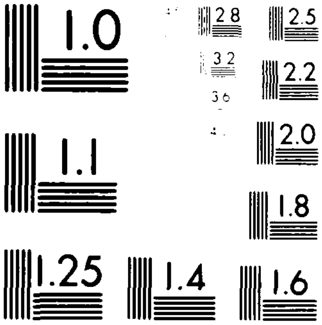
AD-A170 324   ENERGY AND THE BEHAVIOR OF CONNECTIONIST MODELS(U)   1/1
ROCHESTER UNIV NY DEPT OF COMPUTER SCIENCE J A FELDMAN
NOV 85 TR-155 N00014-84-K-0655

UNCLASSIFIED   F/G 9/2   NL


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

Energy and the Behavior of Connectionist Models

Jerome A. Feldman
Computer Science Department
The University of Rochester
Rochester, NY 14627

TR 155
November 1985

Department of Computer Science
University of Rochester
Rochester, New York 14627

Energy and the Behavior of Connectionist Models

Jerome A. Feldman
Computer Science Department
The University of Rochester
Rochester, NY 14627

TR 155
November 1985

Abstract

Massively parallel (connectionist) computational models are playing an increasingly important role in cognitive science. Establishing the behavioral correctness of a connectionist model is exceedingly difficult, as it is with any complex system. For a restricted class of models, one can define an analog to the energy function of physics and this can be used to help prove properties of a network. This paper explores energy and other techniques for establishing that a network meets its specifications. The treatment is elementary, computational, and focuses on specific examples. No free lunch is offered.

This research was supported in part by the Office of Naval Research under Grant N00014-84-K-0655, the Defense Advanced Research Projects Agency under Grant N00014-82-K-0193, and the National Science Foundation (Coordinated Experimental Research) under Grant DCR-8320136.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER TR 155	2. GOVT ACCESSION NO. AD-A176 374	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Energy and the Behavior of Connectionist Models		5. TYPE OF REPORT & PERIOD COVERED technical report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Jerome A. Feldman		8. CONTRACT OR GRANT NUMBER(s) N00014-84-K-0655 N00014-82-K-0193
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Science Department University of Rochester Rochester, NY 14627		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS DARPA / 1400 Wilson Blvd. Arlington, VA 22209		12. REPORT DATE November 1985 13. NUMBER OF PAGES 39
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research Information Systems Arlington, VA 22217		15. SECURITY CLASS. (of this report) unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

simulated annealing
connectionist networks
parallel computing
artificial intelligence
proof of correctness
neural networks
Boltzmann machine
energy minimization

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Massively parallel (connectionist) computational models are playing an increasingly important role in cognitive science. Establishing the behavioral correctness of a connectionist model is exceedingly difficult, as it is with any complex system. For a restricted class of models, one can define an analog to the energy function of physics and this can be used to help prove properties of a network. This paper explores energy and other techniques for establishing that a network meets its specifications. The treatment is elementary, computational, and focuses on specific examples. No free lunch is offered.

Energy and The Behavior of Connectionist Models

Table of Contents

✓

A-1

1. Introduction

Massive parallel (connectionist) models are playing an increasingly important role in cognitive science and are beginning to be employed more widely. The success of initial exploratory studies has led to efforts to systematize and analyze this style of computation. One important aspect of this effort involves the use of formal techniques to specify the required behavior of a model and to verify that the realization meets the stated requirements.

While the verification of complex computational systems has a long history within computer science, these efforts have not led to significant benefits in the design or performance of computer software. Similarly, verification of computer hardware arose as a topic in artificial intelligence and is not part of the normal design cycle. There is currently a mood of greatly reduced expectations among researchers in formal verification.

It is not realistic to assume that we will be able to provide complete specifications for complex connectionistic models any better than we can for traditional programs such as operating systems on digital computers, or for digital circuits. This is certain to hold for the complex connectionist models needed to study such intelligent behaviors as vision and language.

Granting that formal specification and proof will not be feasible for complex models, there are still several reasons for exploring these ideas. It is feasible to characterize sub-systems, and these can then be used with confidence in larger projects. The attempt to formally specify a sub-system is often of considerable heuristic value in itself, and the verification process inevitably leads to insights and often uncovers errors in design. In addition, having an explicit goal of formal specification and verification influences the choice of primitive units used in a model, how they are connected, and the rules of timing and data transmission assumed.

This paper explores a number of methodologies for studying the behavior of connectionist models and a number of strengths and weaknesses of each approach. Special emphasis is placed on the "energy" formulations modeled on statistical mechanics. All of the models considered share the notions of a large network of simple computing units joined by links of varying numeric weights. We are beginning to understand the computational basis of the success of connectionist models (CMs) and that this success is fairly robust over a variety of choices in details. The two central ideas are the use of numerical parameters and the simultaneous utilization of all relevant information at points of decision. Numerical values can be looked upon as providing *evidence* for various propositions or states of the system and have proved to be useful on problems that have proved difficult in formalisms based on logic and symbolic parameters. Evidential inference does not require parallel treatment, but there does seem to be a natural fit which is well captured by connectionist models. The network form emphasizes the *interaction* among contributing factors rather than isolating them as rule-based formulations tend to do. The implied computational rule, shared by all CMs, is that *all* of the inputs to a given unit be combined to yield its output value. The following definitions attempt to capture the key ideas shared by CMs while leaving open the precise form of combination and propagation rules.

The general computational form of a *unit* in our models will be comprised of:

$\{Q\}$ -- a set of discrete *states* < 10

P -- a continuous value in $[-10,10]$ called *potential*

X -- an output *value* $-1 \leq X \leq 10$

D -- a vector of data inputs $d_1 \ldots d_n$

and functions from old to new value of these:

$$(1) \qquad \begin{aligned} P &\leftarrow f(D, P, Q) \\ Q &\leftarrow g(D, P, Q) \\ X &\leftarrow h(D, P, Q) \end{aligned}$$

The form of the *f*, *g*, and *h* functions and the precise rule for updating them will vary within this paper and can entail probabilistic functions. Some of the notation will be suppressed when it is not needed, e.g., we will sometimes have X identically equal to P and will also sometimes refer to the inputs to a unit by X_j, the output of a predecessor unit.

Most models use only potentials in the range [-1, 1] and outputs in the range [0, 1]. Some authors [Smolensky, 1986; Selman and Hirst. 1985] find it technically convenient to use outputs of ± 1 while acknowledging that negative outputs do not map well to biology. Others, particularly at Rochester, emphasize the limited dynamic range of neural signals by restricting outputs to be integers in the range [0,10) while allowing continuous valued potentials. None of the issues addressed in this paper are effected by such considerations.

More generally, there is (already) a significant range of connectionist models with varying goals and assumptions. The work on CM models overlaps a much broader area of parallel algorithms, particularly for constrained optimization problems. The distinguishing characteristic of connectionistic models is the requirement that all decisions be computable in distributed fashion by simple computing units. The biological or electronic plausibility of a given model is of second-order concern here, but will be given some attention.

The definitions above are essentially the same as those given in [Feldman and Ballard, 1982]. The hope there was that the definitions would be sufficiently general to accommodate all connectionist paradigms and, so far, through a rich variety of subsequent efforts, this has essentially held. The major difference here is that I explicitly allow continuous valued output functions, whereas they were discouraged in the earlier version. Continuous output values are unrealistic for most neural computation, but are harmless if the model does not depend on the fine structure of the values. The question of the exact rule for timing the updates of P, Q, and X is a critical one for formal treatment of behavior. The original definitions specified a strictly synchronous rule, and most energy models rely on a strictly asynchronous rule. What one would like is a methodology which is oblivious to the form of the update rule. This issue is discussed further in Section 4. One additional condition that might be made part of the definition is that the *h* function specifying the output be a monotonically increasing function of the potential. All models have satisfied this constraint and it is probably time to canonize it. Potential is used to capture an internal level of activity and the external activity should increase with potential for both computational clarity and biological versimilitude. The only other issue is the scale factors on potential, weights and outputs for which conventions have not yet stabilized.

In this paper, a number of particularizations will be employed because many results hold only for special choices. All of our examples in Sections 1 and 2 will have only one state so that parameter *Q* will be supressed for now. The first system to be examined is composed entirely of binary linear threshold units. Each unit has an output equal to its potential of 0 or 1. The potential and output are computed by the rule:

$$(2) \qquad P_i = \sum w_{ji} X_j - \theta_i$$

$$X_i = \underline{\text{if}}\ P_i > 0\ \underline{\text{then}}\ 1\ \underline{\text{else}}\ 0$$

where w_{ji} are *weights* of either sign. This is essentially the standard perceptron [Minsky and Papert, 1969] and has been used as the basis for much current work, e.g. [Ackley et al., 1985]. A simple illustration of how such units might be employed is given in Figure 1.1, which is a vastly oversimplified version of the word recognition network of [McClelland and Rumelhart, 1981]. The idea is to have recognizing units for letters in different positions (I_1, T_2) that are linked to units that recognize words ("IT"). One could have, e.g., all the weights be 1 and have the rule for "IT" be

$$X_{IT} = X_{I_1} + X_{T_2} - 1$$

so that "IT" would be recognized just when it should be. The major concern of this paper is formalizing the notion of a connectionist network "doing what it should," and we will see that this is not usually straightforward.

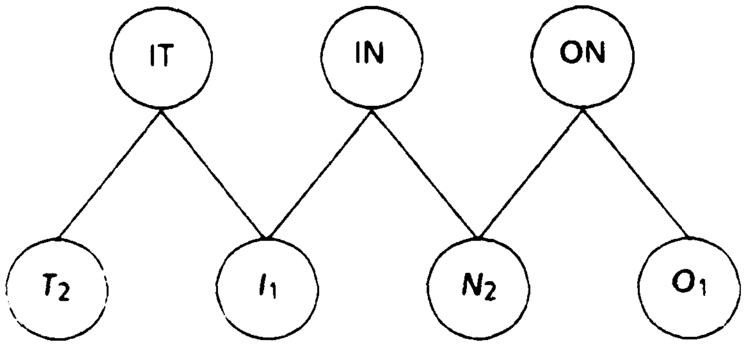


Figure 1.1

1a. The Goodness/Energy Paradigm

One promising approach to proving the correctness of networks is based on the notion of a global "goodness" or "energy" measure [Hopfield, 1982; Ackley et al., 1985; Smolensky, 1986]. These notions are usually motivated by analogies from statistical mechanics but do not require such treatment. The basic idea is quite simple--one would like to find a global measure that could be shown to decrease every time an appropriate change was made to the potential and output of a unit in the network. Should we be able to establish such a measure, and if it is bounded in value, then the networks will always converge. A locally monotonic goodness measure can also be used probabilistically to search for solutions of more complex problems. The remarkable fact is that, under a specific set of assumptions, such a measure can be established. We will proceed by pulling a goodness measure out of a hat and then show how and why it works.

We will first assume the goodness measure G for a network is the sum of contributions G_i from its individual units. For each unit, its contribution will consist of terms describing its interactions with other elements of the network. For convenience we consider the threshold terms and any external inputs to be from extra units whose outputs never change (cf. Figure 1.2). In this case, the contribution of unit i and the total for the network are given by:

$$(3) \qquad G_i = X_i \sum_j w_{ji} X_j \quad \text{and} \quad G = \sum_i G_i$$

First notice that $G_i = 0$ if $X_i = 0$; a unit that is off makes no positive or negative contribution. If $X_i = 1$, then every unit connected to X_i (i.e. $w_{ji} \neq 0$) and that is also on ($X_j = 1$) makes a positive contribution if $w_{ji} > 0$ and a negative one if $w_{ji} < 0$. Intuitively this goodness measure is higher when units linked by positive weights are simultaneously active. Our goodness measure is essentially the same as the *Harmony* of [Smolensky, 1986] and its negative is essentially the *energy* of [Hopfield, 1982; Ackley et al., 1985]. We will talk both of increasing goodness or decreasing energy depending on what sounds more natural in context.

