# Propositional Non-Monotonic Reasoning and Inconsistency in Symmetric Neural Networks *

Gadi Pinkas
Department of Computer Science,
Washington University,
St. Louis, MO 63130, U.S.A.

## Abstract

We define a model-theoretic reasoning formalism that is naturally implemented on symmetric neural networks (like Hopfield networks or Boltzman machines). We show that ev~ery symmetric neural network, can be seen as performing a search for a satisfying model of some knowledge that is wired into the network's weights. Several equivalent languages are then shown to describe the knowledge embedded in these networks. Among them is propositional calculus extended by augmenting propositional assumptions with penalties. The extended calculus 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 into such a network, A sound and complete proof 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 that implements this reasoning paradigm.

## 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], [Hinton, Sejnowski 86]),

We shall demonstrate that symmetric neural networks (SNNs) 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 SNNs, and second, to build a connectionist inference engine capable of reasoning from incomplete and inconsistent knowledge- Proofs and detailed constructions are omitted and will appear in the extended version of the article.

## 2 Reasoning with World Rank Functions

We begin by giving a model-theoretic definition for an abstract reasoning formalism independently of any symbolic language. Later we shall use it to give semantics for the knowledge embedded in SNNs, and for the reasoning mechanism that will be defined.

**DEFINITION 2.1** A ***World Rank Function*** (WRF) with respect to a set of possible worlds (models) $M$ is a function $k : M \rightarrow \mathcal{R}$ that ranks each of the possible worlds with a number that is in $(-\infty \ldots \infty]$.[1] A WRF is ***propositional*** iff it is defined over the set of truth assignments (i.e., $dom(k) = \{0,1\}^n$).

**DEFINITION 2.2** A model $\Omega \in M$ ***satisfies*** a WRF $k$ iff it minimizes the function to a value that is less than $\infty$. Let $\Gamma_k$ 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., $\Gamma_k \subseteq \Gamma_{k'}$.

## 3 Connectionist energy functions

### 3.1 Symmetric connectionist models

Connectionist networks with symmetric weights (SNNs) use gradient descent to find a minimum for quadratic energy functions, A i-order energy function is a function

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

[1]he symbol oo denotes a real positive number that is larger than any other number mentioned explicitly in a formula (practically infinity.

$E : \{0,1\}^n \rightarrow \mathcal{R}$ 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:

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

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

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

We can arbitrarily divide the variables of an energy function into two sets: visible variables and hidden variables. 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 quadratic energy functions and SNNs that minimize them. Variables map into neuron units (nodes), and weighted terms map into weighted connections (arcs). Hyper-graphs with Sigma-Pi units can be used for minimizing high-order energy functions.

### 3.2 WRFs and energy functions

A SNN can be viewed as an implementation of a propositional 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* associated with an energy function $E(\vec{x},\vec{t})$ is the function $rank_E(\vec{x}) = MIN_{\vec{t}}\{E(\vec{x},\vec{t})\}$.

A search performed by the SNN for a global minimum may therefore be interpreted as a search for a satisfying model of the WRF $rank_E$. We will interchangeably use energy functions, rank functions and graph descriptions to represent the functional behavior of SNNs.

## 4 Symbolic languages to describe WRFs

Our next step is to describe symbolically the knowledge that is encapsulated in a network. 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 $\Omega$ *satisfies* $s$ ($\Omega \models s$) iff it satisfies $m(s)$. Similarly, a sentence $s$ *entails* sentence $s'$ ($s \models s'$) iff the WRF $m(s)$ entails the WRF $m(s')$. A conjunction of a background sentence ($k$) with an evidence sentence ($e$) [2] is interpreted as the addition of their corresponding WRFs; i.e., $k, e \models s$ iff $(m(e) + m(k)) \models m(s)$.

[2] NM systems "jump" to conclusions based on evidence given, and later may retract those conclusions based on new evidence. It is convenient therefore to divide the knowledge from which we reason, to background knowledge and evidence (see [Geffner 89].

