Report Number: WUCS-91-02

1991-01-01

Symmetric Neural Nets and Propositional Logic Satisfiability

Gadi Pinkas

Connectionist networks with symmetric weights (like Hopfield networks and Boltman Machines) use gradient descent to find a minimum for quadratic energy functions. We show an equivalence between the problem of satisfiability in propositional calculus and the problem of minimizing those energy functions. The equivalence is in the sense that for an satisfiable Well Formed Formula (WFF) we can find a quadratic function that describes it, such that the set of solutions that minimize the function is equal to the set of truth assignments that satisfy the WFF. We also show that in the same sense every quadratic energy function... Read complete abstract on page 2.

Symmetric Neural Nets and Propositional Logic Satisfiability

Gadi Pinkas

Complete Abstract:

Connectionist networks with symmetric weights (like Hopfield networks and Boltman Machines) use gradient descent to find a minimum for quadratic energy functions. We show an equivalence between the problem of satisfiability in propositional calculus and the problem of minimizing those energy functions. The equivalence is in the sense that for an satisfiable Well Formed Formula (WFF) we can find a quadratic function that describes it, such that the set of solutions that minimize the function is equal to the set of truth assignments that satisfy the WFF. We also show that in the same sense every quadratic energy function describes some satisfiable WFF. Algorithms are given to transform any propositional WFF into an energy function that describes it and vice versa. High-order models that use Sigma-Pi units are shown to be equivalent to the standard quadratic models with additional hidden units. An algorithm to convert high-order networks to low-order ones is used to implement a satisfiability problem-solver on a connectionist network. The results give better understanding of the role of hidden units and of the limitations and capabilities of symmetric connectionist models. The techniques developed for the satisfiability problem may be applied to a wide range of other problems, such as: associative memories, finding maximal consistent subsets, automatic deduction and even non-monotonic reasoning.

Symmetric Neural Nets and Propositional Logic Satisfiability

Gadi Pinkas

WUCS-91-02

January 1991

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

Symmetric Neural Nets and Propositional Logic Satisfiability

Gadi Pinkas

January 23, 1991
WUCS-91-02

Department of Computer Science
Washington University
509 Bryan, Campus Box 1045
One Brookings Drive.
St. Louis, Missouri 63130
pinkas@cics.wustl.edu
(314) 726-7526

Abstract

Connectionist networks with symmetric weights (like Hopfield networks and Boltzman Machines) use gradient descent to find a minimum for quadratic energy functions. We show an equivalence between the problem of satisfiability in propositional calculus and the problem of minimizing those energy functions. The equivalence is in the sense that for any satisfiable Well Formed Formula (WFF) we can find a quadratic function that describes it, such that the set of solutions that minimize the function is equal to the set of truth assignments that satisfy the WFF. We also show that in the same sense every quadratic energy function describes some satisfiable WFF. Algorithms are given to transform any propositional WFF into an energy function that describes it and vice versa.
High-order models that use Sigma-Pi units are shown to be equivalent to the standard quadratic models with additional hidden units. An algorithm to convert high-order networks to low-order ones is used to implement a satisfiability problem-solver on a connectionist network.
The results give better understanding of the role of hidden units and of the limitations and capabilities of symmetric connectionist models. The techniques developed for the satisfiability problem may be applied to a wide range of other problems, such as: associative memories, finding maximal consistent subsets, automatic deduction and even non-monotonic reasoning.

*To appear in "Neural Computation" vol 3-2.

1. Introduction

The problem of satisfiability is deciding whether a truth assignment for the variables of a given propositional WFF exists, so that the formula is evaluated to be true. In many cases the assignment needs also to be found. It is well-known that any of the problems in $\mathcal{NP}$ can be reduced to the satisfiability problem and that satisfiability is $\mathcal{NP}$ complete.

Apart from theoretical importance, satisfiability has direct applications. A satisfiability-problem-solver may be used for example in an inference engine and for solving other hard problems that were reduced to satisfiability during the years.

In this paper we show an equivalence between the satisfiability search problem and the problem of connectionist energy minimization. For every WFF we can find a quadratic energy function such that the values of the variables of the function at the minimum can be translated into a truth assignment that satisfies the original WFF and vice versa. Also, any quadratic energy minimization problem may be described as a satisfiable WFF that is satisfied for the same assignments that minimize the function. More details and formal proofs can be found in [Pinkas90].

