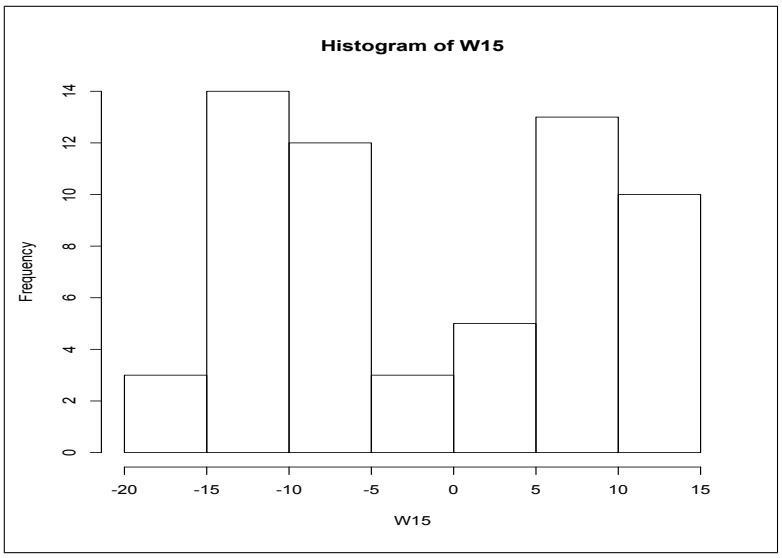


Figure 6.15: A histogram of the hidden node bias weights (W15), following the revision $R[4, 12]$.

the promotion of inconsistent models from the lowest level of preference.

In general, as increasingly inconsistent information is acquired (simple $\rightarrow$ inconsistent revision) greater variation in typicality between models at the same level of preference occurs: increasing incongruence. Under these chaotic circumstances, the output of the machine moves from whole state preference towards atomic typicality.

6.4 Boltzmann Revision

What does the Boltzmann machine representation support as an approach to belief revision? In the subgroup of simple revisions, with partially consistent new information, the machine results fit reasonably well with both of the logical approaches examined (Booth & Meyer or Spohn). However, it cannot support strict prioritisation.

Further, any major inconsistency in the conditional constraints contained within the data expose what we believe is the underlying default of the Boltzmann machine: to learn the atomic distribution (typicality). Such incongruence is manifest by variations in the typicality of states occupying the same level of state preference. It can be present within single data sets before any revision but is most obvious after inconsistent revisions, Section 6.3.3.

The reader is reminded that these experimental results are from a Boltzmann machine utilising a single ILP architecture. The machine is forced to combine or select the

information it learns: state preference and/or atomic typicality. It cannot accurately represent both, unless the involved world is congruent!

None of the current approaches to belief change operators in logic utilise information related to typicality (atomic distribution) to inform the results of revision. Our experiments with the Boltzmann machine indicate that for disordered, inconsistent revisions typicality provides a 2nd dimension or bound, modifying the revision of whole state preference. Further, typicality information is encapsulated in every training set; in contrast to the approach of current two-dimensional revision schemes, which utilise additional meta-information extraneous to the epistemic state (Rott, 2009).

For the reasons given above, we present yet another divergent approach to revision, utilising the scheme which was previously proposed to characterise typicality (Section 5.3.3). A system of spheres diagram is not offered, as the proposed Boltzmann revision is beyond the explanatory capacity of the diagram, which would require dual overlapping spheres for preference and typicality. Recalling our previous description of a state typicality ordering:

1. Let A be a set of atomic propositions: $A = \{a_i : i = 1, 2, ..., k\}$
   – symbolising the atomic composition of states in a world with cardinality k.
   Then: $\overrightarrow{t_a} = a_1^L \prec a_2^L \prec .... \prec a_{k-1}^L \prec a_k^L \prec a_k^M \prec a_{k-1}^M \ .... \ \prec a_2^M \prec a_1^M$
   – is a total pre-order on the typicality of atomic valuations:
   – where a_i^L is less typical than a_j^L and a_i^M is more typical than a_j^M,
   – given complementary truth valuations : $a_i^M,\ a_i^L \in [false - 1, true + 1]$.

   Then a total pre-order on state typicality, $\overrightarrow{T_S}$ given by:

$$\begin{aligned}\overrightarrow{T_S} \quad = \quad & a_k^L \wedge a_{k-1}^L .... \wedge a_2^L \wedge a_1^L \quad \prec \quad a_k^M \wedge a_{k-1}^L .... \wedge a_2^L \wedge a_1^L \quad \prec \\ & a_k^M \wedge a_{k-1}^M .... \wedge a_2^L \wedge a_1^L \quad ..... \ \prec \quad a_k^M \wedge a_{k-1}^M .... \wedge a_2^M \wedge a_1^L \\ & \prec \quad a_k^M \wedge a_{k-1}^M .... \wedge a_2^M \wedge a_1^M\end{aligned}$$

   An extension of this scheme, $\overrightarrow{T_{[\alpha]}}$: denotes a typicality ordering on the models of α. Further, a new typicality operator also needs to be defined:

2. Let $\overrightarrow{h_{\geq\beta} \mapsto T_{[\alpha]}}$ be a re-ordering of the preferred models of α by typicality:
   where models at (preference) ranking $\geq \beta$ are moved (raised or lowered)
   to a new rank based on their atomic typicality.

$$\text{Then } \overrightarrow{h_{\geq\beta} \mapsto T_{[\alpha]}} \quad = \quad \begin{cases} \overrightarrow{T_{[\alpha]}}\ , & \text{if within rank typicality varies} \\ \text{or otherwise} & \\ \overrightarrow{h_{\geq\beta}}\ , & \text{if within rank typicality is uniform} \end{cases}$$

Denoting the typicality operator as an ordering may seem awkward, but it emphasises the unconventional nature of what is actually a replacement of preference by typicality. An advantage of the proposed declaration is that it correctly adjusts for typicality only when required. This becomes necessary when traditional state preferences are insufficient to describe the typicality relationship between models within a ranking, i.e. when models at the same preference level have discrepant typicalities.

In order to define a revision operator based on the results from the Boltzmann Machine, the two classes of revision previously considered need to be encoded separately:

Simple Revision: the new information is partially consistent with the pre-existing knowledge base:

$$\overrightarrow{h} \quad \longmapsto \quad \overrightarrow{h_{<\beta} \vee \neg\alpha} \quad \prec . \ \overrightarrow{h_{=\beta} \mapsto T_{[\alpha]}} \quad \prec . \ \overrightarrow{\alpha}$$

The non-satisfying models are incorporated at lowest preference.

Isolated incongruent models are re-ordered by typicality.

The maximal models are relatively prioritised.

Inconsistent Revision: the new information is extremely inconsistent with the pre-existing knowledge base:

$$\overrightarrow{h} \quad \longmapsto \quad \overrightarrow{h_{<\beta} \vee \neg\alpha} \quad \prec . \ \overrightarrow{h_{\geq\beta} \mapsto T_{[\alpha]}}$$

The non-satisfying models are incorporated at lowest preference.

All the incongruent models are re-ordered by typicality,

Including the maximal models.

Assumption: β is a degree of belief, sufficient to allow the models of α to be relatively prioritised. This is equivalent to Spohn's methodology, specifying an index of one for the agent's commitment to the new information.

When abstracted in this way, the connection between the two cases of revision becomes clearer. In situations where extremely inconsistent new information (α) is received, even the most preferred satisfying models ($Maximal\ Models[\alpha]$) are revised

by their typicality: they may be too inconsistent with the prior beliefs. We have defined the threshold for this revision by typicality based on the divergence in typicality of models at the same level of preference. Although we have not specified the exact metric, it can be appreciated that this definition is a 'proxy for disorder' within the epistemic state. Given sufficient variation in such intra-level typicality, it is likely that overlap occurs with models in higher or lower preference rankings.

Of course, there is a partial solution to the conflict in ordering by preference and typicality, which might mitigate some of the disorder resulting from incrementally inconsistent revisions. That is to represent the two different types of information separately, in architecturally distinct parallel neural systems. This is the 'dual process approach' theorised by Frixione and Lieto (2011, 2014), previously mentioned in Section 5.3.4: Typicality versus Preference. However, even utilising a 'dual approach' still requires some choice be made between the two orderings. Considering the results of belief revision from the experiments with the Boltzmann machine, some of the questions at the end of Section 5.3.4 can now be answered. In situations where belief revision results in chaotic inconsistency: typicality takes precedence over preference.