Now our purpose in developing the goodness function was to show convergence of network computations by establishing that G can be made to always increase. This will be true if its derivative is always greater than zero. Let us assume that at each time slot exactly one unit, X_i, evaluates whether to change its value. We will look at the effect of *continuous* change of X_i on G which will be needed later anyway. The total energy function G can be rewritten to pull out the role of the variable X_i that is being considered:

$$(4) \qquad G = X_i \sum_j w_{ji} X_j + \sum_{j \neq i} (X_j \sum_k w_{kj} X_k)$$

The derivative of G with respect to X_i has two terms -- from the first term of (4) comes the term $\Sigma w_{ji} X_j$. From the remaining sum over $j \neq i$ the derivative is non-zero only where X_i appears in the inner sum (i.e., when $k = i$) and the derivative of each summand is $X_j w_{ij}$, yielding:

$$(5) \qquad \frac{\partial G}{\partial X_i} = \sum_j w_{ji} X_j + \sum_{j \neq i} w_{ij} X_j$$

If we make a further major assumption that $w_{ij} = w_{ji}$ everywhere (symmetric weights) and a minor one that $w_{ii} = 0$ we get the main result:

$$(6) \qquad \frac{\partial G}{\partial X_i} = 2 \sum_j w_{ji} X_j$$

In this case, the entire effect on G of changing X_i can be determined from only the *incoming* weights and values to X_i. The factor of 2 in Equation 6 will be suppressed throughout the paper for simplicity. Now recall that we wanted to guarantee that each choice of X_i makes G increase. This will be assured by the rule

$$(2) \qquad X_i = \underline{\text{if}}\ \Sigma\, w_{ji} X_j > 0\ \underline{\text{then}}\ 1\ \underline{\text{else}}\ 0$$

This is just the original rule for binary linear threshold units--we have provided a rationale for choosing this particular rule [Hopfield, 1982]. That is, our goodness function was chosen so that its derivative with respect to X_i was positive exactly when $X_i = 1$ by Equation 2. The remainder of Section 1 and all of Section 2 explore the goodness/energy formulation and its strengths and weaknesses.

Let us consider how the G function would work for the example of Figure 1.1. Assume that exactly I_1 and T_2 were on and the unit for "IN" was tested to see whether it should go on.

$$\begin{aligned} X_{\text{IN}} &= 1\,X_{I_1} + 1\,X_{N_2} - 1 \\ &= 0 \quad \text{in this case} \end{aligned}$$

and rule (2) says X_{IN} should be zero. We can also compare the value of G with $X_{\text{IN}} =$ 0 or 1.

$$\begin{aligned} G_{\text{IN}} &= X_{\text{IN}}(X_{I_1} + X_{N_2} - 1) \\ &= 0 \quad \text{for } X_{\text{IN}} = 1 \text{ or } X_{\text{IN}} = 0 \end{aligned}$$

Now if the unit for "IT" is tested, we see that

$$\begin{aligned} G_{\text{IT}} &= X_{\text{IT}}(X_{I_1} + X_{T_2} - 1) \\ &= X_{\text{IT}}\ 1 \end{aligned}$$

The value G_{IT} is obviously one where $X_{\text{IT}} = 1$ and zero when $X_{\text{IT}} = 0$. Thus G is higher when $X_{\text{IT}} = 1$ and, of course, equation (2) does specify $X_{\text{IT}} = 1$. We will look very soon at more interesting examples.

A number of assumptions were necessary to establish the increasing goodness relation. The units had to be binary and to compute their output by a linear threshold function. The weights linking any two units had to be symmetric. For future reference, we will refer to networks with (binary) linear threshold elements as (B)LTE networks. If symmetric weights are also required, the networks will be denoted SLTE or BSLTE when outputs are binary. Furthermore, a major assumption was made about the timing of the units' actions: only one unit could act at a time, and its output had to reach all other units before they could act. All of these assumptions are considered further below.

Even under all these assumptions, we have not shown that our networks do what they should do, only that they converge. (They converge because goodness never decreases, if we follow rule (2), and goodness clearly has an upper bound for any specific network.) In order to extend the convergence proof to a correctness one, we must show that the right answer is the one to which the network will converge. This turns out to involve three hard sub-problems. The first sub-problem is to specify formally the desired behavior; this is independent of any particular proof technique and constitutes a recurrent theme. The second sub-problem is to show that the desired state (for each input configuration) has the greatest goodness. The third sub-problem is to show that the system actually reaches this state of maximum goodness. It may seem, at first, that we have already solved the third problem by showing convergence. The difficulty here is that while goodness always increases with each change, the system might converge to a value of goodness which is only a regional maximum, not the greatest possible value. This issue of regional (local) optima is ubiquitous in search problems and is another theme of the paper.

The example in Figure 1.2 depicts a situation in which the regional optimum problem can arise in a tiny system. The additions to Figure 1.1 are a unit for the word "I," an explicit θ node for the thresholds, and some specific weights on the connections (all unlabelled connections have weight = 1). The idea here is that the word "I" is in a mutual inhibition relation to longer words starting with "I" such as "IT" and "IN." The threshold for the word "I" is assumed to be .5 and for "IT" to be 1. This should all seem plausible; the network meets our conditions and the various words seem to be activated when the right letters are activated. There is a bug, however, and it shows up as a regional maximum in goodness. Assume that exactly I_1 and T_2 are active (=1), and consider the contributions to G of the units for the words "I" and "IT."

$$G_I = X_I \cdot (X_{I_1} - X_{IT} - .5)$$

$$G_{IT} = X_{IT} \cdot (X_{I_1} + X_{T_2} - X_I - 1)$$

If unit "I" is considered first, $X_{IT} = 0$ and so $\Sigma w_{ij} x_j = 1 - .5 > 0$ and X_I will be set to 1. This yields a contribution of .5 to G. Now if unit "IT" is considered, its $\Sigma w_{ij} X_j$ term turns out to be $(1 + 1 - 1 - 1) = 0$ (because $X_I = 1$), and so $X_{IT} = 0$. However, if "IT" were chosen before "I" for consideration, X_I would be 0 and X_{IT} would be set to 1. This would inhibit "I" but, more importantly, would make a contribution of + 1 to G. Thus the configuration with $X_{IT} = 1$ is better than the one with $X_I = 1$, but is unreachable from there. This is entirely typical of the general case; an early decision precludes one that would turn out to be better (for G).

In this case, one could fix the bug--for example, by having a negative link from any letter in the second position (e.g., T_2) to the word "I." More generally, it is not at

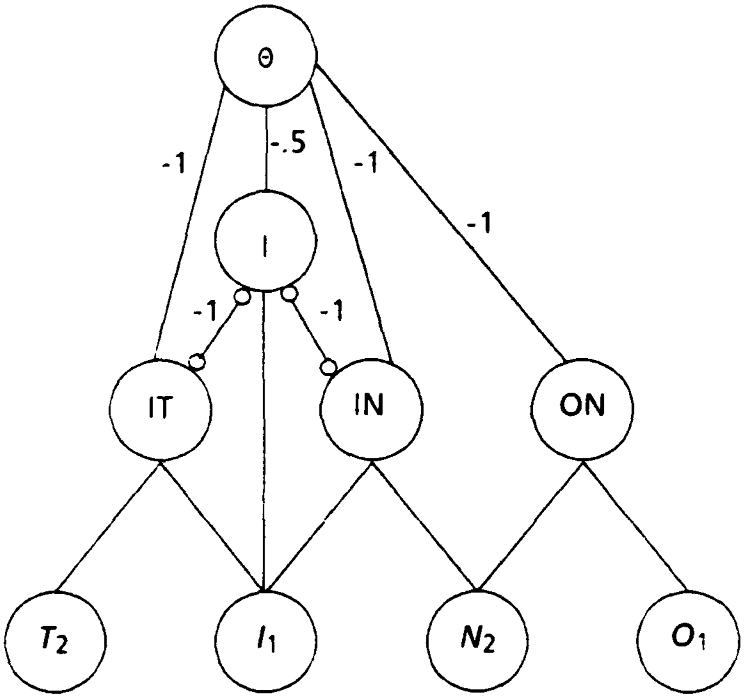


Figure 1.2

all clear how one could avoid the regional extremum problem, and much effort has been devoted to overcoming it, at least partially.

It is also interesting to consider what happens to the example of Figure 1.2 under an updating rule that examines all units simultaneously rather than one at a time. Again assume that exactly I_1 and T_2 are on to start. Now at the first simulation step, both "IT" and "I" will be turned on, according to rule (1). At the second step (since each now has a rival), they would both be turned off, and so on, looping forever. This shows how the choice of timing of the updating can have a major effect on the behavior of a network. An important aspect of modeling is to develop models whose behavior is not dependent on such properties of the simulation. Again, we don't currently know how to do this in general.

One proposed way of evading regional optima is to add a controlled element of randomness to the unit updating process. In the binary output case, we can have the output be 0 or 1, depending on the value of a random variable, which is a function of the unit's potential. The analog with physics suggests the function

$$(7) \qquad prob(X_k = 1) = \frac{1}{(1 + e^{-\Delta G/T})}$$

where ΔG is the difference in goodness between when $X_k = 1$ and when $X_k = 0$. This formulation is called the Boltzmann machine [Ackley et al., 1985]. The parameter T

is analogous to temperature and will be discussed later. For now, we assume $T = 1$, yielding the solid curve in Figure 1.3. Qualitatively, we note that $prob(X_k = 1)$ is 1/2 when its contribution to energy (goodness) is zero and goes up as ΔG_k increases. In our little example, $\Delta G_I = .5$ and $\Delta G_{IT} = 1$, and the better answer will be chosen more often. That is, even if unit "I" is tried first, there is a smaller chance that it will be set to 1. By changing the value of T, one can make the system closer to a random choice (T large, dashed line in Figure 1.3) or to a deterministic system like we started with ($T \sim 0$; dotted line). One could arrange to start T at a high value (to avoid regional extrema) and then lower it (to get exact fit locally). This is the idea of "simulated annealing," and will be discussed below. There are a number of other interesting issues that arise even in this simple example.

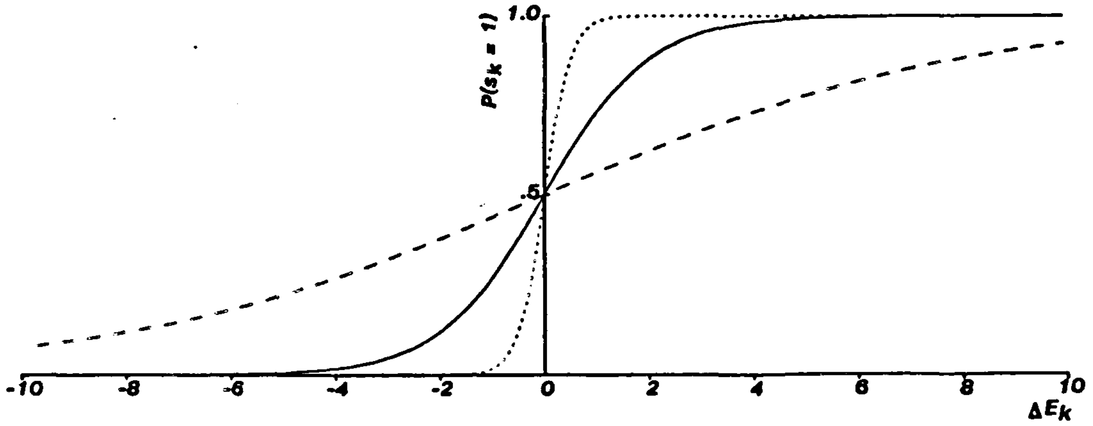


Figure 1.3: Probability $X_k = 1$ as a Function of ΔG_k (from [Ackley et al., 1985])

One important issue is the use of time-averages to characterize the behavior of a network. In our example of Figure 1.2, even with random selection the network would still get the wrong answer on something like 1/3 of the trials. One could try to reduce the fraction of mistakes by changing weights, but as long as ΔG_I is positive, $X_I = 1$ will be chosen at least 1/4 of the time. (The probability of "I" being tried before "IT" = 1/2, and for $\Delta G_I > 0$, the probability of $X_I = 1$ is $\geq 1\ 2$). People differ in their taste for the general idea of time-averages as a criterion for behavior. It seems to me to be difficult to have a system make a decision or choose an action based on a time average without a network that essentially computes the averages. Other workers have made a virtue of necessity by claiming that time averages are a good model of reversible figures or ambiguous words [Selman, 1985], but the time course of phenomena appears to be out of scale with the computational model. An extreme view, which we will not pursue here, is that all mental activity is represented by correlations among the firing patterns of neurons [Bienenstock, 1985].

Much of the work based on goodness and energy takes a different approach to the problem of uncertain behavior in networks that depend heavily on stochastic elements. One can show formally [Ackley et al., 1985] that a system that uses the computation rule (2) with the modification given in (7) will settle into different states with probability related to their energy difference.

$$(8) \qquad \frac{P(B)}{P(A)} = e^{-(E_A - E_B)/T}$$

This doesn't help much by itself unless the desired state (correct solution) is much better than all others, in which case the system will almost always be right. In the more general case, people rely on an "annealing schedule" to increase the probability that the system will settle in the lowest energy state (always assuming that it has been shown to be the correct answer). One can see from equation (8) that as $T \rightarrow 0$, the probability ratio can get arbitrarily small, even for small energy gaps. The problem is that for very small T, the updating rule in (7) is almost deterministic and thus has only a small chance of escaping a regional optimum (cf. also Figure 1.3) and the system will take a very long time to reach the state where it is almost always activating the right answer. A typical annealing schedule for a small problem might start with a fairly high value of T (typically ~20) and lower it in steps down to 2. The results, roughly speaking, are that a moderate number of temperature changes (~10) can lead to a solution that is usually right, but with a significant fraction of errors (~10%). A slower schedule and lower bottom can reduce the residual error. Again, people differ in their taste for computations that require several successive approximations and have significant error probabilities. In the learning paradigms where this idea has been most widely used, it doesn't matter much because the learning typically involves thousands of runs. We will return to the role and plausibility of annealing later in the paper. A particularly clean presentation of the idea and its relation to physics can be found in Smolensky [1986].

