A Connectionist Framework for Reasoning: Reasoning with Examples

Dan Roth*

Dept. of Appl. Math. & CS,
Weizmann Institute of Science,
Israel
danr@wisdom.weizmann.ac.il

Abstract

We present a connectionist architecture that supports almost instantaneous deductive and abductive reasoning. The deduction algorithm responds in few steps for single rule queries and in general, takes time that is linear with the number of rules in the query. The abduction algorithm produces an explanation in few steps and the best explanation in time linear with the size of the assumption set. The size of the network is polynomially related to the size of other representations of the domain, and may even be smaller.

We base our connectionist model on Valiant's Neuroidal model (Val94) and thus make minimal assumptions about the computing elements, which are assumed to be classical threshold elements with states. Within this model we develop a reasoning framework that utilizes a model-based approach to reasoning (KKS93; KR94b). In particular, we suggest to interpret the connectionist architecture as encoding *examples* of the domain we reason about and show how to perform various reasoning tasks with this interpretation. We then show that the representations used can be acquired efficiently from interactions with the environment and discuss how this learning process influences the reasoning performance of the network.

Introduction

Any theory aiming at understanding *commonsense* reasoning, the process that humans use to cope with the mundane but complex aspects of the world in evaluating everyday situations, should account for the flexibility, adaptability and speed of commonsense reasoning.

Consider, for example, the task of language understanding, which humans perform effortlessly and effectively. It depends upon our ability to disambiguate word meanings, recognize speaker's plans, perform predictions and generate explanations. These, and other "high level" cognitive tasks such as high level vision and planning have been widely interpreted as *inference* tasks and collectively comprise what we call commonsense reasoning.

Deductive and abductive reasoning are the basic inference tasks considered in the context of high level cognitive tasks. In this paper we suggest an alternative to the current connectionist account of these tasks.

Connectionist networks have been argued to be better suited than traditional knowledge representations for studying everyday common sense reasoning. Some of the arguments used are that these models have the ability to simultaneously satisfy multiple constraints, dynamically adapt to changes, achieve robustness and provide a useful way to cope with conflicting and uncertain information (Sun95; Pin95; Der90). This should be contrasted with the view that connectionist model are incapable of performing high level cognitive tasks because of their difficulties with representing and applying general knowledge rules (FP88).

The latter opinion, we believe, may reflect on the fact that a lot of the research on understanding high level cognition using connectionist models is actually *trying* to represent and apply general knowledge rules.

Indeed, a lot of the research in this direction is influenced by a research program launched in the fifties, the "*knowledge-base+inference engine*" approach (McC58), which is still the generally accepted framework for reasoning in intelligent systems. The idea is to store the knowledge, expressed in some *representation language* with a well defined meaning assigned to its sentences, in a Knowledge Base (KB). The KB is combined with a reasoning mechanism ("inference engine") that is used to determine what can be inferred from the sentences in the KB. The effort to develop a *logical inference engine* within a connectionist architecture is represented by works such as (BH93; HK91; SA90; SA93; Sun95; LD91; Pin95; Der90).

Given the intractability of the general purpose knowledge base+inference engine approach to reasoning, a significant amount of recent work in reasoning concentrates on (1) identifying classes of limited expressiveness, with which one can still perform reasoning efficiently or (2) resorting to an approximate inference engine. These directions have been pursued both in the knowledge representation and reasoning ($KR\&R$) community and in the connectionism com-

*Research supported by the Feldman Foundation and a Grant from the Israeli Ministry of Science and the Arts.

munity. The former line of research is represented in *KR&R* by many works such as (BL84; Lev92; Rot93; SK90; Cad95) and in the connectionism community by (SA90; BH93; HK91). The latter usually builds on using Hopfield's networks (HT82) or Boltzmann machines (HS86), in an effort to solve optimization problems that are relaxations of propositional satisfiability. This approach is used, for example, in (Pin95; Der90) and is related to approaches suggested in the *KR&R* community (SLM92; MJP90).

None of these works, however, meets the strong tractability requirements required for common-sense reasoning as argued e.g., in (Sha93). Moreover, many of these works have carried out the "knowledge base+inference engine" research program also by neglecting to consider the question of how this knowledge might be acquired[1] and by measuring performance of the reasoning process in absolute terms rather than with respect to the preceding learning process.

