Report Number: WUCS-91-03

1991-04-01

Propositional Non-Monotonic Reasoning and Inconsistency in Symmetric Neural Networks

Gadi Pinkas

We define a notion of reasoning using world-rank-functions, independently of any symbolic language. We then show that every symmetric neural network (like Hopfield networks or Boltzman machines) can be seen as performing a search for a satisfying model of some knowledge that is wired into the network's topology and weights. Several equivalent languages are then shown to describe symbolically the knowledge embedded in these networks. We extend propositional calculus by augmenting assumptions with penalties. The extended calculus (called "penalty logic") is useful in expressing default knowledge, preference between arguments, and reliability of assumptions in an inconsistent knowledge base. Every... **Read complete abstract on page 2.**

Propositional Non-Monotonic Reasoning and Inconsistency in Symmetric Neural Networks

Gadi Pinkas

Complete Abstract:

We define a notion of reasoning using world-rank-functions, independently of any symbolic language. We then show that every symmetric neural network (like Hopfield networks or Boltzman machines) can be seen as performing a search for a satisfying model of some knowledge that is wired into the network's topology and weights. Several equivalent languages are then shown to describe symbolically the knowledge embedded in these networks. We extend propositional calculus by augmenting assumptions with penalties. The extended calculus (called "penalty logic") is useful in expressing default knowledge, preference between arguments, and reliability of assumptions in an inconsistent knowledge base. Every symmetric network can be described by this language and any sentence in the language is translatable to such network. A proof-theoretic reasoning procedure supplements the model-theoretic definitions and gives an intuitive understanding of the non-monotonic behavior of the reasoning mechanism. Finally we sketch a connectionist inference engine for penalty logic and discuss its capabilities and limitations.

Propositional Non-Monotonic Reasoning and Inconsistency in Symmetric Neural Networks

Gadi Pinkas

WUCS-91-03

April 1991

Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
Saint Louis, MO 63130-4899

To appear in the Proceedings of the 12th International Joint Conference on Artificial Intelligence, 1991.

Propositional Non-Monotonic Reasoning and Inconsistency in Symmetric Neural Networks

Gadi Pinkas *

April 29, 1991
WUCS-91-03

Computer Science Dept.,
Campus Box 1045,
Washington University,
St. Louis, MO 63130
E.mail: pinkas@cics.wustl.edu. Tel:(314) 726-7526

Abstract

We define a notion of reasoning using world-rank-functions, independently of any symbolic language. We then show that every symmetric neural network (like Hopfield networks or Boltzman machines) can be seen as performing a search for a satisfying model of some knowledge that is wired into the network's topology and weights. Several equivalent languages are then shown to describe symbolically the knowledge embedded in these networks. We extend propositional calculus by augmenting assumptions with penalties. The extended calculus (called "penalty logic") is useful in expressing default knowledge, preference between arguments, and reliability of assumptions in an inconsistent knowledge base. Every symmetric network can be described by this language and any sentence in the language is translatable to such network. A proof-theoretic reasoning procedure supplements the model-theoretic definition and gives an intuitive understanding of the non-monotonic behavior of the reasoning mechanism. Finally we sketch a connectionist inference engine for penalty logic and discuss its capabilities and limitations.

To appear in the Proceedings of the 12th International Joint Conference on Artificial Intelligence, 1991.

*This research was supported in part by NSF grant 22-1321 57136.

Propositional Non-Monotonic Reasoning and Inconsistency in Symmetric Neural Networks

Gadi Pinkas

1. Introduction

Recent non-monotonic (NM) systems are quite successful in capturing our intuitions about default reasoning. Most of them, however, are still plagued with intractable computational complexity, sensitivity to noise, inability to combine other sources of knowledge (like probabilities, utilities...) and inflexibility to develop personal intuitions and adjust themselves to new situations. Connectionist systems may be the missing link. They can supply us with a fast, massively parallel platform; noise tolerance can emerge from their collective computation; and their ability to learn may be used to incorporate new evidence and dynamically change the knowledge base. We shall concentrate on a restricted class of connectionist models, called symmetric networks ([Hopfield 82],[Hopfield 84], [Hinton, Sejnowski 86]). They are characterized by recurrent network architecture, symmetric weight matrix (with zero diagonal) and a quadratic energy function that should be minimized.

We shall demonstrate that symmetric networks are natural platforms for propositional defeasible reasoning and for noisy knowledge bases. In fact we shall show that every such network can be seen as encapsulating a body of knowledge and as performing a search for a satisfying model of that knowledge.

Our objectives in this paper are first to investigate the kind of knowledge that can be represented by those symmetric networks, and second, to build a connectionist inference engine capable of NM reasoning from incomplete and inconsistent knowledge, expressed symbolically.

2. Reasoning with World Rank Functions

We begin by giving a model-theoretic definition for an abstract reasoning formalism. Our objective is to define such formalism independently of any symbolic language. Later we shall use it to give an

interpretation for the knowledge embedded in symmetric neural nets (SNN), and for the reasoning process performed by them.

DEFINITION **2.1** A *World Rank Function* (WRF) with respect to a set of possible worlds M is a function $k : M \rightarrow \mathcal{R} \cup \{\infty\}$ that ranks each of the possible worlds with a number that is in $(-\infty \ldots \infty]$.
A WRF is *propositional* iff it is defined over the set of truth assignments (i.e., $dom(k) = 2^n$), and it is *strict* iff it ranks the worlds using only 0 and ∞. (i.e., $range(k) = \{0, \infty\}$).

A WRF k *always* carries preference knowledge (a partial order induced over the worlds by the relation $<$ in $\mathcal{R}$; see [Lehmann 89]). We say that Ω_1 is *preferred* over Ω_2 if $k(\Omega_1) < k(\Omega_2)$, and if two worlds are ranked equally, we say that there is no reason to prefer one over the other. In addition, we *may* interpret the world's rank as the utility assigned to the world, or even give it a probabilistic interpretation, like: $P(\Omega_1)/P(\Omega_2) = e^{(k(\Omega_1)-k(\Omega_2))}$ (as in [Derthick 88]).