An alternative to binary outputs is to model the computation with continuous valued outputs whose value might be taken to represent average firing frequency. There have been some very nice results developed for binary output units, particularly in learning, and we discuss these in Section 2, but we will focus on the continuous output case. In general, the restriction to binary outputs makes it difficult to treat many important phenomena. For example, the McClelland and Rumelhart work that was the basis for Figure 1.2 was concerned with visual features and how they might contribute to recognizing a letter like I_1 or T_1 (which share many features). To model this, one would seem to need (as was used) a richer output space than (0,1) to capture the notion of how confident a detecting unit might be that it had seen its target. It turns out that the goodness/energy theory extends neatly to this case.

In fact, our treatment of the goodness function and its derivatives was done assuming continuous valued outputs X_i and then specialized to setting X_i to 0 or 1. Now that we allow continuous valued outputs, the question arises as to what value of output should correspond to a given value of potential $P_j = \Sigma w_{ij} X_j$. Most work uses only non-negative outputs, and in such a system a negative potential should obviously yield an output of zero. But what about positive potentials? Any monotonic function of potential is defensible, and the choice turns out to be a trade-off between speed of convergence and the avoidance of regional extrema. We will look more carefully at a sigmoidal output rule in Section 2b.

Even the simplest continuous output rule

$$X_i = \underline{\text{If}}\ P_i > 0\ \underline{\text{then}}\ P_i\ \underline{\text{else}}\ 0$$

will suffice to fix the regional optimum problem we had in connection with Figure 1.2. Suppose everything is as before and (the worst case) unit "I" is considered first. Now $X_I = P_I = .5$, not 1 as before. The effect of this is that when "IT" is tested, it will have $X_{IT} = (1 + 1 - .5 - 1) = .5$, and at this point both hypotheses will be equally active. However, the next time that unit "I" is tested, X_I will be zero, and this will lead to X_{IT} being 1 on its next test and forevermore. This simple example is illustrative of several general points. A continuous output system can sometimes avoid traps that would catch the binary equivalent. Indeterminate situations are well-modeled by equal potentials. Finally, we see how continuous output systems can settle into a final state with binary outputs. More information and examples on these points can be found in [Hopfield and Tank, 1985; Rumelhart et al., 1986].

1b. Winner-Take-All Networks

In any event, now that we are equipped with continuous output units, we can explore additional questions concerning energy and the behavior of connectionist models. To illustrate the issues, we introduce a second example: a two-unit mutual inhibition network, shown in Figure 1.4. Mutual inhibition is among the most basic computations in neural modeling [Ratliff and Hartline, 1974], and has been studied for many years in various ways. There were mutual inhibitory links in Figure 1.2 between "I" and "IT" and between "I" and "IN." We will look at how one might specify the desired behavior of such a network and whether goodness/energy considerations could be useful in proving that the specification is satisfied.

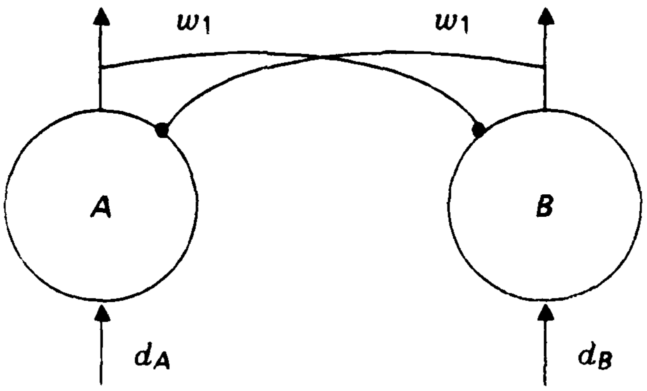


Figure 1.4: Mutual Inhibition Network

As part of this exercise, we will look more seriously at the question of what is involved in specifying the behavior of a network. Informally, mutual inhibition suggests that the output of each unit should decrease as the output of the other increases. For discussion, assume that the data input to unit A, $d_A > d_B$. We almost certainly want the steady state value of $X_B < d_B$ and might well want $X_A \geq d_A$. One simple question would be whether there is a symmetric linear threshold element (SLTE) network that guarantees these two conditions hold.

It turns out that we have not specified nearly enough to make this a well-formed question. For example, we have not specified whether the inputs will be clamped on or will stop, how soon after the inputs appear that the final conditions must hold, how long afterwards they should hold, etc. The assumption that both units start with $X = 0$ can also be a critical one. The issue of whether updating is synchronous, asynchronous, or continuous must also be specified, as well as assumptions on the speed of transmission. One might well want the lower valued input to converge to zero after some time, the winner-take-all (WTA) property. In addition, one would need to know what the network should do in the event that $d_A = d_B$. Even for this small example, one could add a number of other considerations, and the specification problem for scientifically interesting networks is formidable indeed.

In fact, people are more likely to design a network first and later try to establish its properties. This program entails a serious risk of building a Procrustean bed for the original problem, but let us pursue it for the moment. The obvious network is shown as Figure 1.4. The SLTE updating rules for this network, assuming thresholds of zero, are (where $\lfloor X \rfloor$ denotes the greater of X and zero):

$$\text{(9)} \quad \begin{aligned} X_A &= d_A - w_1 \lfloor X_B \rfloor \\ X_B &= d_B - w_1 \lfloor X_A \rfloor \end{aligned}$$

Each unit in Figure 1.4 has an inhibitory link of the same weight w_1 to the other unit, as well as additional input and output. How can we determine acceptable values for w_1 and establish the behavior of the network? For many such networks, it turns out that no SLTE will suffice, and goodness methods are inapplicable. However, the WTA problem does have a nice solution using SLTE. I have not succeeded in using goodness methods to show this, but have done so with simple algebraic considerations. It would be interesting to see if someone else finds goodness measures more informative here.

It is easy to show that the network of Figure 1.4 will have the WTA property if (and only if) $w_1 = 1$. Suppose $d_A > d_B$ in some instance, so that X_B should go to zero. Since $X_A \leq d_A$ always from (9), $X_B \geq d_B - w_1 d_A$. A little algebra shows that for $X_B \to 0$ we need

$$d_B / d_A \leq w_1$$

If $w_1 = 1$, this will be true whenever $d_A > d_B$ as specified. When $w_1 < 1$ there will be cases when $X_B \not\to 0$ and when $w_1 > 1$ there will be cases when both X_A and $X_B \to 0$. Of course values of w_1 close to 1 will only fail for small differences between X_A and X_B. As mentioned, there does not appear to be any way to use goodness arguments to establish the correctness of the small network while simple algebraic considerations work fine.

It is also interesting to ask whether larger WTA networks can be built from SLTE. If so, the mutual inhibition weight must be 1 because it could happen that only two of the rivals are active. It turns out that the obvious extension to Figure 1.4 also works. The updating equation for the n-element WTA is:

$$(10) \qquad X_i = d_i - \lfloor w \sum_{j \neq i}^{n} X_j \rfloor$$

where $w = 1$. Assuming all $X_i = 0$ initially and that the asynchronous update rule is used, this network will result in the unit with the highest input only remaining non-zero. If the unit X_{big} with the highest unit is tried first it will have output = d_{big} and (by (10)) no other unit will be turned on. Moreover, whichever unit is tried first will go on and no smaller or equal input will ever be activated. If a unit with a larger input than the current leader is chosen, it will be activated with its potential (and output):

$$X_{new} = d_{new} - \sum_{old} X_i$$

(from (10)). But then the new value of ΣX_i will be

$$\sum_{new} X_i = \sum_{old} X_i + (d_{new} - \sum_{old} X_i)$$
$$= d_{new}$$

Thus the total rivalry term is the data input of the largest seen so far. Obviously when d_{big} is tested it will be activated and $\Sigma X_i = d_{big}$. From then on, any other unit that is evaluated will have output zero. This result depends on the data inputs, particularly d_{big}, remaining clamped long enough, and can otherwise fail. It will also fail for the synchronous updating rule. This is typical of results with the asynchronous updating model, depending on the persistence of inputs, and the Boltzmann machine experiments have all been done with static, clamped inputs. Again, while the SLTE assumptions behind the goodness model support an elegant solution to the WTA problem, the role of the goodness function is not obvious. In fact, there are simpler and more robust and efficient WTA networks employing, e.g., *maximum* [Shastri, 1985], available if one drops the SLTE restriction. One such example is discussed in Section 4. There are a number of issues concerning the WTA problem, some of which appear in Section 3c, but that is not the focus of this paper (cf. [Feldman and Ballard, 1983]). In Section 2 we will consider a number of cases where the SLTE restrictions preclude desired computations and cases where the formulation has been very helpful. Finally, in Section 2c, we consider some modifications that might extend the range of SLTE networks and goodness techniques. Section 3 deals with some other proof methods and a number of related questions.

2. The Range of Computations Covered by Goodness

2a. Some Cautionary Tales

In this section, we will attempt to explain some of the strengths and limitations of the SLTE/energy model. The model is clearly not universal--the operative question is whether or not it constitutes an adequate basis for essentially all modeling (with minor adjustments) or rather should be treated as one of several special-purpose techniques available in more general formulations. This first subsection points out some computational restrictions on the use of linear combination rules, goodness functions, symmetric links, and the proof techniques that depend on these assumptions.

The idea of using the goodness/energy formulation to understand the behavior of networks is very attractive and has been used successfully in some important cases (cf. Section 2b). It has also been apparent from the outset that not all computations are expressible as energy minimizations. Certainly any behavior that requires a loop or cycle cannot be described by a monotonic goodness function. The system cannot be made to return systematically to the same state while continuously (or probabilistically) increasing the goodness measure. In this section we discuss some other limitations on the computation that arise from the goodness paradigm.

Recall the key assumptions underlying the formulation:

1) linear threshold elements (sometimes with stochastic component);

2) symmetric weights; and

3) an instantaneous asynchronous updating rule.

We first show that linear threshold elements not restricted to symmetric weights on connections are universal, i.e., can compute any computable function. This is trivial because one can easily make a complete set of binary logic devices (e.g., OR and NOT) from binary linear threshold devices. An OR unit has its two weights as one and its threshold as .5.

$$X_{OR} = d_1 + d_2 - .5$$

A negation unit has a weight of -1 and a threshold of -.5.

$$X_{NOT} = -d + .5$$

This is well known and hardly surprising, but serves as a baseline for further study. Notice that universality does not mean that a particular network of LTE will suffice to realize a behavior. There are both computational and biological reasons for using more general elements. With the symmetry condition added, there are many computations that cannot be done at all.

This is true even for the case of asymmetric links of the same sign that occurs routinely in practice. One obvious instance is when the weights represent evidential links (e.g., conditional probabilities) between nodes. These are normally of the same sign, but of different values. For example, the likelihood (in Figure 1.1) of T_2 given the word "IT" is much greater than the likelihood of the word given only the letter position. It has been claimed that one can routinely eliminate asymmetric weights in binary networks by changing the thresholds in one of the units. A typical case is the transformation suggested by Figure 2.1, where $\theta_3 = \theta_2 + (w_1 - w_2)$. This only holds when the unit B has no other inputs; otherwise the changed threshold can obviously effect the behavior of unit B when A is silent. And for the continuous valued model, threshold revision doesn't even work when B has no other inputs. It may be possible in some cases to eliminate an asymmetric weight by substituting more complex symmetric constructions, but this does not seem to be a promising avenue to pursue. Similarly, results showing that isolated boolean functions can be realized with SLTE [Hinton and Sejnowski, 1983b] should not be read to imply that arbitrary boolean networks can be so realized. We will examine below the question of whether there might be a generalization of goodness that would accommodate asymmetric weights of the same sign.

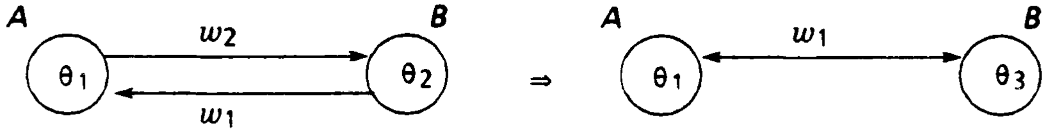


Figure 2.1

For subnetworks with weights of opposite sign (Figure 2.2), the situation is even worse. Suppose the symmetric weight chosen to replace w_1 and w_2 were positive. Then the effect of $B = 1$ on unit A would be the opposite of what it was in the original network. This will only lead to an equivalent computation when the original network had w_2 so small as to have no effect on A. Notice also that the asymmetric situation violates our intuitive notions behind the goodness measure. The idea of mutual consistency doesn't make sense between two nodes with links of opposite sign. This suggests that it might be difficult to find an extension of the goodness/energy paradigm to networks with links of opposite sign. Notice that such a network is the natural way to implement sequencing, unit A activating B and being silenced by it in turn.