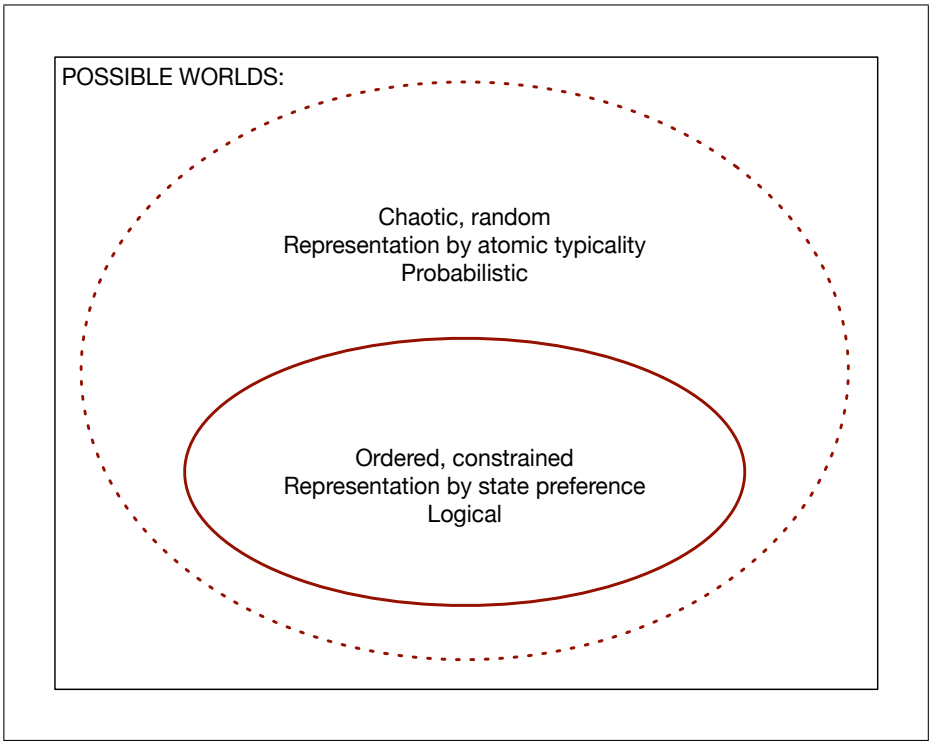


Figure 6.16: A philosophical relationship between chaotic probabilistic typicality and ordered preferential logic.

If the reader can forgive a momentary digression into philosophy (Figure 6.16), on a larger scale too there may be a spectrum of possible worlds with regards learning and adaptation. On the one-hand, there are worlds where revision results in congruence; where gradually accumulated new information (learning), which fits with our view of the world, is merged with our beliefs by small adjustments in preference into a new and rational whole. These ordered systems can be elegantly described by logic. Such conscious, rational learning is a new evolutionary phenomena. On the other-hand, there are worlds where revision results in incongruence; where surprisingly inconsistent, new information which disrupts the order of the prior beliefs, can only be dealt with in the short term by resorting to typicality: the underlying disordered numerical relationships between compositional atoms. The explanation of such chaotic systems has traditionally and phylogenically been probabilistic. Could an extended form of typicality logic be utilised to describe them?

Chapter 7

Conclusions

This chapter presents some open questions and a brief record of intended investigations. It then summarises each thread of the thesis: representation, incongruence and belief revision and attempts to extract the main concepts which consolidate the research.

7.1 Future Directions

It is hoped that this section is a list of aspirations, rather than an acknowledgement of things overlooked.

Unexplained Results: There were a group of permutations in 2-atom micro-worlds (with a single preferred model state and the remaining three states in the least preferred rank) where the ILP Boltzmann machine had difficultly learning a generative model of the input training set. Specifically, the machine over-estimated the preference of the least atomically typical state. We have assumed this problem relates to the very limited architecture of this machine. It didn't occur in any of the larger 3, 4, 5 or 6-atom worlds. There may be a physical limitation on the representation capacity of such a small machine. However, the unexplained results deserve further consideration. It should be possible to examine the internal behaviour of these machines, as in Section 5.2.4 Network Analysis.

Network Behaviour: An internal network analysis of the HLR machine based on a Markov model was avoided because of the complexity of the hidden layer interconnections. However, it would be interesting to see how the representation, in terms of internal node feature detection, differed from the ILP machine.

We would like to investigate the energy minima in the network related to typicality versus preference; to obtain a clearer mathematical insight into when and why the network changes its' representation. We have yet to devise a methodology for carrying out this investigation.

Belief Revision: It would be interesting to test the HLR Boltzmann machine in the context of belief revision. We expect its behaviour would be different from the limited ILP machine. It is possible the additional complexity of the network might enable it to maintain a modified version of preferential ranking in situations where the ILP machine defaulted to an ordering based on atomic typicality.

Further, accepting the common-sense view that the recovery axiom is intuitively correct, we would like to determine if it is possible to achieve an appropriate outcome with the Boltzmann network. Recovery would require two consecutive revisions with the attendant practical difficulty in retaining the original training information.

Variable Binding: As stated earlier, we have not examined the mine-field of predicate calculus and its relationship to the representation of dynamic variable binding. There is a large body of research on this topic (Section 2.4.5: Recent Literature), including some highly complex neural network implementations in the field of Neural-Symbolic Integration (NSI). The problem of dynamic variable binding comes down to an issue of space. Even within the vast size of the dendritic connection tree in an average brain ($\sim 10^{14}$ synapses), there is not enough capacity to represent the almost infinite number of relationships that are present within the physical world. Whilst not constituting an hypothesis, the Boltzmann machine does offer a potential solution for the representation of multiple relationships from a single object. Most networks can only retrieve mathematically functional mappings: many-to-one or one-to-one. These mappings can only consume representational capacity. In the case of the SHRUTI system in NSI (Section 2.4.3), the network requires a large number of ancillary nodes (pairs of 'collector, enabler and utility nodes') and additional connections to represent predicates. The Boltzmann machine can retrieve one-to-many relationships, thus conserving the space within the neural network. As demonstrated in Section 4.3.2: Preference Relation, the machine can cycle through multiple relationships within a single set of neuronal connections: no additional architectural capacity is required.

7.2 Summary

Representation

The thesis makes a contribution to the domain of cognition by advancing a connectionist model for the representation of supra-classical, non-monotonic logic: specifically a Boltzmann neural network.

SCNM logic can be seen as a formalism of common-sense reasoning. It specifically requires a ranking of preferred conclusions in the context of inference under the rational consequence relation. This ranking of model states incorporates the less preferred counter-examples, which are the basis of exception processing and possibly reasoning about causality. In probabilistic terms this ranking is the theoretical equivalent of the energy minima within a symmetric neural network.

Information and Energy:[1] In the context of a SNN these concepts are alternative characterisations. From the previous literature, particularly the research of Pinkas, we know that logical formulae can be mathematically translated to a specific neural network structure, so that conclusions in the logic are represented by energy states in the network. The Boltzmann machine can learn multiple optima which are represented by the partition energy function at equilibrium: the Boltzmann distribution. The Boltzmann machine may be unique in its ability to retrieve a distribution 'strongly equivalent' to its training set.

Probability and Stochastic Activation: Ranking with 'strong equivalence' can only be retrieved by a neural network with stochastic activation functions, sampling from an energy distribution: any such network is analogous to the Boltzmann machine. The default rule and consequent preference relation of SCNM logic are a qualitative counterpart of a generative model in probability. Stochastic activation functions are the underlying basis of the probabilistic representation in neural networks, be they symmetrically recurrent or feed-forward. The stochastic activation allows the network to sample a distribution in the manner of a Markov model.

There were no traditional bench-marks for SCNM logic with which to test the hypothesised Boltzmann model. The thesis utilised logical micro-world environments so that outputs of the network model could be compared to inferential conclusions in the logic: other authors have used similar schemata. The experimental results from a variety of generic micro-worlds, with incremental numbers of atoms, supported the view that the Boltzmann machine is a faithful model of SCNM logic. It was able to learn a

[1] Important themes of the thesis are emphasised

preference relation and able to retrieve appropriately ranked model states entailed by a premiss, in the context of inference under rational consequence. Ideal single machines with very small errors per state are possible, but difficult to efficiently generate during training. The thesis offers a solution of ensemble machines, accumulating a single output from multiple parallel hidden layers. This construction reduces the state errors and enables fast, robust learning.

Further, we have shown that this machine emulates two important example properties of the logic (Section 4.3.4), as a demonstration of the practical utility of the representation. In terms of human cognition, there is evidence from the domain of neuroscience that spiking neural networks, utilising the restricted Boltzmann machine, can be trained using an event-driven variant of Hebbian learning in large neuromorphic systems.