DEFINITION **2.2** A world Ω is called *a model* of a WRF k iff $k(\Omega) < \infty$.
A model Ω *satisfies* a WRF k iff it minimizes the function; i.e., $k(\Omega) = min_k$.
Let Γ_k be the set of all satisfying models of WRF k. We say that k *entails* k' ($k \models k'$) iff all the models that satisfy k satisfy also k'; i.e., $\Gamma_k \subseteq \Gamma_{k'}$.[1]

NM systems "jump" to conclusions based on evidence given, and later may retract those conclusions based on new evidence. It is convenient to divide the knowledge from which we want to reason into "background" knowledge and "evidence" (see [Geffner 89]). The background knowledge is relatively fixed and there should be an easy way to combine evidence with it. In our formalism, combining the evidence is simply done by adding together the two WRFs.

DEFINITION **2.3** Let f, e, k be WRFs. Evidence e entails f with respect to a background knowledge k, written $(e \overset{k}{\models} f)$, iff $k + e \models f$. The evidence function e is usually but not necessarily strict.

[1] We will usually use entailment of strict WRFs.

3. Connectionist energy functions

In the next section we briefly review symmetric connectionist models and explain the concept of energy functions. We then show that every symmetric neural network (SNN) can be viewed as performing a search for a satisfying model of some WRF.

3.1. Symmetric connectionist models

Connectionist networks with symmetric weights use gradient descent to find a minimum for quadratic energy functions. A k-order energy function is a function $E : \{0,1\}^n \rightarrow \mathcal{R}$ that can be expressed in a sum-of-products form with product terms of up to k variables. We denote this sum-of-products form by:

$$\sum_{1\leq i_1<i_2<\cdots<i_k\leq n} w_{i_1,\ldots,i_k} x_{i_1}\cdots x_{i_k} + \sum_{1\leq i_1<\cdots<i_{k-1}\leq n} w_{i_1,\ldots,i_{k-1}} x_{i_1}\cdots x_{i_{k-1}} + \cdots + \sum_{1\leq i\leq n}^{1} w_i x_i.$$

Quadratic energy functions are special cases of energy functions in the form :

$$E(x_1,\ldots,x_n) = \sum_{1\leq i<j\leq n} w_{ij} x_i x_j + \sum_{i\leq n} w_i x_i.$$

We can arbitrarily divide the variables of an energy function into two sets: visible variables and hidden variables. The visible variables represent visible atomic propositions, while the hidden variables are usually not of interest to an external observer and are used for computational reasons. An energy function with $x_1 \ldots x_n$ visible variables and $t_1 \ldots t_k$ hidden variables is denoted by $E(\vec{x},\vec{t})$.

There is a direct mapping between the quadratic energy functions described above and connectionist networks with symmetric weights that minimize them. Each variable of the function maps into a neuron unit - a node in a graph (hidden variables are mapped into hidden units), and is connected by symmetric arcs to other units Unit i is connected to unit j by a weight w iff the energy function includes a term of the form: $-wx_ix_j$. A unit i has a threshold w iff the energy function includes a term of the form: wx_i. Hyper-graphs with Sigma-Pi units can be used for minimizing high-order energy functions with similar mapping ([Sejnowski 86]).

3.2. Reasoning with energy functions

A SNN can be viewed as an implementation of a WRF. A model is a zero/one truth-assignment to the visible variables of the function. For each such assignment we can compute a *rank* by clamping the visible variables to the zero/one values of the assignment and letting the free variables $\vec{t}$ (the hidden units) settle to values that minimize the function.

DEFINITION **3.1** The *rank* of the energy function $E(\vec{x},\vec{t})$ is the minimum value $E(\vec{x},\vec{t})$ gets for all possible assignments of the hidden variables $\vec{t}$; i.e., $Erank_E(\vec{x}) = MIN_{\vec{t}}\{E(\vec{x},\vec{t})\}$.

For every SNN there is an $Erank_E$ function that is associated with its energy function E We consider this function to be the WRF implemented by the network[2] (similarly, every WRF can be approximated by a SNN). A search performed by the SNN for a global minimum is then interpreted as a search for a satisfying model for the WRF $Erank_E$.[3] We will interchangeably use energy functions, rank functions and conventional graph description to represent the functional behavior of symmetric networks.

4. Symbolic languages to describe WRFs

Our next step is to describe symbolically the knowledge that is encapsulated in a network. We shall define several languages and show their equivalence. We shall allow transformation from one form of knowledge representation to another, and for that purpose, we define several types of equivalence relations that preserve basic properties of the knowledge.

DEFINITION **4.1** A calculus is a triple $< \mathcal{L}, m(), M >$ where $\mathcal{L}$ is a language, M is a set of possible worlds and $m : \mathcal{L} \rightarrow \{k \mid k \text{ is a WRF }\}\}$ is a function that returns for each sentence of the language $\mathcal{L}$ a WRF. $m(s)$ is called the *interpretation* of the sentence s.

Let $s, s', e, k \in \mathcal{L}$; a model Ω *satisfies* s ($\Omega \models s$) iff it satisfies $m(s)$.

A sentence s *entails* sentence s' ($s \models s'$) if the WRF $m(s)$ entails the WRF $m(s')$. Similarly, a combination

[2] In [Pinkas 90c] we showed an algorithm that generates for every energy function E an equivalent (possibly) higher order energy function that does not have hidden units. This high-order function is exactly $Erank_E$.

[3] A Boltzman machine implementation of an energy function (without hidden units) also mimics the probabilistic knowledge that is carried by the function, since at equilibrium when the temperature is one, $P(\Omega_1)/P(\Omega_2) = e^{-(E(\Omega_1)-E(\Omega_2))}$.

of a background sentence with an evidence sentence is interpreted as the addition of their corresponding WRFs; i.e., $e \overset{k}{\models} s$ iff $(m(e) + m(k)) \models m(s)$.

Both classic predicate logic and propositional logic can be viewed as calculi whose languages describe strict WRFs.

EXAMPLE **4.1** Propositional calculus is $< \mathcal{L}, m(), 2^n >$, where $\mathcal{L}$ is the language of propositional well formed formulae (WFFs) and $m()$ outputs the function $(\infty(1 - H_s))$, when given a WFF s. $H_s(\vec{X})$ is the characteristic function of the WFFs and is recursively defined as:

$$H_s(\vec{X}) = \begin{cases} X_i & \text{if } s = X_i \text{ is an atomic proposition} \\ 1 - H_{s'}(\vec{X}) & \text{if } s = \neg s' \\ H_{s_1}(\vec{X}) \times H_{s_2}(\vec{X}) & \text{if } s = s_1 \wedge s_2 \\ H_{s_1}(\vec{X}) + H_{s_2}(\vec{X}) - H_{s_1}(\vec{X}) \times H_{s_2}(\vec{X}) & \text{if } s = s_1 \vee s_2 \end{cases}$$

The reader can easily observe that any propositional WFF describes a strict WRF that returns 0 for truth assignments that satisfy the WFF and ∞ for assignments that do not satisfy it.