We utilize a model-based approach to reasoning (KKS93; KR94b) to yield a network that is not a "logical inference engine" but, under some (formally phrased) restrictions, behaves "logically" with respect to a world it interacts with. Our model-based algorithms support instantaneous deduction and abduction, in cases that are intractable using other knowledge representations. The interpretation of the connectionist architecture as encoding *examples* acquired via interaction with the environment, allows for the integration of the inference and learning processes (KR94a) and yields reasoning performance that naturally depends on the process of learning the network.

We develop the reasoning framework within Valiant's Neuroidal paradigm (Val94), a computational model that is intended to be consistent with the gross biological constraints we currently understand. In particular, this is a programmable model which makes minimal assumptions about the computing elements, assumed to be classical threshold elements with states.

In this abstract we focus on presenting the reasoning framework: the architecture, its interpretation as a set of examples and the reasoning algorithms. The learning issues are discussed only briefly.

The Reasoning Framework

This paper considers two inference tasks, Deduction[2] and Abduction. *Deduction*, the basic inference task considered in the context of high level cognitive tasks is usually modeled as follows: given a Boolean function W, represented as a conjunction of rules and assumed to capture our knowledge of the world, and a Boolean function α, a query that is supposed to capture the situation at hand, decide whether W logically implies α (denoted $W \models \alpha$). *Abduction* is a term coined by Peirce (Pei55) to describe the inference rule that concludes A from an observation B and the rule $A \rightarrow B$, given that there is no "better" rule explaining B. The importance of studying abduction became clear in the past few years when some general approaches to Natural Language interpretation have been advanced within the abduction framework (HSME93).

We adopt an alternative, model-based approach to the study of commonsense reasoning, in which the knowledge base is represented as a set of models (satisfying assignments) of the domain of interest (the "world") rather than a logical formula describing it. It is not hard to motivate a model-based approach to reasoning from a cognitive point of view and indeed, most of the proponents of this approach to reasoning have been cognitive psychologists (JL83; JLB91; Kos83), who have alluded to the notion of "reasoning from examples" on a qualitative basis. Building on the work of (KKS93; KR94b) we show that model-based reasoning can be implemented in a connectionist network to yield an efficient reasoning network.

In our framework, when reasoning with respect to the "world" W, information about the W is stored in a network N and is interpreted as a collections of examples observed in W.[3] We present both deduction and the abductive task of *verifying* that an explanation is consistent as a series of forward evaluation tasks. Each takes 5 computational steps. The task of *producing* an explanation utilizes the backwards connections in the networks, and is also instantaneous. In both cases, if the content of the network is a good representation of W, in a well defined sense, for a wide class of queries the network response is provably correct. Interaction with the network for queries presentation and learning the representation is done in a unified manner, via *observations* and the performance of the reasoning is shown to depend naturally on this interaction.

Reasoning Tasks

We briefly present the reasoning tasks and some relevant results. See (KR94b; KR94a) for details. We consider reasoning over a propositional domain. The reasoning queries are with respect to a "world" (domain of interest) that is modeled as a Boolean function (a propositional expression) $f : \{0,1\}^n \rightarrow \{0,1\}$. Let $X = \{x_1, \ldots, x_n\}$ be a set of *variables*, each of which is associated with a world's attribute and can take the value 1 or 0 to indicate whether the associated attribute is true or false in the world. (n is our complexity parameter.) An assignment $x \in \{0,1\}^n$ satisfies f if $f(x) = 1$. (x is also called a *model* of f.) By "f entails (implies) g", denoted $f \models g$, we mean that every model of f is also a model of g.

[1] (Pin95) is an exception.

[2] We emphasize that these terms are used only to give semantics to the network's behavior. The network is not a "logical inference engine" but, under some restrictions on the queries presented, behaves "logically" with respect to a world it had interactions with.

[3] We restrict our discussion to this fragment of the network; in general, this will be part of a larger network and will overlap with network representations of other "worlds".

