A major purpose of the Technical Information Center is to provide the broadest dissemination possible of information contained in DOE's Research and Development Reports to business, industry, the academic community, and federal, state and local governments.

Although a small portion of this report is not reproducible, it is being made available to expedite the availability of information on the research discussed herein.

1

LA-UR -86-1288

CONF-8605108--1

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36

LA-UR--86-1288

DE86 010183

TITLE MACHINE LEARNING USING A HIGHER ORDER CORRELATION NETWORK

AUTHOR(S) Y. C. Lee, Gary Doolen, H. H. Chen, G. Z. Sun, Tom Maxwell, and H. Y. Lee

SUBMITTED TO Proceedings of the "Evolution, Games, and Learning" Conference held at Los Alamos, May 20-24, 1986

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

The Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy

Los Alamos

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

MACHINE LEARNING USING A HIGHER ORDER CORRELATION NETWORK

Y. C. Lee and Gary Doolen
Center for Nonlinear Studies
Los Alamos National Laboratory
Los Alamos, NM 87545

and

H. H. Chen, G. Z. Sun, Tom Maxwell, and H. Y. Lee
University of Maryland

Abstract

A high-order correlation tensor formalism for neural networks is described. The model can simulate auto associative, heteroassociative, as well as multiassociative memory. For the autoassociative model, simulation results show a drastic increase in the memory capacity and speed over that of the standard Hopfield-like correlation matrix methods. The possibility of using multiassociative memory for a learning universal inference network is also discussed.

1. Introduction

Since World War II, scientists have tried to develop techniques which would allow computers to behave more like human beings. The research effort, which usually goes by the fanciful name of "Artificial Intelligence", typically involves abstract problem solving, decision making, planning, machine learning, and natural language understanding. Essential to most AI research efforts is the heavy reliance of high level symbolic manupulation tools which favor serial computation. In fact, it is probably fair to say that the majority of the AI researchers believe that since most high level information processing taking place in our brains are of the "serial terminating search" type, it would be quite unnecessary to resort to massive parallelism to duplicate human intelligence. This should not be construed to mean that AI workers do not believe that parallel computing can play a role in the implementation of AI techniques in hardware. On the contrary, most computer scientists in AI research will readily acknowledge that parallel computing can solve the bottlenecks associated with many kinds of recognition and pattern matching problems that occur in AI. However, the consensus is that when it comes to knowledge processing, parallelism is a poor substitute for sequential heuristic search. The major drawbacks in the traditional AI methodology are their extreme brittleness and the enormous amount of manpower usually required. To wit, even the most complex and the most sophisticated piece of AI software which is capable of performing impressive tasks in its specialized domain will "crash" as soon as it is taken beyond the scope originally contemplated by its designers. On the other hand, most four-year old children can cope with strange or unexpected situations, without having been programmed by an entire army of AI designers!

A drastically different approach to machine intelligence has been attempted by neural modelers who believe that massive parallelism, distributed information storage and associative interconnections, as suggested from biological evidence, should be the base upon which to construct intelligent devices. With the big advances in fast microchip technology, such a connectionist approach has been gaining support among noncomputer-scientists. It can be argued, however, that while certain neural models can indeed perform relatively low level cognitive functions like pattern recognitions and associative memory admirably well, no higher level functions such as planning, decision making, and understanding semantics have been successively demonstrated by this approach. Nevertheless, the idea remains an attractive one, especially from the point of view of nonlinear dynamical systems. Imagine the complex interaction of a huge number of neuron-like processors passing signals to one another. It is not hard to believe that the system will, except in extreme situations, evolve in a most unpredictable way. That this must be so can be inferred from the by now well-established fact that even in a much simpler nonlinear dynamical system of much lower dimensionality there can be sensitivity to both the initial state and the parametric dependence in the most extreme manner; so much so that this dynamical system can be said to behave in an unpredictable fashion. Note that typically AI researchers try to design systems in an algorithmic manner so that the software so produced can usually be proven to perform exactly as prescribed. Any degree of unpredictability that infrequently shows up in the program is usually considered to be a case of malfunctioning and accordingly are treated with great disdain. Indeed, the point of view of wanting the system to perform useful intelligent functions seems incompatible with unpredictability. Yet the behavior of a human being is not exactly predictable either. Still we can count on another human being to perform certain tasks for us. Hence with certain limits, unpredictability does not necessarily mean unreliable performance. In fact, our own free will can be traced back to the unpredictability of our mind. This by no means implies that the free will of the human being can be simulated by merely introducing a large dose of randomness externaly into the system, just as we cannot argue that the turbulent ocean waves have any more "free will" than that of the human being simply because they are more unpredictable.

The main characteristic difference between the thought process of the human brain and the motion of the turbulent ocean seems to come from the ability of the human brain to store, process, and retrieve information. Try to feed information to the ocean waves and it is quickly lost. Many physical systems driven far from thermal equilibrium also display memory effects. However, these tend to be of very short time scale so they are more akin to the short term memory phenomenon of the human brain than the long term memory one, the latter being of much greater importance to higher level intelligence of humans. One physical system capable of long term memory is the Ising spin glass model, its relationship to the influential model of Hopfield having already been pointed out by numerous authors [1]. Another spin-glass-like model with nonMarkovian spin-spin interaction functions has been demonstrated by Fukushima [1a] to be capable of spatio-temporal associative memory. The information in both cases can be stored by modifying the nonlocal spin-spin interaction strength (to model the neuronal synaptic strength) according to Hebbian rules which, in effect, "learn" patterns by performing weighted summation of exterior products of pattern vectors. The latter can be shown to be the binary correlation functions of the pattern vectors in the statistical mechanical sense. The major drawback of the binary correlation function models can be seen from the fact that if we desire spatial translational invariance (temporal translational invariance is automatic, absolute time simply has no meaning), then the binary correlation functions are simply the Fourier transform of the power spectrum. Hence all information about the angular dependence of the individual Fourier component of the pattern vectors is totally lost. The lack of angular dependence severely limits the capability of such systems to perform pattern discrimination tasks. In fact, the memory capacity of the Hopfield model is extremely low, being of the order of $n/4\log n$ in most cases, where n is the total number of neurons.

The preceding discussion is mostly pertinent to the so called "autoassociative memory" modelling [1]. In the heteroassociative case [1], the research is usually directed toward classification of patterns according to their membership in respective equivalence classes. Here the application of Hebbian learning rules leads to the "correlation matrix" method [1,2]. Unfortunately the correlation matrix can only store the "average" pattern vector of any given equivalence class. It certainly will not learn the equivalence class of, say, your grandmother, because the "average" of grandma's pattern vector can only be a big blurr, quite indistinguishable from that of your pet octopus, for example.

In this paper, we propose to improve the memory capacity as well as to enhance the pattern discriminating capability of the neural system by the introduction of higher order tensorial memory functions (correlations). In so doing, we depart significantly from the Hebbian doctrine. Since there does not seem to be any neurophysiological evidence to support the assumption that the synaptic connections are higher order in nature, we will focus our attention on whether such a system can display intelligent behavior. The possibility of implementing higher order correlation by using hidden neurons will be discussed in a separate paper. We consider neural models which are capable of spatial as well as spatio-temporal associative memory. For the spatial memory problems, the memory pattern vectors are stored in the form of stable fixed points of neural dynamics, each with a large basin of attraction. A large basin of attraction means that even relatively noisy or incomplete information can be used to evoke the full memory. The memory recall process consists of evolving the neural system in discrete time steps from the starting state which corresponds to the incomplete pattern information vector until it converges to a fixed point.

One of the most striking aspects of the higher order correlation models is that even when relatively large sets of patterns are stored, it usually takes no more than one discrete time step for the initial vector to converge to the proper fixed point, thus making the high order scheme very attractive from the point of view of computing efficiency. For the spatio-temporal models, a multi-associative neural network is used together with a shifting operator. Since the pattern vectors are now functions of the time, it is no longer possible to use the time axis for error correcting iterations. Fortunately, because of the excellent convergent properties of the higher order scheme, it is possible to introduce a very small number of intermediate time steps (two or three are usually enough) to allow each pattern vector to iterate several times before the next pattern vector comes in. For the more realistic situation where the successive pattern vectors stay close to one another, additional iteration steps can be gained because the basins of attraction of the individual points of the attracting orbit tend to merge together, forming a single basin, thus allowing the state vector to converge toward the attracting orbit adiabatically. Multiassociative memory can also be used as both a spatial and a spatio-temporal pattern classifier. The latter clearly has its application in speech recognition.

The multiple attractor model as described basically implements the "learning from positive examples" paradigm. However, perhaps just as important is the "learning from negative examples" paradigm, which can be implemented by inserting repellers that correspond to negative example pattern vectors into the neural dynamics. Note that the introduction of repellers is very different from the "unlearning negative example" paradigm, which simply corresponds to the eradication of a previously learned pattern attractor but does nothing to prevent the same mistake from being made in the future. Weak repellers can also be employed to perform minor surgery (sculpting) of the basins of attractions without affecting the basins of neighboring attractors [3].

The learning paradigms encountered so far can be broadly categorized as "rote" learning, or learning by memorization. In fact, with few exceptions, most of the neural network reserach to date can be said to belong to this general category. The main drawback of this approach is that, while it is possible in theory for the system to memorize all possible situations that the system is likely to encounter and the appropriate responses therein, it would require a tremendous amount of memory storage and the computing power. For example, in the case of learning chess-playing skill, it simply does not make any sense to try to memorize the responses to every possible move that the opponent is likely to throw at you! Nor, in pattern recognition problems, is it reasonable to store every conceivable profile of your grandma's face! The predominant view in the AI community on these matters is that the difficulties associated with information storage and processing can be solved by using information compression methods utilizing feature detectors, thus rendering it possible to process the reduced information on a higher level using symbolic processing techniques.

The problem with the AI approach is that the selection of features for detection, the actual implementation of feature extraction, and the heuristic rules for symbolic processing of the abstract features all have to be programmed explicitly, a monumental task. For example, even though we've all learned to recognize our own grandma's face, it is hard to determine exactly what set of feature detectors we use in our brains to enable us to make the proper identification. The point is that the processing associated with feature detection are all done at very low levels, below our conscious level. Because of our inability to recognize our own feature-detecting subsystems, it is very difficult to transfer our own experience into the design of proper software and/or hardware tools to implement the feature-space paradigm. In our attempt to overcome these obstacles, we propose the following alternative: instead of trying to find out how to detect salient features, we can simply overload the neural network with patterns well beyond the capacity of the neural system to store them without causing severe mutual interference. Simple arguments borrowed from statistical mechanics can be used to convince us that what actually gets stored in the neural dynamics are no longer the individual patterns since these tend to get washed away through mutual destructive interference of input pattern vectors, and all that remains is, in some sense, certain representations of the statistical or causal invariants of the ever-changing environment, much in the same way the holograms are constructed.

The preceding remark should not be taken to mean that our approach is holographical in nature. Far from it, our proposal does not require treating the neural network as a optical-grade media, nor is there any need to introduce coherent wavefronts. About the only thing in common between these two approaches is the point we just mentioned. In our higher-order correlation model, the statistical/causal invariants are encoded into the higher-order spatio-temporal correlation connections directly. It is clear that since binary correlation models can only learn binary correlations, by definition they have about as little predictive ability as that of the binary correlation functions of fully developed fluid turbulence, once the memory is overloaded. It is also clear that, the higher the order of the interconnections, the more complex are the correlations the network can learn. The ultimate limitation of this approach is in hardware implementation, which generally favors lower order interconnection.