Finding minima for quadratic functions is the essence of symmetric connectionist models [Hopfield82] [Hinton,Sejnowski86] [Hinton89]. They are characterized by a recurrent network architecture, a symmetric weight matrix (with zero diagonal) and a quadratic energy function that should be minimized. Each unit asynchronously computes the gradient of the function and adjusts its activation value, so that energy decreases monotonically. The network eventually reaches equilibrium, settling on either a local or a global minimum. [Hopfield,Tank85] demonstrated that certain complex optimization problems can be approximated by these kind of networks and some of the work done in connectionist reasoning and knowledge representation has used energy-minimization models ([Ballard86], [Touretzky,Hinton88],[Derthick88]...).

There is a direct mapping between these models and quadratic energy functions, and most of the time we will not distinguish between the function and the network that minimizes it. Thus, the equivalence between energy minimization and satisfiability means that everything that can be stated as satisfiability of some WFF *and nothing else* can also be expressed in symmetric models. The techniques described are used (in this paper) for the direct implementation of a satisfiability problem-solver on connectionist networks; however they may also be applied to automatic deduction, abduction, construction of arbitrary associative memories and more.

2. Satisfiability and models of propositional formulas

A WFF is an expression that combines atomic propositions (variables) and connectives $(\vee, \wedge, \neg, \rightarrow, (,))$. A model (truth assignment) is a vector of binary values that assigns 1 ("true") or 0 ("false") to each of the variables. A WFF φ is satisfied by a model $\vec{x}$ if its characteristic function H_φ evaluates to "one" given the vector $\vec{x}$.

The characteristic function is defined to be $H_\varphi : 2^n \rightarrow \{0,1\}$ such that:
- $H_{x_i}(x_1,\ldots,x_n) = x_i$
- $H_{(\neg\varphi)}(x_1,\ldots,x_n) = 1 - H_\varphi(x_1,\ldots,x_n)$
- $H_{(\varphi_1\vee\varphi_2)}(x_1,\ldots,x_n) = H_{\varphi_1}(x_1,\ldots,x_n) + H_{\varphi_2}(x_1,\ldots,x_n) - H_{\varphi_1}(x_1,\ldots,x_n) \times H_{\varphi_2}(x_1,\ldots,x_n)$
- $H_{(\varphi_1\wedge\varphi_2)}(x_1,\ldots,x_n) = H_{\varphi_1}(x_1,\ldots,x_n) \times H_{\varphi_2}(x_1,\ldots,x_n)$
- $H_{(\varphi_1\rightarrow\varphi_2)}(x_1,\ldots,x_n) = H_{(\neg\varphi_1\vee\varphi_2)}(x_1,\ldots,x_n)$

The satisfiability search problem for a WFF φ is to find an $\vec{x}$ (if one exists) such that $H_{\varphi(\vec{x})} = 1$.

3. Equivalence between WFFs

We call the atomic propositions that are of interest for a certain application "visible variables" (denoted by $\vec{x}$). We can add additional atomic propositions called "hidden variables" (denoted by $\vec{t}$) without changing the set of relevant models that satisfy the WFF. The set of models that satisfy φ projected onto the visible variables is then called "the visible satisfying models" ($\{\vec{x} \mid (\exists\vec{t})H_{\varphi}(\vec{x},\vec{t}) = 1\}$).

Two WFFs are equivalent if the set of visible satisfying models of one is equal to the set of visible satisfying models of the other.

A WFF φ is in Conjunction of Triples Form (CTF) if $\varphi = \bigwedge_{i=1}^{m} \varphi_i$ and every φ_i is a sub-formula of at most three variables.[1]

Every WFF can be converted into an equivalent WFF in CTF by adding hidden variables. Intuitively, we generate a new hidden variable for every binary connective (eg: $\vee, \rightarrow$) except for the top most one, and we "name" the binary logical operation with a new hidden variable using the connective ($\leftrightarrow$).