In deduction (entailment), given Boolean functions f (assumed to capture our knowledge of the world) and α (a query that is supposed to capture the situation at hand) we need to decide whether f implies α (denoted $f \models \alpha$). For abduction, we refer here to one of the propositional formalisms in which abduction is defined as the task of finding a *minimal* explanation, given a knowledge base f (the *background theory*), a set of propositional letters A (the *assumption set*), and a query letter q. An *explanation* of q is a minimal subset $\mathcal{E} \subseteq A$ such that (1) $f \wedge (\wedge_{x \in \mathcal{E}} x) \models q$ and (2) $f \wedge (\wedge_{x \in \mathcal{E}} x) \neq \emptyset$. Thus, abduction involves tests for entailment (1) and consistency (2), but also a search for a minimal[4] explanation that passes both tests.

Reasoning with Models

The model based strategy for the deduction problem $f \models \alpha$ is to try and verify the implication relation using model evaluation. In doing so, the knowledge base consists of a set Γ of models of f rather than a Boolean function. When presented with a query α the algorithm evaluates α on all the models in Γ. If a counterexample x such that $\alpha(x) = 0$ is found, then the algorithm returns "No". Otherwise it returns "Yes".

Clearly, the model based approach solves the inference problem if Γ is the set of *all* models of f. However, the set of all models might be too large, making this procedure infeasible computationally. A model-based approach becomes useful if one can show that it is possible to use a fairly small set of models as the test set Γ, and still perform reasonably good inference.

Exact Reasoning using models is based on a theory developed in a series of papers (KKS93; KR94b; KR94a; KR95) where a characterization of when a model based approach to reasoning is feasible is developed. An important feature of the theory is that the correctness of reasoning depends on the type of *queries* presented and not so much on the world we reason about (provided that the reasoner holds a "good" description of the world). The class of queries which allows efficient model-based reasoning is called the class of *common queries* $(\mathcal{Q}_c)$. It contains a rich class of theories and, in particular, all Horn and all $\log n$CNF functions. Proving the feasibility of model-based reasoning involves showing that for the purpose of reasoning with respect to $\mathcal{Q}_c$, a Boolean function f can be represented using a polynomial size set of models, Γ_f.

Theorem 1 ((KR94b)) *For any knowledge base f there exists a set Γ_f of models whose size is polynomially[5] related to the DNF size of f. Deduction (with respect to $\mathcal{Q}_c$) and Abduction (given a query q and assumption set A) can be performed correctly in polynomial time, using Γ_f.*

Approximate Reasoning is related to the notion of pac learning (Val84) and was developed in (KR94a). We assume that the occurrences of observations in the world is governed by a fixed but arbitrary and unknown probability distribution D defined on $\{0,1\}^n$.

A query α is called (f,ϵ)*-fair* if either $f \subseteq \alpha$ or $Prob_D[f \setminus \alpha] > \epsilon$. An algorithm for approximate deduction will is to err on non-fair queries. (Intuitively, it is allowed to err in case $f \not\models \alpha$, but the weight (under D) of f outside α is very small.) Along with the *accuracy* parameter ϵ, we use a *confidence* parameter δ which stands for the small probability that the reasoning algorithm errs on fair queries.

Theorem 2 *Let $\mathcal{Q}$ be a class of queries of interest, and let $0 < \delta, \epsilon$ be given confidence and accuracy parameters. Suppose that we select $m = \frac{1}{\epsilon}(\ln|\mathcal{Q}| + \ln\frac{1}{\delta})$ independent examples according to D and store in Γ all those samples that satisfy f. Then the probability that the model-based deduction procedure errs on an (f,ϵ)-fair query in $\mathcal{Q}$ is less than δ.*

Since the queries in $\mathcal{Q}$ are Boolean functions of polynomial size, the number m of samples required is polynomial. Moreover, given a set of possible explanations as input, this approach efficiently supports the entailment and consistency stages of abductive reasoning.

The Connectionist Framework

The architecture investigated is based on Valiant's Neuroidal model (see (Val94) for details). We present just the few aspects we need to describe the knowledge representation that supports the reasoning tasks.

Valiant's Neuroidal model is a programmable model which makes minimal assumptions on the computing elements, assumed to be classical threshold elements with states. We make a few minor abstractions for methodological purposes. Most importantly, we abstract away the important notion that in the localist representation assumed, every item is represented as a "cloud" of nodes rather than a single node.