DEFINITION **4.2** Let $s \in \mathcal{L}_1$ and $s \in \mathcal{L}_2$ be sentences of two (possibly different) calculi $< \mathcal{L}_1, m_1, M >$ and $< \mathcal{L}_2, m_2, M >$; we define equivalence between them:

1. s is *strongly equivalent* to s' ($s \overset{s}{\approx} s'$) iff their corresponding WRFs are equal, up to a constant difference; i.e., $m(s) = m'(s') + c$. We call this equivalence "magnitude preserving" or s-equivalence.

2. s is p-equivalent to s' ($s \overset{p}{\approx} s'$) iff their associated WRFs induce the same ordering over the set of worlds; i.e., $\forall \Omega_1, \Omega_2, m(s)(\Omega_1) < m(s)(\Omega_2)$ iff $m'(s')(\Omega_1) < m'(s')(\Omega_2)$. We call this equivalence "preference preserving" or p-equivalence.

3. s is weakly equivalent to s' ($s \overset{w}{\approx} s'$) iff their corresponding WRFs have the same sets of satisfying models; i.e., $\Gamma_{m(s)} = \Gamma_{m'(s')}$. We call this equivalence "minima preserving" or w-equivalence.

OBSERVATION **4.1** *1. If two background sentences are strongly equivalent, then for any given evidence function e, the two corresponding WRFs entail the same set of conclusions; i.e., if e is* any *evidence and* $k \overset{s}{\approx} k'$, *then for every WRF f,* $(m(k) + e) \models f$ *iff* $(m'(k') + e) \models f$.[4]

[4] In addition, two strongly equivalent WRFs have the same probabilistic interpretation since $P(\Omega_1)/P(\Omega_2) = e^{(k(\Omega_1) - k(\Omega_2))} = e^{((k'(\Omega_1)+c) - (k'(\Omega_2)+c))}$.

2. *If two background sentences are p-equivalent, then for every* strict *evidence e, the two corresponding WRF's entail the same set of conclusions. I.e: if* $dom(e) = \{0, \infty\}$ *and* $k \overset{p}{\approx} k'$, *then for every WRF* f, $(m(k) + e) \models f$ *iff* $(m'(k') + e) \models f$. *We can't guarantee this property for any non-strict evidence.*

3. *If two sentences* k, k' *are weakly equivalent, then for every WRF* f, $m(k) \models f$ *iff* $m'(k') \models f$. *We can't guarantee this property to hold once we try to add evidence to both* k *and* k'.

If all we want is to preserve the set of conclusions achievable from a piece of knowledge, we may use transformations which only preserve the minima (weak equivalence). If however we would like to be able to combine strict evidence to our transformed knowledge, we need to perform "preference preserving" transformations. We need "magnitude preserving" transformations (strong equivalence) if we want to combine *any* evidence or give probabilistic/utility interpretation to our transformed knowledge. Most of our transformations in the reminder of this paper are "magnitude preserving".

We define now an equivalence between two calculi.

DEFINITION 4.3 A calculus $C_1 =< \mathcal{L}, 2^n, m >$ is (s-/p-/w-) equivalent to a calculus $C' =< \mathcal{L}', 2^n, m' >$ iff for every $s \in \mathcal{L}$ there exists a (s-/p-/w-) equivalent $s' \in \mathcal{L}'$ and for every $s' \in \mathcal{L}'$ there exists a (s-/p-/w-) equivalent $s \in \mathcal{L}$.

We thus can use the language $\mathcal{L}$ to represent every WRF that is representable using the language $\mathcal{L}'$, and vice versa. In the sections to come we shall present several equivalent calculi and show that all of them describe the knowledge embedded in SNNs.

5. Calculi for describing symmetric neural networks

The algebraic notation that was used to describe energy functions as sum-of-products can be viewed as a propositional WRF. The *calculus of energy functions* is therefore $< \{E\}, 2^n, m() >$, where $\{E\}$ is the set of all strings representing energy functions written as sum-of-products, and $m(E) = Erank_E$. Two special cases are of particular interest: the calculus of quadratic functions and the calculus of high-order energy functions with no hidden variables.

In [Pinkas 90a],[Pinkas 90c] we gave algorithms that 1) Convert high-order energy functions to strongly equivalent[5] low-order ones with additional hidden variables, and 2) Convert energy functions with hidden variables to strongly equivalent (possibly) higher order ones without those hidden variables. These algorithms allow us to trade the computational power of Sigma-Pi units with additional hidden units. We therefore conclude that the calculus of high-order energy functions with no hidden units is strongly equivalent to the calculus of quadratic functions. Thus, we can use the language of high-order energy functions with no hidden units to describe any symmetric neural network (SNN) (with arbitrary number of hidden units). For the remainder of this paper we will use this high-level language to describe symmetric networks without being concerned about hidden units.

In [Pinkas 90a] we also proved that the satisfiability of propositional calculus is equivalent (weakly) to quadratic energy minimization. We gave algorithms to convert any satisfiable WFF to a weakly equivalent quadratic energy function (of the same order of length), and every energy function to a weakly equivalent satisfiable WFF. As a result, propositional calculus is weakly equivalent to the calculus of quadratic energy functions and can be used as a high-level language to describe SNNs. However, two limitations exist: 1) The algorithm that converts an energy function to a satisfiable WFF may generate an exponentially long WFF; and 2) The equivalence is *weak*; although the WFF and the energy function have the same set of satisfying models, neither evidence can be added nor a utility or probabilistic interpretation is preserved. (Examples of p-equivalent calculi exist but they are out of the scope of this paper).

In the next section we will define a new logic calculus that is strongly equivalent to the calculus of energy functions and that does not suffer from the limitations discussed above.

6. Penalty calculus

We now extend propositional calculus by augmenting assumptions with penalties (like in [Derthick 88]). The extended calculus is able to deal with an inconsistent knowledge base (due to noise, errors in observations, *etc...*) and will be used as a framework for defeasible reasoning.