The saturation learning paradigm just presented affords a tremendous compression of information content because it eliminates the bulk of redundant noninformation which does not have any causal consequence and is therefore devoid of any contextual meaning. Note also that because of the nature of the correlation functions in a highly varying environment, signal patterns tend to get fused into localized (both spatially and temporally) chunks and are stored as such. The coherent spatio-temporal information chunks are in many ways reminiscent of the soliton structures in nonlinear continuum systems. Furthermore, since there are interconnections (or nonlinear coupling coefficients), it follows that the neural network will behave like a multi-adaptive filter which is sensitized to incoming signals containing only those information chunks. In fact, with proper thresholding, it is possible for the neural network to allow only those signals having the proper chunks to pass through for further processing. Obviously the threshold behavior itself must also be adaptive. For example, a threshold can be lowered by just "paying attention". Otherwise no new information will be learned. Often, though, new information can be just a new combination of already chunked information. Thus we can identify the localized invariant spatio-temporal chunks to be the output of some sort of local feature detectors (filters).

The concept can be generalized even further. For example, let us assume that there is another layer of slower neural network which monitors the highly thresholded output from the first layer (the adaptive feature-detecting filter layer) so that only the strongly resonant filtered feature chunks can get through to the second neural layer. Then the feature chunks now become the smallest irreducible information units for the input signals of the second layer. By analogy, the intercorrelation among various feature units can be further integrated into higher level chunks, thus leading to a even higher level representation of the knowledge. In fact the relationships (both spatial and causal) among different feature units can be reasonably called rules. In consequence, a two-level neural network system can be said to be capable of learning rules. The reason for using a slower (longer time constant) neural network for the second layer is that since the spatio-temporal correlations of the feature units are by definition weaker but may have much longer spatio-temporal range, therefore, only an accumulator with sufficiently long time constant can pick up such weak correlations. The situation is very similar to that in particle physics where very strong correlation generated by the strong force leads to highly localized chunks known as nuclei; the weaker electromagnetic force leads to much longer range but weaker correlations; and the far weaker gravitational force in turn leads to tiny but extremely long-range correlations which can be detected only by integrating (averaging) over macroscopic space-time volume.

Once the network system can be taught rules, it becomes capable of recognizing pattern classes instead of just the individual patterns. This gives the system strong immunity to noisy environment as well as great flexibility. This ability to synthesize rules also makes it possible to design networks which can perform neural programming of almost arbitrary complexity.

2. Fundamental network dynamics

Neural networks are often modeled by approximating neurons by threshold elements. The neurons interact with one another through an interconnection matrix which simulates neuronal synaptic connections. Learning can be achieved through Hebbian-like modification of the interconnection matrix [1,2], which has the effect of either creating new attractors or changing the locations of existing attractors so that the system can produce appropriate responses to a set of external stimuli. More generally, a neural network can be considered to be a nonlinear dynamical system with a large number of degrees of freedom together with a large set of adjustable parameters which in turn are controlled by other nonlinear dynamical equations with very long characteristic time constants. Of course, the importance of interaction between two time scales has long been recognized in physics. Yet none of these physical systems display any behavior which could be characterized as being "intelligence". Clearly, there is more to an intelligent system than just two-timescale interactions. To simulate synaptic plasticity, the long-time subsystem will have to behave in some way like an accumulator. The requirement of large memory capacity seems to dictate the existence of a large number of attractors with nontrivial basins for the short time subsystem. It is not obvious whether or not any sufficiently complicated system having the above mentioned properties can display some sort of recognizable "intelligent" behavior, even if "intelligence" is interpreted in the loosest sense.

The particular model we consider is heavily influenced by a recent article by Hopfield [1]. The main difference being that the interconnection in our model is of tensorial character, rather than the matrix-type interconnection of the Hopfield model. Specifically, we define a Lyapunov (energy) function

$$E = \sum_{\nu_1}^{N} \sum_{\nu_2}^{N} \cdots \sum_{\nu_k}^{N} T_{\nu_1 \nu_2 \cdots \nu_k} S_{\nu_1} S_{\nu_2} \cdots S_{\nu_k} \qquad (1)$$

where $\underline{S} = (S_1, S_2, \cdots S_N)$ is a state vector whose components, $S_j, j = 1, \cdots N$, can only assume the values of 1 or -1, and k specifies the order of the interconnections. The dynamical evolution of the fast subsystem is governed by the following discrete map:

$$S_i^{(n+1)} = W\Big[\sum_{\nu_2 \nu_3 \cdots \nu_k} T^{(n)}_{i, \nu_2 \cdots \nu_k} S^{(n)}_{\nu_2} S^{(n)}_{\nu_3} \cdots S^{(n)}_{\nu_k} \Big] \qquad (2)$$

where $W(x)$ is a step function whose value is 1 whenever $x > 0$ and -1 otherwise, and $S_i^{(n)}$ is just the ith component of the neuron "spin" at the nth discrete time step. The modifiable synaptic interconnection tensor $T_{\nu_1 \cdots \nu_k}^{(n)}$ satisfies the following long time evolution dynamics:

$$T_{\nu_1 \cdots \nu_k}^{(n+1)} = (1-\alpha) T_{\nu_1 \cdots \nu_k}^{(n)} + \alpha \sum_{\mu_1 \cdots \mu_k} D_{\nu_1 \cdots \nu_k}^{\mu_1 \cdots \mu_k} S_{\mu_1}^{(n)} S_{\mu_2}^{(n)} \cdots S_{\mu_k}^{(n)} \tag{3}$$

where $D_{\nu_1 \cdots \nu_k}^{\mu_1 \cdots \mu_k}$ is a positive definite matrix which has the property that all permutations of $\{\mu_1 \cdots \mu_k\}$ and $\{\nu_1 \cdots \nu_k\}$ leave it unchanged. Hence it follows that $T_{\nu_1 \cdots \nu_k}^{(n)}$ is also a symmetrix tensor provided that $T_{\nu_1 \cdots \nu_k}^{(0)}$ is initially symmetric. α^{-1} in eq. (3) is of the order of the characteristic time scale for long term memory. For very small values of α, $T_{\nu_1 \cdots \nu_k}^{(n)}$ can be considered to be essentially constant in time (namely it is independent of n). Thus eq. (2) can be treated as a nonlinear discrete map with constant coefficients, $T_{\nu_1 \cdots \nu_k}$. As a special case, we can take $T_{\nu_1 \cdots \nu_k}$ to be of the form

$$T_{\nu_1 \cdots \nu_k} = \alpha \sum_{P=1}^{m} \sum_{\mu_1 \cdots \mu_k}^{N} D_{\nu_1 \cdots \nu_k}^{\mu_1 \cdots \mu_k} \xi_{\mu_1}^{(P)} \xi_{\mu_2}^{(P)} \cdots \xi_{\mu_k}^{(P)} \tag{4}$$

where $\underline{\xi}^{(P)} = (\xi_1^{(P)}, \xi_2^{(P)}, \cdots \xi_N^{(P)})$ is a pattern vector which corresponds to the Pth input pattern and m is the total number of patterns stored in $T_{\nu_1 \cdots \nu_k}$. Here again, we assume that $\xi_i^{(P)}$ can have values of ±1. With suitable choice of the D tensor, it can be demonstrated that eq. (2) has more than m attractors. Indeed, if we consider the simplest case of D = 1, the energy function E becomes (see eq. (1)),

$$E = \sum_{P=1}^{m} (\underline{\xi}^{(P)} \cdot \underline{S})^k. \tag{5}$$

For sufficiently large values of k, E has very sharp maxima whenever $\underline{S}$ is close to one of the pattern vectors, $\underline{\xi}^{(P)}$, provided that m is of the order of $N^{k-1}/\ell n N$ or less [4], where N is the total number of neurons. (Note that $\underline{S} \cdot \underline{S} = \underline{\xi}^{(P)} \cdot \underline{\xi}^{(P)} = N$).

A very interesting property concerning the discrete dynamical eq. (2) for D = 1 and k=even is that the energy function defined in eq. (5) is a nondecreasing function of time. To prove this, we note that

$$\begin{aligned} \Delta E &\overset{\text{def}}{=} E(\underline{S} + \Delta \underline{S}) - E(\underline{S}) \\ &= k \sum_{\nu_1 \cdots \nu_k}^{N} T_{\nu_1 \cdots \nu_k} S_{\nu_1} \cdots S_{\nu_{k-1}} \Delta S_{\nu_k} + R \end{aligned} \tag{6}$$

where

$$R \overset{\text{def}}{=} \sum_{p=1}^{m} \sum_{j=2}^{k} \binom{k}{j} (\underline{\xi}^{(P)} \cdot \underline{S})^{k-j} (\underline{\xi}^{(P)} \cdot \Delta \underline{S})^j. \tag{7}$$

From eq. (2) and the definition of the W-function, it follows that

$$\Delta S_i \sum_{\nu_2 \cdots \nu_k} T_{i,\nu_2 \cdots \nu_k} S_{\nu_2} \cdots S_{\nu_k} \geq 0. \tag{8}$$

The equality holds only for $\Delta S_i = 0$. From (8) we can see that the first term on the right hand side of eq. (6) must be nonnegative. It turns out that R can also be shown to be nonnegative. To show this, all we need to demonstrate is that the function,

$$f_q(x) \overset{\text{def}}{=} \sum_{j=2}^{2q} x^j \binom{2q}{j} = (1+x)^{2q} - (1+2qx) \tag{9}$$

is nonnegative. Using the method of mathematical induction, we first establish that $f_1(x) = x^2 \geq 0$. Now assume that $f_q(x) \geq 0$ for all integers $q \leq \bar{q}$, we will demonstrate that $f_{q+1}(x)$ is also nonnegative. In fact, we can show that

$$\begin{aligned} f_{q+1}(x) - f_q(x) &= (1+x)^{2q+2} - (1+x)^{2q} - [1 + (2\bar{q}+2)x] + 1 + 2\bar{q}x \\ &= [(1+x)^{2q} - 1][(1+x)^2 - 1] + x^2 \\ &\geq [(1+x)^{2q} - 1][(1+x)^2 - 1] \geq 0, \end{aligned} \tag{10}$$

where we have made use of the fact that $(y^q - 1)(y - 1) \geq 0$ for all nonnegative values of y. Q.E.D.

Although the nondecreasing property of E has been shown only for the case D = 1 and k=even, it is not hard to generalize to cases when the D's are positive definite matrices. Dynamically, this implies that the discrete dynamics governed by eqs. (2) and (5) admits only stable fixed point attractors. This combined with the fact that E contains sharp maxima for $\underline{S} = \underline{\xi}^{(P)}$ implies that the pattern vectors $\underline{\xi}^{(P)}$ are indeed the stable attractors for the neural dynamics. It should be noted that Hopfield first proved that both the asynchronous scheme and this scheme has the hill climbing property for the binary correlation model [2]. Although it can be argued that an asynchronous firing model is probably a more realistic representation for biological neural networks, it should be pointed out that the synchronous scheme has several advantages over the asynchronous one. For example: (a) synchronous maps have the semi-group property in that the composition of two synchronous maps is a synchronous map; this facilitates a group theoretical approach, (b) the synchronous scheme is more amenable to parallel computation, (c) the deterministic nature of the synchronous scheme makes the results easier to interpret (the basins of attraction can be uniquely defined), (d) asynchronous maps can move the state at most one Hamming distance at a time, whereas synchronous maps allow the state to jump more than one Hamming distance at each discrete time step; thus it is harder for the synchronous maps to get stuck at a local maximum. However, for odd order correlation models, the dynamics is no longer strictly hill climbing in the synchronous scheme. In our research, both the synchronous and asynchronous schemes are studied.