A 5-tuple $(G(G,E), W, M, \delta, \lambda)$ defines a network N. Here $G = G(G,E)$ is a *directed graph* describing the topology of the network, W is the set of possible *weights* on edges of G, M is the set of *modes* a node can be in at any instant, δ is the *update function of the mode* and λ is the *update function of the weights.*

We view the nodes of the net as a set G of propositions. The set E is a set of directed edges between the nodes. The set of weights W is a set of numbers. e_{ij} denotes the edge directed from i to j, and its weight is w_{ij}. Sometimes both $e_{ij}, e_{ji} \in E$. The *mode*, (s,T), of the node describes every aspect of its instantaneous condition other than the weights on its incoming edges. $s \in S$ is a finite set of states and T is a threshold. In particular, S consists of two kinds of states F and Q,

[4] Here minimal means that no subset of it is a valid explanation. In general this is not, by itself, adequate for choosing among explanations and more general schema can be discussed in our framework.

[5] Thus, Γ_f is in general exponentially smaller than the number of satisfying assignments of f, and sometimes even exponentially smaller than the DNF representation.

which stand for *firing* (that is, the node is active at this time), and *quiescent* (a non-active state).

The *mode transition function* δ specifies the updates that occur to the mode of the node from time t to $t+1$. δ depends on the current state of the node and the sum of weights $w_i = \sum_k \{w_{ki} | e_{ki} \in E, k \in F\}$, of its active parents. Similarly, the *weight transition function* λ defines for each weight w_{ij} at time t the weight to which it will transit at time $t+1$. The new value may depend on the values of the weights on the edges between node i and its parents, their firing state and the mode of i, all at time t. Two default transitions are assumed. First, a *threshold transition* by default occurs whenever $w_i > T_i$ at the ith node, provided that no explicit condition that overrides the default is stated. The second default assumed is that a node in a firing state ceases firing at the end of the time unit.

To further specify a network we need to define the *initial conditions IC*, (i.e., initial weights and modes of the nodes) and *input sequence IS*. The interaction of the network with the outside world is modeled by assuming the existence of *peripherals.* They have the power to cause various sets of nodes in the network to fire simultaneously at various times. Every interaction like that we call here an *observation.* It specifies the set of nodes that the peripherals activate at an instant, i.e., the set of propositions that are observed to be active in the environment. The actual choices of the sets and the times in which the observations are presented to the network determine the input sequences *IS*.

Timing is crucially important to the model. After the peripherals prompt the network and cause some subset of nodes to fire simultaneously, a cascade of computation follows, and the algorithm has to ensure that it terminates in a stable situation, before the time unit has elapsed. Typically, the peripherals will prompt low level nodes and the algorithm being executed may need to modify nodes representing higher level concepts, that are separated in the network from the prompted ones by several intermediate nodes.

Knowledge Representation

To emphasize the correspondence between the network and propositional reasoning, we consider a subset of the nodes in N which are controlled by the peripherals and view it as a set $X = \{x_1, \ldots, x_n\}$ of propositions. For simplicity, in order to describe both the presence and the absence of an attribute x_i, it is duplicated in the representation: one node describes x_i and another describes $\overline{x_i}$. We represent each interaction with the network as an observation $v = (x_{i_1} = v_{i_1}, x_{i_2} = v_{i_2}, \ldots, x_{i_d} = v_{i_d})$, with $d \leq n, v_i \in \{0,1\}$, and this is translated to a corresponding node activation by the peripherals. For example, when the observation is $(x_1 = 1, x_2 = 1, x_3 = 0)$, the peripherals activate nodes corresponding to x_1, x_2 and $\overline{x_3}$. An observation v can be interpreted also as a *query* presented to the network in the reasoning stage. The presentation of v is interpreted as the Boolean query $\alpha = l_{i_1} \wedge \ldots \wedge l_{i_d}$, where $l_j = x_j$ if $v_j = 1$, $l_i = \overline{x_i}$ if $v_j = 0$.

Definition 1 *Let y be a node in N, $E_y = \{z | e_{zy} \in E\}$ its set of parents. A node $z \in E_y$ is called a* model *of y and $e_z = \{i \in E_z | w_{iz} = 1\}$ its set of* components. *The* model-based representation *of y, $M_y = \{(z, e_z) | z \in E_y\}$, is the set of models and their components.*

We assume also that the positive and negative literals of each proposition are connected via a *relay node.* Figure 1 depicts a model-based representation of y. The edges are assumed to be bidirectional (i.e., each line represents two edges) and all the weights on the edges drawn are assumed to be 1. Every model is connected to all $2n$ literals, and the n not drawn are assumed to have weight 0. Initially, all the thresholds in the representation are set to a high value, denoted by ∞. The algorithms also assume a specific set of initial modes of the nodes in the representation.