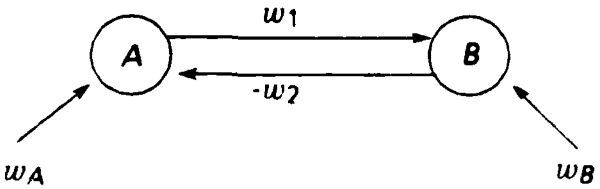


Figure 2.2

Another assumption of the standard SLTE/goodness model is that no unit has a link to itself or retains any "memory" of its previous values (hysteresis). This is natural for the thermodynamic situation, but is not appropriate for neurons. There are also a number of computational problems that become easier when memory across firings is permitted. For example, the usual way to prevent regional maxima problems like those of Figure 1.2 is to have units accumulate evidence internally for some time before outputting values that might short-circuit the computation [Sabbah, 1985]. Another use of unit memory would be to support models that did not require the inputs to be clamped on, i.e., to implement what is called a "latch" in circuit design. Fortunately, there does seem to be a systematic modification of the goodness paradigm that will accommodate unit memory for some versions of the continuous output model. The idea is to augment any unit requiring memory with an auxiliary linked only to it by some positive weight, w.

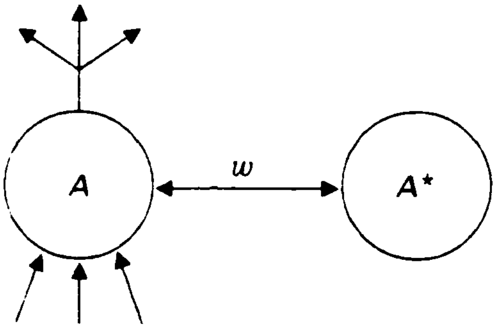


Figure 2.3

The conventional goodness model has the output of A^* always available to unit A, so A^* acts as internal memory. If the model being modified has output of units equal to their potential, the value w can be set to 1. If the memory is intended to decay with time, the weight w can be set to less than 1. In the event that output is not equal to potential ($X \neq P$), the memory of P can be approximated by using a value of w and a threshold in A^* to approximate memory. The relation between output and memory input for A is:

$$\text{memory input} = w\,(w \cdot \text{output} - \text{threshold}).$$

In addition, one would probably want to arrange the simulation so that A^* was evaluated between evaluations of A without exception. To the best of my knowledge, no one has tried this. It may also be possible to use a different goodness function that incorporates self-links, but I have not found a way to do so.

While it is not possible to design loops within the SLTE restriction, one can design sequences. The idea is to use successively smaller weights and thresholds so that activation proceeds in sequence (cf. Figure 2.4). Thus the starting input value of 80 is greater than the first unit's threshold of 70, but the backward linking output of 60 is not. The sequencer would work by the input with weight 80 being turned on for some time and then stopped. Assuming the simulation timing worked out, the first unit (threshold 70) would fire and activate whatever it was controlling as well as the next sequencing unit. If the scale of values is much greater than the values used in the controlled network, the symmetric links back to the sequencer will have no effect. This construction is presented for intuition only and is not suggested for use. A practical sequencer would need either a timer or a completion signal from the controlled network signaling the time to step ahead. A timer requires a loop and cannot be built with SLTE units. A completion signal would be symmetric and would restart the completed action unless great care was taken.

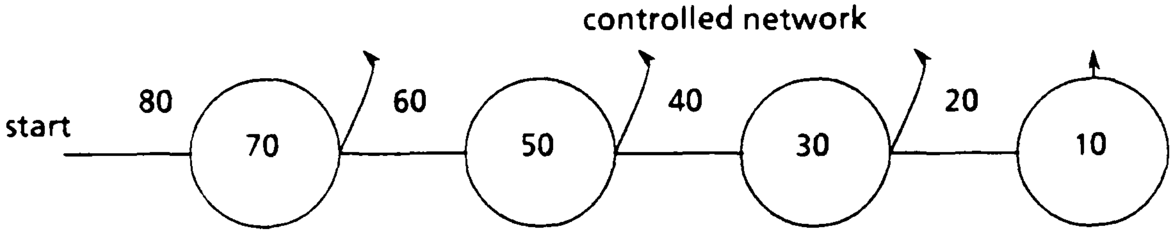


Figure 2.4

Thus far we have seen that the symmetric weight requirement imposes several computational restrictions, but that asymmetric linear threshold units are universal. The notion of increasing goodness could be extended to networks with unequal weights of the same sign, but not without a heavy price. The crucial step in going from (5) to (6) in the derivation of monotonicity exploited symmetric weights and, without them, the effect on global goodness of a local change cannot be computed at that unit. There are some applications where having the system compute the global effect of a change might be effective, but this takes us out of the range of connectionist models. Even so, one could not extend the goodness paradigm to networks with links of opposite sign.

One can also question the possibility of using symmetric weights but non-linear rules of combination. Many investigators have used non-linear rules such as product, maximum, or logic functions to combine inputs, and there is no biological reason to assume only linear combinations. Now the goodness derivation depends in a crucial way on both the unit combination rule and on summation as the global measure of goodness. There is a natural extension to conjunctive connections that are symmetric in all three directions using an update rule analogous to (2), namely:

$$(11) \qquad P_i = \sum_{kj} w_{kji} x_k x_j - \theta_i$$

and the appropriate energy function [Hinton, personal communication]. It is not at all obvious how to extend the goodness model to more general combination rules while preserving the crucial property that the global effect on goodness be computable at the unit contemplating change. The derivation of local computability depends heavily on the form of the combination function and no direct extension to functions like maximum or logic functions goes through.

In addition to the computational restrictions like those described above, the goodness/SLTE paradigm imposes severe control limitations on models. For example, one scientific goal in the McClelland and Rumelhart work behind our first example (and other models) was to explain why some letter identifications were faster than others. The whole question of computational time becomes problematical in the goodness formulation. The standard version relies upon random asynchronous activation. instantaneous transmission, and potentially unbounded memory at each incoming site for the activation level of its source. Versions that employ simulated annealing add another level of time variation in using several separate settlings of the system. Quite aside from the electronic or biological plausibility of these assumptions, there does not appear to be a mapping to psychological time. A different and promising treatment of time in an energy *cum* annealing paradigm is the bipartite harmony model of Smolensky [1986], as depicted in Figure 2.5. The current version uses BSLTE, a goodness function, temperature, and annealing, but restricts connectivity to a bipartite graph. The idea is that each unit in the lower layer connects only to units in the upper layer, and vice-versa. The advantage of this is that the system can be shown to have good convergence properties (and can be simulated) with *synchronous* updating. This can be viewed as inserting a layer of synchronization units between levels of a tree of representational units. This and some other possible modifications to the paradigm will be discussed in Section 2c, after we look at some successes of the methodology.

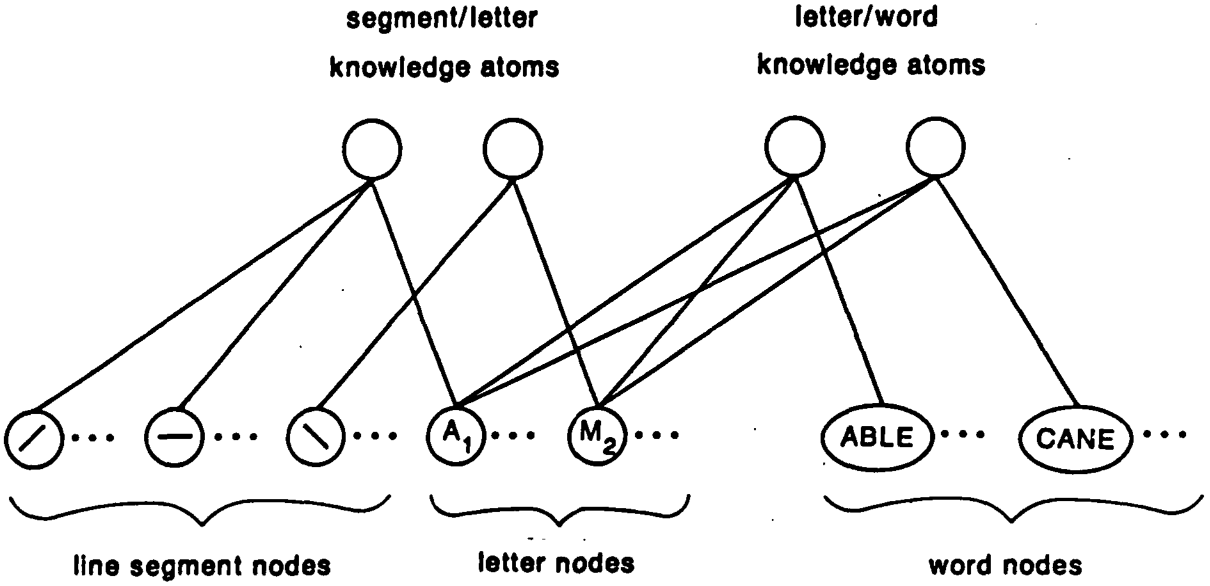


Figure 2.5: Bipartite Harmony Network for Words (from [Smolensky, 1986])

One question that arises about all these limitations is: do they matter? Perhaps the SLTE formalism can compute all functions of interest, or a learning mechanism can produce good approximations to any important computation. Computer science has extensive experience with underpowered formalisms, and the prognosis is not good for attempts to get by with one. A particularly clear example arises in the case of formal grammars, where finite-state grammars (FSGs) play a role analogous to SLTE networks. There are all sorts of lovely decidability and optimality theorems for FSGs, and some practical problems where they are clearly the method of choice (cf. Section 3b). However, for problems with a slightly richer structure (e.g., matching parentheses), FSG techniques fall apart. Any attempt to construct or learn an FSG for an inappropriate language leads to ever larger approximate models that (necessarily) fail to incorporate the structure of the domain. It is true that investigators who favor the energy paradigm tend to believe in (although not necessarily to use) massively distributed representations, but I fail to see how this could effect the basic computational limitations of the technique.

2b. Some Success Stories

We have seen that many computational problems have no formulation in goodness/energy terms. The problems that have thus far been handled effectively by goodness techniques all fall into the general category of constrained optimizations. The additive goodness function, uniform combination rule, and symmetric weights comprise a natural vocabulary for expressing problems in which the individual choices have degrees of compatibility and where the best solution maximizes the sum of these individual measures. Several important questions have this character and, for some of these, goodness formulations have provided valuable insights. Smolensky [1986] suggests that mutual compatibility (harmony) is exactly the domain of goodness methods. With this restriction on the domain, Smolensky can present a coherent story covering all of my behavior criteria. The correct answer to a compatibility problem is (by definition) the maximum entropy solution, and this is found (with high probability) by an annealing schedule. Other work has focused on solving problems for which the answer is specified by external criteria.

Both the word recognition and WTA problem discussed earlier can be at least partially understood in goodness terms. For discrete assignment problems, the fit can be even better. A simple example can be found in [Rumelhart and McClelland, 1986], where the Necker cube is treated as a problem of assigning to each vertex a labeling such as "front lower left." Each vertex has two mutually incompatible labels, and these partition into mutually compatible subsets in the obvious way. However, by plotting the goodness function over the various values for the number of units on in the two alternative coalitions, one can get a much better feel for the structure of the computation (Figure 2.6). The local peaks correspond to interpretations that are not three-dimensional objects.

The Necker cube example was one where the goodness map functioned as an aid for a problem whose solution was known. There are some other cases in which the goodness/energy formulation is central. Perhaps the most interesting of these is the recent work of Hopfield and Tank [1985] on the Traveling Salesman problem (TSP). The TSP is to find the shortest path through a graph in which each node is visited exactly once. This is clearly a constrained optimization problem, and one of considerable theoretical and practical importance. A key trick in the Hopfield and Tank formulation is to represent the solution as a binary square matrix with a 1 only in the entry corresponding to the sequential position in the tour (column) of a node (row). Thus an acceptable answer is a binary matrix with a single 1 in each row and each column. An energy function is constructed which is much better for such configurations than for other binary matrices. This is done with strong mutual inhibition links in the usual way. Finally, another term representing the total distance of a tour is added to the energy function. The problem is now in the form where minimizing energy would provably yield the shortest tour.