EXAMPLE 3.1 Converting $\varphi = ((\neg((\neg A) \wedge B)) \rightarrow ((\neg C) \rightarrow D))$ into CTF:
From $(\neg((\neg A) \wedge B))$ we generate: $((\neg((\neg A) \wedge B)) \leftrightarrow T_1)$ by adding a new hidden variable T_1,
from $((\neg C) \rightarrow D)$ we generate: $(((\neg C) \rightarrow D) \leftrightarrow T_2)$ by adding a new hidden variable T_2,
for the top most connective ($\rightarrow$) we generate: $(T_1 \rightarrow T_2)$.
The conjunction of these sub-formulas is :
$((\neg((\neg A) \wedge B)) \leftrightarrow T_1) \wedge (((\neg C) \rightarrow D) \leftrightarrow T_2) \wedge (T_1 \rightarrow T_2)$. It is in CTF and is equivalent to φ.

4. 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: $E^k(x_1, \ldots, x_n) =$

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

Quadratic energy functions are special cases:

$$\sum_{1 \le i < j \le n} w_{ij} x_i x_j + \sum_{i \le n} w_i x_i + w.$$

We can arbitrarily divide the variables of an energy function into two sets: visible variables and hidden variables.

We call the set of minimizing vectors projected onto the visible variables, "The visible solutions" of the minimization problem. ($\mu(E) = \{\vec{x} \mid (\exists\vec{t})E(\vec{x},\vec{t}) = min_{\vec{y},\vec{z}}\{E(\vec{y},\vec{z})\}\}$).

We can always translate back and forth [Hopfield82] between a quadratic energy function and a network with symmetric weights that minimize it (see fig 1). Further, we can use high-order networks [Sejnowski86] to minimize high-order energy functions (see figure 2). In the extended model each node is assigned a Sigma-Pi unit that updates its activation value using:

[1] CTF differs from the familiar Conjunctive Normal Form (CNF). The φ_i's are WFFs of up to 3 variables that may include any logical connective and are not necessarily a disjunction of literals as in CNF. To put a bidirectional CTF clause into a CNF we would have to generate two clauses, thus $(A \leftrightarrow B)$ becomes $(\neg A \vee B) \wedge (A \vee \neg B)$.

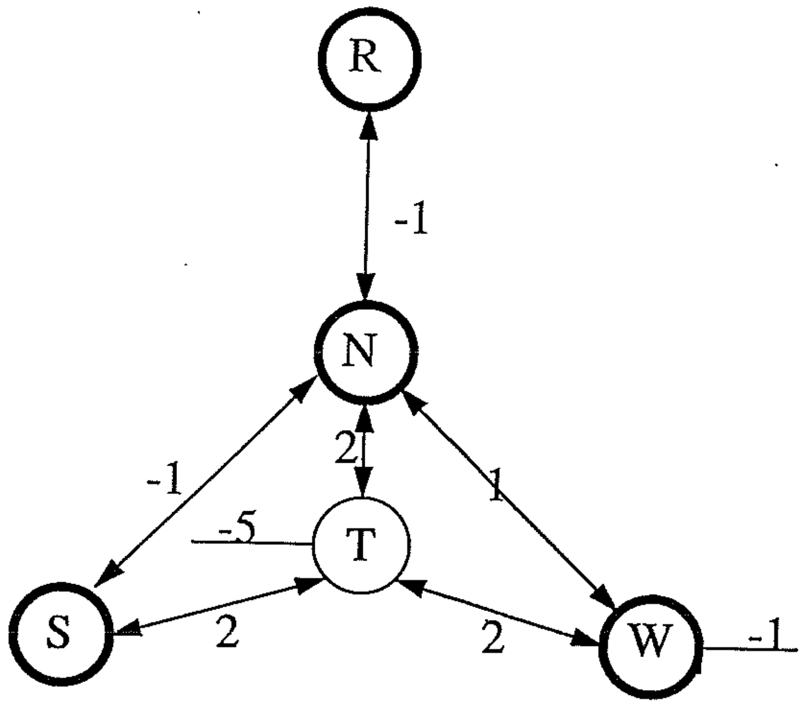

Figure 1: A symmetric network that represents the function $E = -2NT - 2ST - 2WT + 5T + NS + RN - WN + W$, describing the WFF: $(N \wedge S \rightarrow W) \wedge (R \rightarrow (\neg N)) \wedge (N \vee (\neg W))$. T is a hidden unit.

$$a_i = F\left(\sum_{i_1 \cdots i \cdots i_k} -w_{i_1,\ldots,i,\ldots,i_k} \prod_{1 \leq j \leq k, i_j \neq i} x_{i_j} \right)$$

Like in the quadratic case, there is a translation back and forth between k-order energy functions and symmetric high-order models with k-order Sigma-Pi units (see figure 2).