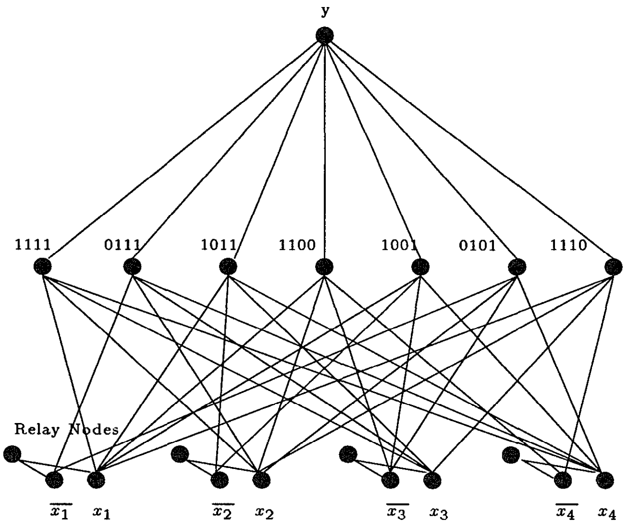

Figure 1: A Connectionist Model-Based Representation

A model z can be represented as a Boolean vector. If $e_z = (l_1, l_2, \ldots l_n)$ is a representation of z as a set of its components ($l_i \in \{x_i, \overline{x_i}\}$), than $e_z = [b_1, b_2, \ldots b_n]$ is its Boolean representation, where $b_i \in \{0,1\}$ is defined by: $b_i = 1$ if $l_i = x_i$, $b_i = 0$ if $l_i = \overline{x_i}$. It can be verified that the model-based representation presented in Figure 1 is the representation of the function $f = \{\overline{x_1} \wedge \overline{x_2} \rightarrow x_3,\ \overline{x_1} \wedge \overline{x_4} \rightarrow x_2,\ x_1 \wedge x_2 \wedge x_4 \rightarrow x_3\}$ with respect to all Horn queries. (See (KR94b).)

In general, a network N will be a collection of such model-based representations. These can share nodes and any input to the network may influence many of them. Thus, although we discuss "logical" behavior, no global consistency is required. Note that while n is our complexity parameter, it is not related to the size of the whole network, but only to the number of propositions "related" to y in its local network.

Reasoning in the Network

We briefly describe the reasoning algorithms, for lack of space. A complete description appears in (Rot96a).

We note that within this framework, there are quite a few other ways to achieve the same goal. In particular, we could define other modes and use other ways to evaluate the queries on the models stored in the model-based representation. We emphasize two design decisions that we view as important to the approach. First, queries are presented to the network as conjunctions. Thus, consistently with the natural interface considered when learning the network, queries are viewed as *observations* – a list of propositions that are active (or non-active) in the environment. Second, in our algorithms, the top node, where the decision is being made, need not know the size of its input domain (the number of propositional letters). This is essential also to the extension to reasoning with incomplete information.

Let N be a network which contains a model-based representation for f. That is, there exists a network structure as in Definition 1 and Figure 1. We imply nothing on the models stored in the network (i.e., which are the components of the models). We also assume that various nodes are in suitable initial states.

Algorithms are described in the format of a sequence of steps (following (Val94)). First, we describe the initial (pre-)conditions assumed in the network. The input ("prompt") is orchestrated by the peripherals, which also "collect" the algorithm's response, represented as a pattern of firings of one or more nodes. At each step, "prompt" describes the input at this stage - the set of nodes that the peripherals force to fire this time. Then, we define the transitions that are invoked during the following time unit at the relevant nodes. All other aspects of the algorithm are fully distributed. The effect of the algorithm on any node not directly prompted is completely determined by the transition rules and by the conditions at this node and at its parents. The overall algorithm can be invoked at any time, by having the preconditions of the first step satisfied as a result of an appropriate prompt.

Deduction Consider the deduction problem $f \models \alpha$. Queries are presented to the network as conjunctions of rules, $\alpha = C_1 \wedge \ldots \wedge C_k$. Every rule has the form $C = A \rightarrow B$, where A and B are conjunctions of literals. Since $f \models C_1 \wedge \ldots \wedge C_k$ iff $f \models C_i \ \forall i \in (1, k)$, it is sufficient to consider the single[6] rule case.