The advantages of using repellers to represent negative examples have already been stated in the introduction. Naively, it would seem that a repeller can be created for pattern vectors $\underline{\xi}^{(n)}$ simply by changing the energy function (5) to

$$E = \sum_{p=1}^{m_p} (\underline{\xi}^{(P)} \cdot \underline{S})^k - \sum_{n=1}^{m_n} (\underline{\xi}^{(n)} \cdot \underline{S})^k, \tag{11}$$

since then $\underline{\xi}^{(n)}$ will be minima instead of maxima and therefore will be avoided in any hill climbing algorithm. The trouble with (11) is that even for even order ($k = 2q$) correlation models, the presence of negative weights invalidates the proof of the nondecreasing property for E. In fact if we assume that $\underline{S} \simeq \underline{\xi}$, then for sufficiently large k, E can be approximated by $-(\underline{\xi} \cdot \underline{S})^k$. The discrete dynamic eq. (2) now becomes

$$S_i^{(n+1)} = W[-(\underline{\xi} \cdot \underline{S}^{(n)})^{k-1}\underline{\xi}_i + \text{small remainder}]. \tag{12}$$

Clearly, for $\underline{S}^{(0)} \simeq \underline{\xi}$, we find $\underline{S}^{(1)} = -\underline{\xi}$, which in turn means that $\underline{S}^{(2)} = \underline{\xi}, \cdots etc.$ Thus it is found that the introduction of $-(\underline{\xi} \cdot \underline{S})^k$ in E merely adds a 2-cycle attractor to the neural dynamics, not the repeller as one might have anticipated. This difficulty can be circumvented by modifying (11) in the following way:

$$E = \sum_{p=1}^{m_p}(\underline{\xi}^{(P)} \cdot \underline{S})^k - \sum_{n=1}^{m_n}[(\underline{\xi}^{(n)} \cdot \underline{S})^k + \frac{k}{k-1} \cdot N \cdot C\underline{\xi}^{(n)} \cdot \underline{S})^{k-1}]. \tag{13}$$

The corresponding dynamical equation becomes

$$S_i^{(n+1)} = W\left\{\sum_{p=1}^{m_p}(\underline{\xi}^{(P)} \cdot \underline{S}^{(n)})^{k-1}\underline{\xi}^{(P)} - \sum_{m=1}^{m_n}[(\underline{\xi}^{(m)} \cdot \underline{S}^{(n)})^{k-1} + N(\underline{\xi}^{(m)} \cdot \underline{S}^{(n)})^{k-2}]\underline{\xi}^{(m)}\right\}. \tag{14}$$

Again let us assume that $\underline{S}^{(0)} \simeq \underline{\xi} \in \{\underline{\xi}^{(m)}\}$. Equation (14) can be written as

$$S_i^{(n+1)} = W\{-(\underline{\xi} \cdot \underline{S}^{(n)})^{k-1}\underline{\xi} - N(\underline{\xi} \cdot \underline{S}^{(n)})^{k-2}\underline{\xi} + \text{remainder}\}. \tag{15}$$

This time, the time sequence of $\underline{S}$ becomes

$$\underline{S}^{(0)} \simeq \underline{\xi}, \underline{S}^{(1)} = -\underline{\xi}, \underline{S}^{(2)} \not\simeq \underline{\xi}, \cdots etc. \tag{16}$$

The reason that $\underline{S}^{(2)}$ is no longer near $\underline{\xi}$ is that $(\underline{\xi} \cdot \underline{S}^{(1)})^{k-1}$ and $N(\underline{\xi} \cdot \underline{S}^{(1)})^{k-2}$ exactly cancel each other when $\underline{S}^{(1)} = \underline{\xi}$, consequently the subsequent dynamics is no longer governed by $\pm\underline{\xi}$.

The behavior just described provides a very interesting model for the conditioned avoïdance-escape response to negative stimulus, such as shock. If the subject inadvertently touches an exposed hot wire and suddenly realizes it, the first reaction is to take a step backward and then hesitate before deciding to move in a path which does not intercept the exposed wire. Actually we are getting a little bit ahead of ourselves in the above example, since we have not yet discussed heteroassociative memory which is more appropriate for the above case; neither have we talked about how to insert input into the neural network (the neural net defined by eqs. (2) and (3) is strictly autonomous!).

To address the problem of input, it is necessary to modify eq. (2) as follows:

$$S_i^{(n+1)} = W\left[\beta I_i^{(n)} + \sum_{\nu_2\nu_3\cdots\nu_k} T_{i,\nu_2\cdots\nu_k}^{(n)} S_{\nu_2}^{(n)} S_{\nu_3}^{(n)} \cdots S_{\nu_k}^{(n)}\right], \tag{17}$$

where $\underline{I}^{(n)}$ is the input vector at the nth discrete time step and β is the gain control parameter; i.e., if we set $\beta = 0$, then the input signal is turned off. The evolution of the state vector $\underline{S}$ when the input is turned off can be compared with hallucination.

We can now understand how eq. (4) is derived. Assuming that m (the total number of input time steps) is sufficiently small that $(1-\alpha)^m$ is still very close to one, and that the first m input signals are

$$\underline{I}^{(n)} = \underline{\xi}^{(n)} \quad , n = 1, 2, \ldots m, \tag{18}$$

then eq. (4) follows immediately from eqs. (3) and (18) provided that $\beta = 1$ (the input channel is open) during the learning period and that we are considering the dynamics of the system not long after the learning period (i.e., $m + \epsilon m > n > m, \epsilon << 1$). Since the factor, $(1-\alpha)$, in eq. (3) represents forgetting, the long term memory as described by eq. (3) essentially operates in a "first in, first out" fashion. To ensure that a particular item (pattern vector) is retained in the memory bank, it will have to be revisited from time to time. Also an item which has been frequently visited will have a large weighting factor attached to it, which tends to give it a higher energy peak as well as increasing the corresponding attractor basin. Again this seems to correlate well with the behavior of long term memory in human.

As a nonlinear dynamical system, however, there is always the danger that if the attractor basin of a particular pattern vector becomes large enough to dominate the dynamics, then its weight will grow even further, making it even more dominant. This will lead to a vicious cycle with the end result that the attractor basin of this particular pattern vector will encompass the entire state space. One way to avoid this catastrophe is to allow the weighting factor to saturate at a value low enough that it cannot dominate the neural dynamics. There are several ways to implement this. One is by introducing attentional feedback control to the learning dynamics (3), namely

$$T^{(n+1)}_{\nu_1 \cdots \nu_k} = (1-\alpha) T^{(n)}_{\nu_1 \cdots \nu_k} + \gamma\alpha \sum_{\mu_1 \cdots \mu_k} D^{\mu_1 \cdots \mu_k}_{\nu_1 \cdots \nu_k} S^{(n)}_{\mu_1} \cdots S^{(n)}_{\mu_k} \tag{19}$$

where γ is the attentional feedback gain control (i.e., when γ is larger than one, the system can be said to be "paying attention" which enhances learning speed, on the other hand, when γ is less than one, than the system is not "paying attention", and the rate of learning slows down). The problem is to find an attention controller which is intelligent enough that it can automatically reduce γ to zero whenever a particular pattern vector begins to dominate the dynamics. One way to do this is to use a so called "novelty filter". However, this will be the topic of another article and will not be discussed any further here.

A more direct way is to introduce nonlinear terms which can saturate the runaway instability. One of the simplest ways to accomplish this is to consider the following learning equation

$$T^{(n+1)}_{\nu_1 \cdots \nu_k} = (1-\alpha) T^{(n)}_{\nu_1 \cdots \nu_k} + \alpha \sum_{\mu_1 \cdots \mu_k} D^{\mu_1 \cdots \mu_k}_{\nu_1 \cdots \nu_k} S^{(n)}_{\mu_1} \cdots S^{(n)}_{\mu_k} - \in T^{(n+1)}_{\nu_1 \cdots \nu_k} \cdot E^2_n + \delta T^{(n)}_{\nu_1 \cdots \nu_k} E^2_n \tag{20}$$

where

$$E_n = \sum_{\nu_1\cdots\nu_k} T^{(n)}_{\nu_1\cdots\nu_k} S^{(n)}_{\nu_1} S^{(n)}_{\nu_2} \cdots S^{(n)}_{\nu_k}$$

is just the energy (Lyapunov) function defined in eq. (1) and evaluated at the nth discrete time step; δ and $\in$ are small parameters which have to be chosen so that the nonlinearity is sufficiently small that it does not slow down all learning and yet not so small that it is unable to arrest the growth of a particular attractor before it is too late. The reason for using E^2_n instead of E_n in (20) is to allow repellers to be included in E_n. The way the nonlinear saturation mechanism works is as follows: whenever a particular pattern attractor becomes large, then E^2_n becomes very large when $\underline{S}^{(n)}$ happens to be in its neighborhood (namely, inside the attractor basin). This has the effect of dramatically slowing down learning and preventing further growth of the attractor basin. This can be seen by rewriting eq. (20) for the special case $\delta = \in (1-\alpha)$:

$$T^{(n+1)}_{\nu_1\cdots\nu_k} = (1-\alpha)T^{(n)}_{\nu_1\cdots\nu_k} + \alpha(1+\in E^2_n)^{-1} \sum_{\mu_1\cdots\mu_k} D^{\mu_1\cdots\mu_k}_{\nu_1\cdots\nu_k} S^{(n)}_{\mu_1}\cdots S^{(n)}_{\mu_k}. \tag{21}$$

Equation (21) can be seen to be simply a special case of the attentional feedback learning (eq. (19)) by identifying γ with $(1+\in E^2_n)^{-1}$. In this respect, one might even generalize eq. (21) in such a way that $(1+\in E^2_n)^{-1}$ is replaced by a new function $\gamma(E_n)$ which goes to zero whenever E_n becomes too large, e.g., $\gamma(E_n) = (1-\in E^4_n)/(1+\in E^2_n)$.

To incorporate repellers into the learning mechanism, we need to introduce a critic. This can either be an external teacher or a built-in internal monitor which acts according to some established set of rules. For example, we can have an internal probe which monitors the physical well-being of the system. If any physical damage has been detected, then a pain message will be sent to the learning subsystem. To see how learning rules have to be changed, we take, for simplicity, $D^{\underline{\mu}}_{\underline{\nu}} = 1$ and ignore the saturation terms. Again a gain control parameter λ is called for, and eq. (3) is changed to

$$\begin{aligned} T^{(n+1)}_{\nu_1\cdots\nu_k} &= (1-\alpha)T^{(n)}_{\nu_1\cdots\nu_k} + \alpha[f_+(\lambda) - f_-(\lambda)]S^{(n)}_{\nu_1}\cdots S^{(n)}_{\nu_k} \\ \hat{T}^{(n+1)}_{\nu_1\cdots\nu_{k-1}} &= (1-\alpha)\hat{T}^{(n)}_{\nu_1\cdots\nu_{k-1}} - \alpha\frac{kN}{k-1} f_-(\lambda)S^{(n)}_{\nu_1}\cdots S^{(n)}_{\nu_{k-1}} \end{aligned} \tag{22}$$

where

$$\begin{aligned} &f_+(\lambda) > 0 \text{ when } \lambda > 0\ ,\ f_-(\lambda) > 0 \text{ when } \lambda < 0 \\ &f_+(\lambda) = 0 \text{ when } \lambda < 0\ ,\ f_-(\lambda) = 0 \text{ when } \lambda > 0 \end{aligned} \tag{23}$$

The energy function in this case becomes

$$E = -\sum_{\nu_1\cdots\nu_k} T_{\nu_1\cdots\nu_k} S_{\nu_1}\cdots S_{\nu_k} + \frac{k-1}{k}\sum_{\nu_1\cdots\nu_{k-1}} \hat{T}_{\nu_1\cdots\nu_{k-1}} S_{\nu_1}\cdots S_{\nu_{k-1}} \tag{24}$$

and the dynamical equation is changed to

$$S_i^{(n+1)} = W\left[\beta I_i^{(n)} + \sum_{\nu_2\cdots\nu_k} T^{(n)}_{i,\nu_2\cdots\nu_k} S^{(n)}_{\nu_2}\cdots S^{(n)}_{\nu_k} + \sum_{\mu_2\cdots\mu_{k-1}} \tilde{T}^{(n)}_{i,\mu_2\cdots\mu_{k-1}} S^{(n)}_{\mu_2}\cdots S^{(n)}_{\mu_{k-1}}\right]. \qquad (25)$$

Note that as soon as the repellers are introduced, the dynamics is no longer strictly of the hill climbing type, even with $\beta = 0$. This allows the system to admit periodic attractors in addition to fixed point attractors. Aperiodic attractors of course are not possible since there are only a finite number of distinct states, 2^N to be exact. However, as long as k is sufficiently large, any pattern vector with sufficient weight is still a stable fixed point because for $\underline{S} \simeq \underline{\xi}$, E is still dominated by $(\underline{\xi}\cdot\underline{S})^k$.