Plausibility versus Efficiency: The biological and engineering domains often have competing requirements. Hinton's original version of the Boltzmann machine had a rich hidden layer consistent with the role of inhibitory constraint within the biological cortex. It implements simulated annealing to achieve optimisation. Simulated annealing can be viewed as the parent algorithm of threshold optimisation, inherently suitable for a disordered biological environment. Both these implementations make the standard machine less computationally efficient than the restricted machine. However, in the context of a massively parallel, biological system they may not be such a disadvantage.

Incongurence

Closer examination of our initial results with a HLR Boltzmann machine revealed that every training set of vectors contains dual information about: whole state frequencies and atomic activation frequencies. The whole state frequency distribution is equivalent to whole state preference in logic. However, a separate state distribution can be reconstructed from the product of atomic frequencies. These dual information distributions are often incongruent. Experimentation with the architecture of the Boltzmann machine produced a restricted ILP machine which more closely selected the whole state preference ranking of traditional logic. As previously this machine architecture was utilised in the context of an ensemble: parallel hidden layers with an accumulated single output.

Further, exploration of the internal 'mechanics' of the ILP machine architecture was completed in an individual specific environment with a superimposed semantic analogy. This analysis as a Markov model, involved examination of the learned weights

and activation states in the network, when presented with a neutral (null) premiss. It demonstrated that the machine's hidden node acted as a linear conceptual space, representing the micro-world semantics.

Preference versus Typicality: The thesis proposes that the distribution of states reconstructed from atomic frequency is the probabilistic equivalent of typicality in logic. Although the term typicality is widely used in the literature, the traditional view of preferential semantics in non-monotonic logic does not utilise it. We believe, there is no adequate definition in logic based on compositional, atomic characteristics. We have argued, by counter-example, that traditional representation of typicality by 'minimal model semantics' is incorrect and that atomic typicality requires a separate ranking from whole state preference. The thesis attempts to provide an atomic definition of typicality. The selection of typicality versus preference by architectural alteration of a Boltzmann network provides a practical way of implementing a theorised 'dual process approach' to reasoning.

One theme of the thesis is the inherent conflict between whole state preference and atomic typicality in these incongruent information distributions. Chapter 4 Representation models the traditional view of logic based solely on whole state preference. Chapter 5 Incongruence presents the conflicting information based on atomic typicality, which this traditional logic largely ignores. The final Chapter 6 Belief Revision demonstrates, from the machine perspective, that a process of dual adaptation in both preference and typicality is required to model a range of belief revisions: in the extreme involving prioritisation of information completely inconsistent with the pre-existing default rule.

Belief Revision

Any cognitive agent must be able, not only to draw inferences from the environment but also, to adapt to changes in the environment. Belief revision in logic formalises the structure of adaptation. It is a relatively young domain; the landmark paper of AGM was published in 1985 and there are still a surfeit of competing theories.

In general, neural networks have difficulty retaining previous learning when exposed to new data. They are by default irrationally non-monotonic. The thesis utilised a variation of the Boltzmann machine learning algorithm to implement pseudo-rehearsal, which allowed the network to maintain past learning during re-training. It enabled experimental examination of the machine plausibility of two current theories of belief revision, Booth & Meyer and Spohn.

We considered experimental results from three representative categories of revision:

- In situations where new information was partially consistent with the previously held beliefs, the machine was able to be re-trained to reflect the view held in logic. The preference ranking of the satisfying and non-satisfying models was able to be appropriately adjusted, incorporating the new information. The internal, linear conceptual space representation by the hidden node was preserved from the exemplar 4-atom micro-world, $O[0, 1, 8, 9]$.

- In situations where new information was entirely inconsistent with the previously held beliefs, the results of machine re-training did not reflect the logic: the entire preference ranking was replaced by an ordering on atomic typicality. The internal representation of a linear conceptual space by the hidden node was lost.

- In situations where there was a mixture of partially consistent and entirely inconsistent new information, the Boltzmann machine was not able to strictly prioritise individual model states. These machine results indicate that logical theories holding strict prioritisation are not plausible.

Order versus Chaos: The results from re-training the Boltzmann machine suggest that the current rational approaches to belief revision in logic only apply in ordered, congruent worlds. In chaotic, incongruent worlds typicality provides a ranking of states based on their individual atomic probabilities. In these disordered worlds inconsistencies in whole state preference, between pre-existing rules and the revised models, cannot be rationally accommodated with reference to a unitary state exemplar.

It is hypothesised that exception processing using typicality is the evolutionary basis of biological adaptation and that rational reasoning is phylogenetically recent. A definition of atomic typicality (as attempted in the thesis) could extend the formality of logic to adaptation in chaotic environments, where inconsistent information can only be assimilated with reference to individual compositional atoms.

In conclusion; the thesis offers the Boltzmann machine as a practical representation of SCNM logic. It validates the place of a Boltzmann-like mechanic in human cognition and re-enforces current research in neuroscience into spiking neural networks with stochastic Hebbian learning. Further, the model enables the experimental investigation of domains, such as typicality and belief revision, which are current areas of mathematical conjecture in logic.

References

Aarts, E. and Korst, J. (1990). *Simulated Annealing and Boltzmann Machines*. Interscience Series in Mathematics and Optimization. John Wiley and Sons.

Aarts, E. and Korst, J. (1997). *Local Search in Combinatorial Optimization*, Chapter Simulated Annealing, 91–120. John Wiley and Sons.

Adams, E. (1965). The Logic of Conditionals. *Inquiry*, *8*, 166–197.

Adeli, H. and Ghosh-Dastidar, S. (2009). Spiking Neural Networks. *International Journal of Neural Systems*, *19*(4), 295–308.

Alchourron, C., Gärdenfors, P., and Makinson, D. (1985). On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. *Journal of Symbolic Logic*, *50*, 510 – 530.

Alchourron, C. and Makinson, D. (1982). On the Logic of Theory Change: Contraction Functions and their Associated Revision Functions. *Theoria*, *48*, 14 – 37.

Annis, C. (2014). Central Limit Theorem (Summary). *Statistical Engineering - http://www.statisticalengineering.com/central_limit_theorem_(summary).htm*.

Baader, F. (2003). *The Description Logic Handbook: Theory, Implementations and Applications*. (3rd ed.). Cambridge University Press.

Bacchus, F. (1988). *Representing and Reasoning with Probabilistic Knowledge*. Ph. D. thesis, University of Alberta.

Bacchus, F. (1990). A Logic for Representing Reasoning with Statistical Knowledge. *Computational Intelligence*, *6*, 209–231.

Bacchus, F. (1991). Default Reasoning from Statistics. In *Proceedings AAAI*, 392–398.

Bacchus, F. (1996). From Statistical Knowledge Bases to Degrees of Belief. *Artificial Intelligence*, *87*, 75–143.

Balkenius, C. and Gärdenfors, P. (1991). Non-Monotonic Inferences in Neural Networks. In *Principles of Knowledge Representation and Reasoning*, 32–39.

Barsalou, L. (1985). Continuity of the Conceptual System across Species. *Trends in Cognitive Sciences*, *9*(7), 305–307.

Barsalou, L. (1999). Perceptual Symbol Systems. *Behavioural and Brain Sciences*, *22*, 577–660.

Benferhat, S., Bonnefon, J., and Neves, S. (2005). An Overview of Possibilistic Handling of Default Reasoning, with Experimental Studies. *Synthese*, *146*, 53–70.

Blanchette, G., McCane, B., Labuschagne, W., and Robins, A. (2015). Towards a Representation of Non-Monotonic Inference in an Artificial Neural Network. Technical report, Otago University Press, Computer Science.

Bochman, A. (2011). *Essays in Non-Monotonic Reasoning, 30th Anniversary*, 25–61. College Publications.

Boden, M. and Niklasson, L. (2000). Semantic Systematicity and Context in Connectionist Networks. *Connection Science*, *12*(2), 111–142.

Booth, R., Chopra, S., Meyer, T., and Ghose, A. (2004). A Unifying Semantics for Belief Change. In R. Lopez de Mantaras and L. Saitta (Eds.), *European Conference of Artificial Intelligence*, Volume 16, 793–797.

Booth, R. and Meyer, T. (2011). How to Revise a Total Preorder. *Journal of Philosophical Logic*, *40*, 193–238.

Booth, R., Meyer, T., and Varzinczak, I. (2012). PTL: A Propositional Typicality Logic. *13th European Conference on Logics in Artificial Intelligence (LNAI)*, *7519*, 107–119.

Booth, R., Meyer, T., and Varzinczak, I. (2013). PTL: A Propositional Typicality Logic (extended version). Self publication.

Booth, R., Meyer, T., and Wong, K. (2006). A Bad Day Surfing is Better than a Good Day Working: How to Revise a Total Preorder. In *International Conference on the Principles of Knowledge Representation and Reasoning*, Volume 10, 230–238.