[5] In these papers we were concerned only with weak equivalence, but it is easily shown that strong equivalence holds.

DEFINITION **6.1** A *Penalty Logic WFF* (PLOFF) ψ is a finite set of pairs. Each pair is composed of a real positive number (including ∞), called *penalty*, and a standard propositional WFF, called *assumption* ; i.e., $\psi = \{< \rho_i, \varphi_i > | \ \rho_i \in \mathcal{R}^+ \cup \{\infty\}, \varphi_i \text{ is a WFF}, i = 1...n\}$.
The *violation-rank* of a PLOFF ψ is the function ($Vrank_\psi()$) that assigns a real-valued rank to each of the truth assignments. The $Vrank_\psi$ function is computed by summing the penalties for the assumptions of ψ that are violated by the assignment; i.e., $Vrank_\psi(\vec{x}) = \sum_i \rho_i H_{\neg\varphi_i}(\vec{x})$.[6]
Penalty Logic calculus is the triple $< \mathcal{L}, 2^n, m() >$, where $\mathcal{L}$ is the set of all PLOFFs and $(\forall \psi \in \mathcal{L})$ $m(\psi) = Vrank_\psi$.

We may conclude that a truth assignment $\vec{x}$, satisfies a PLOFF ψ iff it minimizes the violation-rank of ψ to a finite value. i.e., $Vrank_\psi(\vec{x}) = MIN_{Vrank_\psi} < \infty$. A sentence ψ therefore entails φ iff any model that minimizes the violation-rank of ψ, also minimizes the violation-rank of φ. The operator "union" ($\cup$) in the meta-language, plays the role of $\wedge$ (AND) in classic predicate and propositional logic. It allows us to combine two PLOFFs simply by merging them. The reader may check that $Vrank_{\psi \cup \psi'} = Vrank_\psi + Vrank_{\psi'}$ (when $\psi \cap \psi' = \phi$).[7] The evidence in penalty calculus is therefore cumulative; we can combine evidence with background simply by merging the two PLOFFs together.

7. Proof-theory for penalty calculus

Although our inference engine will be based on the model-theoretic definition, a proof procedure still gives us valuable intuition about the reasoning process and about the role of the penalties.

DEFINITION **7.1** T is a *sub-theory* of a PLOFF ψ if T is a consistent subset (in the normal sense) of the assumptions in ψ; i.e., $T \subseteq \{\varphi_i \ | < \rho_i, \varphi_i > \in \psi\} = \mathcal{U}_\psi$, (note that $\mathcal{U}_\psi$ may be inconsistent).
The *penalty* of a sub-theory T of ψ is the sum of the penalties of the assumptions in ψ that are not included in T; i.e., $penalty_\psi(T) = \sum_{\varphi_i \in (\mathcal{U}_\psi - T)} \rho_i$ and is called the *penalty function* of ψ.
A *Minimum Penalty* sub-theory (MP-theory) of ψ is a sub-theory T that minimizes the penalty function of ψ; i.e., $penalty_\psi(T) = MIN_S\{penalty_\psi(S) \mid S \text{ is a maximal consistent sub-theory of } \psi\}$.

[6] We may consider some of the atomic propositions of ψ as "hidden" (free). $Vrank_\psi()$ is then computed by letting these hidden propositions to settle so that the penalty is minimized. i.e., $Vrank_\psi(\vec{x}) = MIN_{\vec{t}}\{\sum \rho_i H_{\neg\varphi}(\vec{x}, \vec{t})\}$.

[7] We need to alter the definition of $\cup$ if we want the property to hold also when $\psi \cap \psi' \neq \phi$. The union $\cup$ should double the penalties of the assumptions participating in $\psi \cap \psi'$:
$\psi_1 \cup \psi_2 = (\psi_1 - \psi_2) \cup (\psi_2 - \psi_1) \cup \{< 2\rho_i, \varphi_i > | \ < \rho_i, \varphi_i > \in \psi_1 \cap \psi_2\}$

The reader may observe that any MP-theory is a maximal consistent -sub-set of $\mathcal{U}$ and therefore a model that satisfies a MP-theory, T, violates all the assumptions not in T (and only these assumptions). We therefore conclude the following lemma:

Lemma 1 *The penalty of a MP-theory, T, is equal to the violation-rank of all the models that satisfy T; i.e., $penalty_\psi(T) = Vrank_\psi(\Omega)$ if $\Omega \models T$.*

This allows us to use a proof-theoretic ranking function ($penalty_\psi$) instead of the model-theoretic function ($Vrank$).

DEFINITION **7.2** Let ψ and φ be two PLOFFs and let $T_1, T_2 \ldots T_k$ and $T'_1, T'_2 \ldots T_{k'}$ their MPMC-theories respectively. We say that $\psi \vdash \varphi$, iff $\bigvee_{i=1}^{k} T_i \vdash \bigvee_{j=1}^{k'} T_j$.

We can device a *Proof theoretic procedure* for $\psi \vdash \varphi$ by first identifying the MPMC-theories of both ψ and φ, and then checking that the disjunction of all the MPMC-theories of φ can be proved (using a standard propositional calculus proof procedure) from any MPMC-theory of ψ. Note that when the set of assumptions of φ is consistent, there is only one MPMC-theory which is $\mathcal{U}_\varphi$ so we have to check only that $T_i \vdash \mathcal{U}_\varphi$ for all MPMC-theories T_i of ψ. Intuitively, we can look at the process as if conflicting sub-theories compete among themselves and those who win are those who have the maximum sum of penalties. T' must follow from all the winning sub-theories.

Theorem 1 *The proof procedure is sound and complete; i.e., $\psi \models \varphi$ iff $\psi \vdash \varphi$.*

This mechanism is useful both for dealing with inconsistency in the knowledge base and for defeasible reasoning. When we detect inconsistency we usually want to adopt a sub-theory with maximum cardinality (we assume that only a minority of the observations are erroneous). When all the penalties are one, minimum penalty means maximum cardinality. Penalty logic is therefore a generalization of the maximal cardinality principle. For defeasible reasoning, the notion of conflicting sub-theories can be used to decide between conflicting sets of arguments. Intuitively, a set of arguments A_1 defeats a conflicting set of arguments A_2 if A_1 is supported by a "better" sub-theory than all those that support A_2 (A_1 wins if it is included in a MP-theory and A_2 is not).