We respond to $f \models (A \rightarrow B)$ using the following version of the reasoning with models algorithm. Given the set Γ of models, filter out the models that do not satisfy A. Respond *no* (y inactive) iff one of the remaining models (which satisfied A) does not satisfy B.

Only the top node y and the example nodes take part in the deduction algorithm *AlgD*. It takes five steps: in the first two steps, the A part of the query is presented by the peripherals and is evaluated on all the models; in the next two steps, the B part of the query is presented by the peripherals and is evaluated on all the models that satisfied the A part; finally, the top node fires if all the models that satisfied A satisfy also B.

In the first step, an example node that receives activity wakes up and stores the total incoming weight for later comparison. A weight flip is used to evaluate the query presented to the network on the examples stored in it. In the second step, the same propositional nodes are prompted. This time, due to the weight flip, an example satisfies the observation (query) presented iff the input it sees doubles. In this case it fires and changes its mode to wait for the second part of the query. The same mechanisms works for the second part of the query, but applies only to examples which satisfied A. Therefore, it is sufficient for the top node to record (by setting its threshold) the *number* of these examples and make sure they all satisfy B also. Finally, the peripherals also prompt the target node y and this is used for the case where no model satisfies A, in which the response should also be "yes". The algorithm also makes sure that all the nodes return to their original states. Depending on the content of the representation we can prove:

Theorem 3 *Let y be a node in the network N, and let M_y be its model-based representation. (1) If M_y consists of the set of models $\Gamma_f^{\mathcal{Q}_c}$ then AlgD performs correct deduction whenever presented with a common query. (2) If M_y consists of a set of models of f acquired by sampling the environment according to distribution D then, with high probability, AlgD performs correct deduction whenever presented with an (f, ϵ)-fair query with respect to D.*

Abduction The algorithms for abductive reasoning, are not presented here. They perform the following tasks: (i) Given a candidate explanation $\mathcal{E}$ and a query, verify that $\mathcal{E}$ is a valid explanation. (ii) Provided that candidate explanations are represented as dedicated nodes in the network, given a query, the algorithm fires a valid explanation $\mathcal{E}$. All these tasks can be performed in constant time. In addition, the peripherals can use (i) to greedily present (subsets of) the collected output of (ii), in search for a minimal explanation.

The algorithm is similar to the deduction algorithm with the main distinction being that in this case we utilize the relay nodes and the backwards connections in order to communicate information down the network.

Learning to Reason

An essential part of the developed framework is that reasoning is performed by a network that has been learned from interaction with the environment (KR94a). For this purpose we have defined the interac-

[6] It is easy to extend the algorithm to handle sequentially the presented rules, timed by the peripherals, and respond only after seeing the last rule. The thing to note is that it takes constant time to respond to a single rule, and the total time is linear in the number of rules.

tion with the network via queries that are represented as *observations*. This allows for combining the interfaces to the world used by known learning models with the reasoning task. For example, the main avenue of interaction with the world used in the formal study of learning is an *Example Oracle*. When accessed, this oracle returns $v \in \{0,1\}^n$, drawn at random according to a distribution D; v can be viewed as an *observations* $v = (v_{i_1}, \ldots, v_{i_d})$ and interpreted also as a query. Examples presented in this way can be "memorized" into our network (Val94), and in combination with *AlgD* this provides a Learning to Reason algorithm that interacts with the environment, learns a model-based representation and supports correct entailment. Furthermore, using (2) of Theorem 3, the dependence of the reasoning performance on the learning process can be stated qualitatively. This type of interaction is supported also by the *on-line* L2R models (KR94a; Rot95) and can be shown to support other reasoning tasks, when augmented with membership and reasoning queries.

Conclusion

This paper develops a new approach to reasoning in connectionist networks. We suggest to interpret the connectionist architecture as encoding *examples* and show how to perform various reasoning tasks with this interpretation. Assuming the network encodes a (reasonably small) set of *representative* examples of a "world", we proved that our algorithms perform correct deduction and abduction, tasks that were considered intractable under other knowledge representations. Moreover, our framework naturally supports Learning to Reason and the representations used can be efficiently acquired by interaction with the world.