So far we have mostly specialized to the case $D^{\mu}_{\underline{\nu}} = 1$. In terms of hardware implementation, this requires, in addition to the N primary processors, an additional $\binom{N}{k} \propto N^k$ secondary processors with attendent interconnections to all primary processors. Even for a moderate k value of 4, and a small number ($N \sim 10^2$) of primary processors, we need about 10^7 secondary processors and a like number of interconnections, a true hardware nightmare! A drastic reduction of the number of interconnections as well secondary processors can be achieved by choosing $D^{\mu}_{\underline{\nu}}$ to be a sparse connection tensor.

One way of accomplishing this is to have a randomly selected subset of all possible interconnections, as suggested by Kohonen [2]. The number of interconnections in this method needs only be directly proportional to the number of primary neurons, instead of $O(N^k)$ when all possible interconnections are used. If we take M to be the average number of interconnections per primary neuron, then MN is the total number of interconnections as well as the number of "hidden" (secondary) neurons. The random sparse interconnection model seems to have some support from neurophysiological evidence [2]. From a theoretical point of view, though, it is rather difficult to predict its behavior strictly on an analytical basis. It should be pointed out that the term "random interconnections" pertains only to the "spatial" structure of the network; it says absolutely nothing about the possible existence or the absence of "chaotic" temporal behavior of the network. Another interesting idea is to allow the interconnections be made in a self-similar fashion, making it easier for the system to correlate events which can be transformed to one another by a change in spatial scales as well as by time renormalization. The interconnections are arranged in a hierarchical manner with only nearest neighbor interconnections at the lowest level. Each kth order nearest neighbor interconnection cluster is fed to a "hidden neuron" [3]. This "hidden neuron" processes the information passed from the primary neurons according to eqs. (2) and (3), passes the processed information back down to the primary neurons, updates the connectivity coefficient (which can be considered to be the state vector for the "hidden" neuron), and also passes the averaged spin value (order parameter) of the primary neurons to the next level, where it undergoes another threshold operation to suppress noise. The situation now becomes identical to that at the previous level and the same clustering operation can be applied. The operation which transforms any given level to the next level can be described in terms of the renormalization group operator, R. Let $D^{h,\underline{\mu}}_{\underline{\nu}}$ be a connection tensor between the hth level and h+1st level:

$$D^{h,\underline{\mu}}_{\underline{\nu}} \equiv \sum_{\underline{t}\in g} \delta_{\underline{\mu}-\underline{\nu}-\underline{t}}, \tag{26}$$

where $\underline{\nu}+\underline{t} = (\nu_1+t_1, \nu_2+t_2, \cdots \nu_k+t_k)$ represents a "near neighbor" of $\underline{\nu}$ for $\underline{t} \in g$, and $\delta_{\underline{\nu}} = \delta_{\nu_1}\delta_{\nu_2}\cdots\delta_{\nu_k}(\delta_\nu = 1$ for $\nu = 0$ and 0 otherwise). The h+1st state vector $T^{h+1}_{\underline{\nu}}$ is updated as follows:

$$T^{h+1,(n+1)}_{\nu_1\cdots\nu_k} = (1-\alpha_{n+1})T^{h+1,(n)}_{\nu_1\cdots\nu_k} + \alpha_{h+1}\sum_{\mu_1\cdots\mu_k} D^{h,\mu_1\cdots\mu_k}_{\nu_1\cdots\nu_k} S^{h,(n)}_{\mu_1} S^{h,(n)}_{\mu_2} \cdots S^{h,(n)}_{\mu_k}, \tag{27}$$

where $\alpha_{h+1} = \alpha^{h+1}$, and $\underline{S}^h$ is defined recursively as

$$S^{h+1}_{\nu_i} = W\left[\sum_{t_i\in g} < S^h_{\nu_i+t_i} >_h\right] \equiv \mathrm{R}S^h_{\nu_i}. \tag{28}$$

$< \cdots >_n$ is the time average operator for the hth time scale, and $S^o_\nu \equiv S_\nu$. In eq. (27), we have conveniently ignored the nonlinear saturation terms to simplify the equation. From the construction of the renormalization group transformation (27) and (28), we notice that there is no longer a single short time scale. In fact, other things being equal, larger objects tend to evolve more slowly in time, and the self-similar scheme seems to capture this spatio-temporal scale invariance quite nicely. The number of interconnections (and interneurons) in this scheme scales as $N\ell nN$, which again is much more favorable than the N^k scaling of the D=1 scheme. While the renormalization group scheme seems to have an overshelming advantage in areas where scale-invariance play an important role, it shares the disadvantage with the random connection scheme in its opacity to analytical treatment. Perhaps the simplest scheme is the subspace projection method in which the connection tensor $D^{\underline{\mu}}_{\underline{\nu}}$ is defined as

$$D^{\mu_1\cdots\mu_k}_{\nu_1\cdots\nu_k} = \sum_{\ell=1}^{N}\sum_{q=-M}^{M}\prod_{i=1}^{k}\delta_{\mu_i-q-\ell}\delta_{\nu_i-q-\ell}. \tag{29}$$

It can be seen from (29) that only local interconnections are allowed in this scheme. Using eq. (4) for $T_{\underline{\nu}}$, the resulting expression for energy is

$$E = -\alpha\sum_{\ell=1}^{N}\sum_{P}(\xi^{(P)}\cdot P_\ell S)^k, \tag{30}$$

where P_ℓ is the projection operator for the ℓth subspace. Specifically,

$$P_\ell S = P_\ell(S_1, S_2, \cdots S_N) = (0, 0\cdots 0, S_{\ell-M}, S_{\ell-M+1}, \cdots, S_\ell, \cdots S_{\ell+M}, 0, 0\cdots 0). \tag{31}$$

The number of interconnections scales as MN, the same as in the random interconnection scheme. It is evident from eq. (30) that the energy E becomes large whenever there is a large overlap between $\underline{S}$ and a particular pattern vector $\underline{\xi}$, and attains maximum value when $\underline{S} = \underline{\xi}$, provided that k is sufficiently large. One interesting way of looking at eq. (30) is that each term $(\underline{\xi}^{(P)} \cdot P_\ell \underline{S})^k$ in E can be considered to be the score of the local feature detector in the ℓth subspace for the candidate $\underline{S}$, and the energy function E can be considered to be the total votes cast by all the local feature detectors on $\underline{S}$. This interpretation is particularly interesting from the point of view of statistical inference schemes [5]. In fact, if we interpret $\underline{S}$ as a candidate hypothesis, then E represents the total score registered for this particular hypothesis; and the hypothesis which maximizes E is the most plausible one. Another interesting aspect of this particular scheme is its relationship with cellular automata of radius M. Further discussion of this scheme will be given later in this paper.

Although the preceding discussion has been focused on autoassociative memory, it can be extended to the heteroassociative case with slight modification. For the sake of clarity, let us for the moment drop the connection tensor D and assume that the primary neurons are fully interconnected. We will also ignore learning and simply take the correlation tensor T to be

$$T_{\nu_1 \cdots \nu_k \, \nu_{k+1} \cdots \nu_\ell} = \alpha \sum_{P=1}^{m} \xi^{(P)}_{\nu_1} \xi^{(P)}_{\nu_2} \cdots \xi^{(P)}_{\nu_k} \eta^{(P)}_{\nu_{k+1}} \eta^{(P)}_{\nu_{k+2}} \cdots \eta^{(P)}_{\nu_\ell}, \tag{32}$$

where $\underline{\xi}^{(P)} = (\xi^{(P)}_1, \cdots, \xi^{(P)}_{N_i})$ and $\underline{\eta}^{(P)} = (\eta^{(P)}_1, \cdots, \eta^{(P)}_{N_o})$ are two pattern vectors which represent, respectively, the input vector of dimension, N_i, and the output vector of dimension, N_o. Equation (32) is a straightforward generalization of eq. (5) for D=1. Denoting the N_i-dimensional input neuronal state vector by $\underline{S}$ and the N_o-dimensional output neuronal state vector by $\underline{U}$, the "interaction" energy E_{int} between the input and output states is

$$\begin{aligned} E_{int} &= \sum_{\nu_1 \cdots \nu_\ell} T_{\nu_1 \cdots \nu_k \cdots \nu_\ell} S_{\nu_1} \cdots S_{\nu_k} U_{\nu_{k+1}} \cdots U_{\nu_\ell} \\ &= \alpha \sum_{P=1}^{m} (\underline{\xi}^{(P)} \cdot \underline{S})^k (\underline{\eta}^{(P)} \cdot \underline{U})^{\ell - k}. \end{aligned} \tag{33}$$

If k is sufficiently large,then it is not hard to see that for $\underline{S} \sim \underline{\xi}$, where $\underline{\xi}$ is one of the stored input pattern, E_{int} is dominated by a single term

$$E_{int} \sim \alpha(\underline{\xi} \cdot \underline{S})^k (\underline{\eta} \cdot \underline{U})^{\ell - k}, \tag{34}$$

provided all other stored patterns are sufficiently different from $\underline{\xi}$. "Sufficiently different" here means

$$\max_{\underline{\xi}^{(P)} \neq \underline{\xi}} \left\{ \frac{(\underline{\xi}^{(P)} \cdot \underline{\xi})^k}{(\underline{\xi} \cdot \underline{\xi})^k} \right\} \ll 1. \tag{35}$$