Boutlier, C. (1996). Iterated Revision and Minimal Change of Conditional Beliefs. *Journal of Philosophical Logic*, *25*, 263–305.

Bowers, J. (2009). On the Biological Plausibility of Grandmother Cells: Implications for Neural Network Theories in Psychology and Neuroscience. *Psychological Review*, *116*(1), 220–251.

Bowers, J. (2011). What is a Grandmother Cell? And how would you know if you found one? *Connection Science*, *23*(2), 91–95.

Burch, R. (2014). Charles Sanders Peirce. *Stanford Encyclopedia of Philosophy - http://plato.stanford.edu/archives/win2014/entries/peirce/*.

Byrne, R. (1989). Suppressing Valid Inferences with Conditionals. *Cognition*, *31*, 61–83.

Cantwell, J. (1997). On the Logic of Small Changes in Hypertheories. *Theoria*, *63*(1), 54–89.

Carnap, R. (1950). *Logical Foundations of Probability*. University of Chicago.

Chan, H. and Darwiche, A. (2002). When do Numbers Really Matter? *Journal of Artificial Intelligence Research*, *17*, 265–287.

Chang, F. (2002). Symbolically Speaking: A Connectionist Model of Sentence Production. *Cognitive Science*, *93*, 1–43.

Chen, H. and Murray, A. (2003). Continuous Restricted Boltzmann Machine with an Implementable Training Algorithm. *IEEE Proc. Visual Image Processing*, *150*(3), 153–158.

Cheng, P. and Holyoak, K. (1985). Pragmatic Reasoning Schemas. *Cognitive Psychology*, *17*, 391–416.

Chopra, S., Ghose, A., and Meyer, T. (2002). Iterated revision and the Axiom of Recovery: a Unified Treatment via Epistemic States. In F. van Harmelen (Ed.), *European Conference of Artificial Intelligence*, Volume 15, 541–545.

Chopra, S., Ghose, A., and Meyer, T. (2003). Non-Prioritized Belief Change. *Journal of Philosophical Logic*, *32*, 417–443.

Courville, A., Bergstra, J., and Bengio, Y. (2011). A Spike and Slab Restricted Boltzmann Machine. *Artificial Intelligence and Statistics*, *1*, 233–241.

CRAN (2014). A Language and Environment for Statistical Computing. *www.r-project.org/*.

Creath, R. (2014). Logical Empiricism. *Stanford Encyclopedia of Philosophy - https://plato.stanford.edu/archives/spr2014/entries/logical-empiricism/*.

Darwiche, A. and Pearl, J. (1997). On the Logic of Iterated Belief Revision. *Artificial Intelligence*, *89*, 1–29.

d'Avila Garcez, A., Lamb, L., and Gabbay, D. (2007). Connectionist Modal Logic: Representing Modalities in Neural Networks. *Theoretical Computer Science*, *371*, 34–53.

d'Avila Garcez, A., Lamb, L., and Gabbay, D. (2009). *Neural-Symbolic Cognitive Reasoning*. Cognitive Technologies. Springer.

Davis, E. and Morgenstern, L. (2004). Introduction: Progress in Formal Common-Sense Reasoning. *Artificial Intelligence*, *153*, 1–12.

Delgrande, J. (1987). A First-Order Logic for Prototypical Properties. *Artificial Intelligence*, *33*, 105–130.

Egger, M. (1988). The Boltzmann Machine: a Survey and Generalization. Technical Report TR 805, Massachusetts Institute of Technology.

Eslami, S., Heess, N., and Win, J. (2012). The Shape Boltzmann Machine: A Strong Model of Object Shape. In *IEEEComputer Vision and Pattern Recognition*, 406–413.

Fodor, J. and Pylyshyn, Z. (1988). Connectionism and Cognitive Architecture: A Critical Analysis. *Cognition*, *28*, 3–71.

Ford, M. and Billington, D. (2000). Strategies in Human Non-Monotonic Reasoning. *Computational Intelligence*, *16*, 446–469.

Franca, M., Zaverucha, G., and d'Avila Garcez, A. (2014). Fast Relational Learning Using Bottom Clause Propositionalization with Artificial Neural Networks. *Machine Learning*, *94*, 81–104.

Frank, S., Haselager, W., and van Rooij, I. (2009). Connectionist Semantic Systemicity. *Cognition*, *110*, 358–379.

Freeman, H. (1994). *Neural Networks, Algorithms, Applications and Programming Techniques*. Addison-Wesley.

Frixione, M. and Lieto, A. (2011). Representing Concepts in Artificial Systems: A Clash of Requirements. *CEUR Proceedings 4th HCP Workshop Human Processes*, *4*, 75–82.

Frixione, M. and Lieto, A. (2014). Towards an Extended Model of Conceptual Representations in Formal Ontologies: A Typicality-Based Proposal. *Journal of Universal Computer Science*, *20*(3), 257–276.

Gabbay, D., Hogger, C., and Robinson, J. (1994). *Handbook of Logic in Artificial Intelligence and Logic Programming*, Volume Volume 3: Nonmonotonic Reasoning and Uncertain Reasoning. Oxford UP, Oxford.

Gärdenfors, P. (1994). *Logic and Information Flow*, Chapter How Logic Emerges from the Dynamics of Information, 49–77. Cambridge MIT Press.

Gärdenfors, P. (2004). *Conceptual Spaces*. MIT Press, Cambridge.

Gärdenfors, P. and Makinson, D. (1988). Revisions of Knowledge Systems using Epistemic Entrenchment. In *Theoretical Aspects of Reasoning about Knowledge*, 83–95. Morgan Kaufmann.

Geman, S. and Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images. In *IEEE Transactions on Pattern Analysis and machine Intelligence*, Volume 6, 721–741.

Getoor, L. and Taskar, B. (2007). *Introduction to Statistical Relational Learning*. MIT Press Cambridge.

Giordano, L., Gliozzi, V., Olivetti, N., and Pozzato, G. (2007). Preferential Description Logics. *Artificial Intelligence and Reasoning. LNAI*, *4790*, 257–272.

Giordano, L., Gliozzi, V., Olivetti, N., and Pozzato, G. (2013). A Non-Monotonic Description Logic for Reasoning about Typicality. *Artificial Intelligence*, *195*, 165–202.

Girard, P. and Tanaka, K. (2016). Paraconsistent Logics. *Synthese*, *193*, 1–14.

Giugino, R. and Lukasiewciz, T. (2002). P-SHOQ(D): A Probabilistic Extension of-SHOQ(D) for Probabilistic Ontologies in the Semantic Web. *8th European Conference on Logics in Artificial Intelligence (LNAI)*, *2424*, 86–97.

Gomez-Torrente, M. (2015). Alfred Tarski. *Stanford Encyclopedia of Philosophy - http://plato.stanford.edu/archives/win2015/entries/tarski/*.

Grove, A. (1988). Two Modelings for Theory Change. *Journal of Philosophical Logic*, *17*, 157–170.

Hadley, R. and Cardei, V. (1999). Language Acquisition from Sparse Input without Error Feedback. *Neural Networks*, *12*, 217–235.

Hadley, R., Rotaru-Varga, A., Arnold, D., and Cardei, V. (2001). Syntactic Systemicity Arising from Semantic Predictions in Hebbian-Competitive Networks. *Connection Science*, *13*(1), 73–94.

Hansson, S. (1991). Belief Contraction without Recovery. *Studia Logica*, *50*, 251–260.

Hansson, S. (1993). Changes of Disjunctively Closed Bases. *Journal of Logic, Language and Information*, *2*(4), 255–284.

Hansson, S. (2011). Logic of Belief Revision. *Stanford Encyclopedia of Philosophy - https://plato.stanford.edu/entries/logic-belief-revision/*.

Harnad, S. (1987). *Categorical Perception: The Groundwork of Cognition.* Cambridge University press.

Harnad, S. (1990). The Symbol Grounding Problem. *Physica*, *42*, 335–346.

Hartshorne, C. and Weiss, P. (1958). *Collected Papers of Charles Sanders Peirce: Volumes I - VIII, 1931-1935*, Volume I -VIII. Harvard University Press.

Haykin, S. (2008). *Neural Networks and Learning Machines* (3rd Edition ed.). Learning Tasks p38-45, Stochastic Methods p580-621, Hopfield Model p690-703: Pearson.

Hebb, D. (1949). *The Organisation of Behaviour.* John Wiley and Sons.

Heidema, J. and Labuschagne, W. (2001). *Culture in Retrospect*, Chapter Knowledge and Belief: The Agent-Oriented View, 194–214. UNISA Press.