Another interesting aspect of this paper is the method used to "search for" the energy minimum. Even though the final answer has only binary entries in the assignment matrix, Hopfield and Tank allow the corresponding units to take on any

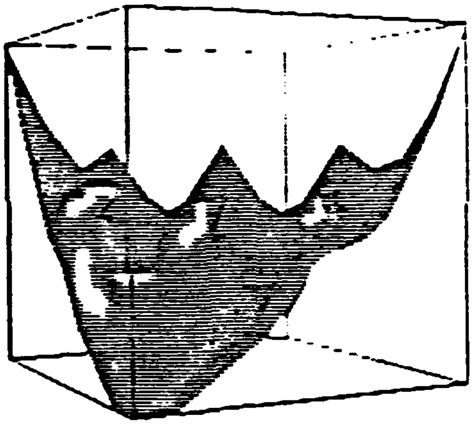


Figure 2.6: The goodness of fit surface for the Necker cube network (from Rumelhart et al., 1986]). The low point at the (0,0) corner corresponds to the start state. The peaks on the right and left correspond to the standard interpretations of the cube.

value between 0 and 1. They view the binary matrix as corners of a phase space and the continuous values as specifying interior points. One can also view the approach as avoiding local minima by moving more cautiously through energy space than strictly binary units would permit. This idea was illustrated in Section 1 and was also used in the Necker cube example above. It is interesting that Hopfield and Tank employ a sigmoid function to map potentials to output values: this tends to push high and low values to the extremes, but is more nearly linear in the middle. With some additional adjustments, like adding small random bias to initial values to break symmetry, the system turns out to do pretty well, and would be very fast on appropriate hardware. The paper discusses performance and a number of heuristics that might improve it. Some of the assumptions of the model are oriented around electronic circuits and are questionable for neurons; we will discuss this further in Section 4.

Another important paper whose success is at least partially due to energy considerations is that of [Geman and Geman, 1984]. They are concerned with the restoration of gray-scale images that have been corrupted by noise of a known statistical character. Given this knowledge and the statistical character of the domain, the best Bayesian restoration is the image which would most likely have led to what we started with. The problem is that this maximum a posteriori (MAP) estimate has been computationally intractable. Geman and Geman exploit the equivalence of Markov Random Fields and the Gibbs distribution in thermodynamics to convert the MAP problem into one of minimizing an energy functional. Using essentially the same ideas as we discussed earlier, they show that local decisions which reduce energy are possible. The results, within the limited domain approached, are quite good.

We briefly discussed simulated annealing in the energy paradigm in Section 1. The general idea of simulated annealing in optimization problems has attained considerable success. Some formulation of problems in vision, such as the Gemans' work (above) and [Poggio et al., 1985], result in constrained maximization problems, and simulated annealing will be one of the solution techniques employed. But most of this work is not concerned with connectionist models. I will briefly discuss the biological plausibility of annealing in Section 4. As a computational construct, annealing has been primarily used to avoid regional optima, and doesn't add any functionality to the models. One potentially interesting exception to this generalization arises in Selman's Master's thesis [Selman, 1985; Selman and Hirst, 1985].

Selman was concerned (like several others) with building a connectionist parser for context-free grammars. One nice aspect of Selman's work is an automatic method of constructing the connectionist parser from a limited context-free grammar. Figure 2.7 shows a tiny grammar and the resulting network. The construction algorithm also supplies weights and thresholds and is one of several elegant new model builders that will be discussed in Section 4. For our current purposes, the interest centers on the mutually inhibitory "binder" nodes labeled 1-4 in Figure 2.7. The idea is that only one of the three alternative constructions for VP can be present in a sentence. (The three alternatives are represented by four binders for technical reasons.) One could build a deterministic parsing network using enough nodes and memory or state within the computational units [Cottrell, 1985a; Waltz and Pollack, 1984]. What Selman does instead is to minimize the number of units in his network and employ an annealing schedule to find good matches. The interesting aspect of this is that annealing is playing a role similar to the sequential search and backtracking of conventional parsers. Unfortunately, the Master's thesis only involved simple examples, and it is not clear how far the analogy will take us. The thesis itself [Selman, 1985] contains interesting discussions of design decisions and other considerations concerning the energy paradigm, and some possible modifications.

Some of the greatest successes with the goodness/energy paradigm have come in studying abstract computational questions, particularly learning. Learning is a central issue of intelligence and is particularly important for connectionist models, which reject the notion of an interpreter and a store of encapsulated knowledge. It turns out that SLTE systems, particularly the binary Boltzmann machine, have remarkable properties of adaptation and learning. These have been explored in a series of papers by Hinton, Sejnowski, and their collaborators [Ackley et al., 1985; Derthick, 1984; Sejnowski et al., 1985].

The learning algorithm for Boltzmann machines again is based on the average occupancy rate of various states when the system has reached equilibrium. Learning is designed to produce a system that will spontaneously produce the same statistics in input/output units as was used in training. If later some partial input is specified, the system will generate maximum entropy estimates of the unspecified inputs.

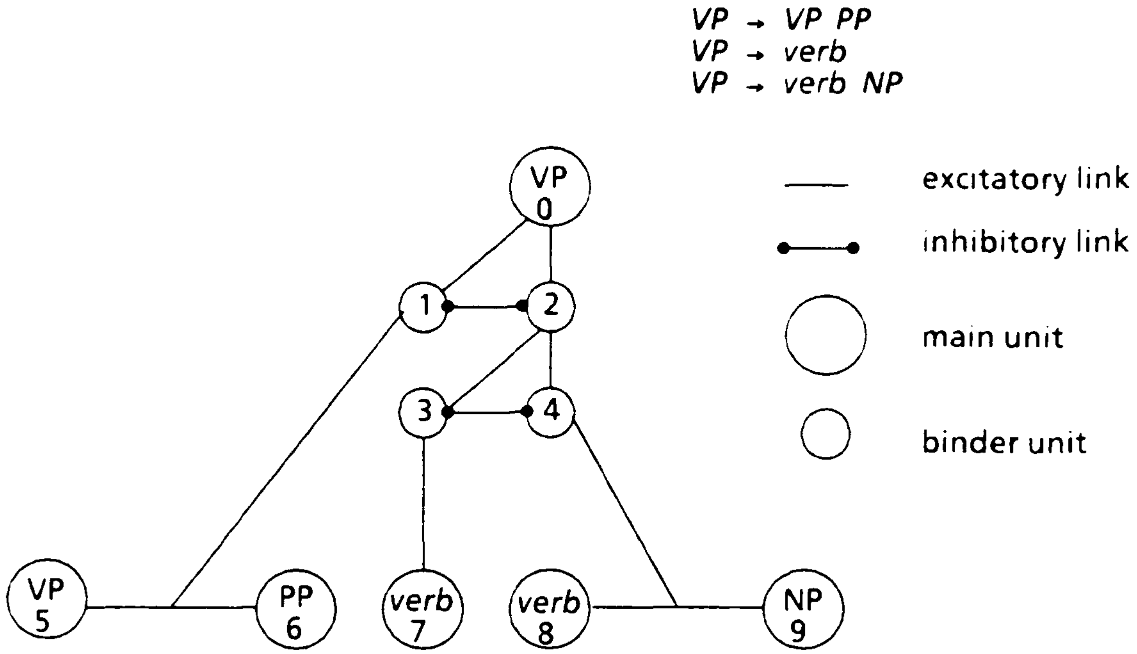


Figure 2.7: A Small Grammar and its Network (from [Selman, 1985])

Given that one accepts this model of learning, a very powerful local learning rule obtains. Since relative energy among various states determine their frequency, any errors in frequency must be due to some weights in the network being set sub-optimally. But the energy function is a sum of local energy functions at each unit, which is in turn a sum of contributions by each link. Therefore one can use the following simple update rule. Let p_{ij} be the probability of units i and j both being on in the clamped situation and p'_{ij} be the joint probability when the system is free running. Then the weight-updating rule:

$$\Delta w_{ij} = \varepsilon \, (p_{ij} - p_{ij})$$

where ε is a scale factor, will converge to the correct expected values. The proof of this is given in [Hinton et al., 1984] and uses partial derivative techniques like our treatment in the introduction. A distributed version of this algorithm would require that the system go through successive cycles of stochastically accurate training and free-running simulation. The free-running state has been likened to dreams (cf. also [Crick and Mitchison, 1983; Hopfield et al., 1983]). The value of the learned weight p_{ij} would have to be stored (for each connection) during the simlation runs and recalled for normal activity. Again, reasonable men can differ on the plausibility of all this.

One way of looking at these results is to notice that the Boltzmann learning algorithm will eventually converge (in the appropriate sense) for any function computable by a Boltzmann machine, assuming enough units and connections are available. Of course, the range of learnable functions is limited, but it does provide an excellent vehicle for theoretical study. Very recently some powerful results on learning in more general networks have begun to appear [Rumelhart et al., 1985; Parker, 1985]. Although they do not employ SLTE, energy, or annealing, there is a clear intellectual link with the Boltzmann formulation. The course of true science does not run smooth.

2c. Some Possible Extensions

The results of the previous section show that there are some modeling problems for which the SLTE goodness/energy formulation can be directly applied to good effect. All of the ones discovered so far are optimal assignment problems with symmetric constraints and an additive quadratic objective function. In this section we will look at the question of regional and global maxima, of classes of problems that might fit directly into the paradigm, and of modifications that might extend its applicability.

The results of the previous sections enable us to provide a clear answer to the primary question raised in the paper. Linear threshold elements, symmetric weights, and asynchronous updating support a powerful paradigm in which termination and sometimes correctness can be established. They do not, however, come close to covering the range of computational problems of interest in connectionist models. The situation is no different than it is with any other mathematical formalism; it requires considerable judgement to decide whether a problem might fit into the energy paradigm and how best to do it. To start a modeling project on the assumption that it must be made to fit the energy model is to court disaster. Of course, one must also consider whether any connectionist model is appropriate for the task at hand.

For those with appropriate background, the analog with statistical mechanics can be quite suggestive. Smolensky [1986] has the best treatment of this. What one must realize is that statistical mechanics is most powerful in systems lacking significant structure. For structured domains, e.g., protein folding, statistical considerations play a minor role. It seems to me that cognition is much more like the interaction of complex structures than an ideal gas. Moreover, the symmetric weight (interaction strength) condition does hold in complex physical situations but not in many cognitive domains. The physics model would have to be expanded greatly to explicate the nature of parallel computation. On the other hand, the general metaphorical notion of a parallel system settling into a stable state could have lasting value.

The principal attraction of the energy paradigm is the property that it is guaranteed to converge. To ensure that a network will deterministically converge to the right answer, one must also show that the right answer has the lowest energy and that the energy surface is convex (uni-modal). We assume for now that the former is accomplished, and examine the latter question. It should be clear from the Necker cube example (Figure 2.1) that it will not generally be feasible to construct a uni-modal energy or goodness map. This does not mean that networks cannot be proven to converge, only that the simplest methods do not suffice. The most common way of addressing the problem of regional minima has been the use of random perturbation, usually becoming less random as the computation progresses. This idea of "annealing" is taking its place in the general repertoire of combinatorial optimization techniques [Aragon et al., 1985], but does not seem to be a useful idea for connectionist models, for several reasons, some of which were outlined earlier. The annealing process has been much too slow and uncertain to form the computational basis for basic neural perception, action, etc..

Given a problem that will fit the SLTE energy model, there are a variety of other ways to combat the problem of regional optima. One method was discussed in connection with Hopfield and Tank's TSP network, where the amount of descent (in the steepest direction) was varied. Of course, no such deterministic technique will escape an energy valley, but it should work fairly well for neutral (non-perverse) initial conditions.

Another interesting possibility arises from making a virtue of the transmission delay present in neurons but missing from all the models. If, in a real or simulated system, we allow units to be evaluated with information that might be out of date, this will have the effect of adding noise. If one assumes that the probability of outdated information is random (as would occur with random choice of units to activate), then there will be random noise. Moreover, as Francis Crick first pointed out, the noise will be greater in the early stages of computation when things are changing a lot, and will be less later in the computation; thus simulating simulated annealing. Some preliminary experiments by Terry Sejnowski along these lines have had moderate success. It will also be interesting to see how the bipartite synchronous formulation [Smolensky, 1986] extends to more complex problems.

Another way of improving the pure energy method is to develop the analogy to multi-grid and resolution hierarchy techniques. The idea there is to first solve a (spatially) crude approximation to the optimum and then refine it. One way to carry this idea over to the SLTE world is to put in extra units that explicitly represent areas of the solution space and establish heavily weighted inhibitions among them. In an ideal case one could even make the solution space uni-modal.