For k = 8 and $\underline{\xi}^{(P)}$ different from $\underline{\xi}$ by 10%, then $\frac{(\underline{\xi}^{(P)} \cdot \underline{\xi})^8}{N^8} \sim 0.16$ which is already quite small. If we assume that $\underline{\xi}^{(P)}$ is statistically independent of $\underline{\xi}$, then the expectation value of $(\underline{\xi}^{(P)} \cdot \underline{\xi})^8$ is $105N^4 + O(N^3)$, hence

$$(\underline{\xi}^{(P)} \cdot \underline{\xi})^8 / N^8 \sim 105N^{-4}, \tag{36}$$

which is a very small number for any reasonable value of N. Hence eq. (34) can easily be satisfied. The equations analogous to eq. (17) are

$$S_i^{(n+1)} = W\left[\beta^{(n)} I_i^{(n)} + \sum_{\nu_2,\cdots\nu_k,\nu_{k+1}\cdots\nu_\ell} T_{i,\nu_2\cdots\nu_k\nu_{k+1}\cdots\nu_\ell} S_{\nu_2}^{(n)} \cdots S_{\nu_k}^{(n)} U_{\nu_{k+1}}^{(n)} \cdots U_{\nu_\ell}^{(n)}\right], \tag{37}$$

and

$$U_i^{(n+1)} = W\left[\sum_{\nu_1\cdots\nu_k,i,\nu_{k+2},} T_{\nu_1\cdots\nu_k,i,\nu_{k+2},\cdots\nu_\ell} S_{\nu_1}^{(n)} \cdots S_{\nu_k}^{(n)} U_{\nu_{k+2}}^{(n)} \cdots U_{\nu_\ell}^{(n)}\right], \tag{38}$$

Thus, if eq. (34) is true, then eqs. (37) and (38) are approximately

$$S_i^{(n+1)} = W\left[\underline{\xi}_i(\underline{\xi}\cdot\underline{S}^{(n)})^{k-1}(\underline{\eta}\cdot\underline{U}^{(n)})^{\ell-k} + \text{remainder}\right], \tag{39}$$

$$U_i^{(n+1)} = W\left[\underline{\eta}_i(\underline{\xi}\cdot\underline{S}^{(n)})^{k}(\underline{\eta}\cdot\underline{U}^{(n)})^{\ell-k-1} + \text{remainder}\right]. \tag{40}$$

Clearly, $\underline{S}$ and $\underline{U}$ will quickly converge to $\underline{\xi}$ and $\underline{\eta}$ respectively. Therefore, the heteroassociative network that we have just modeled has the desirable property that for any input signal which is sufficiently close to one of the stored pattern, $\underline{\xi}^{(\Gamma)}$, not only will the input state converge to $\underline{\xi}^{(\Gamma)}$ quickly, but it will also elicit a response in the output state which converges rapidly to $\underline{\eta}^{(\Gamma)}$. Such a property is of course extremely important in pattern recognition problems where $\underline{\eta}^{(\Gamma)}$ could represent the name for an equivalence class of which $\xi^{(\Gamma)}$ is one of the members.

An even more interesting application of the heteroassociative model is in problems of drawing inferences and resolving hypothesis from a mass of uncertain and incomplete evidence. The standard approach is to use either a Bayesian inference network, where *ad hoc* scoring functions and certainty factors for inference rules are supplied by domain experts. The trouble with this approach is that the joint probabilities provided by the experts are usually inconsistent and inaccurate, and very complicated global relaxation processes are typically required to strike a balance between conflicting evidence. In contrast, the reasoning process exhibited by humans usually progresses in a narrowly focused incremental manner along prescribed pathways. And the speed and ease with which our brains perform low level cognitive and interpretive functions seem to indicate that there are far better ways to approach inference problems. Unlike the pattern recognition problem, where there exists a many-to-one correspondence between $\{\xi^{(\Gamma)}\}$, and $\eta^{(i)}$, $\{\xi^{(\Gamma)}\}$, is the set of pattern vectors which belong to the ith equivalence class. In inference problems, the probabilistic inference rules are typically of the form:

$$\begin{array}{lll} \text{If } \xi^{(1)}, \text{ then } \eta^{(1)} & \text{with conditional probability} & P(\eta^{(1)}|\xi^{(1)}) \\ \quad\text{or}\quad \eta^{(2)} & \text{with probability} & P(\eta^{(2)}|\xi^{(1)}) \\ & \vdots & \\ \quad\text{or}\quad \eta^{(r)} & \text{with probability} & P(\eta^{(r)}|\xi^{(1)}). \end{array} \tag{41}$$

Hence the correspondence is of the one-to-many type.

More generally, there may exist an equivalence class $\{\underline{\xi}^{(P)}\}_i$ of evidence with the corresponding set $\{\underline{\eta}^{(P)}\}_i$ of alternative hypotheses. Anytime $\underline{S}$ is close to $\underline{\xi} \in \{\underline{\xi}^{(P)}\}_i$, E_{int} can be shown to be dominated by

$$E_{int} \simeq \alpha(\underline{\xi} \cdot \underline{S})^K \sum_{P \in M_i} W_{(P)}(\underline{\eta}^{(P)} \cdot \underline{U})^{\ell-k} \simeq \alpha N^K \sum_{P \in M_i} W_{(P)}(\underline{\eta}^{(P)} \cdot \underline{U})^{\ell-k} \tag{42}$$

where $\{\underline{\xi}^{(P)} | P \in M_i\} = \{\underline{\xi}^{(P)}\}_i$, and statistical weights $W_{(P)}$ have been introduced to represent the fact that the hypotheses, $\underline{\eta}^{(P)}$, in general are not equally probable.

The relationship between the statistical weight $W_{(P)}$ and the probability can be gleaned from the fact that the probability that $\underline{U}$ will converge to one of the hypotheses, $\underline{\eta}^{(P)}$, is simply the ratio of the number of states within the attractor basin of $\underline{\eta}^{(P)}$ to the total number of states ($= 2^N$), and that the basin always increases in size with the increase of $W_{(P)}$. However, a precise determination of the functional dependency of basin size and $W_{(P)}$ is extremely difficult for all but the simplest cases. Nevertheless, in most cases even a crude estimate of $W_{(P)}$ based on subjective probability (for example, simply equate $W_{(P)}$ with the probability) is probably no worse than the likelihood ratio produced by experts. Once a set of rules is constructed, we can allow those rules to interact with one another, thus enabling complex inferences to be made. A simple way to couple the various inference rules together is to feed the output state directly to the input state in order to form a recurrent inference network.

To understand this in dynamical terms, let us consider eqs. (37) and (33) to be a nonlinear mapping T between $(\underline{S}^{(n)}, \underline{U}^{(n)})$ and $(\underline{S}^{(n+1)}, \underline{U}^{(n+1)})$:

$$T\begin{pmatrix} \underline{S}^{(n)} \\ \underline{U}^{(n)} \end{pmatrix} = \begin{pmatrix} \underline{S}^{(n+1)} \\ \underline{U}^{(n+1)} \end{pmatrix}, \tag{43}$$

with the initial condition $\underline{S}^{(0)} = \underline{I}$ and $\underline{U}^{(0)} = \underline{U}_0$. Note that we have assumed that $\beta^{(0)}$ is large and $\beta^{(n)} = 0$ for $n > 0$. After interating the map (43) a sufficient number of times, the initial state vector (I, U_0) is mapped to an attractor which will be taken to be a fixed point $(\underline{\xi}, \underline{\eta})$:

$$T^\ell \begin{pmatrix} I \\ U_0 \end{pmatrix} = T^{(\infty)} \begin{pmatrix} I \\ U_0 \end{pmatrix} = \begin{pmatrix} \xi \\ \eta \end{pmatrix} = \begin{pmatrix} \xi \\ O \end{pmatrix}. \tag{44}$$

where $O = \eta$ is defined to be the final output vector. Obviously both $\underline{\xi}$ and O depend on I and U_0. Hence we can define the $\underline{U}_0$-dependent mapping between the input and output vectors:

$$O = F_{U_0} \circ I = F(U_0, I), \tag{45}$$

where F_{U_0} is the U_0-dependent map and F is a vector-valued function defined by F_{U_0}.

It should be pointed out that, in general, F is a function of enormous complexity and in the limit, $2^N \to \infty$, F cannot be defined in terms of elementary functions because of the recursive nature of the definition. (Although $2^N \to \infty$ and $N \to \infty$ mathematically means the same thing, we nevertheless choose the former to emphasize the fact that even for a modest N value of 10^3, 2^N can be considered to be infinite for all practial purposes). The complexity of $\underline{F}$, we believe, is the biggest difference between our approach and those using the so called "linear associative mapping" where $\underline{F}$ is assumed to be a linear function of $\underline{I}$ and independent of $\underline{U}_0$.

Having defined $F_{\underline{U}_0}$, we now proceed to identify the output vector $\underline{O}$ as the next input. We will also assume that $\underline{U}_0$ is randomly generated for simplicity. Evidently, if we had more information, we might be able to choose $\underline{U}_0$ judiciously to obtain an optimal result. There are other possibilities; for example, $\underline{U}_0$ can be taken to be $\underline{\xi}$ from the previous calculation. The problem with the last method is its deterministic nature. Therefore it is not readily amenable to statistical treatment. Assuming that $\underline{U}_0$ has uniform statistics, we can take $F_{\underline{U}_0}$ to be a stochastic mapping function having the property that it will map $\underline{I}$ to one of the admissible attractors compatible with $\underline{I}$ with a probability which is directly proportional to the size of the basin of that particular attractor. Renaming $F_{\underline{U}_0}$ as $\hat{F}$, we can define a stochastic recurrent inference network according to

$$\underline{I}^{(n+1)} = \hat{F} \circ \underline{I}^{(n)} \ , \ \underline{I}^{(0)} = \underline{I}. \tag{46}$$

Equation (46) can be considered to be a Monte Carlo version of the Bayesian network because the probabilities associated with alternative hypotheses cannot be obtained in a single pass and can only be obtained by averaging over many passes. This, in fact, is reminiscent of the inference process of human beings because of our rather limited short-term memory and our general inability to switch rapidly between alternative hypotheses. It can even be argued that under normal circumstances it is unnecessary to access to all alternative hypotheses. The first couple of most plausible hypotheses usually suffice. Equation (46) still needs a termination condition whenever the top of the net has been reached or when additional pieces of evidence are required. The former can be achieved by mapping the top level hypothesis to an action which could, for example, post a message on the monitor screen. Similarly, in the latter case, the inference processing can be interrupted momentarily by sending a message posed as a question. Once a proper answer has been received, the new piece of evidence can be introduced as input, and the inference propagation can be resumed. There is another possibility of nontermination, namely circular reasoning. Fortunately, since $\hat{F}$ is a stochastic operator, strictly periodic attractors are not possible, and sooner and later the system will break the cycle.

Equations (32), (33), (37) and (38) can be further generalized to include multiple association. For example, the interaction energy E_{int} can be written as

$$\begin{aligned} E_{int} = & -\sum_{\nu_1 \cdots \nu_{k_\ell}} T_{\nu_1 \cdots \nu_{k_1} \nu_{k_1+1} \cdots \nu_{k_2} \cdots \nu_{k_2} \cdots \nu_{k_\ell}} U_{1,\nu_1} \cdots U_{1,\nu_{k_1}} U_{2,\nu_{k_1+1}} \cdots U_{2,\nu_{k_2}} \cdots U_{\ell,\nu_{k_\ell}} \\ = & -\alpha \sum_{p=1}^{m} (\xi_1^{(p)} \cdot U_1)^{k_1} (\xi_2^{(p)} \cdot U_2)^{k_2 - k_1} \cdots (\xi_\ell^{(p)} \cdot U_\ell)^{k_\ell - k_{\ell-1}}, \end{aligned} \tag{47}$$

and the neural dynamics is governed by

$$U_{ji}^{(n+1)} = W\left[\sum_{p=1}^{m} \xi_{ji}^{(P)}(\underline{\xi}_1^{(P)} \cdot \underline{U}_1^{(n)})^{k_1} \cdots (\underline{\xi}_{j-1}^{(P)} \cdot \underline{U}_{j-1}^{(n)})^{k_{j-1}-k_{j-2}} (\underline{\xi}_j^{(P)} \cdot \underline{U}_j^{(n)})^{k_j-1-k_{j-1}} (\underline{\xi}_{j+1}^{(P)} \cdot \underline{U}_{j+1}^{(n)})^{k_{j+1}-k_j} \cdots \right.$$

$$\left. \cdots (\underline{\xi}_\ell^{(P)} \cdot \underline{U}_\ell^{(n)})^{k_\ell-k_{\ell-1}}\right], \tag{48}$$

where $\underline{\xi}_j^{(P)} = (\xi_{j1}^{(P)}, \xi_{j2}^{(P)} \cdots \xi_{jN_j}^{(P)})$ and $\underline{U}_j^{(n)} = (U_{j1}^{(n)}, \cdots U_{jN_j}^{(n)})$ are, respectively, the Pth pattern vector of the jth layer and the nth state vector of the jth layer. Equations (47) and (48) reduce to the autoassociative and heteroassociative cases by simply setting ℓ equal to 1 and 2, respectively. Therefore, autoassociative model can be considered to have only a single layer of primary neurons; and heteroassociative, in turn, has two layers, namely, an input layer and an output layer.