Hendler, J. (1989). Marker Passing over Microfeatures: Towards a Hybrid Symbolic Connectionist Model. *Cognitive Science*, *13*, 79–106.

Heyes, C. (2012). New Thinking: the Evolution of Human Cognition. *Philosophical Transactions of the Royal Biological Society*, *367*, 2091–2096.

Hinton, G. (1989). Deterministic Boltzmann Learning Performs Steepest Descent in Weight Space. *Neural Computation*, *1*, 143–150.

Hinton, G. (1990). Preface to the Special Issue on Connectionist Symbol Processing. *Artificial Intelligence*, *46*, 1–4.

Hinton, G. (2010). A Practical Guide to Training Restricted Boltzmann Machines. Technical Report TR 2010-003, University of Toronto, Machine Learning.

Hinton, G. and Brown, A. (1999). Spiking Boltzmann Machines. *NIPS*, 122–128.

Hinton, G., Osindero, S., and Teh, Y. (2000). What Kind of Graphical Model is the Brain? *www.cs.toronto.edu/ hinton/talks/ijcai3.ppt*.

Hinton, G., Sejnowski, T., and Ackley, D. (1984). Boltzmann Machines: Constraint satisfaction Networks that Learn. Technical Report TR 84-119, Carnegie-Mellon University, Computer Science.

Hinton, G., Sejnowski, T., and Ackley, D. (1985). A Learning Algorithm for Boltzmann Machines. *Cognitive Science*, *9*, 147–169.

Hopfield, J. (1982). Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings Natural Academy of Science*, *79*, 2554–2558.

Hsu, F. (2002). *Behind Deep Blue: Building the Computer that Defeated the World Chess Champion*, Volume ISBN 0-691-09065-3. Princeton University Press.

Hummel, J. (2011). Getting Symbols Out of a Neural Architecture. *Connection Science*, *23*(2), 109–118.

Irvine, A. (2015). Bertrand Russell. *Stanford Encyclopedia of Philosophy - http://plato.stanford.edu/archives/win2015/entries/russell/*.

Izhikevich, E. (2004). Which Model to Use for Cortical Spiking Neurons? In *IEEE Transactions on Neural Networks*, Volume 15, 1063–1070.

Jagota, A. (1994). *Representing Discrete Structures in a Hopfield-Style Network*, Volume Neural Networks for Knowledge Representation and Inference, 123–142. Lawrence Erlbaum.

Kennedy, J. (2016). Kurt Gödel. *Stanford Encyclopedia of Philosophy - http://plato.stanford.edu/archives/win2016/entries/goedel/*.

Kersting, K., De Raedt, L., and Raiko, T. (2006). Logical Hidden Markov Models. *Journal of Artificial Intelligence Research*, *25*, 425–456.

Khosravi, H. and Bina, B. (2010). A Survey on Statistical Relational Learning. *Canadian Artificial Intelligence LNAI*, *6085*, 256–268.

Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). Optimisation by Simulated Annealing. *Science*, *220*, 671–680.

Knott, A. (2012). *Sensorimotor Cognition and Natural Language Syntax*. MIT Press.

Koller, D. and Pfeffer, A. (1998). Probabilistic Frame Based Systems. In *Proceedings AAAI*, Volume 15th, 580–587.

Koons, R. (2014). Defeasible Reasoning. *Stanford Encyclopedia of Philosophy - https://plato.stanford.edu/archives/spr2014/entries/reasoning-defeasible/*.

Kraus, S., Lehmann, D., and Magidor, M. (1990). Non-Monotonic Reasoning, Preferential Models and Cumulative Logics. *Artificial Intelligence*, *44*, 167–207.

Krogh, A. and Hertz, J. (1995). *Advances in Neural Information Processing Systems*, Volume 4, Chapter Simple Weight Decay can improve Generalization, 950–957. Morgan Kaufmann.

Kullback, S. and Leibler, R. (1951). On Information and Sufficiency. *Annals of Mathematical Statistics*, *22*, 79–86.

Labuschagne, W. and Heidema, J. (2010). Towards Agent-Oriented Logic: (I - II) Variations on the Theme of Logical Consequence. Technical report, Otago University.

Labuschagne, W., Heidema, J., and Britz, K. (2013). Supra-classical Consequence Relations: Tolerating Rare Counter-examples. *Advances in AI in Springer LNAI*, 326–337.

Lehmann, D. and Magidor, M. (1992). What Does a Conditional Knowledge Base Entail? *Artificial Intelligence*, *55*, 1–60.

Leitgeb, H. (2001). Nonmonotonic Reasoning by Inhibition Nets. *Artificial Intelligence*, *128*, 161–201.

Leitgeb, H. (2004). *Inference on a Low Level. An Investigation into Deduction, Non-Monotonic Reasoning and the Philosophy of Cognition*, Volume 30 of *Applied Logic Series*. Kluwer Academic.

Leitgeb, H. (2007). Neural Network Models of Conditionals: an Introduction. In X. Arrazola and J. Larrazabal (Eds.), *International Workshop on Logic and Philosophy of Knowledge*, 191–223.

Lenat, D. (2016). The Cyc Project. *www.Cyc.com*.

Levi, I. (1977). Subjunctives, Dispositions and Chances. *Synthese*, *34*, 423–455.

Lewis, S., Blake, J., Cherry, J., Sternberg, P., and Thomas, P. (2016). The Gene Onotology. *http://www.geneontology.org*.

Linnainmaa, S. (1970). The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors. Master's thesis, University of Helsinki.

Lukasiewciz, T. (1999). Probabilistic Deduction with Conditional Constraints over Basic Events. *Journal of Artificial Intelligence Research*, *10*, 199–241.

Makinson, D. (2003). Bridges between Classical and Nonmonotonic Logic. *Journal of the IGPL*, *11*(1), 69–96.

Makinson, D. (2005). *Bridges from Classical to Nonmonotonic Logic*. King's College Publications.

Markram, H. (2014). Human Brain Project. *www.humanbrainproject.eu*.

Mass, W. (1997). Networks of Spiking Neurons: The Third Generation of Neural Network Models. *Neural Networks*, *10*, 1659–1671.

Mazzoni, P., Anderson, R., and Jordan, M. (1991). A More Biologically Plausible Learning Rule for Neural Networks. *Proceedings Natural Academy of Science*, *88*, 4433–4437.

McCarthy, J. (1959). Programs with Common Sense. In *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, 75–91.

McCarthy, J. (1980). Circumscription - A Form of Non-Monotonic Reasoning. *Artificial Intelligence*, *13*, 27–39.

McCarthy, J. and Hayes, P. (1969). *Machine Intelligence*, Chapter Some Philosophical Problems from the Standpoint of Artificial Intelligence, 463–502. Edinburg University Press.

McCloskey, M. and Cohen, N. (1989). *The Psychology of Learning and Motivation*, Volume 23, Chapter Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem., 109–164. New York: Academic Press.

McCulloch, W. and Pitts, W. (1943). A Logical Calculus of the Ideas Imminent in Nervous Activity. *Bulletin of Mathematical Biophysics*, *5*, 115–133.

McDermott, D. and Doyle, J. (1980). Non-Monotonic Logic I. *Artificial Intelligence*, *13*, 41–72.

Metropolis, N. and Rosenbluth, A. (1953). Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, *21*(6), 1087–1092.

Meyer, T., Labuschagne, W., and Heidema, J. (2000). Refined Epistemic Entrenchment. *Journal of Logic, Language and Information*, *9*, 237–259.

Minsky, M. (1974). A Framework for Representing Knowledge. Tech. report 306, MIT AI Laboratory.

Minsky, M. and Papert, S. (1969). *Perceptrons*. MIT Press, Cambrige MA.

Minsky, M. and Papert, S. (1971). Progress Report on Artificial Intelligence. *web.media.mit.edu/ minsky/papers/PR1971.html*.

Myllymaki, P. (1999). Massively Parallel Probabilistic Reasoning with Boltzmann Machines. *Applied Intelligence*, *11*, 31–44.

Nayak, A. (1994). Foundational Belief Change. *Journal of Philosophical Logic*, *23*, 495–533.

Nayak, A., Pagnucco, M., and Peppas, P. (2003). Dynamic Belief Revision Operators. *Artificial Intelligence*, *146*, 193–228.

Neftci, E., Das, S., Pedroni, B., Kreutz-Delgado, K., and Cauwenberghs, G. (2014). Event-Driven Contrastive Divergence for Spiking Neuromorphic Systems. *Frontiers in Neuroscience*, *7*, 74–87.

Neves, S., Bonnefon, J., and Raufaste, E. (2002). An Empirical Test of Patterns for Non-Monotonic Inference. *Annals of Mathematics and Artificial Intelligence*, *34*, 107–130.