EXAMPLE **7.1** Two levels of blocking (from [Brewka 89]):

1	meeting	I usually go to the Monday meeting.
10	sick $\rightarrow$(¬ meeting)	If I'm sick I usually don't go to the meeting.
100	cold-only $\rightarrow$ meeting	If I have only a cold then I tend go to the meeting.
1000	cold-only $\rightarrow$ sick	If I have a cold it means I'm sick.

Without any additional evidence, all the assumptions are consistent, and we can infer that "meeting" is true (from the first assumption). However, given the evidence that "sick" is true, we prefer models that falsify "meeting" and "cold-only", since the second assumption has greater penalty than the competing first assumption (The only MP-theory, does not include the first assumption). If we include the evidence that "cold-only" is true, we prefer again the models where "meeting" is true, since we prefer to defeat the second assumption rather than the third or the fourth assumptions.

EXAMPLE **7.2** Nixon diamond:

10	$N \rightarrow Q$	Nixon is a quaker.
10	$N \rightarrow R$	Nixon is a republican.
1	$Q \rightarrow P$	Quakers tend to be pacifists.
1	$R \rightarrow \neg P$	Republicans tend to be not pacifists.

When Nixon is given, we reason that he is both republican and quaker. We cannot decide however, whether he is a pacifist or not, since in both possible models (those with minimal $Vrank$) either the third or fourth assumption is violated; i.e., there are two MP-theories: one, does not include the third assumption, while the other, does not include the fourth. The first MP-theory entails $\neg P$ whereas the second entails P. We cannot decide therefore about P even though we can conclude $Q \wedge R$.

8. Penalty logic and energy functions

In this section we show that penalty calculus is strongly equivalent to the calculus of quadratic energy functions. We shall give algorithms to convert a PLOFF into a strongly equivalent quadratic energy function and vice-versa.

We first show that every PLOFF can be reduced into a quadratic energy function.

Theorem 2 *For every PLOFF* $\psi = \{< \rho_i, \varphi_i > | \; i = 1 \ldots n\}$ *there exists a strongly equivalent quadratic energy function* $E(\vec{x}, \vec{t})$ *such that* $(\forall \vec{x}) Vrank_\psi(\vec{x}) = Erank_E(\vec{x})$.[8]

We can construct E from ψ using the following procedure:

[8] This is just an approximation since we translate infinite penalties into sufficiently large but finite weights.

1. "Name" all φ_i's using new hidden atomic propositions T_i[9] and construct the set $\{< \infty, T_i \leftrightarrow \varphi_i >\}$. The high penalty guarantees that these WFFs will always be satisfiable.

2. Construct $\psi' = \{< \infty, T_i \leftrightarrow \varphi_i >\} \cup \{< \rho_i, T_i >\}$ so that the T_i's (with the original penalty) compete with each other (ψ' is strongly equivalent to ψ).

3. Construct the energy function $\sum_i \beta E_{T_i \leftrightarrow \varphi_i} - \sum_j \rho_j T_j$, where β is chosen to be sufficiently large so that it is is greater then the sum of all finite penalties (β is practically ∞), and E_φ is the energy function obtained from φ using the following algorithm from [Pinkas 90c]:

   (a) convert the WFF into a conjunction of sub-formulas, each of at most three variables.[10] This is done by adding some additional hidden atomic propositions;

   (b) assuming the result is of the form $\bigwedge_j \varphi_{i_j}$, the energy function is computed to be $\sum_j H_{\neg\varphi_{i_j}}$;

   (c) convert the cubic terms in the result to quadratic ones using a high-order to low-order procedure [Pinkas 90a].[11]

Note, that 1) if $\rho_i = \infty$ we substitute $T_i = 1$ (clamping it); and 2) if φ_i has three variables or less, there is no need to "name" it; the pair $< \rho_i, \varphi_i >$ can be used directly in ψ'.

EXAMPLE 8.1 Converting the "meeting" example; we first show the general case with "naming" (as if the assumptions were of more then three variables):

Penalty	WFF	$E_{\varphi_i}(\vec{x})$
1000	$T_1 \leftrightarrow$ meeting	$1000(T_1 - 2T_1M + M)$
1000	$T_2 \leftrightarrow$ (sick $\rightarrow$($\neg$ meeting))	$1000(T_2SM - 2T_2 - S - M + T_2S + T_2M)$
1000	$T_3 \leftrightarrow$ (cold-only $\rightarrow$ meeting)	$1000(-T_3 - C + 2T_3C + M - T_3M - T_3CM)$
1000	$T_4 \leftrightarrow$ (cold-only $\rightarrow$ sick)	$1000(-T_4 - C + 2T_4C + S - T_4S - T_4CS)$
1	T_1	$-1T_1$
10	T_2	$-10T_2$
100	T_3	$-100T_3$
1000	T_4	$-1000T_4$

The energy function we get by summing the energy of the assumptions is:

$1000T_2SM - 1000T_3CM - 1000T_4CS - 2000T_1M + 1000T_2S + 1000T_2M + 2000T_3C - 1000T_3M + 2000T_4C - 1000T_4S + 1000M - 2000C + 999T_1 - 2010T_2 - 1100T_3 - 2000T_4$. It is shown as a cubic

[9] Similar to Poole's default naming [Poole 88].

[10] In contrast to the familiar 3-SAT, connectives in a sub-formula are not limited to disjunctions of literals.

[11] This step is not necessary if we allow for high-order connectionist networks [Sejnowski 86].

symmetric network in fig 1-a and as a quadratic network in fig 1-b. Since the assumptions in our example have less then three variables each, we can generate a simpler (strongly equivalent) network from the energy function of $\sum_i \rho_i E_{\neg\varphi_i} = 1(-M) + 100(C - CM) + 1000(C - CS)$ (see fig 1-c).