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

EXAMPLE **4.1** Propositional calculus is a triple $< \mathcal{L}, m(), \{0,1\}^n >$, where $\mathcal{L}$ is the language of propositional well formed formulae (WFFs) and $m()$ outputs the function $(\infty \times (1 - H_s))$, when given a WFF $s$. $H_s(\vec{x})$ is the characteristic function of the WFFs and is recursively defined as:
$H_{X_i}(\vec{x}) = X_i$
$H_{\neg s}(\vec{x}) = 1 - H_s(\vec{x})$
$H_{s_1 \vee s_2}(\vec{x}) = H_{s_1}(\vec{x}) + H_{s_2}(\vec{x}) - H_{s_1}(\vec{x}) \times H_{s_2}(\vec{x})$

The reader can easily observe that any propositional WFF describes a WRF that returns 0 for truth assignments that satisfy the WFF and $\infty$ 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 $< \mathcal{L}_1, m, M >$ and $< \mathcal{L}_2, m', M >$; 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 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. [3]

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 $k \overset{s}{\approx} k'$ and $e$ is any evidence, then for every WRF $f$, $(m(k) + e) \models f$ iff $(m'(k') + e) \models f$.* [4]

*2. 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 evidence to our transformed knowledge, we need to perform "magnitude preserving" transformations (strong equivalence). The transformations we use in the reminder of this paper are all "magnitude preserving".
We define now an equivalence between two calculi.

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

[3] A third equivalence, one that preserves the order of the worlds, is also possible, but is beyond the scope of this article.

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

We thus can use the language *C* to represent every WRF that is represent able using the language 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 there-' fore < *{E},rn(),{0,1}^n* >, where *{E}* is the set of all strings representing energy functions written as sum-of-products, and *rn{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.

Using the algorithms given in [Pinkas 90] we can 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.

In [Pinkas 90] we also 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) Although the WFF and the energy function have the same set of satisfying models, evidence can not be added and the a probabilistic interpretation is not preserved.

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

## 6 Penalty calculus

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

DEFINITION 6.1 A *Penalty Logic WFF* (PLOFF) $\psi$ is a finite set of pairs. Each pair is composed of a real positive number (including $\infty$), called *penalty*, and a standard propositional WFF, called an *assumption* ; i.e., $\psi = \{< \rho_i, \varphi_i > | \ \rho_i \in \mathcal{R}^+, \varphi_i \text{ is a WFF}, i = 1...n\}$.
The *violation-rank* of a PLOFF $\psi$ ($Vrank_\psi(\vec{x})$) assigns a real-valued rank to each of the truth assignments. It is computed by summing the penalties for the assumptions of $\psi$ that are violated by the assignment; i.e., $Vrank_\psi(\vec{x}) = \sum_i \rho_i H_{\neg\varphi_i}(\vec{x})$. *Penalty calculus* is the triple $< \mathcal{L}, m(), \{0,1\}^n >$, where $\mathcal{L}$ is the set of all PLOFFs and $m(\psi) = Vrank_\psi$.

We may conclude that a truth assignment $\vec{x}$, satisfies a PLOFF $\psi$ iff it minimizes the violation-rank of $\psi$ to a finite value (we call such models, "preferred models"). A sentence $\psi$ therefore semantically entails $\varphi$ iff any preferred model of $\psi$ is also a preferred model of $\varphi$.

## 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 $\psi$ if $T$ is a consistent subset of the assumptions in $\psi$; 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 $\psi$ is the sum of the penalties of the assumptions in $\psi$ that are not included in $T$; The *penalty function* of $\psi$ is: $penalty_\psi(T) = \sum_{\varphi_i \in (\mathcal{U}_\psi - T)} \rho_i$.
A *Minimum Penalty* sub-theory (MP-theory) of $\psi$ is a sub-theory $T$ that minimizes the penalty of $\psi$; i.e., $penalty_\psi(T) = MIN_S\{penalty_\psi(S) \mid S \text{ is a sub-theory of } \psi\}$.

DEFINITION 7.2 Let $T_\psi = \{T_i\}$ the set of all MP-theories of $\psi$, and let $T_\varphi = \{T'_j\}$ the set of all MP-theories of $\varphi$. We say that $\psi$ entails $\varphi$ ($\psi \vdash \varphi$) iff all the MP-theories of $\psi$ entails (in the classic sense) the disjunction of all the MP-theories of $\varphi$; i.e, $\psi \vdash \varphi$ iff $\bigvee T_i \vdash \bigvee T'_j$ (when $\varphi$ is consistent then $\psi \vdash \varphi$ iff all MP-theories of $\psi$ entail $\varphi$).

Intuitively, conflicting sub-theories compete among themselves and those who win are the preferred sub-theories (with the minimum sum of penalties). $\varphi$ must follow from all the preferred (winning) sub-theories.