Nilsson, N. (1986). Probabilistic Logic. *Artificial Intelligence*, *28*, 71–87.

O'Reilly, R. (1998). Six Principles for Biologically Based Computational Models of Cortical Cognition. *Trends in Cognitive Sciences*, *11*(2), 455–462.

Ortega, J. and Parrilla, J. (1999). Adaptive Cooperation between Processors in a Parallel Boltzmann Machine Implementation. In *Lecture Notes in Computer Science*, Volume 1607, 208–218. Springer.

Pearl, J. (1997). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Representation and Reasoning. Morgan Kaufmann.

Pearl, J. (2015). An Introduction to Causal Inference. ISBN: 1507894295.

Pearl, J. and Geffner, H. (1988). Probabilistic Semantics for a Subset of Default Reasoning. Technical Report CSD-8700XX, R-93-III, Computer Science, UCLA.

Peirce-Edition-Project (2010). *Writings of Charles S. Peirce: a Chronological Edition: Volumes I - VIII, 1857 - 1892*, Volume I - VIII. Indiana University Press.

Pelletier, F. and Elio, R. (1997). What Should Default Reasoning Be, by default? *Computational Intelligence*, *13*, 165–187.

Peppas, P. (2008). *Handbook of Knowledge Representation*, Chapter 8 Belief Revision, 317–359. Elsevier.

Pfeifer, N. and Kleiter, G. (2005). Coherence and Non-Monotonicity in Human Reasoning. *Synthese*, *146*, 93–109.

Pfleger, K. (1998). Categorical Boltzmann Machines. Technical Report TR 98-05, Stanford University, Knowledge Systems Laboratory.

Pinkas, G. (1994). *Neural Networks for Knowledge Representation and Inference*, Chapter Propositional Logic, Non-Monotonic Reasoning and Symmetric Networks - on Bridging the Gap between Symbolic and Connectionist Knowledge Representation, 175–203. Lawrence Erlbaum.

Pinkas, G. (1995). Reasoning, Non-Monotonicity and Learning in Connectionist Networks that Capture Propositional Knowledge. *Artificial Intelligence*, *77*, 203–247.

Pinkas, G. and Dechter, R. (1995). Improving Connectionist Energy Minimization. *Journal of Artificial Intelligence Research*, *3*, 223–248.

Pinkas, G., Lima, P., and Cohen, S. (2012). A Dynamic Binding Mechanism for Retrieving and Unifying Complex Predicate-Logic Knowledge. *ICANN 2012, LNCS*, *7552*, 482–490.

Popper, K. (1959). *The Logic of Scientific Discovery*. Hutchinson.

Priest, G., Tanaka, K., and Weber, Z. (2016). Paraconsistent Logic. *Stanford Encyclopedia of Philosophy - https://plato.stanford.edu/archives/win2016/entries/logic-paraconsistent/*.

Radermacher, F. (1996). Cognition in Systems. *Cybernetics and Systems*, *27*(1), 1–41.

Ratcliff, R. (1990). Connectionist Models of Recognition Memory: Constraints Imposed by Learning and Forgetting Functions. *Psychological Review*, *97*(2), 285–308.

Reimers, M. (2011). Local or Distributed Activation? The View from Biology. *Connection Science*, *23*(2), 155–160.

Reiter, R. (1980). A Logic for Default Reasoning. *Artificial Intelligence*, *13*, 81–132.

Rives, B. (2010). Internet Encyclopedia of Philosophy: Jerry Fodor. *http://www.iep.utm.edu/fodor/*.

Robins, A. (1995). Catastrophic Forgetting, Rehearsal and Pseudo-Rehearsal. *Connection Science: Journal of Neural Computing, Artificial Intelligence and Cognitive Research*, *7*, 123–146.

Rosales, R. and Sclaroff, S. (2006). Combining Generative and Discriminative Models in a Framework for Articulated Pose Estimation. *International Journal of Computer Vision*, *67*, 251–276.

Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain. *Psychological Review*, *65*(6), 386–408.

Rott, H. (1993). Belief Contraction in the Context of a General Theory of Rational Choice. *Journal of Symbolic Logic*, *58*.

Rott, H. (2007). Bounded Revision: Tow-Dimensional Belief Change between Conservatism and Moderation. *http://www.fil.lu.se/hommageawlodek/site/abstra.htm*.

Rott, H. (2009). Shifting Priorities: Simple Representations for Twenty-Seven Iterated Theory Change Operators. In D. Makinson, J. Malinowski, and H. Wansing (Eds.), *Towards Mathematical Philosophy: Trends in Logic*, Volume 28, 269–296. Springer, Business Media.

Rumelhart, D., Smolensky, P., McCelland, J., and Hinton, G. (1986a). *Parallel Distributed Processing: An Exploration in the Microstructure of Cognition*, Volume 1: Foundations. MIT Press.

Rumelhart, D., Smolensky, P., McCelland, J., and Hinton, G. (1986b). *Parallel Distributed Processing: An Exploration in the Microstructure of Cognition*, Volume 2: Psychological and Biological Models. MIT Press.

Russell, S. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Prentice Hall.

Sathasivam, S. (2011). Boltzmann Machine and New Activation. *Applied Mathematical Sciences*, *78*(5), 3853–3860.

Saygin, A., Cicekli, I., and Akman, V. (2000). Turing Test: 50 Years later. *Minds and Machines*, *10*, 463–518.

Schurz, G. (2005). Non-Monotonic Reasoning from an Evolution-Theoretic Perspective: Ontic, Logical and Cognitive Foundations. *Synthese*, *146*, 37–51.

Sejnowski, T. (1986). High Order Boltzmann Machines. In *Neural Networks for Computing*, Volume 151 of *American Institute of Physics Conference Proceedings 151*, 398–395.

Sejnowski, T. and Destexhe, A. (2000). Why Do We Sleep? *Brain Research*, *886*(1), 208–223.

Shastri, L. (2007). *SHRUTI: A Neurally Motivated Architecture for Rapid, Scalable Inference*, Volume 77 of *Perspectives in Neural Symbolic Integration*, Chapter 8, 183–204. Springer.

Shastri, L. and Ajanagadde, V. (1993). From Simple Associations to Systematic Reasoning: A Connectionist Representation of Rules, Variables and Dynamic Bindings using Temporal Synchrony. *Behavioural and Brain Sciences*, *16*, 417–494.

Shastri, L. and Wendelken, C. (2000). Probabilistic Inference and Learning in a Connectionist Causal Network. Technical report, International Computer Science Institute.

Sholam, Y. (1987). *Readings in Non-Monotonic Reasoning*, Chapter A Semantical Approach to Non-Monotonic Logics, 227–249. Morgan Kaufmann.

Singh, P. (2002). The Open Mind Common-Sense Project. *http://www.kurzweilai.net/the-open-mind-common-sense-project*.

Spekle, E. and Kinzer, K. (2007). Core Knowledge. *Developmental Science*, *10*(1), 89–96.

Spohn, W. (1988). Ordinal Conditional Functions: A Dynamic Theory of Epistemic States. In H. . Skyrms (Ed.), *Causation in Decision, Belief Change and Statistics*, Volume 11, 105–134. Kluwer Academic.

Spohn, W. (1999). Ranking Functions, AGM Style. *50th Birthday for Peter Gärdenfors, http://www.lucs.lu.se/spinning/*.

Spohn, W. (2009). *Degrees of Belief*, Volume 342 of *Synthese Library*, Chapter A Survey of Ranking Functions, 185–228. Springer.

Stenning, K. and Van Lambalgen, M. (2008). *Human Reasoning and Cognitive Science*. Cambridge MIT Press.

Strasser, C. and Antonelli, G. (2016). Non-Monotonic Logic. *Stanford Encyclopedia of Philosophy - https://plato.stanford.edu/archives/win2016/entries/logic-nonmonotonic/*.

Striedter, G. (2016). *Neurobiology*. Oxford University press.

Swingler, K. (1996). *Applying Neural Networks: A Practical Guide*. Academic Press Inc.

Tamminga, A. (2003). A Critical Exposition of Isaac Levi's Epistemology. *Logique and Analyse*, *183*, 447–478.

Tang, Y. and Salakhutdinov, R. (2013). Learning in Stochastic Feedforward Neural Networks. *Advances in Neural Information Processing Systems*, *2013*, 530–538.

Tarski, A. (1956). *Logic, Semantics, Meta-Mathematics: Papers from 1923 to 1938*. Clarendon Press.

Thornton, S. (2016). Karl Popper. *Stanford Encyclopedia of Philosophy - http://plato.stanford.edu/archives/win2016/entries/popper/*.