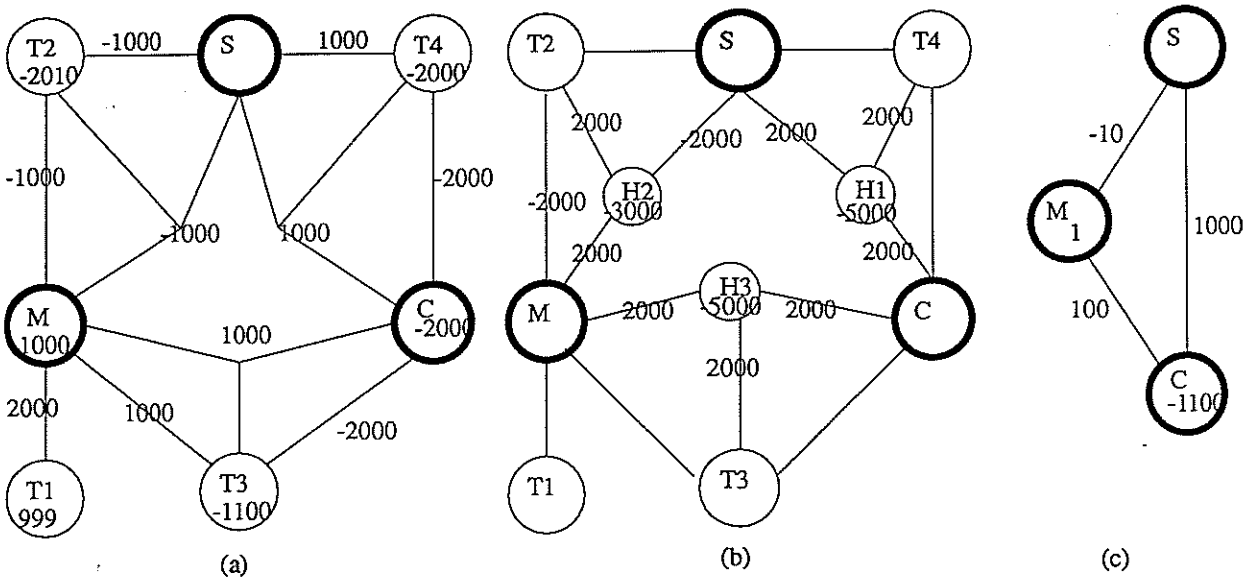

Figure 1: Equivalent symmetric networks for the meeting example (the numbers in the circles are thresholds): a) Cubic; b) Quadratic; and c) quadratic for the simple conversion (no naming).

Theorem 3 *Every energy function E is strongly equivalent to some PLOFF ψ, such that $Erank_E = Vrank_\psi + c$.*

The following algorithm generates a strongly equivalent PLOFF from an energy function:

1. Eliminate hidden variables (if any) from the energy function using the algorithm of [Pinkas 90a].

2. The energy function (with no hidden variables) is now brought into a sum-of-products form and is converted into a PLOFF in the following way:

   Let $E(\vec{x}) = \sum_{i=1}^{m} w_i \prod_{n=1}^{k_i} x_{i_n}$ be the energy function.

   We construct a PLOFF $\psi = \{< -w_i, \bigwedge_{n=1}^{k_i} x_{i_n} > | \; w_i < 0\} \cup \{< w_l, \neg \bigwedge_{n=1}^{k_l} x_{l_n} > | \; w_l > 0\}$.

$Vrank_\psi$ is equal to the sum of the positive weights when $\bigwedge x_{l_n}$ is true ($\prod X_{l_n} = 1$) minus the sum of the negative weights when $\neg \bigwedge X_{i_n}$ is true ($\prod X_{i_n} = 0$). On the other hand $E(\vec{x})$ is:

$$E(\vec{x}) = \sum_{w_l>0 \wedge \prod x_{l_n}=1} w_l + \sum_{w_i<0 \wedge \prod x_{i_n}=1} w_i = \sum_{w_l>0 \wedge \bigwedge x_{l_n}} w_l + \sum_{w_i<0} w_i - \sum_{w_i<0 \wedge \neg \bigwedge x_{i_n}} w_i$$

Therefore, $E(\vec{x}) = Vrank_\psi(\vec{x}) + \sum_{w_i<0} w_i$. The size of the PLOFF that is generated is of the same order as the size of the original function (in its sum-of-products form or in its network form).

COROLLARY 8.1 *Penalty calculus is strongly equivalent to the calculus of quadratic energy functions.*

As a result, penalty logic can be used as a language to describe symmetric neural networks; it is expressive enough to represent in a compact way every such network, and for every sentence in this language we can generate a compact symmetric neural network that represents the same WRF.

9. A sketch of a connectionist inference engine

Suppose a background PLOFF ψ, an evidence PLOFF e, and a standard logic WFF φ. We would like to construct a connectionist network to answer one of the possible three answers: 1) $\psi \cup e \models \varphi$; 2) $\psi \cup e \models (\neg\varphi)$; or 3) both $\psi \not\models \varphi$ and $\psi \not\models (\neg\varphi)$ ("unknown"). For simplicity let us first assume that the evidence e is a strict conjunction of literals (atomic propositions or their negation) and that φ is an atomic proposition. Later we'll describe a general solution.

Intuitively, our connectionist engine is built out of two sub-networks, each that is trying to find a satisfying model for $\psi \cup e$. The first sub-network is biased to search for a model which satisfies also φ, whereas the second sub-network is biased to search for a model which satisfies $\neg\varphi$. If two such models exist then we conclude that φ is "unknown" ($\psi \cup e$ entails neither φ nor $\neg\varphi$). If no model also satisfies φ, we conclude that $\psi \cup e \models \neg\varphi$, and if no model also satisfies $\neg\varphi$, we conclude that $\psi \cup e \models \varphi$.

To implement this intuition we first need to duplicate our background knowledge ψ and create its copy ψ' by naming all the atomic propositions A using A'. For each atomic proposition Q that might participate in a query, we then add two more propositions: "$QUERY_Q$" and "$UNKNOWN_Q$". $QUERY_Q$ is used to initiate a query Q: it will be externally clamped by the user, when he or she

inquires about Q. $UNKNOWN_A$ represents the answer of the system. It will be set to TRUE if we can conclude neither that ψ entails φ nor that ψ entails φ.

Our inference engine can be therefore described (using the high-level language of penalty logic) by:

ψ	searches for a model that satisfies also Q
$\cup\psi'$	searches for a model that satisfies also $\neg Q$
$\cup\{< \epsilon, (QUERY_Q \rightarrow Q) >\}$	bias ψ to search for a model that satisfies Q
$\cup\{< \epsilon, (QUERY_Q \rightarrow (\neg Q')) >\}$	bias ψ' to search for a model that satisfies $(\neg Q')$
$\cup\{< \epsilon, (Q \wedge \neg Q') \rightarrow UNKNOWN_Q >\}$	if two satisfying models exist that do not agree on Q, we conclude "UNKNOWN"
$\cup\{< \epsilon, (Q \leftrightarrow Q') \rightarrow (\neg UNKNOWN_Q) >\}$	if despite the bias we are unable to find two such satisfying models we conclude "$UNKNOWN_Q$"

Using the algorithm of Theorem 2, we generate the corresponding network.