The heteroassociative model of the Monte Carlo inference network probably will perform satisfactorily when different inference rules can be assumed to be statistically and/or causally independent. Indeed, the same assumption has been made by AI workers for a Bayesian network in order to make the probabilistic algorithm computationally managable. However, in the real world the various inference rules are usually both statistically and causally related, and the Markovian approximation is no longer valid. The multi-layered multi-associative dynamics can be modelled by replacing the stochastic mapping (46) by its nonMarkovian counterpart:

$$(\underline{I}^{(n+1)}\ ,\ \underline{I}^{(n)}\ ,\ \underline{I}^{(n-1)}\ , \cdots \underline{I}^{(n-\ell+3)}) = \hat{F} \circ (\underline{I}^{(n)}\ ,\ \underline{I}^{(n-1)}\ ,\ \cdots\ ,\ \underline{I}^{(n-\ell+2)}), \tag{49}$$

or

$$\underline{I}^{(n+1)} = \underline{F}(\underline{I}^{(n)}\ ,\ \underline{I}^{(n-1)}\ , \cdots \underline{I}^{(n+2-\ell)}), \tag{50}$$

where $\underline{F}$ is a stochastic vector-valued function defined by the nonMarkovian map $\hat{F}$ (eq. (49)).

In general, it is expected that the causal and statistical links grow weaker as we travel further into the past, therefore ℓ need not be very large. Also N_j, the number of primary neurons at the jth layer, can be made to decrease rapidly with j if some data compression scheme can be used to reduce the amount of evidence which needs to be retained for the jth layer. One possible scheme is to progressively weed out weaker evidence. Equation (50) represents a kind of voting convention in which the most likely hypothesis is decided not just by current members but also by the voting of the past members, although the votes cast by the past members tend to get counted less.

It can be said that the influence exerted by past events is largely contextual. As such, it allows contextual dependency to play an important role. For example, if the majority of the past evidence and the inference rules invoked to deal with them all have something to do with a football game, then even if the presently acquired new evidence seems to have nothing whatsoever to do with football game, it is a safe bet that the most likely new hypothesis still will have a lot to do with the football game. Hence, if we simply take the current event out of context, then most probably, we will draw a wrong conclusion. Furthermore, even if the current event indeed has absolutely nothing to do with the football game, it is overwhelmingly possible that it is merely a distraction.

For example, imagine yourself in a football game with a friend during the half-time intermission, the topics of conversation can momentarily shift to national politics. If the inference process does not take into account the past conversation, then clearly it will be misled to explore regions which are irrelevant and of no interest to the main event. Mathematically, because of the integral relationship between the joint probability distributions:

$$P(\underline{I}^{(n+1)}, \underline{I}^{(n)}) = \sum_{\underline{I}^{(n-1)}} \sum_{\underline{I}^{(n-2)}} \cdots \sum_{\underline{I}^{(n+2-\ell)}} P(\underline{I}^{(n+1)}, \underline{I}^{(n)}, \cdots \underline{I}^{(n+2-\ell)}), \tag{51}$$

where $\sum_{\underline{I}^{(i)}}$ denotes the summation over all possible vectors $\underline{I}^{(i)}$, given $\underline{I}^{(n-2)}, \cdots \underline{I}^{(n+2-\ell)}$, the joint distribution $P(\underline{I}^{(n+1)}, \cdots \underline{I}^{(n+2-\ell)})$ tends to be more localized. Going back to the example of the football game, if most of the already received or deduced evidence (once a hypothesis has been selected, it automatically becomes deduced evidence) tends to be weakly connected with the football game so that they are biased by at least ϵ amount, then the combined bias should be like $(1+\epsilon)^{\ell}$, which, for $\ell > \frac{1}{\epsilon}$, would strongly support the hypothesis that you are in a football game, even though "football game" has never been presented as direct evidence. Thus, a criminal can be convicted by overwhelming circumstancial evidence for precisely the same reason.

The advantage of having a strongly localized probability distribution is evident; having known that you are very likely to be in a particular situation certainly allows you to "zoom in" and greatly reduces the search space. In fact, we can call the nonMarkvian search the "context-directed beam search" to paraphrase AI jargon. Another bonus of this approach is that hidden concepts can be represented through a distributed correlation among a fair number of seemingly unrelated events or objects. Note the similarity between "distributed concept" and distributed memory. The possibility of being able to lift a high level concept right out of multiple correlation is an intriguing one.

It should be remarked here that there already exists a technique in traditional AI to handle nested contexts. The technique, which is in vogue in AI currently, has been dubbed "frame" by Marvin Minsky [6] who has contributed many ideas about frames. The important point here is that frames make explicit the contextual dependence by having frames nested within frames. Each frame has many slots inside which either subframes can be placed, or else the slots are filled with default values. The frames have nice property inheritance rules which allow default properties of a frame to be passed down to all its subframes. Hence, for example, a football game frame will come with its own set of expectations which are inherited by all its instantations.

The frame technique recently has come under attack because of its inability to deal with variability and exceptions and conflict resolution. For that matter, the Monte Carlo inference net can deal with exceptions and conflicts in a very natural way. Imagine that the rule:

$$\text{If } A \text{ and } B \text{ , then } D \text{ with probability 1,} \tag{52}$$

has initially been implemented. If, at a later time a new rule is found, which states that

$$\text{If } A \text{ and } C \text{ and } B \text{ , then } E \text{ with probability 1.} \tag{53}$$

Obviously these two rules are not compatible. However, it is quite likely that the condition $A \wedge B \wedge C$ was never encountered in the previous tests. Hence the entire region $A \wedge B$ gets mapped to the basin of D. As soon as the region of exception $A \wedge B \wedge C$ has been found and learned, a new attractor E is created inside the original attracting basin for D. On a frame-based system, conflicts of the type just mentioned can only be resolved by a major revision of the program.

Another way to extend the capability of the Markovian inference net is by adding an internal state vector $\underline{Q}$ to the inference rules as well as rules to update $\underline{Q}$'s. The extended rules are of the form:

$$\begin{array}{lllllll} & \text{If } \underline{I} \text{ and } \underline{Q}_1 \text{ , then change } \underline{Q}'_1 \text{ , and } \underline{O}_1 & & & & & \\ \textit{Else} \quad \textit{If} & \underline{Q}_2 \text{ ,} & \text{then change} & \underline{Q}_2 & \textit{to} & \underline{Q}'_2 & \textit{and} \quad \underline{O}_2 \\ & & \vdots & & & & \\ \textit{Else} \quad \textit{If} & \underline{Q}_\ell \text{ ,} & \text{then change} & \underline{Q}_\ell & \textit{to} & \underline{Q}'_\ell & \textit{and} \quad \underline{O}_\ell \end{array} \tag{54}$$

The internal states, $\underline{Q}_i$, can be considered to be a set of fancy flags. Their introduction can drastically enhance the computational capability of the inference net. For example, they could be used to keep track of which rules have been used and how many times, etc. As a matter of fact, each rule now is as powerful as a finite state automaton. This can be seen from the following generalization of eq. (43):

$$T \begin{pmatrix} \underline{S}^{(n)} \\ \underline{Q}^{(n)} \\ \underline{U}^{(n)} \end{pmatrix} = \begin{pmatrix} \underline{S}^{(n+1)} \\ \underline{Q}^{(n+1)} \\ \underline{U}^{(n+1)} \end{pmatrix} , \tag{55}$$

with the initial condition $\underline{S}^{(0)} = \underline{I}$, $\underline{Q}^{(0)} = \underline{Q}$, and $\underline{U}^{(0)} = \underline{U}_0$.

Following our previous derivation, we can write

$$T^{(\infty)} \begin{pmatrix} \underline{I} \\ \underline{Q} \\ \underline{U}_0 \end{pmatrix} = \begin{pmatrix} \xi \\ Q' \\ O \end{pmatrix} , \tag{56}$$

where ξ is the next state vector. Again, a U_0-dependent map can be defined

$$\begin{pmatrix} O \\ Q' \end{pmatrix} = F_{U_0} \begin{pmatrix} I \\ Q \end{pmatrix} = \begin{bmatrix} \delta_{U_0} & (I, & Q) \\ \lambda_{U_0} & (I, & Q) \end{bmatrix} , \tag{57}$$

where $\underline{\delta}_{U_0}$ is the next output function and $\underline{\lambda}_{U_0}$ is the next state function. Therefore the extended rule satisfies the definition of a finite state machine. Of course, since (55) can implement a large set of extended rules at the same time, what we have here is a stochastic inference network whose nodes are themselves finite state automata.

Even more power can be gained by allowing the extended rule finite automata to interact through a separate associative memory message list (or "blackboard"), thus turning the system into a universal computer. The advantage of having a message list or a blackboard in the inference net is that it will enable the net to deal with multiple evidence (or multiple input). Typically at any time during the reference process, the message list will contain both initial and deduced evidence, as well as additional new external evidence which has just been entered, and the rules can be considered to be active agents which can enter and retrieve messages.

An interesting metaphor is viewing the inference engine as a master craftsman and the message list as the master craftsman's work bench on which unfinished materials (initial evidence), parts and products in various stages of completion (deduced evidence) are sitting. What the master craftsman does is pick up something from the workbench, find the proper tool to apply, work on it, and change tools as necessary until he is finished with that subtask. Then he puts the partially finished product back on the bench and proceeds to pick up another object from the bench. The message list is an autoassociative storage with intermediate time constant. Its pointer is usually at the attractor which has just been entered. If the pointer happens to be in the basin of attraction of the input vector of one of the inference rules (meaning that a "best match" is found between the new message and the condition part of that particular rule), then that particular rule will be triggered and the message processed. The processed information is then dumped back into the message list to be processed by other rules. In order to prevent the same message from being picked up by the same rule more than once, one can either erase the message from the autoassociative store by writing in the negative image of that message, or one can change the internal state of the rule to keep track of those messages which have been processed by a given rule.