Tichnor, A. and Barret, H. (1987). Optical Implementations in Boltzmann Machines. *Optical Engineering*, *26*(1), 16–21.

Turing, A. (1950). Computing Machinery and Intelligence. *Mind*, *59*, 433–460.

UCI (2013). Machine Learning Repository. *http://archive.ics.uci.edu/ml/datasets.html*.

Uebel, T. (2016). Vienna Circle. *Stanford Encyclopedia of Philosophy - https://plato.stanford.edu/archives/spr2016/entries/vienna-circle/*.

van der Velde, F. and de Kamps, M. (2006). Neural Blackboard Architectures of Combinatorial Structures in Cognition. *Behavioural and Brain Sciences*, *29*, 37–108.

von der Malsburg, C. and Willshaw, D. (1981). Co-operativity and the Brain. *Trends in Neurosciences*, *4*(4), 80–83.

Wason, P. (1966). *New Horizons in Psychology I*, Chapter Reasoning, 135–151. Harmondsworth: Penguin.

Wason, P. (1969). Regression in Reasoning? *British Journal of Psychology*, *60*, 471–480.

Werbos, P. (1975). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Harvard University Press.

Wermter, S. (1997). Hybrid Approaches to Neural Network-Based Language Processing. Technical Report TR 97-030, ICSI Berkley.

Zadeh, L. (1986). A Simple View of the Dempster-Shafer Theory of Evidence and its Implication for the Rule of Combination. *Artificial Intelligence Magazine*, *7*, 85–90.

Zalta, E. (2016). Gottlob Frege. *Stanford Encyclopedia of Philosophy - http://plato.stanford.edu/archives/win2016/entries/frege/*.

Zufferey, C., Floreano, D., van Leeuwen, M., and Merenda, T. (2002). Evolving Vision-Based Flying Robots. In *International Workshop on Biologically Motivated Computer Vision*. Springer Berlin Heidelberg.

Appendix A

Vocabulary

A.1 Abbreviations

Table A.1: **Abbreviations/Acronyms used in the Text**

Abbreviation	Full Name
AGM	Alchourron, Gärdenfors & Makinson (postulates of belief revision)
ANN	Artificial Neural Network
DL	Description Logic
HLR	'Hidden Layer Rich' (modified Boltzmann machine)
ILP	'Input Layer Poor' (modified Boltzmann machine)
KLM	Kraus, Lehmann & Magidor (non-monotonic preferential semantics)
MLP	Multi Layer Perceptron (neural network)
NSI	Neural-Symbolic Integration
SCNM	Supra-Classical, Non-Monotonic (logic)
SNN	Symmetric (recurrent) Neural Network
SRL	Statistical Relational Learning

A.2 Glossary

Table A.2: **Terms used in the Text**

Term	Description
Artificial Intelligence	Rational cognitive function exhibited by machines. Involving: perception, learning and adaptation, knowledge representation, language processing, reasoning, planning, motion and object control. Section 2.2 Common-Sense Reasoning. Section 4.4 Discussion on Human Cognition.
Atomic Typicality	A total pre-order on state models representative of their specificity within a class. Based on the individual contributions of compositional characteristics, rather than a proto-typical exemplar. Section 5.3 Typicality.
Belief Revision	Formalisation of adapting an existing knowledge base, to new information, based on the AGM postulates. Including operations: contraction, expansion and revision. Chapter 6 Belief Revision.
Binary Logic	Two valued Boolean logic. [-1 false; +1 true] Section 3.2 Micro-World Schemata.
Biological Plausibility	Credible explanation related to biological causality. In the context of neuroscience, a neural network with particular biologically favourable attributes. Section 3.3.2 Neurobiology.
Boltzmann Machine	A symmetrically recurrent neural network with stochastic activation functions: Implementing Hebbian learning with alternating phases of clamped and free sampling, from a Boltzmann equilibrium distribution, achieved by simulated annealing. Section 2.3.1 Boltzmann Machine. Sections 4.1 & 4.2 Representation.

Term	Description
Catastrophic Forgetting	The loss or disruption of previously learned information when a neural network is exposed to new information. Section 2.1.2 Preferential Semantics. Section 6.2 Belief Revision: Machine Perspective.
Classical Symbolic Logic	A formal, recursively defined language for knowledge representation, together with a deductive system for codifying valid inference and model-theoretic semantics codifying truth conditions. An approach to artificial intelligence based on mathematics and philosophy, involving: propositional, first-order and modal systems. Chapter 1 Introduction. Chapter 2 Literature.
Conjunctive Normal Form	A Boolean formula with conjuncted clauses which are disjunctions of literals. Section 2.4.1 Early Symmetric Networks.
Connectionism	An approach to artificial intelligence based on biology, that models cognition as an emergent process in neural networks. Chapter 1 Introduction. Chapter 3 Background.
Contrastive Divergence	Gradient descent learning based on log-likelihood. Utilised in Markov models with limited connectivity. Computationally more efficient than simulated annealing. Section 4.1 Network Selection. Section 4.4.1 Place of the Boltzmann Machine.
Discriminative Model	A model of conditional probability, $P(h\|x)$ The probability of variable h given observations x. Section 4.3.3 Model Selection.
Energy Minima	A low energy state which represents a learned pattern. A local or global solution in terms of optimisation. A resonant state, Balkenius and Gärdenfors (1991). Section 2.3.1 The Boltzmann Machine. Section 2.4.1 Early Symmetric Networks.

Term	Description
Generative Model	Model of the joint probability distribution, $P(x, h)$ A complete description of the relationship between observations x and variable h. Section 4.3.2 Preference Relation.
Inductive or Predictive Inference	Argument based on a premiss to a probable conclusion. Applied in this thesis to inference, relying on a learned default rule and preference relation. Chapter 2 Literature.
Neural Network	Inter-connected network of simple processing units, whose activation functions take a sum of inputs. Learning algorithms achieve knowledge representation via weight adaptation. Section 2.3 Artificial Neural Networks. Chapter 3 Background.
Non-Monotonic Logic	Formalisation of defeasible inference enabling tentative conclusions which my be retracted. Based on KLM preferential semantics. Section 2.1 Supra-Classical, Non-Monotonic Logic Chapter 3 Background.
Pseudo-Rehearsal	The virtual re-training of previously learned information, by the re-creation of old training patterns. Chapter 1 Introduction. Section 6.2 Belief Revision: Machine Perspective.
Rational Consequence	An approach to inference in supra-classical logic, which selects those conclusions at maximal preference. Section 2.1 Supra-Classical, Non-Monotonic Logic
Restricted Boltzmann Machine	A symmetrically recurrent neural network with stochastic activation functions but with limited connectivity. Learning without simulated annealing, by contrastive divergence. Section 4.1 Network Selection. Section 4.4.1 Place of the Boltzmann Machine.

Term	Description
State Preference	A total pre-order on whole state models representative of their frequency within a training set, as an exemplar. The basis of preferential semantics in KLM. Section 2.1.2 Preferential Semantics. Section 3.1 Logical Preliminaries.
Supra-Classical Logic	A collection of logics which tolerate counter-examples and exceptions to a classically inferred conclusion. Encompasses non-monotonic logic. Section 2.1 Supra-Classical, Non-Monotonic Logic
Ternary Logic	Three valued logic. [-1 false, 0 unknown / not observed, +1 true] Section 3.2 Micro-World Schemata.

Appendix B

CD: Implementation

A CD containing the code implementation of the Boltzmann networks is attached with the thesis. The files included are:

- C++ code for the Boltzmann machine as an object-oriented project, with some ancillary files of data sets and machine weight matrices.
- The support files necessary for running the machine implementation under XCode in a MacOS X environment.

B.1 C++ Code for Boltzmann Machine

The C++ code for the Boltzmann networks can be found in the folder 'BoltzMachine'. The current version is **v7.5**. The code is object-oriented and needs to run in an environment like XCode or be complied as a stand alone project using an instruction such as:

```
g++ -W -Wall -ansi -pedantic -pthread MLibrary.cpp
Node.cpp Weight.cpp Layer.cpp WghtLayer.cpp NodeLayer.cpp
BoltzMach.cpp BMTestApp.cpp -o mach
```

Ancillary files are contained in the folder 'User': some data sets and machine weight matrices as text files.

B.1.1 XCode Support Files

To use the project in XCode, double click the 'BoltzMachine.xcodeproj' file to start the project (XCode greater than version v8.2). The 'Build/', 'Index/' and 'DerivedData/' folders are created by the XCode development. The 'Build/' folder provided contains a

'Products/Debug/' folder with the actual complied machine code, a folder for the input data sets used during runs and a results file for recording text output. It is important to give XCode preferences for the correct location; use the advanced tab to specify the location relative to the workspace in the 'Build/' folder.