THEOREM 7.1 *The proof procedure is sound and complete; i.e., $\psi \models \varphi$ iff $\psi \vdash \varphi$.*

The theorem follows from the observation that the penalty of a maximal consistent subset $T \subseteq \mathcal{U}_\psi$ is equal to the violation rank ($Vrank$) of the models that satisfy $T$.

This entailment mechanism is useful both for dealing with inconsistency in the knowledge base and for defeat sible reasoning. For example, in a noisy knowledge base, 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 arguments. Intuitively, an argument $A_1$ defeats a conflicting argument $A_2$ if $A_1$ is supported by a "better" sub-theory than all those that support $A_2$.

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

| | | |
|---|---|---|
| 1 | meeting | I tend to go to the meeting. |
| 10 | sick $\rightarrow$(¬ meeting) | If sick, I don't go, |
| 100 | cold-only $\rightarrow$ meeting | If only a cold, I still go. |
| 1000 | cold-only $\rightarrow$ sick | If I've 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 (skeptical reasoning):

1 0 $N \rightarrow Q$ o n i s a quaker.
1 0 $N \rightarrow R$ o n i s a republican.
1 Q $Q \rightarrow P$ e r s 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 pacifist or not, since in both preferred models (those with minimal *Vrank)* either the third or fourth assumption is violated; i.e., there are two MP-theories: one that entails *-P,* whereas the other entails P.

## 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 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 8.1 *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* $Vrank_\psi = Erank_E$.

We construct $E$ from $\psi$ using the following procedure:

1. "Name" all $\varphi_i$'s using new hidden atomic propositions $T_i$ and construct $\psi' = \{< \infty, T_i \leftrightarrow \varphi_i >\} \cup \{< \rho_i, T_i >\}$. The high penalty guarantees that the "naming" will always be satisfied, while the $T_i$'s (with the original penalty) compete with each other.
2. Construct the energy function $\sum_i \infty E_{T_i \leftrightarrow \varphi_i} - \sum_j \rho_j T_j$, where $E_\varphi$ is the energy function that describes $\varphi$ (using the algorithm from [Pinkas 90]).

EXAMPLE 8.1 Converting the "meeting" example:

<1000, $T_1 \leftrightarrow$ meeting>,
<1000, $T_2 \leftrightarrow$ (sick $\rightarrow$($\neg$ meeting))>,
<1000, $T_3 \leftrightarrow$ ( cold-only $\rightarrow$ meeting)>,
<1000, $T_4 \leftrightarrow$ (cold-only $\rightarrow$ sick)>,
<1, $T_1$ >, <10, $T_2$ >, <100, $T_3$ >, <1000, $T_4$ >

The energy function we get by summing the energy terms of the assumptions is: $1000(T_1 - 2T_1M + M) + 1000(T_2SM - 2T_2 - S - M + T_2S + T_2M) + 1000(-T_3 - C + 2T_3C + M - T_3M - T_3CM) + 1000(-T_4 - C + 2T_4C + S - T_4S - T_4CS) - 1T_1 - 10T_2 - 100T_3 - 1000T_4$. The corresponding network appears in fig 1.

THEOREM 8.2 *Every energy function* $E$ *is strongly equivalent to some PLOFF* $\psi$, *such that* $rank_E = Vrank_\psi + c$.