Perhaps it is worthwhile at this point to note the strong similarity between our workbench metaphor and the classifier system of Holland [7] as well as the immune system of Farmer, Packard and Perelson [9]. In fact, in addition to the more superficial similarities between our inference rules and Holland's condition-action pairs as well as the employment of a message list in both cases, there is a much deeper correspondence. By the very nature of the associative learning algorithm, the inference rules in our system constantly evolve on the long time scale. Rules which are frequently applied gain strength, thereby increasing their respective attractor basins which in turn makes them even more useful. This, of course, is just the learning instability we have alluded to earlier and nonlinear terms need to be added to prevent rule condensation. Weaker rules will gradually lose their strength because of the forgetting effect; most will ultimately disappear. New rules can either be imported from the outside, or they can be created internally. The latter is made possible by (a) the stochastic nature of the rules, and (b) the nonlinearity of the learning subsystem. The stochasticity of the rules is similar to mutation because it allows the mapping to differ slightly each time the same rule is applied. Consequently the rules can perform a random walk in the rule space (the attractors and attractor basins evolve by executing a random walk). Needless to say, if a rule has proved its worth, then it tends either not to wander very far from the winning formula, or not at all!

Potentially a much faster way of evolution is through the nonlinear effect of the learning process. Although nonlinear evolution of a complex system is in general not well-understood, it is nevertheless useful here to make a few speculations: first of all, if sparse local interconnections similar to those described by eqs. (27), (28) or eq. (30) are used for learning, then the rules in general can be thought of as tightly coupled collections of local feature operators. Since the local feature operators usually have the tightest bond, it follows that they are usually evolved as single units. Hence, nonlinear interactions of rules usually lead to rearrangement of feature operators instead of a wholesale transformation of the latter, provided the nonlinearity is not too strong. From the energy point of view, it can be said that the binding energy of the individual feature operator is too strong to be perturbed by weak interaction. Here again, the analogy of chemical reactions and immune system is relevant. If the interaction is even weaker, then only weak links binding feature operators can be broken, leading to a genetic operation which is known as "cross-over". Now if we can differentiate between strong bond and moderately strong bond, then it is not unreasonable to believe that there also exists weaker bonds among inference rules. In fact, this has already been alluded to when we address the problem of contextual dependence and long range correlations among rules. This leads to our second speculation, that there exists a hierarchical structure of rules, with local features being the atoms, larger subrule units being the tightly bonded molecules, rules themselves being the molecules of molecules, and contextual and/or causal collections of rules, or tasks, being the macromolecules, and so on, and so forth. In other words, there exists a hierarchy of knowledge representations with varying degrees of cohesiveness and complexity in approximately mutually exclusive fashion. Whereas strong nonlinear interactions can result in drastic genetic surgery of the rules, weaker interactions tend to cause a slow evolution of the hierarchical structure itself. Once again, higher order multiple associations cause "level hopping", allowing the evolution to be carried out at the "meta" level. The third speculation is that there may exist a critical transition point which is a fixed point of some renormalization group, beyond which long range order can develop across all complexity scales and across all levels. In statistical mechanics and in nonlinear dynamics, the renormalization group equation is usually expressed mathematically as

$$R' = SRS^{-1}, \qquad (58)$$

where R is the renormalization semi-group transformation defined in terms of block averaging (in statistical mechanics), p is a positive integer (usually 2), and S is a scaling group transformation with group parameter λ. The fixed points for eq. (58) correspond to nontrivial solutions of eq. (58) for some λ, the latter being the eigenvalue of the renormalization group equation, (58). In our case, such an equation is clearly not very meaningful since it implies that there exists a fixed scaling relationship across all levels which simply cannot be true for systems of this complexity. Instead the situation is probably slightly more akin to that of the random fractal (Mandelbrot [7]) where the relationship between different levels can only be described in a statistical sense. Of course "random" is probably not a good description of the situation either. A better description is that there is a synergistic relationship across all levels. At the individual level, the evolution of the rules would all seem to be random, but when the system is examined as a whole, over a sufficiently long period of time, we find that the truly random part of the evolution tends to average out rather quickly. On the other hand, the coherent part tends to grow exponentially (algebraically at the fixed point) out of the noise because of the existence of coherent structure at higher levels, which in turn triggers the evolution of higher level rules.

The discussion of these speculations is a bit vague. However, that does not mean such synergistic behavior between different levels cannot be found in physical systems. We wish to point our that just such behavior is present in plasma physics (among many other examples). Consider the following situation: an electric field is applied to set up a relative drift between electrons and ions in a uniform plasma. After the electric field is turned off, the macroscopic drift will cause the plasma to develop small scale (microscopic) instabilities, meaning that the coherent part of the ever present noise will get amplified (the coherent part is defined to be a superposition of unstable linear eigenfunctions). The nonlinear evolution of the microinstabilities in turn induces a "quasilinear" evolution of the macroscopic distributions. The main difference between the plasma physics example and the neural system is that in the former, there are only two levels, microscopic and macroscopic; whereas in neural net there is a hierarchy of levels.

Another way of looking at multiple associative memory is to consider it to be either an autoassociator or heteroassociator with multiple conjunctive/disjunctive switches. Take triple association, for example. We can write

$$S_i^{(n+1)} = W\left[\sum_{\rho=1}^{m} \xi_i^{(\rho)}(\underline{\xi}^{(\rho)} \cdot \underline{S}^{(n)})^{k_1-1}(\underline{\eta}^{(\rho)} \cdot \underline{U}^{(n)})^{k_2}(\underline{\varsigma}^{(\rho)} \cdot \underline{V})^{k_3}\right], \tag{59}$$

$$U_i^{(n+1)} = W\left[\sum_{\rho=1}^{m} \eta_i^{(\rho)}(\underline{\xi}^{(\rho)} \cdot \underline{S}^{(n)})^{k_1}(\underline{\eta}^{(\rho)} \cdot \underline{U}^{(n)})^{k_2}(\underline{\varsigma}^{(\rho)} \cdot \underline{V})^{k_3}\right]. \tag{60}$$

It is not hard to see that eqs. (59) and (60) are the same as the heteroassociative eqs. (37) and (38) provided that all $\varsigma^{(\rho)}$ are the same (in which case the extra factor $(\underline{\varsigma} \cdot \underline{V})^{k_3}$ in (59) and

(60) simply does not matter). To make (59) and (60) more interesting, we define the equivalence classes

$$C_q = \left\{(\underline{\xi}^{(P)}\ ,\ \underline{\eta}^{(P)})|P \in I_q\right\}\ ,\ q = 1, 2, \cdots M \tag{61}$$

where I_q's are disjoint sets of integers. By choosing a different $\underline{\varsigma}^{(q)}$ for each equivalence class, C_q, it is possible to switch from one equivalence class to another one just by changing $\underline{V}$, since for sufficiently large K_2, $(\underline{\varsigma}^{(q)} \cdot \underline{V})^{K_2}$ is a sharp function of $\underline{\varsigma}^{(q)} \cdot \underline{V}$. For $M < N$, it may even be possible to pick $\underline{\varsigma}^{(q)}$'s which are mutually orthogonal.

The multiple switch idea is related to the subject of frames. Indeed, it is straightforward to construct a hierarchy of nested equivalence classes using the above mentioned procedure. Default values and property inheritance come naturally in this scheme. However, there is no reason to restrict oneself to the hierarchical representation favored by frame enthusiasm, because most knowledge does not lend itself to strict hierarchical representation.

Attentional feedback gain control is another area where the switch paradigm is useful. The ability to focus, or to "pay attention" to some particular feature of the information being processed (be it external input or internally generated) to the exclusion of all other information, is an important attribute of human intelligent endeavor. Through focusing, we can filter out irrelevant or distracting factors. This enables us to discover connections which would have been obscured or masked by "noise". Deliberately filtering out certain characteristic features also allows the formation of analogical reasoning and concept generalization.

Spatio-temporal memory is another important aspect of human information processing activity which can be implemented by multiple associative memory. Perhaps the simplest method is to use the heteroassociative version of eq. (50)

$$\underline{I}^{(n+1)} = \underline{F}(\underline{I}^{(n)}). \tag{62}$$

However, the heteroassociative network cannot deal with the situations where the same spatial pattern $\underline{A}$ may be followed by more than one pattern, e.g., $\underline{B}_1, \underline{B}_2$, ... etc. In fact, if a given spatio-temporal sequence contains one to many maps in many places, then because of the stochastic nature of the association, the recalled sequence can get all tangled up. If there are more than one spatio-temporal sequences having this property, then the recalled memory will jump from one spatio-temporal sequence to another in a random manner.

An obvious way to remedy the situation is to use the full multiple associative version, eq. (48), directly with sufficiently large ℓ to minimize multiple mapping. The problem with this approach is that in the "real world", time is continuous, and the rate at which events unfold in time may differ from one instant to the next and on successive trial. For small ℓ, such variability can be dealt with automatically because consecutive patterns tend to be close to each other. For large ℓ, however, the same spatio-temporal act played at two distinctly different rates will not match well with each other over the time interval specified by ℓ. Clearly some sort of time-warping is needed. The other alternative is to encode spatio-temporal acts at all possible rate changes within a certain range. Again this will work for small ℓ. For large ℓ, the possible combinations will simply explode (combinatorial explosion).

To see how dynamical time-warping can be implemented, let us examine the two-step version of eq. (50)

$$\underline{X}^{(1)} = \underline{F}(\underline{X}^{(0)}, \underline{X}^{(-1)}, \underline{X}^{(-2)}, \cdots, \underline{X}^{(2-\ell)}), \tag{63}$$

$$\underline{X}^{(1)} \rightarrow \underline{X}^{(0)}, \underline{X}^{(0)} \rightarrow \underline{X}^{(-1)}, \cdots \cdots, \underline{X}^{(3-\ell)} \rightarrow \underline{X}^{(2-\ell)}. \tag{64}$$

Equation (64) corresponds to a shift operation. To recall a spatio-temporal pattern, we first input

$$\underline{I}^{(\ell-1)} \rightarrow \underline{X}^{(0)}, \underline{I}^{(\ell-2)} \rightarrow \underline{X}^{(-1)}, \cdots \underline{I}^{(0)} \rightarrow \underline{X}^{(2-\ell)}, \tag{65}$$

where $\underline{I}^{(0)}, \underline{I}^{(1)}, \cdots, \underline{I}^{(\ell-1)}$ need not be complete nor accurate. After the initial condition is loaded into the network, we take turn executing eqs. (63) and (64) and output the successive values of $\underline{X}^{(1)}$. So far there is no difference between this procedure and eq. (50). To allow time-warping, at any particular time step, we can either suspend the shift operation, (64), to slow down the tempo and do nothing, or execute the shift operation twice to accelerate the tempo. In order to be able to perform the last option, we will need an intermediate value of $\underline{X}^{(1)}$. This can be accomplished by simply using the previous value of $\underline{X}^{(1)}$ for the intermediate $\underline{X}^{(1)}$. A total time-warping factor of 4 or more can be achieved using this procedure. The decision of which one of the three alternative operations should be performed at each time step can be made by using a three-way switch (eqs. (60) and (61)). For spatio-temporal pattern or speech recognition problems, the 3-way switch can be controlled by a low ℓ (short time) spatio-temporal pattern matcher which previews the input and decides which of the three alternatives constitutes the best match. Note that the low ℓ temporal pattern matcher, being inferior to the high ℓ one, might make a decision error and decide to pick the "do nothing" option at a particular time step. This is not a problem since, at the next time step, a sufficiently large discrepancy will show up that the pattern matcher will make the right choice.

5. Learning Statistical Invariants

Even though experts differ in their estimates of the information capacity of a human being, it is generally agreed (Kohonen) [2] that the human brain just does not have enough memory capacity to hold the information which floods us everyday. The usual argument is that information can only be transferred into long term memory under attentional control. However, even if only one configurational sensory pattern were stored every ten seconds or so, the estimated human mem ry capacity would still be exceeded rather quickly.