To initiate a query about propositional Q the user externally clamps the unit $QUERY_Q$. This causes a small positive bias ϵ to be sent to unit Q and a small negative bias $-\epsilon$ to be sent to Q'. Each of the two sub-networks ψ and ψ', searches for a global minimum (a satisfying model) of the original PLOFF. The bias (ϵ) is small enough so it does not introduce new global minima. It may however, constrain the set of global minima: if a satisfying model that also satisfies the bias exists then such model is in the new set of global minima.

The network therefore tries to find models that satisfy also the bias rules. If it succeeds, we conclude "UNKNOWN", otherwise we conclude that all the satisfying models agree on the same truth value for the query. The "UNKNOWN" proposition is then set to "false" and the answer whether $\psi \models \varphi$ or whether $\psi \models \neg\varphi$ can be found in the proposition Q. If Q is "true" then the answer is $\psi \models \varphi$ since Q holds in all satisfying models. Similarly, if Q is false, we conclude that $\psi \models \varphi$

When the evidence is a strict conjunction of literals we can add it to the background network simply by clamping the appropriate atomic propositions. In the general case we need to combine an arbitrary evidence e and query an arbitrary WFF φ: We do this by building an inference network for $\psi \cup e \cup \{< \infty, Q \leftrightarrow \varphi >\}$ and querying Q, the new atomic proposition.

The network that was generated converges to the correct answer if it manages to find a global minimum. An annealing schedule like in [Hinton, Sejnowski 86] may be used for such search. A slow enough annealing is certain to find a global minimum and therefore the correct answer, but it might take exponential time. Since the problem is NP-hard, we will probably not find an algorithm that will give

us always the correct answer in polynomial time. Traditionally in AI, knowledge representation systems traded the expressiveness of the language they use with the time complexity they allow [Levesque 84].[12] The accuracy of the answer is usually not sacrificed. In our system, like in [Derthick 88] we trade the time with the accuracy of the answer. We are given limited time resources and we stop the search when this limit is reached. The annealing schedule can be planned to fit the time limitation and an answer is given at the end of the process. Although the answer may be incorrect, the system is able to improve its guess as more time resources are given.

10. Related work

Derthick [Derthick 88] was the first to observe that weighted logical constraints (which he called "certainties") can be used for non-monotonic connectionist reasoning. We follow his direction and there are many similarities and differences that will be discussed in the longer version of this paper. There are however, two basic differences: 1) Derthick's "Mundane" reasoning is based on finding the most likely model that satisfies a WFF; his system is never skeptical; 2) Our system can be implemented using standard low-order units, and we can use models like Hopfield nets or Boltzman machines that were relatively well studied (e.g., a learning algorithm exists).

[Shastri 85] is a connectionist non-monotonic system for inheritance nets that uses evidential reasoning based on maximum likelihood. Our approach is different; we use low-level units and we are not restricted to inheritance networks.[13] Shastri's system is guaranteed to work, whereas we trade the correctness with the time.

Our WRFs have a lot in common with Lehmann's ranked models [Lehmann 89]. His result about the relationship between rational consequence relations and ranked models can be applied to our paradigm; yielding a rather strong conclusion: for every conditional knowledge base we can build a ranked model (for the rational closure of the knowledge base) and implement it as a WRF using a symmetric neural net. Also, any symmetric neural net is implementing some ranked model and therefore induces a rational consequence relation.

[12] Connectionist systems like [Shastri,Ajjanagadde 90] and [Hölldobler 90] trade expressiveness with time complexity.

[13] We can easily extend our approach to handle inheritance nets, by looking at the atomic propositions as predicates with free variables. Those variables are bound by the user during query time.

Our penalty logic has some similarities with systems that are based on the user specifying priorities to defaults (like [Brewka 89], [Lifschitz 85]). The closest is Brewka's system that is based on levels of reliability. In fact Brewka's system for propositional logic can be mapped to penalty logic by selecting large enough penalties. Systems like [Poole 88] (with strict specificity) can be implemented using our architecture, and the penalties can therefore be generated automatically from knowledge bases described in simple conditional languages (that do not force the user to state explicitly numbers or priorities to the assumptions). There are however several differences: 1) Our language is more expressive and we can implicitly state a preference relation between sets of assumptions. 2) Brewka is concerned with maximal consistent sets in the sense of set inclusion, while we are interested in sub-theories with maximum cardinality (generalized definition). As a result we prefer theories with "more" evidence. For example consider the Nixon diamond of example 7.2 when we add $< 10, N \rightarrow FF >$ and $< 1, FF \rightarrow \neg P >$ (Nixon is also a football fan and football fans tend to be not pacifists). Most other NM systems [Touretzky 86], [Simari,Loui 90], [Geffner 89] will still not decide whether P is true or false. Our system decides that $\neg P$ holds (based on more evidence[14]) since it is better to defeat the one assumption that quakers tend to be pacifists, than the two assumptions supporting $\neg P$. We can correct this behavior by multiplying the penalty for $Q \rightarrow P$ by two. Further, a network with learning capabilities can adjust the penalties autonomously and thus develop its own intuition.

Because we do not allow for arbitrary partial orders ([Shoham 88] [Geffner 89],[Lehmann 89]) of the models, there are other fundamental problematic cases where our system concludes the truth (or falsity) of a proposition while other systems are skeptical. For the following example[15] we have a clear intuition of what should be the preference relation; however, there is no WRF that induces such order. E.g. $A \supset B$(strict implication), $B \rightarrow C, A \rightarrow \neg C, D \rightarrow C, E \rightarrow \neg C$. Given $A \wedge D$ as evidence, C should be "unknown" and therefore $f(A, B, C, D, \neg E) = f(A, B, \neg C, D, E)$. When given $E \wedge D \wedge B$, C is "unknown" and therefore $f(A, B, \neg C, D, E) = f(\neg A, B, C, D, E)$. Given $B \wedge E$, C is "unknown" and therefore $f(\neg A, B, C, D, E) = f(A, B, \neg C, \neg D, E)$. However, given A as evidence we prefer to conclude $\neg C$ since $A \rightarrow \neg C$ is more specific then $B \rightarrow C$, and therefore $f(A, B, \neg C, \neg D, E) < f(A, B, C, D, \neg E)$. Of course no such function f exists (otherwise we had: $f(A, B, C, D, \neg E) < f(A, B, C, D, \neg E)$).