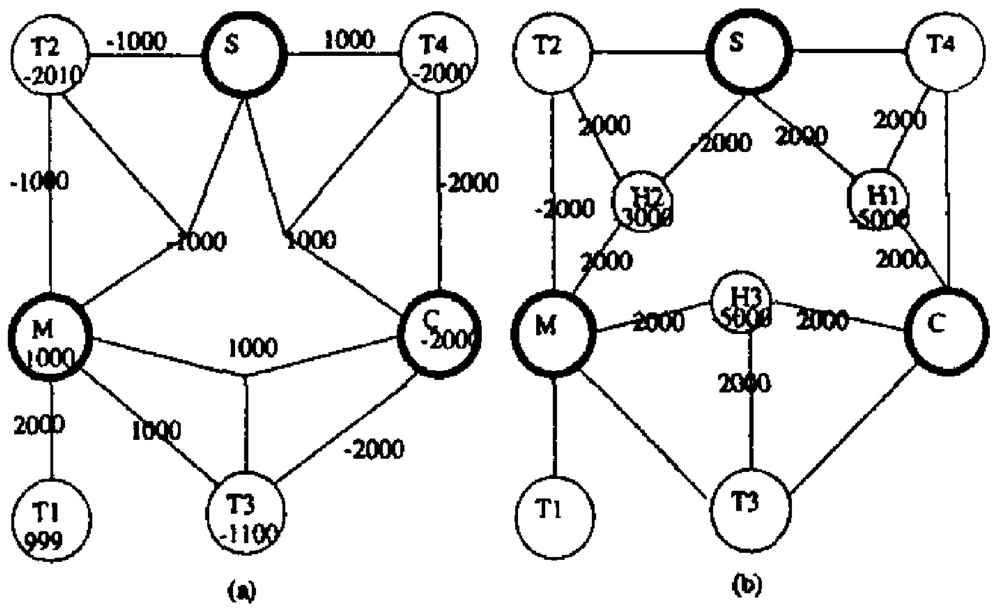


**Figure 1: Equivalent SNNs (the meeting example). The numbers in the circles are thresholds. a) Cubic network; b) Quadratic network after adding hidden units.**

The following algorithm generates a strongly equivalent PLOFF $\psi$ from an energy function $E$:

1. Eliminate hidden variables from the energy function using the algorithm of [Pinkas 90].
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 $\psi = \{< -w_i, \bigwedge_{n=1}^{k_i} x_{i_n} > | \ w_i < 0\} \cup \{< w_l, \neg \bigwedge_{n=1}^{k_l} x_{l_n} > | \ w_l > 0\}$.

The size of the generated PLOFF is of the same order as the size of the original function.

Penalty logic is therefore strongly equivalent to the calculus of quadratic energy functions and can be used as a language to describe SNNs. It is expressive enough to represent in a compact way every such network, and for every sentence in this language we can generate a SNN that represents the same WRF. The transformations are efficient and generate a linear size output.

## 9 A sketch of a connectionist inference engine

Suppose a background PLOFF $\psi$, an evidence PLOFF $e$, and a WFF $\varphi$. We would like to construct a 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 monomial ( a conjunction of literals) and that $\varphi$ is a single literal. 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* $\varphi$, while the second is biased to search for a model which satisfies *also* $\neg\varphi$. If two such preferred models exist then we conclude that $\varphi$ is "unknown" ($\psi \cup e$ entails neither $\varphi$ nor $\neg\varphi$). If no preferred model of $\psi \cup e$ satisfies $\varphi$, we conclude that

$\psi \cup e \models \neg\varphi$, and if no preferred model satisfies $\neg\varphi$, we conclude that $\psi \cup e \models \varphi$.

To implement this intuition we first need to duplicate our background knowledge $\psi$ and create a copy $\psi'$ by naming all the atomic propositions $A$ using $A'$. For each 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 inquiring about $Q$. $UNKNOWN_A$ represents the answer of the system. It will be set to TRUE if we can conclude neither that $\psi$ entails $\varphi$ nor that $\psi$ entails $\varphi$.

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

$\psi$ /* 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 $Q$ */
$\cup\{< \epsilon, (QUERY_Q \rightarrow (\neg Q')) >\}$ /* bias $(\neg Q')$ */
$\cup\{< \epsilon, (Q \wedge \neg Q') \rightarrow UNKNOWN_Q >\}$
$\cup\{< \epsilon, (Q \leftrightarrow Q') \rightarrow (\neg UNKNOWN_Q) >\}$

Using the algorithm of Theorem 8.1, we generate the corresponding energy function and network.

To initiate a query about propositional Q the user externally clamps the unit *QUERYQ.* This causes a small positive bias E to be sent to unit Q and a negative bias —*i* to be sent to *Q'.* Each of the two sub-networks w and $\psi'$, searches for a global minimum (a satisfying model) of the original PLOFF. The bias (e) 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 it is in the new set of global minima. The network tries to find preferred models that satisfy also the bias rules. If it succeeds $(Q \wedge \neg Q')$, we conclude "UNKNOWN", otherwise we conclude that all the satisfying models agree on the same truth value for the query. The "UNKNOWN" unit 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.*

When the evidence is a monomial, 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 an arbitrary WFF <*p* as a query. We do this by adding to $rp_j$ the energy terms that correspond to $e \cup \{< \infty, Q \leftrightarrow \varphi >\}$ and querying *Q.*

The network that is generated converges to the correct answer if it manages to find a global minimum. An annealing schedule as in [Hinton, Sejnowski 86] may be used for such search. A slow enough annealing will find a global minimum and therefore the correct answer, but it might take exponential time. Since the problem is NP-hard, we shall probably not find an algorithm that will always give us 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.[5] The accuracy of the answer is usually not sacrificed. In our system, 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. Although the answer may be incorrect, the system is able to improve its guess as more time resources are given.