One major problem with the energy method is that (except for noise) it must improve the metric on each and every calculation. One trivial, but powerful, modification would be to allow some fore-play period of time in which energy was ignored. One way to realize this idea is to use the state variable, Q, from our original definition. One could allow enough time for information to spread through the network and then have units switch to a state where the energy minimization became operative. In addition, one could restrict the energy minimization to a selected subset of the units. The early phases and non-energy units could have asymmetric links and would presumably be characterized by different methods, like those of the next section. Recall that the whole idea of the method is to show convergence and, hopefully, correctness. There is no reason why these results can't be established considering part of a network for a restricted part of the computation. In fact, these are some of the kinds of techniques that have been used to construct connectionist models in the past. A number of other alternative ideas for specifying and proving the behavior of parallel systems are discussed in the next section.

3. Other Specification and Verification Techniques

3a. Evidential Reasoning

There are, of course, many ways of specifying computations and verifying the networks that carry them out. One useful idea is to specify the desired computation in another computational language and then show its equivalence with a particular network. Ideally, one would like to be able to do this generally for a class of problems. For example, one could try to devise a "compiler" from standard [Ballard and Hayes, 1984] or non-monotonic [Cottrell, 1985b] logic to connectionist networks and prove that it always yields the right results. No one has succeeded in doing this, but some interesting results have been achieved. In a more restricted domain, Fanty [1985] has a system that will produce a provably correct parser (over fixed length input) for any context-free language.

Some of the best results in network specification have been achieved with evidential reasoning systems. There are a number of conceptually similar formalisms for specifying how pieces of evidence should be combined to reach the "best" conclusion. We could view our first example as a specification (via the network) of rules for combining evidence for letters (e.g., I_1, T_2) to compute the likelihood or confidence of the appearance of words. There is a natural correspondence between levels of evidence, probability, etc., and the levels of activation in network models. This has been obvious from the outset and appears to have been understood by Freud in his (abandoned) neural net project [Pribram and McGill, 1976]. Certainly the "spreading activation" models of cognitive science [Collins and Loftus, 1975] had an implicit evidence theory.

More recently, there have been a number of efforts to examine formal theories of evidence as computational engines, and some of this effort has been done using network models. Formal network models embodying Bayesian [Pearl, 1985] and Dempster/Shafer [Lowrance, 1982; Wesley et al., 1984] evidence theories have been studied. These networks require complex calculations at each node and do not fall directly into our topic, but many of the ideas carry over to connectionist models. A recent thesis by L. Shastri [1985] carries out a detailed analysis of connectionist networks for solving inheritance and categorization problems using an evidence theory derived from maximum entropy. We will examine this effort as another paradigm for producing connectionist models with provable behavior.

Shastri's model is a conventional hierarchical semantic network augmented with relative frequency information (cf. Figure 3.1). The network is required to compute the most likely answer to inheritance queries (e.g., pacifism of DICK) and categorization problems (e.g., who is a pacifist and Republican). Shastri argues that the appropriate notion of "most likely" derives from maximum entropy. This is not relevant to our discussion and, in the cases considered, conventional Bayesian inference would give identical results. The relevant point is that a tight specification of the required behavior preceded the design of the network.

The specification of the semantic network that exhibits the required behavior and the proof that it does so are given in [Shastri, 1985]. We will just try to convey the

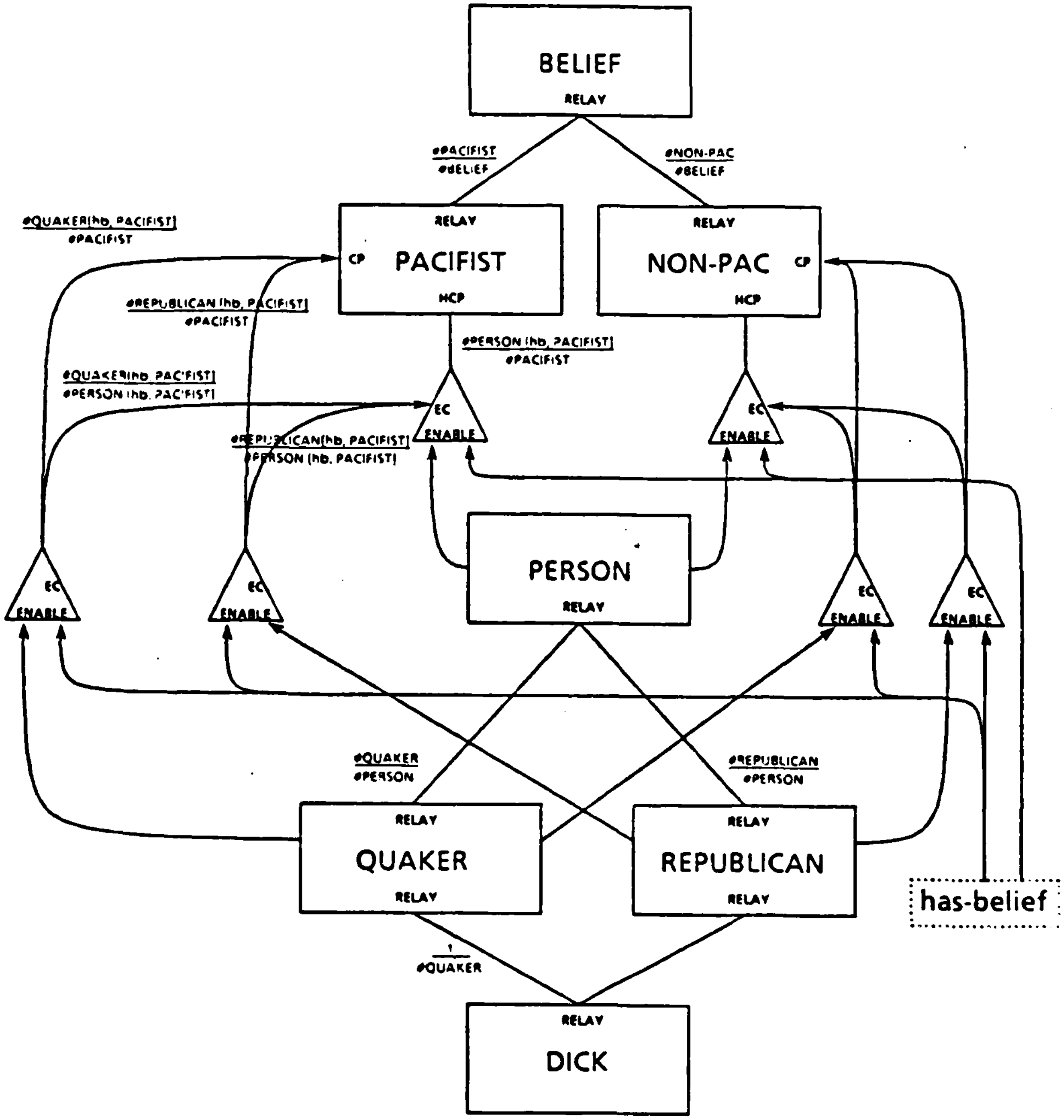


Figure 3.1: An example of inheritance (from [Shastri, 1985]).
All inputs incident at side enable of δ-node have a weight of 1.0
Not all sites and weights have been shown. hb = has-belief.

computational flavor of the design and the correctness proof for inheritance. Shastri's problem is made easier by the fact that his sub-networks have no negative links or loops. The simulation rule is parallel--all units recompute at each cycle. Convergence is assured because activation spreads in a controlled fashion to the top of the hierarchy and back down. Explicit control units (triangular nodes in Figure 3.1) are designed into the network as in conventional logic design. Different "enable" and "relay" units are activated, depending on the type of query being processed. It is then fairly straightforward to show that activation will flow through the network in a controlled fashion and that the results of a query will stabilize in time proportional to the depth of the semantic hierarchy.

Given that the network will converge, it remains to show that it will yield the correct answer. Shastri's evidence model specifies that a query such as "Do Republicans tend to be pacifists or not?" be answered by comparing two ratios:

$$L(Republican \Rightarrow pacifist) = \frac{\#\ Republican\ pacifists}{\#\ pacifists} \times \frac{\#\ pacifists}{\#\ people\ with\ beliefs}$$

and

$$L(Republican \Rightarrow non-pacifist) = \frac{\#\ Republican\ non-pacifists}{\#\ non-pacifists} \times \frac{\#\ non-pacifists}{\#\ people\ with\ beliefs}$$

The remainder of the proof consists of showing that networks like that in Figure 3.1, with the various ratios as weights and appropriate (multiplicative) combining rules, do produce activation levels proportional to the required L values. This gets somewhat complicated for multiple and conflicting inheritance questions, but this is due to the nature of the calculation more than its network implementation. Shastri then goes on to show how the same network, units, and weights will also answer categorization questions (e.g., "Who is a Republican pacifist?"), using different enabling rules.

The Shastri results also provide the clearest instance of why evidential considerations have been at the core of much of the success of connectionist models. From an abstract computational viewpoint, there is no reason why a parallel formulation should have descriptive advantages over a serial specification such as a set of rules. Two factors, one of them incidental, has led to the descriptive success of connectionist models. It turns out that categorical (all-or-none) rules are not appropriate for describing many cognitive tasks and that some sort of evidential or probabilistic reasoning is required. This has no inherent connection to parallelism, as Shastri clearly shows. Where parallelism does come in is that it permits (and encourages) the simultaneous consideration of all the relevant evidence. The sequential application of individual probabilistic rules does not capture the same notion. Of course, Shastri's problem is enormously simplified by the assumption that the evidential network has no cycles. The general case involves "relaxation" and problems analogous to the problem of regional optima in goodness/energy models.

There are some attractive and some problematical aspects of these results. The idea of exploiting a powerful theory of the domain (here, maximum likelihood evidence theory) is clearly of great importance to connectionist models. Many other recent efforts have used linguistic theories to specify networks (e.g., [Dell, 1985; Cottrell. 1985a; Selman, 1985]). Shastri is the first to prove formally that his networks realize the theory, and it is important to see how this can be done. On the other hand, there is something overly rigid about the precision with which the network is designed and controlled. An earlier version of Shastri's model [Shastri and Feldman, 1984] seems much more natural, but it had no underlying theory and was intractable. Another problem, not directly relevant here, is that the frequency ratios currently require far more precision than one can reasonably expect in nature. We will say more about the biological plausibility of various models in Section 4.

3b. Finite Automata and Related Methods of Computer Science

We will now consider, for the first time, the symbolic state variable Q that was included in the original unit definition (1). There are excellent biological and computational reasons for allowing a unit (neuron) to have different computational rules that apply in varying circumstances. Fatigue and habituation are two simple state-dependent phenomena. There is good reason to suspect that learning and perhaps dreaming involve distinct computational states. At the biochemical level, the discovery of a rich variety of peptide modulators has also strengthened the notion that an unchanging computational device is not the best model of a neuron. The state variable in definition (1) provides a formal mechanism for incorporating state dependence without much bias as to how it is used.

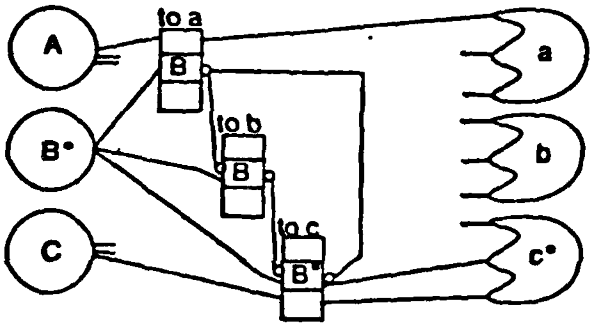
From a computational point of view, finite-state machines (FSMs) provide a well explored formalism for specifying and verifying the behavior of networks. One major advantage here is that the FSM formulation allows one to formalize systems that have cyclic behavior--which none of the previously discussed techniques can. Another use of FSM techniques is in defining discrete behavior transitions which are harder to see in a continuous formulation. We will present an example from (the errata sheet of) [Feldman, 1982] that uses FSM techniques in two related ways.

The problem here is to develop a model of short-term memory that can quickly (without weight change) learn to connect paired associates, e.g., (B-c), in an originally uniform network. The idea is to use dedicated inter-units, shown as rectangular arrays, to represent each pairing. The behavioral requirements are:

1) high inputs to both ends of a pair activate the inter-unit;
2) this activation will decay in the absence of new input;
3) while the inter-unit is active, an input to either associate will lead to activation of the other.

A major complication arises if one wants the network to support several associations simultaneously without crosstalk. This requires WTA networks among the inter-units, and these are indicated by inhibitory links as in Figure 3.2.

The FSM tables for the end and inter-units given in Figure 3.2 are essentially the complete specification of the network. For example, the second entry in the first row specifies that an idle inter-unit, upon receiving input from both ends, will switch to *high* and block its rivals. Columns headed by a dash describe what happens in the absence of input, and X denotes impossible situations. It is possible to formulate a one-state, continuous potential version of this network, but it is much harder to describe and prove its behavior. Moreover, the blocked state of inter-units conveys the intended idea that inputs are ignored better than very large negative weights. A similar, but more complex, use of discrete states can be found in the work of Sabbah [1985]. In this case, the discrete states with different computational rules appear to be necessary to make the system work.