[14] and not because it is inherently ambiguity-blocking.
[15] Thanks to Hector Geffner for suggesting this example.

11. Conclusions

We have developed a notion of reasoning using world-rank-functions independently of the use of symbolic languages. This concept suits well the connectionist sub-symbolic approach. We showed that any symmetric neural network can be viewed as if it is searching for a satisfying model of such a function, and every such function can be approximated using these networks.

Several equivalent high-level languages can be used to describe symmetric neural networks: 1) quadratic energy functions [Hopfield 82]; 2) high-order energy functions with no hidden units [Pinkas 90a]; 3) propositional logic (only weakly equivalent), and finally 4) penalty logic. All these languages are expressive enough to describe any symmetric network and every sentence of such languages can be translated into a symmetric network. We gave algorithms that perform these transformations, which are magnitude preserving (except for propositional calculus).

We have developed a calculus based on assumptions augmented by penalties that fits very naturally the symmetric models' paradigm. This calculus can be used as a platform for defeasible reasoning and inconsistency handling. Some NM systems can be mapped to this paradigm and therefore suggest settings of the penalties. When the right penalties are given (for example using algorithms like in [Brewka 89] that are based on specificity), penalty calculus features a non-monotonic behavior that (usually) matches our intuition. Penalties do not necessarily have to come from a syntactic analysis of a symbolic language; since those networks can learn, it is theoretically possible for such networks to adjust their WRFs and develop their own intuition.[16] It is possible to show, though, that some intuitions cannot be expressed in world-rank-functions.

Revision of the knowledge base and adding new evidence are easy tasks if we use penalty logic to describe the network: adding (or deleting) a PLOFF is simply computing the energy function of the new PLOFF and then adding (deleting) it to the background energy function. A local change to the PLOFF describing the network is translated to a local change in the network.

We sketched a connectionist inference engine for penalty calculus. When a query is clamped, the global minima of such network correspond exactly to the correct answer. Using massively parallel

[16] By intuition I mean: the set of conclusions drawn from a piece of knowledge. This set is dependent only on the WRF. By adjusting the WRF a network can therefore develop its own intuition.

hardware convergence should be very fast, although the worse case for *correct* answer is still exponential.

The mechanism however trades the soundness of the answer with the time given to solve the problem.

References

[Brewka 89] G. Brewka, Preferred sub-theories: An extended logical framework for default reasoning., *IJCAI* 1989, pp. 1043-1048.

[Derthick 88] M. Derthick, "Mundane reasoning by parallel constraint satisfaction", PhD Thesis, TR. CMU-CS-88-182, Carnegie Mellon 1988.

[Geffner 89] H. Geffner, "Defeasible reasoning: causal and conditional theories", PhD Thesis, TR UCLA, 1989.

[Hinton, Sejnowski 86] G.E Hinton and T.J. Sejnowski "Learning and Re-learning in Boltzman Machines" in J. L. McClelland, D. E. Rumelhart *"Parallel Distributed Processing:Explorations in the Microstructure of Cognition"* Vol I pp. 282 - 317 MIT Press 1986

[Hölldobler 90] S. Hölldobler, "CHCL, a connectionist inference system for horn logic based on connection method and using limited resources", *International Computer Science Institute* TR-90-042, 1990.

[Hopfield 82] J.J. Hopfield "Neural networks and physical system with emergent collective computational abilities," *Proceedings of the National Academy of Sciences* USA, 1982,79,2554-2558.

[Hopfield 84] J.J. Hopfield "Neurons with graded response have collective computational properties like those of two-state neurons". *Proceedings of the National Academy of Sciences* USA, 1984,Vol 81, pp. 3088-3092.

[Lehmann 89] D. Lehmann, "What does a conditional knowledge base entail?", KR-89, *Proc. of the international conf. on knowledge representation,* 1989.

[Levesque 84] H.J Levesque, "A fundamental tradeoff in knowledge representation and reasoning." *Proc. CSCSI-84,* London, Ontario, pp 141-152, 1984.

[Lifschitz 85] V. Lifschitz, Computing circumscription. *Proc. IJCAI-85,* 1985.

[Pinkas 90a] G. Pinkas, "The Equivalence of Connectionist Energy Minimization and Propositional Calculus Satisfiability " *Technical Report: WUCS-90-03, Department of Computer Science, washington University* 1990.

[Pinkas 90b] G. Pinkas, " Symmetric neural networks and propositional logic satisfiability", to appear in Neural Computation Vol 3-2, 1991.

[Pinkas 90c] G. Pinkas, "Energy minimization and the satisfiability of propositional calculus", In Touretzky,D.S., Elman, J.L, Sejnowski, T.J., and Hinton, G.E. (eds), *Proceedings of the 1990 Connectionist Models Summer School.* San Mateo, CA: Morgan Kaufmann.

[Poole 88] D. Poole , "A logical framework for default reasoning", *Artificial Intelligence* 36,1988.

[Sejnowski 86] T.J. Sejnowski, "Higher-order Boltzman machines", Neural Networks for Computing", *Proc. of the American Institute of Physics", 151,* Snowbird (Utah) pp 3984, 1986.

[Shastri 85] L. Shastri, "Evidential reasoning in semantic networks:A formal theory and its parallel implementation", *PhD thesis, TR 166, University of Rochester,* Sept. 1985.

[Shastri,Ajjanagadde 90] L. Shastri, V. Ajjanagadde, "From simple associations to systematic reasoning: a connectionist representation of rules, variables and dynamic bindings" TR. MS-CIS-90-05 *University of Pennsylvania, Philadelphia,* 1990.

[Shoham 88] Y. Shoham, "Reasoning about change" *The MIT press*, Cambridge, Massachusetts, London, England 1988.

[Simari,Loui 90] G. Simari, R.P. Loui , "Mathematics of defeasible reasoning and its implementation", to appear in *AI Journal.*

[Touretzky 86] D.S. Touretzky,"The mathematics of inheritance systems", Pitman, London, 1986.