We believe that these results make this model suitable for studying *reflexive* reasoning (Sha93; Val94).

This work is part of a project in which we are trying to understand how networks of simple and slow neuron-like elements can encode a large body of knowledge and perform a wide range of interesting inferences almost instantaneously. It provides the theoretical framework for a system that learns knowledge representations for natural language understanding tasks.

References

A. Beringer and S. Holldobler. On the adequateness of the connection method. In *AAAI-93*, pages 9–14.

R. Brachman and H. Levesque. The tractability of subsumption in framebased description languages. *AAAI-84*.

M. Cadoli. *Tractable Reasoning in Artificial Intelligence*. Springer-verlag, 1995. Lec. notes in AI, vol 941.

M Dertthick. Mundane reasoning by settling on a plausible model. *Artificial Intelligence*, 46:107–157, 1990.

J. A. Fodor and Z. W. Pyilshyn. Connectionism and cognitive architecture: a critical analysis. *Cognition*, 28, 1988.

S. Holldobler and F. Kurfeb. CHCL *Parallelization in Inference Systems*. Springer-Verlag, 1991.

G. E. Hinton and T. J. Sejnowski. Learning and relearning in Botzmann machines. *PDP (Volume I: Foundations)*, pages 282–317. MIT Press, 1986.

R. Hobbs, J, M. Stickel, P. Martin, and D. Edwards. Interpretation as abduction. *Art. Intell.*, 63:69–142, 1993.

J. J. Hopfield and D. W. Tank. Neural computation of decisions in optimization problems. *Biol. Cyber.*, 1982.

P. N. Johnson-Laird. *Mental Models*. Harvard Press, 1983.

P. N. Johnson-Laird and R. M. J. Byrne. *Deduction*. Lawrence Erlbaum Associates, 1991.

H. Kautz, M. Kearns, and B. Selman. Reasoning with characteristic models. In *AAAI-93*, pages 34–39.

S. M. Kosslyn. *Image and Mind*. Harvard Press, 1983.

R. Khardon and D. Roth. Learning to reason. In *AAAI-94*, pages 682–687.

R. Khardon and D. Roth. Reasoning with models. In *AAAI-94*, pages 1148–1153.

R. Khardon and D. Roth. Default-reasoning with models. In *IJCAI-95*, pages 319–325.

T. E. Lange and M. G. Dyer. High-level inferencing in a connectionist network. *Connection Science*, 1991.

H. Levesque. Is reasoning too hard ? In *Proceeding of the 3rd NEC research Symposium*. 1992.

J. McCarthy. Programs with common sense. In R. Brachman and H. Levesque, *Readings in KR, 1985.*

S. Minton, M. D. Johnson, and A. B. Phillips. Solving large scale constraint satisfaction and scheduling problems using a heuristic repair method. In *AAAI-90.*

C. S. Peirce. *Abduction and Induction*. Dover, NY, 1955.

G. Pinkas. Reasoning, nonmonotonicity and learning in connectionist network that capture propositional knowledge. *Artificial Intelligence*, 77:203–247, 1995.

D. Roth. On the hardness of approximate reasoning. In *IJCAI-93*, pages 613–618.

D. Roth. Learning to reason: The non-monotonic case. In *IJCAI-95*, pages 1178–1184.

D. Roth. A connectionist framework foı reasoning: Reasoning with examples. Tech. report, Dept. of App. Math. and CS, Weizmann Inst. of Science, 1996.

L. Shastri and V. Ajjanagadde. An optimally efficient limited inference system. In *AAAI-90*, pages 563–570.

L. Shastri and V. Ajjanagadde. From simple associations to systematic reasoning: A connectionist representation of rules, variables and dynamic binding using temporal synchrony. *Behavioral and Brain Sciences*, 1993.

L. Shastri. A computational model of tractable reasoning - taking inspiration from cognition. In *IJCAI-93*.

B. Selman and H. Kautz. Model-preference default theories. *Artificial Intelligence*, 45:287–322, 1990.

B. Selman, H. Levesque, and D. Mitchell. A new method for solving hard satisfiability problems. In *AAAI-92.*

R. Sun. Robust reasoning: Integrating rule-based similarity-based reasoning. *Artificial Intelligence*, 1995.

L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, November 1984.

L. G. Valiant. *Circuits of the Mind*. Oxford University Press, November 1994.