B.2 Running the Implementation

For reference, we have repeated the diagram of the object-oriented implementation of the Boltzmann machine given in Chapter 4 (Figure B.1).

The project is organised as a hierarchy of objects each with a header file. The highest level 'BoltzMach' object contains the methods for creating, training and testing a machine. Its header file has some information about the versioning history. Below this level a machine is constructed from 'Weight Layers' and 'Node Layers' which inherit from a parent 'Layer' class. They in turn run methods which utilise more primitive objects, 'Nodes' and 'Weights'. There is an overarching application 'BMTestApp' which reads the input data files, sets up individual machines within threads and collates the output from them into a text file.

Input & Output

Perhaps the most likely problem with running the implementation will be with reading and writing data. The project expects the location of input and output data to be set relative to the path:

$$/BoltzmannMachine/Build/Products/Debug/$$

Input data sets should be placed in 'in-ptrns/' and machine weight matrices should be placed in 'machines/'. Output text data is written to 'results.txt'

Parameter Setting

Optional parameters for the project are set within the header files, 'Param.h' and 'PMSpf.h'. They control a huge variety of variables, from the most abstract (example: the purpose of a particular use of the project) to the most concrete (example: the scaling for weight decay). There is a generic parameter header 'Param.h' which contains information related to the primitive elements of any Boltzmann machine. That information doesn't usually have to be altered. It links to a specific micro-world and

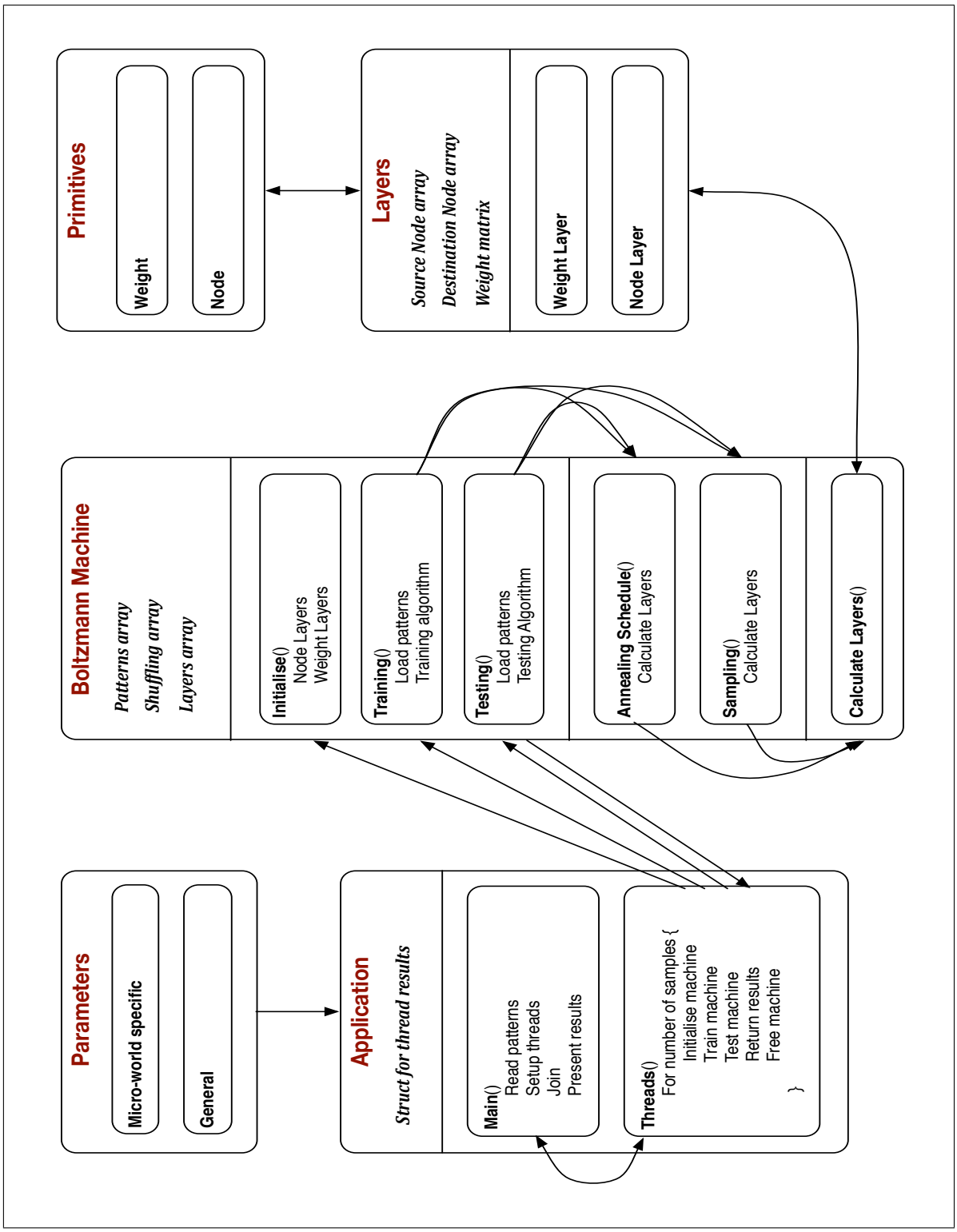


Figure B.1: Object-oriented implementation of a Boltzmann machine. This overview of of the project structure should aid understanding of the location of specific code.

machine header, 'PMSpf.h', which adjusts the details of use for individual machines. This header is attached to the generic header by an '#include' instruction.

The **'Param.h'** header contains the following generic information:

- Details of the ensemble size: number of threads and sequences within a thread.
- Learning constants: Number of training epochs, learning rate and momentum.
- Details of annealing schedules, training and testing can specified separately: threshold (temperature) range, cycles per temperature point and final sampling numbers.
- Printing of intermediate results can be specified.
- Constants specifying primitive weight parameters. These should not normally require adjustment. Initialisation, update, decay, ..
- Constants specifying primitive node parameters. These should not normally require adjustment. Biases, transition information, sparsity, ..

The micro-world and machine specific header, **'PMSpf.h'**, contains the following information which will need to be adjusted for each individual circumstance:

- Output constants, the file paths for text results and export of machine weight matrices.
- Input constants related to training and testing:
  - Multiple training runs can be specified from a single data set in which case multiple cohorts of machines will be trained **OR** from separately listed files in which case each machine will train from a separate source.
  - Multiple testing runs can be specified from a single data set in which case each trained machine will be tested against the same data multiple times **OR** tested on separately listed files in which case each machine will be tested against these separate sources.
- The architecture can be specified for a particular micro-world and machine, both nodes and connections. A comment at the top of the file indicates the current machine configuration.

- Nodes: the number of layers and the number of nodes in each layer. Two hidden layers are possible. The number of state combinations needs to be set as a constant, related to the atomic size of each micro-world.
- Connections within and between each layer of nodes need to be specified.
- For example:
  - For the 4-atom micro-world: NUMLY=3, NVINP=4, NVOUT=4 and NHID1=6 (HLR architecture).
  - For the HLR architecture: Only connections CHID1=true.
  - For the ILP architecture: NHID1=1 and only connections CVINP=true.

- At a more abstract level, the current utilisation of the project can be changed. Different modes of training and testing can be specified, for example:
  - In the most simple circumstance, machines are trained and then tested from scratch, parameters should be set as follows: TRAIN=true, EXPRT=false, IMPRT=false, RHRSL=0, SPLIT=true, XFLDS=false. SNPSH and PRINT will usually be set to false. The PCAPCT should be set to more than the number of patterns in the largest data set.
  - Weight matrices of trained machines (tested to a set tolerance) can be exported by setting EXPRT=true and specifying the appropriate path for the text files.
  - Weight matrices of previously trained and exported machines can be imported by setting IMPRT=true and specifying the appropriate path for the text files. If TRAIN=false these weight matrices are then used to create new machines for testing without retraining.
  - However, if TRAIN=true these newly imported machines will be retrained, potentially with any new data and with a mixture of re-cycling / pseudo-rehearsal versus retraining, specified by RHRSL=x(%).
  - A single input data set can be used for both training and testing using a cross-fold by specifying, SPLIT=false and XFLDS=x, the number patterns to be cross-folded. This must be an even integer division into the total number of patterns or the result will be meaningless.

I hope that the reader will not be too critical of my programming style and realise that this project evolved over years, in directions that could not have been foreseen at the beginning. Forgive me, when I have deliberately forsaken the principle of object encapsulation by utilising global variables; instead of re-writing every object interface multiple times as each new task was discovered.