5. The equivalence between high-order models and low-order models

We call two energy functions *equivalent*, if they have the same set of visible solutions.

Any high-order energy function can be converted into an equivalent low-order one with additional hidden variables. In addition, any energy function with hidden variables can be converted into a (possibly) higher one by eliminating some or all of the hidden variables. These algorithms allow us to trade the computational power of Sigma-Pi units for additional simple units and vice versa.

- Any k-order term $(w \prod_{i=1}^{k} x_i)$, with NEGATIVE coefficient w, can be replaced by the quadratic terms: $\sum_{i=1}^{k} 2wX_iT - (2k-1)wT$ generating an equivalent energy function with one additional hidden variable T.

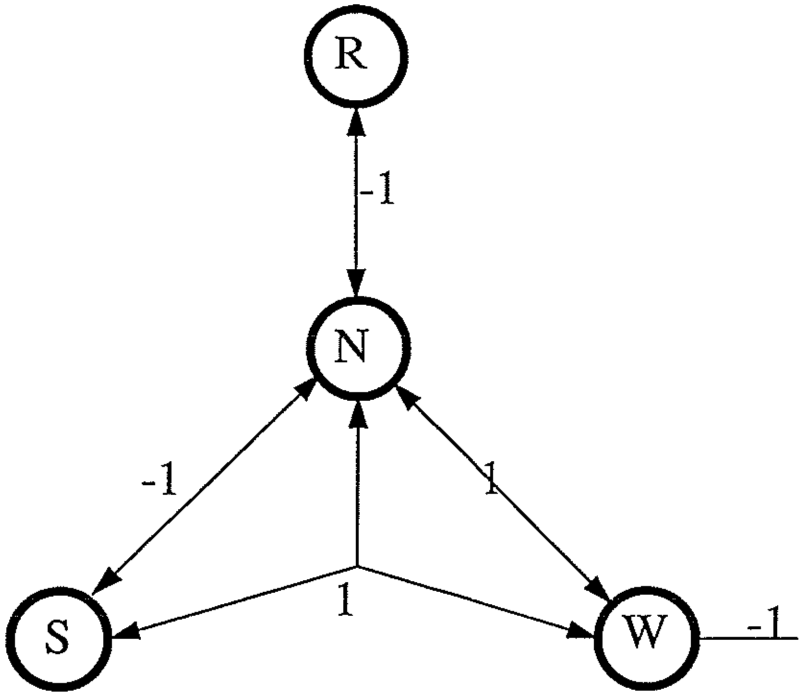

Figure 2: A cubic network that represents $E = -NSW + NS + RN - WN + W$ using Sigma-Pi units and a cubic hyper-arc. (Its is equivalent to the network of Figure 1 without hidden units)

- Any k-order term ($w \prod_{i=1}^{k} x_i$), with POSITIVE coefficient w, can be replaced by the terms: $w \prod_{i=1}^{k-1} x_i - (\sum_{i=1}^{k-1} 2wX_iT) + 2wX_kT + (2k-3)wT$, generating an equivalent energy function of order $k-1$ with one additional hidden variable T.[2]

EXAMPLE 5.1 The cubic function $E = -NSW + NS + RN - WN + W$ is equivalent to $-2NT - 2ST - 2WT + 5T + NS + RN - WN + W$, (introducing T). The corresponding high-order network appears in figure 2 while the equivalent quadratic one in figure 1.