Inter-unit

	One-end	Dual	Block	—
Idle	Low	High Block	Blocked	
Low	High Block	High Block	Blocked	Idle
High	(Low)		X	Low
Blocked		X		Idle

End-unit

	Start	From inter	—
Idle	Low	Low	
Low	High	High	Idle
High		(Low)	Low

Figure 3.2: State and Output Tables
for Dynamic Connections

Thus far in the paper, all of the systems studied have converged to an answer configuration, at least probabilistically. Many computations do not have this character at all, but rather cycle through various states. Obvious examples of this include rhythmic behavior like walking or breathing and, in computation, systems like computer networks and operating systems. Continuous methods for cyclic systems basically lie in the domain of control theory and will be discussed briefly in Section 3c. In computer science, people have developed a variety of techniques, often based on FSM, for specifying and proving properties of cyclical systems [Filman and Friedman, 1984]. Much of the detailed specification efforts have come in the area of distributed systems where the designer must protect against perverse timing effects and lost, garbled, or duplicated messages. A major concern in the analysis of networks of computing devices is to show that the system never enters certain erroneous states. Typical problems include deadlock, where each unit is awaiting information from another, and lockout, where one unit is precluded from operating. There is some similarity to the issue of regional optima, and the design and analysis techniques may carry over. In particular, the issue of behavior that is proof against communication glitches has received considerable attention. This question of robustness over vagaries of timing and communication is important in connectionist modeling and will recur in Section 4. A recent and potentially interesting development is the use of the module-message paradigm from distributed computing [Filman and Friedman, 1984] as a specification language for connectionist models.

3c. Control Theory

The computational formalism with the longest and strongest record of success in modeling neural systems (as well as many other applications) is control theory. There is ample evidence that negative feedback control loops are widely used in nature, and conventional linear control theory has proved to be very useful in analyzing them [Arbib, 1975; Cannon and Robinson, 1985]. Almost all of this work has been with "lumped" models where no direct map to neurons and firings is provided. There have been some attempts (e.g., [Addanki, 1983]) to develop neural-level, connectionist models of motor behavior, and conventional control theory will continue to play a major role in these efforts. There is also some recent work directed at bringing lumped models closer to neural behavior [Cannon and Robinson, 1985], and there is hope that these efforts will converge.

A more general question concerns the role of mathematical control theory in exploring the behavior of connectionist models. Since control theory is well developed for linear systems with loops, one might hope that it would complement goodness theory, which deals with linear systems without loops. The difficulty is that most of the interesting behavior in connectionist models occurs at the saturation limits of the units and not in their linear range. We will illustrate this with a very simple example: the 2-unit WTA network of Figure 1.4.

The mutual inhibition network appears to be a simple feedback system that should be easy to analyze using control theory, but there are several difficulties. For one thing, the upper and lower saturation limits of our units makes linear systems theory problematical. We could try to examine the behavior of the system in its linear range, using the unbounded variant of equation (12):

$$(12) \qquad X_A = d_A - wX_B$$

$$X_B = d_B - wX_A$$

If we attempt to treat these equations literally (with $X_A(t)$ and $X_B(t)$ as functions of time), bizarre results obtain. One can substitute the expression for $X_A(t)$ in the second equation of (12) and solve for $X_B(t)$, yielding

$$(13) \qquad X_B(t) = \frac{d_B - wd_A}{1 - w^2}$$

First of all, $X_B(t)$ is constant, which certainly violates the WTA condition , as well as our intuition. And the value of the constant approaches ∞ as w nears 1, the value that works for the WTA network. The basic difficulty here is that equation (12) is really a shorthand notation for a mutual assignment statement, not a true functional relation that should hold for all times (t). A more accurate rendering would be

$$(14) \qquad X_A(t) = d_A - wX_B(t-\varepsilon)$$

$$X_B(t) = d_B - wX_A(t-\varepsilon)$$

where ε is a small transmission delay. From this we can write down the differential equations (using the chain rule):

$$(15) \qquad \dot{X}_A(t) = -w\dot{X}_B(t-\varepsilon)$$

$$\dot{X}_B(t) = -w\dot{X}_A(t-\varepsilon)$$

Equation (15) at least conveys the intuitions that the outputs of the two units change in opposite directions and that larger w yield faster change. One could proceed from here by employing some approximation of $\dot{X}_A(t\text{-}\varepsilon)$ as a function of $\dot{X}_A(t)$ and solving the resulting system. But, even ignoring the saturation bounds, there does not seem to be any point to it. Control theory is excellent at exploring the detailed temporal characteristics of linear systems, but the questions of interest in most connectionist models are not of this form.

What all this suggests is that (at least conventional) control theory is not an appropriate vehicle for studying the detailed behavior of most connectionist models. The major exception is in the study of motor control both at the system level and as we move to more detailed neural models. It does appear that (at least the subcortical) neurons involved in motor control do operate largely in their linear range and that units that saturate are "replaced" by ones operating in a higher range. But for most connectionist models of perception, representation, inference, etc., there are better formalisms. For cyclic behavior in these domains, the finite state methods of the previous section appear to be the best current choice.

4. Related Issues and Conclusions

Computational models involving very large numbers of simple computing units are certain to be the subject of considerable effort for some time. Animal nervous systems clearly have this character, and neuroscience is rapidly advancing to the point where computational models (theories) are a central aspect of the field. People are already building computer hardware designs that allow for a million or more simple processors [Hillis, 1985]. At the abstract level, many investigators have found massively parallel formulations of their problems more effective than descriptions in more traditional computational formalisms. Even on computers without enormous parallelism, algorithms expressed in connectionist language may prove to be efficient to write and run.

With all of this existing and potential achievement, the deep understanding of massively parallel computation takes on increasing importance. No one is satisfied with a large network having ad hoc units, connections, and simulation rules. Progress in all aspects of massively parallel computation will depend on systematic treatment of underlying computational questions. The formal analysis and proof techniques discussed in this paper represent one line of approach to structuring connectionist systems. A few points that transcend these particular methods are pursued in this section.

One major source of confusion (not surprising in this early stage) is that the scientific goals of a particular modeling effort are often not clear. For many of us, the ultimate goal is detailed models of intelligent behavior that are directly testable down to the single neuron level. At present, this perfectly explicit goal provides direct guidance on modeling requirements only for the simplest systems. Essentially all cognitive level models deal at a level of abstraction at which units of the model cannot be identified with neurons or groups of them or parts of them. The question becomes: what should "biologically plausible" mean in this setting? My view of this is that there are basic computational constraints which are sufficient to keep the models reasonable. The foremost of these are the restrictions on the number of units and connections, the processing and information transfer rate limitations, and the absence of an interpreter that can operate above the level of the network. An important additional consideration, which is often ignored, is that the simulation itself be robust over variations in exact values transmitted and timing of updates. Neither the purely synchronous or purely asynchronous model is biologically plausible, and we do not have good tests for the robustness of models over variations. None of the proof techniques discussed in the paper take this question seriously. We have done some simulations using a synchronous rule with random noise added to each output. This appears to be a robust and biologically realistic methodology, but nothing is known of its formal properties. Insightful recent discussions of the biological role of connectionist models can be found in [Ballard, 1986] and [Sejnowski, 1986]. There has been some work in computer science on asynchronous systems with variable timing. The problems encountered have been sufficiently difficult to establish a strong bias towards synchronous computers and to restrict inherently asynchronous systems, such as networks, into very simple modes of processing. Nevertheless, it does seem possible to develop connectionist models that are robust over simulation details.

Consider, for example, another variation on the Winner-Take-All task discussed earlier. Suppose each unit in the WTA network got input from all the others and computed the maximum activity level of its rivals. If each unit simply turned itself off when it saw a rival of greater activity, the WTA property would obtain. Moreover, this construction will work for any reasonable simulation method. A variant of this network that does not require N^2 connections was used in [Shastri and Feldman, 1984]. Each rival fed its output to a single MAX unit which fed back to all the rivals. The units computed their new output by subtracting MAX from their previous output; again this is stable over all reasonable simulation strategies. There is no compelling evidence on the biological plausibility of a MAX function, but it will be surprising if nature did not evolve something of this sort to solve computational problems like those suggested by the WTA example.

In addition to biological motivations, several groups studying massive parallelism are interested in hardware realizations. The physical constraints here are quite different. Speed of computation and transmission become less critical and the number and length of connections much more so. The robustness of simulations also takes on a different flavor, getting into the standard problems of synchronous and asynchronous circuits. Another important issue is the physical plausibility of models that assume continuously valued output functions. Such models are biologically realistic for only the restricted range of problems involving non-spiking neural signals. For most neural processing of interest in cognitive science, the limited range of neural firing frequencies limits outputs effectively to a few bits, and this has major consequences in modelling. For electronic circuits, the continuous output assumption is fine, and this may turn out to change our ideas of "digital" computation. It is almost certainly a mistake to take the same model as an engineering proposal and a description of neural functioning.

In addition to the robustness and physical realizability issues discussed above, there are important questions about connectionist networks as expressive formalisms. While it is clear that the connectionist framework is the right *kind* of formalism for many tasks, the understandability of a particular model is less obvious. For example, the McClelland and Rumelhart model [1981] was helpful in understanding a number of experimentally important effects, but did not incorporate a great many known constraints on the structure and interaction of lexical items. There is a danger that connectionist models can grow so complex that it is difficult or impossible to recognize the governing principles behind the behavior. Notice that if some system learned such a network for itself, we would be in an even worse position to extract the principles of organization. The specification and proof techniques discussed in this paper can only help if the abstract specification itself contains manifest information characterizing the structure of the problem.

This general idea of a higher-level description which captures the structure of a domain is one of the most promising recent developments in connectionist modeling. We saw in Section 2b how Selman transformed a restricted context-free grammar into a connectionist parser. The point, of course, is that we can consider the linguistic adequacy of the grammar independent of its realization. Cottrell [1985a] has indicated how automated construction could be employed to cover semantic case-roles of words in English. There appears to be no problem, in principle, prohibiting the development of a "compiler" that will convert a natural language specification (including relative frequencies, etc.) into a connectionist recognizer. Were we able to do this, it would constitute a major scientific advance, because it would allow the direct test of linguistic models against psycholinguistic data. Once we developed confidence in the underlying implementation and translation, differences between predictions and findings could be traced to the theory. The role of the connectionist realization would be a functioning map from competence to performance. Returning to the current paper, a critical step in this ambitious program would be to prove that some translation realized the given formal specification.

In a simpler domain, this is just what Shastri (Section 3a) was able to do for his evidence theory. In this case the input to the system is just semantic network knowledge with relative frequency information on properties, values, and sub-types. Given this input, Shastri's constructor builds a connectionist network that produces optimal answers to inheritance and categorization queries, by the maximum entropy criterion. The point, here again, is that the theory of evidence and the knowledge base structure and content can be studied independently of the realization. And again, the connectionist implementation is claimed to be sufficiently realistic to support direct behavioral tests.

Formal specification and proof techniques will become increasingly important as connectionist modeling matures. The current flurry of interest in energy models is motivated by the right goals, but the formalism itself is too weak to carry us very far. Professional theoretical computer scientists are beginning to take a serious interest in some of these problems and are providing valuable insights [Valiant, 1985; Goldschlager, 1984]. Automatic learning is a central issue and can be fruitfully studied independently of domain. For direct modeling of intelligent behavior, the greatest promise appears to lie with methods for automatically translating formal specifications into connectionist networks.

Acknowledgements

The connectionist clique communicates continually, and many of the ideas in this paper arose in conversation. Valuable assistance with the current version has come from Gary Cottrell, Gary Dell, Mark Fanty, Steve Small, Mandayam Srinivas, Terry Sejnowski, and Paul Smolensky.

References

Ackley, D.H., G.E. Hinton, and T.J. Sejnowski, "A learning algorithm for Boltzmann machines," *Cognitive Science 9*, 147-169, 1985.

Addanki, S., "A connectionist approach to motor control," Ph.D. thesis, Computer Science Dept., U. Rochester, 1983.

Andrews, G.B. and F.B. Schneider, "Concepts and notations for concurrent programming," *Computer Survey 15*, 1, 3-43, March 1983.

Aragon, C.R., D.S. Johnson, and L.A. McGeoch, "Optimization by simulated annealing: an experimental evaluation," manuscript in preparation, 1985.

Arbib, M.A., "Artificial intelligence and brain theory: unities and diversities," *Annals of Biomedical Engineering 3*, 238-274, 1975.

Ballard, D.H., "Cortical connections and parallel processing: structure and function," TR 133, Computer Science Dept., U. Rochester, revised January 1985; to appear, *Behavioral and Brain Sciences*, 1986.