Connectionist systems like [Shastri,Ajjanagadde 90] trade expressiveness with time complexity, while systems like [Holldobler 90] trade time with size.

## 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. There are however, two basic differences: 1) Derthick's "Mundane" reasoning is based on finding a most likely single model; his system is never skeptical. Our system is more cautious and closer in its behavior to recent symbolic NM systems. 2) Our system can be implemented using standard low-order units, and we can use models like Hopfield nets or Boltzman machines that are relatively well studied (e.g,, a learning algorithm exists).

Another connectionist non-monotonic system is [Shastri 85]. It uses evidential reasoning based on maximum likelihood to reason in inheritance networks. Our approach is different; we use low-level units and we arc not restricted to inheritance networks.[6] 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 rational consequence relation.

Our penalty logic has some similarities with systems that are based on the user specifying priorities to defaults. The closest system is [Brewka 89] that is based on levels of reliability. 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 conditional languages that do not force the user to associate explicitly numbers or priorities to the assumptions. Brewka however 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 (like [Touretzky 86], [Geffner 89],[Simari,Loui 90]) will still be skeptical about $P$. Our system decides $\neg P$ since it is better to defeat the one assumption sup-

[6] 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.

porting P, than the two assumptions supporting -P. We can correct this behavior however, 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 and non-monotonic behavior.

Because we do not allow for arbitrary partial orders ([Shoham 88] [Geffner 89]) of the models, there are other fundamental problematic examples where our system (and all systems with ranked models semantics) concludes the truth (or falsity) of a proposition while other systems are skeptical- Such examples are beyond the scope of this article. On the positive side, every skeptical reasoning mechanism with ranked models semantics can be mapped to our paradigm,

## 11 Conclusions

We have developed a model theoretic notion of reasoning using world-rank-functions independently of the use of symbolic languages. We showed that any SNN 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 SNNs: 1) quadratic energy functions; 2) high-order energy functions with no hidden units; 3) propositional logic, and finally 4) penalty logic. All these languages are expressive enough to describe any SNN and every sentence of such languages can be translated into a SNN. We gave algorithms that perform these transformations, which are magnitude preserving (except for propositional calculus which is only weakly equivalent).

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. Several recent NM systems can be mapped into this paradigm and therefore suggest settings of the penalties. When the right penalties are given, penalty calculus features a non-monotonic behavior that matches our intuition. Penalties do not necessarily have to come from a syntactic analysis of a symbolic language; since those networks can learn, they can potentially adjust their WRFs and develop their own intuition.

Revision of the knowledge base and adding evidence are efficient if we use penalty logic to describe the knowledge: adding (or deleting) a PLOFF is simply computing the energy terms of the new PLOFF and then adding (deleting) it to the background energy function. A local change to the PLOFF is translated into 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. Although the worst case for the *correct* answer is still exponential, the mechanism however, trades the soundness of the answer with the time given to solve the problem.

Acknowledgment Thanks to John Doyle, Hector Geffner, Sally Goldman, Dan Kimura, Stan Kwasny, Fritz Lehmann and Ron Loui for helpful discussions and comments.

## References

[Brewka 89] G. Brewka, "Preferred sub-theories: An extended logical framework for default reasoning.", *IJ-*<7>i/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, UCLA, 1989.

[Hinton, Sejnowski 86*]* G.E Hinton and T.J. Sejnowski "Learning and Re-learning in Boltzman Machines" in McClelland, Rumelhart, "*Parallel Distributed Processing"* , Vol I MIT Press 1986

[Holldobler 90] S. Holldobler, "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," *Proc. of the Nat Acad, of Sciences* , 79,1982.

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

[Pinkas 90] G. Pinkas, "Energy minimization and the satisfiability of propositional calculus", *Neural Computation* Vol 3-2, 1991.

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

[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", *Artificial Intelligence,* to appear.

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