The symmetric transformation, from low-order into high-order functions by eliminating any subset of the variables, is also possible (of course we are interesting in eliminating only hidden variables). To eliminate T, bring the energy function to the form: $E = E' + oldterm$, where $oldterm = (\sum_{j=1}^{k} w_j \prod_{i=1}^{l_j} X_{j_i})T$.
Consider all assignments S for the variables ($\hat{X} = x_{i_1} \cdots x_{i_l}$) in *oldterm* (not including T), such that $\beta_S = \sum_{j=1}^{k} w_j \prod_{i=1}^{l_j} x_{j_i} < 0$.
Each negative β_S represents an energy state of the variables in $\hat{X}$ that pushes T to become "one" and decreases the total energy by $\mid \beta_S \mid$. States with positive β_S cause T to become zero, do not reduce the total energy, and therefore can be ignored. Therefore, the only states that matter are those that reduce the energy; i.e β_S is negative.
Let $L_S^j = \begin{cases} \text{"}X_{i_j}\text{"} & \text{if } S(X_{i_j}) = 1 \\ \text{"}(1 - X_{i_j})\text{"} & \text{if } S(X_{i_j}) = 0 \end{cases}$ it is the expression "X_i" or "$(1 - X_i)$" [3] depending whether

[2] A symmetric (but less efficient) transformation for the positive case also exists.
[3] L_S^j is like a macro and is replaced by either "X_j" or "$1 - X_j$" once it is used in *newterm*.

the variable is assigned 1 or 0 in S.
The expression $\prod_{j=1}^{l} L_S^j$ therefore determines the state S, and the expression

$$newterm = \sum_{S \text{ such that } \beta_S < 0} \beta_S \prod_{j=1}^{l} L_S^j$$

represents the disjunction of all the states that cause a reduction in the total energy.
The new function $E' + newterm$, is therefore equivalent to $E' + oldterm$ and does not include T.

With this technique, we can convert any network with hidden units into an equivalent network without any such units.

6. Describing WFFs using penalty functions

An energy function E describes a WFF φ if the set of visible satisfying models of φ is equal to the set of visible solutions of the minimization of E.[4]

The penalty function E_φ of a WFF φ is a function $E_\varphi : \{0,1\}^n \to \mathcal{N}$, that penalizes sub-formulas of the WFF that are not satisfied. It computes the characteristic of the negation of every sub-formula φ_i in the upper level of the WFF's conjunctive structure.
If $\varphi = \bigwedge_{i=1}^{m} \varphi_i$ then,

$$E_\varphi = \sum_{i=1}^{m} (H_{\neg\varphi_i}) = \sum_{i=1}^{m} (1 - H_{\varphi_i})$$

If all the sub-formulas are satisfied, E_φ gets the value zero; otherwise, the function computes how many are unsatisfied.

It is easy to see that φ is satisfied by $\vec{x}$ iff E_φ is minimized by $\vec{x}$ (the global minima have a value of zero). Therefore, every satisfiable WFF φ has a function E_φ such that E_φ describes φ.

EXAMPLE **6.1**

$$\begin{aligned} E_{((N\wedge S)\to W)\wedge(R\to(\neg N))\wedge(N\vee(\neg W))} &= H_{\neg((N\wedge S)\to W)} + H_{\neg(R\to(\neg N))} + H_{\neg(N\vee(\neg W))} \\ &= H_{N\wedge S\wedge(\neg W)} + H_{R\wedge N} + H_{(\neg N)\wedge W} \\ &= (NS(1-W)) + (RN) + ((1-N)W) \\ &= -NSW + NS + RN - WN + W \end{aligned}$$

The corresponding network appears in figure 2.

The following algorithm transforms a WFF into a quadratic energy function that describes it, generating $O(length(\varphi))$ hidden variables:

- Convert into CTF (section 3).
- Convert CTF into a cubic energy function and simplify it to a sum of products form (section 6).
- Convert cubic terms into quadratic terms. Each of the triples generates only one new variable. (section 5).

The algorithm generates a network whose size is linear in the number of binary connectives of the original WFF. The fan-out of the hidden units is bounded by a constant.

[4] Note, that it is only the minima (and not the details of the function's surface) that cause E to describe φ.

7. Every energy function describes some satisfiable WFF.

In section 5 we saw that we can convert any energy function to contain no hidden variables. We show now that for any such function E with no hidden variables there exist a satisfiable WFF φ such that E describes φ.