Ballard, D.H. and P.J. Hayes, "Parallel logical inference," *Proc., Sixth Annual Conf. of the Cognitive Science Society*, 286-292, Boulder, CO, June 1984.

Barnden, J.A., "On short-term information-processing in connectionist theories," *Cognition and Brain Theory 7*, 1, 25-59, 1984.

Bienenstock, E., "Dynamics of central nervous system," *Proc., Workshop on Dynamics of Macrosystems*, Laxenburg, Austria, September 1984; to be published by Springer-Verlag, J.P. Aubin and K. Sigmund (Eds), 1985.

Cannon, S.C. and D.A. Robinson, "An improved neural-network model for the neural integrator of the oculomotor system: more realistic neuron behavior," *Biological Cybernetics 53*, 1-16, 1985.

Cohen, M.A. and S. Grossberg, "Absolute stability of global pattern formation and parallel memory storage by competitive neural networks," *IEEE Trans. on Systems, Man, and Cybernetics SMC-13*, 5, 815-826, September/October 1983.

Collins, A.M. and E.F. Loftus, "A spreading-activation theory of semantic processing," *Psychological Review 82*, 407-429, November 1975.

Cottrell, G.W., "A connectionist approach to word sense disambiguation," TR 154 and Ph.D. thesis, Computer Science Department, U. Rochester, May 1985a.

Cottrell, G.W., "A model of lexical access of ambiguous words," *Proc., Natl Conf. on Artificial Intelligence*, Austin, TX, August 1984.

Cottrell, G.W., "Parallelism in inheritance hierarchies with exceptions," *Proc., 9th Intl. Joint Conf. on Artificial Intelligence*, 194-202, Los Angeles, CA, August 1985b.

Cottrell, G.W. and S.L. Small, "A connectionist scheme for modeling word sense disambiguation," *Cognition and Brain Theory 6*, 1, 89-120, 1983.

Crick, F. and G. Mitchison, "The function of dream sleep," *Nature 304* (5922), 111-114, 1983.

Dell, G.S., "Positive feedback in hierarchical connectionist models: applications to language production," *Cognitive Science (Special Issue on Connectionist Models and their Applications)*, pp. 3-23, January-March 1985.

Derthick, M., "Variations on the Boltzmann machine learning algorithm," CMU-CS-84-120, Computer Science Dept., Carnegie-Mellon U., August 1984.

Fahlman, S.E. *NETL: A system for Representing and Using Real-World Knowledge*. Cambridge, MA: The MIT Press, 1979.

Fahlman, S.E., "Three flavors of parallelism," *Proc., Fourth Natl. Conf. of the Canadian Society for the Computational Studies of Intelligence*, Saskatoon, Saskatchewan, May 1982.

Fanty, M., "Context-free parsing in connectionist networks," TR 174, Computer Science Dept., U. Rochester, November 1985.

Feldman, J.A., "Connectionist models and parallelism in high level vision," TR 146, Computer Science Dept., U. Rochester, January 1985; *CVGIP 31, Special Issue on Human and Machine Vision*, 178-200, 1985a.

Feldman, J.A., "Dynamic connections in neural networks," *Biological Cybernetics 46*, 27-39, 1982.

Feldman, J.A., "Four frames suffice: a provisional model of vision and space," *Behavioral and Brain Sciences 8*, 265-289, June 1985b.

Feldman, J.A. and D.H. Ballard, "Computing with connections," in A. Rosenfeld and J. Beck (Eds). *Human and Machine Vision*. Academic Press, 1983.

Feldman, J.A. and D.H. Ballard, "Connectionist models and their properties," *Cognitive Science 6*, 205-254, 1982.

Filman, R.F. and D.P. Friedman. *Coordinated Computing*. New York: McGraw-Hill, 1984.

Fukushima, K., S. Miyake, and T. Ito, "Neocognitron: a neural network model for a mechanism of visual pattern recognition," *IEEE Trans. on Systems, Man, and Cybernetics SMC-13*, 5, 826-834, September/October 1983.

Geman, S. and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. on Pattern Analysis and Machine Intelligence 6*, 6, 721-741, November 1984.

Goldschlager, L.M., "A computational theory of higher brain function," technical report, Computer Science Dept., Stanford U., April 1984.

Hammerstrom, D., D. Maier, and S. Thakkar, "A proposal for developing cognitive architectures based on connectionist models," Computer Science and Engineering, Oregon Graduate Center, 1985.

Hillis, D.W. *The Connection Machine*, Cambridge, MA: The MIT Press, 1985.

Hinton, G.E. and T.J. Sejnowski, "Analyzing cooperative computation," *Proc., Fifth Annual Conf. of the Cognitive Science Society*, Rochester, NY, May 1983a.

Hinton, G.E. and T.J. Sejnowski, " Optimal perceptual inference," *Proc., IEEE Conf. on Computer Vision and Pattern Recognition*, Washington, DC, 1983b.

Hinton, G.E., T.J. Sejnowski, and D.H. Ackley, "Boltzmann machines: constraint satisfaction networks that learn," TR CMU-CS-84-119, Computer Science Dept., Carnegie-Mellon U., May 1984.

Hopfield, J.J., "Neural networks and physical systems with emergent collective computational abilities," *Proc., Natl. Acad. Sciences USA 79*, 2554-2558, 1982.

Hopfield, J.J., "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc., Natl. Acad. Sci. 81*, 3088-3092, May 1984.

Hopfield, J.J., D. Feinstein, and R.G. Palmer, "Unlearning has a stabilizing effect in collective memories," *Nature 304*, p. 158, 1983.

Hopfield, J.J. and D.W. Tank, "'Neural' computation of decisions in optimization problems," to appear, *Biological Cybernetics*, 1985.

Kirkpatrick, S., C.D. Gelatt, Jr., and M.P. Vecchi, "Optimization by simulated annealing," *Science 220*, 4598, 671-680, 1983.

Koch, C., T. Poggio, and V. Torre, "Nonlinear interactions in a dendritic tree: localization, timing, and role in information processing," *Proc., Natl. Acad. Sciences USA 80*, 2799, 1983.

Lowrance, J.D., "Dependency-graph models of evidential support," Ph.D. thesis, Dept. of Computer and Information Science, U. Mass., 1982.

McClelland, J.L. and D.E. Rumelhart, "An interactive activation model of context effects in letter perception, Part 1: an account of basic findings," *Psychological Review 88*, 375-407, 1981.

Metropolis, N. A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, "Equation of state calculation by fast computing machines," *Journal of Chemical Physics 21*, 1087, 1953.

Minsky, M. and S. Papert. *Perceptrons*. Cambridge, MA: MIT Press, 1969.

Mjolsness, E., "Fingerprint hallucination with neural networks," working document, Computer Science Dept., California Inst. of Technology, December 1984.

Parker, D.B., "Learning-logic," TR 47, Center for Computational Research in Economics and Management Science, MIT, 1985.

Pearl, J., "Fusion, propagation and structuring in Bayesian networks," TR CSD-850022, Cognitive Systems Laboratory, U. California, Los Angeles, April 1985; also presented at the *Symposium on Complexity of Approximately Solved Problems*, Columbia U., April 1985.

Poggio, T., V. Torre, and C. Koch, "Computational vision and regularization theory," *Nature 317*, 26 September 1985.

Pollack, J.B. and D.L. Waltz, "Natural Language processing using spreading activation and lateral inhibition," *Proc., Fourth Annual Conf. of the Cognitive Science Society*, 50-53, Ann Arbor, MI, August 1982.

Posner, M.I. *Chronometric Explorations of Mind.* Hillsdale, NJ: Lawrence Erlbaum Associates, 1978.

Pribram, K.H. and M. McGill. *Freud's Project Re-assessed.* New York: Basic Books, 1976.

Quillian, M.R., "Semantic memory," in M.L. Minsky (Ed). *Semantic Information Processing.* Cambridge, MA: MIT Press, 1968.

Ratliff, K. and F. Hartline. *Studies on Excitation and Inhibition in the Retina.* New York: The Rockefeller U. Press, 1974.

Riley, M.S. and P. Smolensky, "A parallel model of (sequential) problem solving," *Proc., Sixth Annual Conf. of the Cognitive Science Society*, 286-292, Boulder, CO, June 1984.

Rumelhart, D.E., G.E. Hinton, and R.J. Williams, "Learning internal representations by error propagation," ICS Report 8506, Institute for Cognitive Science, U. California, San Diego, September 1985.

Rumelhart, D.E. and J.L. McClelland, "An interactive activation model of context effects in letter perception, Part 2: the contextual enhancement effect and some tests and extensions of the model," *Psychological Review 89*, 60-94, 1982.

Rumelhart, D.E. and J.L. McClelland (Eds). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations.* Cambridge, MA: Bradford Books/MIT Press, 1986.

Rumelhart, D.E., P. Smolensky, J.L. McClelland, and G.E. Hinton, "PDP models of schemata and sequential thought processes," in D.E. Rumelhart and J.L. McClelland (Eds). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations.* Cambridge, MA: Bradford Books/MIT Press, 1986.

Sabbah, D., "Computing with connections in visual recognition of Origami objects," *Cognitive Science 9*, 1985.

Sejnowski, T.J., "Open questions about computation in cerebral cortex," in McClelland, J.L. and D.E. Rumelhart (Eds). *Parallel Distributed Processing:*

Explorations in the Microstructure of Cognition. Vol. 2: Applications. Cambridge, MA: Bradford Books/MIT Press, 1986.

Sejnowski, T.J. and G.E. Hinton, "Separating figure from ground with a Boltzmann machine," in M.A. Arbib and A.R. Hanson (Eds). *Vision. Brain. and Cooperative Computation.* Cambridge: MIT Press, 1985.

Sejnowski, T.J., P.K. Kienker, and G.E. Hinton, "Learning symmetry groups with hidden units: beyond the perceptron," manuscript in preparation, 1985.

Selman, B., "Rule-based processing in a connectionist system for natural language understanding," TR CSRI-168 and Master's thesis, Computer Systems Research Institute, U. Toronto, April 1985.

Selman, B. and G. Hirst, "A rule-based connectionist parsing system," *Proc., 7th Annual Conf. of the Cognitive Science Society*, 1985.

Shastri, L., "Evidential reasoning in semantic networks: a formal theory and its parallel implementation," Ph.D. thesis and TR 166, Computer Science Dept., U. Rochester, September 1985.

Shastri, L. and J.A. Feldman, "Evidential reasoning in semantic networks: a formal theory," *Proc., 9th Intl. Joint Conf. on Artificial Intelligence*, 465-474, Los Angeles, CA, August 1985.

Shastri, L. and J.A. Feldman, "Semantic networks and neural nets." TR 131, Computer Science Dept., U. Rochester, June 1984.

Shepard, R.N., "Ecological constraints on internal representation: resonant kinematics of perceiving, imagining, thinking, and dreaming," *Psychological Review 91*, 417-447, 1984.

Sivilotti, M., M. Emerling, and C. Mead, "A novel associative memory implemented using collective Ccomputation," *Proc., Chapel Hill Conf. on VLSI*, 329-342, May 1985.

Small, S.L., G.W. Cottrell, and L. Shastri, "Towards connectionist parsing," *Proc., Natl. Conf. on Artificial Intelligene*, 247-250, Pittsburgh, PA, August 1982.

Small, S.L., L. Shastri, M.L. Brucks, S.G. Kaufman, G.W. Cottrell, and S. Addanki, "ISCON: a network construction aid and simulator for connectionist models," TR 109, Computer Science Dept., U. Rochester, April 1983.

Smolensky, P., "Foundations of harmony theory: cognitive dynamical systems and the subsymbolic theory of information processing," in D.E. Rumelhart and J.L. McClelland (Eds). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations.* Cambridge, MA: Bradford Books/MIT Press, 1986.

Smolensky, P., "Schema selection and stochastic inference in modular environments," *Proc., Natl Conf. on Artificial Intelligence*, Washington, DC, 1983.

Touretzky, D.S. and G.E. Hinton, "Symbols among the neurons: details of a connectionist inference architecture," *Proc., Int. Joint Conf. on Artificial Intelligence*, August 1985.

Valiant, L.G., "Learning disjunctions of conjunctions," *Proc., 9th Intl. Joint Conf. on Artificial Intelligence*, Los Angeles, CA, August 1985.

Waltz, D.L. and J.B. Pollack, "Phenomenologically plausible parsing," *Proc., Natl. Conf. on Artificial Intelligence*, 335-339, Austin, TX, August 1984.

Wesley, L.P., J.D. Lowrance, and T.D. Garvey, "Reasoning about control: an evidential approach," Technical Note 324, AI Center, SRI International, 1984.

Wolfram, S., "Cellular automata as models of complexity," *Nature 311*, 419-424, 4 October 1984.

END

DTIC

8–86