The procedure is first to find the set $\mu(E)$ of minimum energy states (the vectors that minimize E). For each such state create an n-way conjunctive formula of the variables or their negations depending whether the variable is assigned 1 or 0 in that state. Each such conjunction $\bigwedge_{i=1}^{n} L_S^i$ where $L_S^i = \begin{cases} \text{"}X_i\text{"} & \text{if } S(X_i) = 1 \\ \text{"}(\neg X_i)\text{"} & \text{if } S(X_i) = 0 \end{cases}$ represents a minimum energy state. Finally the WFF is constructed by taking the disjunction of all the conjunctions: $\varphi = \bigvee_{S\epsilon\mu(E)}(\bigwedge_{i=1}^{n} L_S^i)$. The satisfying truth assignments of φ correspond directly to the energy states of the net.

8. Conclusions

We have shown an equivalence between the search problem of satisfiability and the problem of minimizing connectionist energy functions. Only those problems that can be stated as satisfiability search problems (and every such problem) can be stated in symmetric neural networks.

Any satisfiable WFF can be described by an n-order energy function with no hidden variables, or by a quadratic function with $O(length(WFF))$ hidden variables. Using the algorithms described we can efficiently determine the topology and the weights of a connectionist network that represents and approximates[5] a given satisfiability problem.

Several other applications may benefit from the techniques developed here. Two are: associative memory, and finding maximal consistent subset.

Given a set of binary vectors we wish to store in an associative memory, we can construct a WFF φ that is satisfied for all and only these vectors. (φ is the boolean implementation of the function that outputs "one" for all memory vectors and "zero" otherwise). By implementing the network that describes φ, we get an associative memory which performs completion when only a portion of the input is supplied.

As a second application consider a set of possibly contrary beliefs. We can construct a network that will search for a maximal consistent subset (adding degree of belief or certainty as in [Derthick88], is a simple extension of the penalty function). Subsets of beliefs compete among themselves and some are defeated in favor of others [Pinkas91]. The network searches for a maximal consistent subset of beliefs such that the total penalty is minimized.

9. Acknowledgments

I thank Bill Ball, Sally Goldman, Dan Kimura, Arun Kumar, Stan Kwasny, Ron Loui and Dave Touretzky for helpful comments and suggestions. This work was supported by McDonnell Douglas, Southwestern Bell, and Mitsubishi America at the Center for Intelligent Computing Systems and by NSF grant 22-1321 57136.

[5]The networks are not always capable of escaping from local minima.

References

[Ballard86] D. H. Ballard "Parallel Logical Inference and Energy Minimization" *Proceedings of the 5th National conference on Artificial Intelligence*, Philadelphia, Pa., August 1986, pp. 203-208

[Derthick88] M. Derthick "Mundane reasoning by parallel constraint satisfaction" PhD thesis, CMU-CS-88-182 *Carnegie Mellon University*, Sept. 1988

[Hinton89] G.E Hinton "Learning in Mean Field Theory" *Neural Computation* vol 1-1, 1989

[Hinton,Sejnowski86] 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

[Hopfield82] J. J. Hopfield "Neural networks and physical systems with emergent collective computational abilities", *Proceedings of the National Academy of Sciences* USA, 1982 , 79,2554-2558.

[Hopfield,Tank85] J.J. Hopfield, D.W. Tank "Neural Computation of Decisions in Optimization Problems" *Biological Cybernetics*, Vol 52, pp. 144-152.

[Pinkas90] G. Pinkas, "Energy Minimization and the Satisfiability of Propositional Calculus " *Technical Report:* WUCS-90-03, Department of Computer Science, Washington University 1990

[Pinkas91] G. Pinkas, " Propositional Non-Monotonic Reasoning and Inconsistency in Symmetric Neural Networks", *Technical Report:* WUCS-91-03, Department of Computer Science, Washington University, 1990.

[Sejnowski86] T. J. Sejnowski "Higher-Order Boltzman Machines" Neural Networks for Computing, *Proceedings of the American Institute of Physics* 151, Snowbird (Utah) pp 3984.

[Touretzky,Hinton88] D.S Touretzky, G.E. Hinton "A distributed connectionist production system" *Cognitive Science* 12(3):423-466.