This has led us to ask the following question: can anything useful be learned after memory capacity of the neural network is completely saturated? For simplicity, let us consider the simple spatial memory model described by the learning rule eq. (3) with the additional assumption that $D_{\underline{\nu}}^{\mu} = 1$. Equation (3) can be readily integrated to give

$$T_{\nu_1 \cdots \nu_k}^{(n)} = \alpha \sum_{\Gamma=1}^{n} (1-\alpha)^{\Gamma-1} S_{\nu_1}^{(n-\Gamma)} S_{\nu_2}^{(n-\Gamma)} \cdots S_{\nu_k}^{(n-\Gamma)}. \tag{66}$$

Unlike eq. (4), however, $\underline{S}^{(\Gamma)}$ cannot be approximated by the input patterns, $\underline{\xi}^{(\Gamma)}$, because, except for early time, the state vectors $\underline{S}^{(\Gamma)}$, to which the neural dynamics converge, depend on the patterns which the system has already learned. If we model noise by independent random variables, then for odd k, according to eq. (4), noise tends to get washed out because of its random nature. Thus neural learning has the effect of noise suppression; i.e., it cannot learn meaningless patterns! Furthermore, once certain pattern attractors are formed, similar input patterns usually get attracted to the respective attractors rather quickly. Hence the formation of new pattern attractors is inhibited. The location as well as the size of the attractors may drift slowly because of the transient effects and because of the input component, $\beta I_i^{(n)}$ (see eq. (17)), which may alter the dynamics somewhat. Such slow evolution should allow the system to adapt to the slowly changing environment adiabatically.

When the input pattern is sufficiently different from stored patterns, the input either may not converge to one of the stored patterns, or even if it does, it usually does so in a very deliberate manner. In the former case, it may get attracted to a spurious attractor which is probably a "recombinant" attractor (i.e., an attractor which has pieces of other attracting patterns), or if the input has been on for sufficiently long time and/or appears frequent enough, then a new attractor is formed. Whereas in the latter case, the long transient may be sufficient for the learning subsystem to either significantly alter the attractor basin or to create a new attractor altogether.

Given the highly nonlinear and varying nature of the evolution of the state vector $S^{(n)}$, it would seem that no conclusion could be drawn regarding the characteristics of $T_{\nu_1 \cdots}^{(n)}$. However, it can be argued that $\underline{S}^{(n)}$ always mirrors the input patterns in one way or another, hence for the purpose of illustration, we will simply replace $\underline{S}^{(P)}$ in (66) by $\underline{\xi}^{(P)}$:

$$T^{(n)}_{\nu_1\cdots\nu_k} = \alpha\sum_{P=0}^{n-1}(1-\alpha)^{n-P}\xi^{(P)}_{\nu_1}\xi^{(P)}_{\nu_2}\cdots\xi^{(P)}_{\nu_k}. \tag{67}$$

For large n and small α, we can replace the discrete sum by an integral,

$$T_{\nu_1\cdots\nu_k}(t) = \alpha\int_0^t d\tau e^{-\Gamma(t-\tau)}\xi_{\nu_1}(\tau)\xi_{\nu_2}(\tau)\cdots\xi_{\nu_k}(\tau), \tag{68}$$

where $n\to t$ and $\alpha\to\Gamma$. Invoking the "reasonable" assumption that the time average can be approximated by ensemble average, we have

$$T_{\nu_1\cdots\nu_k}(t) = \alpha\int_0^t d\tau e^{-\nu(t-\tau)} < \xi_{\nu_1}(\tau)\xi_{\nu_2}(\tau)\cdots\xi_{\nu_k}(\tau) >_\tau, \tag{69}$$

where the time-dependent ensemble average is taken over one statistical ensemble accumulated during time $t+\Delta > \tau > t-\Delta$ with $1 << \Delta << \frac{1}{\alpha}$. Thus $T_{\underline{\nu}}$ can be physically interpreted as time-weighted correlation functions.

Perhaps what differentiates highly saturated memory patterns from sparsely stored patterns is that, whereas the energy landscape of the latter consists strictly of isolated peaks, the former usually consists of peaks merged to form ridges. To see this, let us consider a trajectory, $\underline{\xi}(t)$, which we take to be

$$\underline{\xi}(t) = cos\theta(t)\underline{\xi}^{(1)} + sin\theta(t)\underline{\xi}^{(2)}, \tag{70}$$

where $\theta(t_0) = 0, \theta(t_f) = \frac{\pi}{2}$, and $\dot{\theta} = \Omega = \frac{\pi}{2}(t_f - t_0)$. The state vector can be similarly expressed:

$$\underline{S} = (cos\phi\underline{\xi}^{(1)} + sin\phi\underline{\xi}^{(2)})cos\theta + sin\theta\underline{\xi}^{(3)}, \tag{71}$$

where $\underline{\xi}^{(3)}\perp\underline{\xi}^{(2)}$, $\underline{\xi}^{(1)}$. The form of $\underline{\xi}$ and $\underline{S}$ is dictated by the requirement that $\underline{S}\cdot\underline{S} = \underline{\xi}\cdot\underline{\xi} = N$. The contribution of the trajectory to E is

$$\begin{aligned}\Delta E &= \alpha\left[\int_{t_0}^{t_f} d\tau(\underline{\xi}(\tau)\cdot\underline{S})^k\right]e^{-\Gamma t_f}\\ &= \alpha\left\{\int_{t_0}^{t_f} d\tau cos^k\theta[cos(\theta(\tau)-\phi)]^k\right\}e^{-\Gamma t}\\ &= f(\phi)e^{-\Gamma t}cos^k\theta,\end{aligned} \tag{72}$$

where $t\sim t_0\sim t_f$ and $f(\phi)$ is very nearly constant when $\frac{\pi}{2} > \phi > 0$ and drops to zero rapidly whenever ϕ lies outside of $(0,\frac{\pi}{2})$. Note also the sharp decline when $\underline{S}$ starts to move out of the plane of $\underline{\xi}_1, \underline{\xi}_2$. The trajectory also makes a contribution to $T_{\nu_1\cdots\nu_k}$ which is of the form:

$$\Delta T_{\nu_1\cdots\nu_k} = \alpha_{11\cdots1}\xi^{(1)}_{\nu_1}\xi^{(1)}_{\nu_2}\cdots\xi^{(1)}_{\nu_k} + \alpha_{21\cdots1}\xi^{(2)}_{\nu_1}\cdots\xi^{(1)}_{\nu_k} + \cdots + \alpha_{2\cdots2}\xi^{(2)}_{\nu_1}\xi^{(2)}_{\nu_2}\cdots\xi^{(2)}_{\nu_k}, \tag{73}$$

where $\xi^{(1)} = (\xi_1^{(1)}\ ,\ \xi_2^{(1)}\ ,\cdots\xi_N^{(1)})$ etc.

The preceding example shows that a dense distribution of patterns can change the dimension of the attractor from zero to finite dimension (in an approximate sense, if one can consider 2^N to be infinite). The merging of simple attractors to form an invariant attractor manifold certainly results in the disappearance of the identity of the individual patterns. Nevertheless, this does not imply that the resulting attractors are any less useful. On the contrary, any piece of information which cannot survive the averaging process should be considered irrelevant, and the elimination of irrelevent information contributes immensely to the effective utilization of the memory storage. This becomes clear when one realizes that the expression (73) for $\Delta T_{\underline{k}}$ contains fewer than 2^k coefficients, even though it is obtained by summing over a very large number (of order N) of patterns!

Since the invariant attractor manifold summarizes the characteristics of a whole class of spatial (or spatio-temporal) patterns, they can no longer be said to describe individual patterns but must be considered as rules. In fact, it is entirely possible for the saturation learning algorithm to learn laws of physics empirically. The ability to learn common sense (naive) physics empirically could potentially be of some benefit to research in cognitive psychology.

6. Numerical Studies of the Autoassociative Map

As a simple demonstration of the capabilities of the autoassociative scheme, we have developed a small computer code which implements eqs. (2)-(4) with $D = a = 1$ on a CRAY XMP, a machine which performs logical operations at the rate of 5×10^{10} bits per second. An initial pattern set, $\xi^{(r)}$, was chosen and many state vectors, $S^{(n)}$, were iterated through the algorithm, always converging to some pattern. A fraction of the initial state vectors converged to members of the pattern set. The dependence of this fraction upon the power (k in eq. 5) and upon the number, I, of patterns in the pattern set was calculated for several different k and I. Also the average number of iterations required for convergence was calculated.

7. Dependence on Exponent and Size of Pattern Set

One qualitative result is that the number of iterations needed to converge decreases with increasing exponent; however, the number of attractors not in the initial set of patterns increases with increasing exponent. Another qualitative result is that the number of iterations required to converge generally increases with the number of initial patterns.

Specifically, an initial pattern set containing I elements was chosen. Each pattern had 64 randomly chosen bits. Eighteen pattern sets were used with $2 \leq I \leq 512$. For each set, 100,000 random initial states were iterated to convergence. If the state converged to a pattern not in the initial set, we called the converged pattern a trap. The following quantities were calculated: average number of iterations to convergence, number of traps, and fraction of the states which converge to a trap. The results are shown in Figs. 1 and 2 for four different values of the exponent.

In Fig. 1, the average number of iterations to convergence is seen to decrease with increasing exponent and it is observed to increase with increasing pattern number. In Fig. 2, the number of traps is observed to increase with increasing pattern number and with increasing exponent.

The ability to associate each input state with a unique attractor is closely related to the concept of memory. If we define the memory capacity of a given map to be the number of initial patterns stored for which 90% of the input states are correctly identified with the stored pattern from which the state was generated by changing N bits, then we find that memory capacity depends strongly on the power, P, of the energy function and on the initial Hamming distance between the pattern and the initial state. In Fig. 3a, the fraction of input states which converge correctly is plotted as a function of the number of randomly chosen stored patterns. The curve for P = 2 has a memory capacity of ten; the curve for P = 3 has a memory capacity of about 120; and the curve for P = 5 has a memory capacity greater than 10,000. The 2048 randomly chosen initial states were each exactly 2 Hamming units away from an initial pattern.

For states which are exactly 12 Hamming distances away from one of the stored patterns, we see in Fig. 3b that the memory capacities decrease to 5 for P = 2, 100 for P = 3, and about 2000 for P = 5. One can conclude from these calculations that the memory capacity increases considerably with increasing P. One can also conclude that the basin of attraction for each attractor is quite large.

If the definition of memory capacity is modified to only require that the initial state converge to the nearest pattern instead of the pattern from which the state was generated by changing N bits, then we see in Fig. 3c that memory capacity increases. Here for an initial Hamming distance of 20 from one of the patterns, the newly defined memory capacity becomes 2 for P = 2, 40 for P = 3, 1200 for P = 5 and greater than 2000 for P = 9.

It is possible that a polynomial map might combine the advantages of the monomial maps studied here. For example, a linear combination of P = 3 and P = 17 might provide rapid convergence and fewer traps.

Limit cycles with periods of up to 6 iterations were observed. It is also interesting that no traps were observed for I = 2.

8. Dependence on Initial Hamming Distance

The choice of random initial states simulates a choice of initial states which are ~ 32 Hamming distances away from elements in the initial pattern set. Studies which examine the convergence rate as a function of the initial Hamming distance indicate more rapid convergence and fewer traps for smaller initial Hamming distance.

9. Dependence on Mask Overlap

The above studies took the dot product of a state vector with a pattern vector and raised the dot product to a power to obtain a weight. All bits contribute to this weight so it is of some interest to study the effect of "masking" the dot product. We define "masking" of a dot product to be taking the dot product of a subset of the pattern bits. Certainly no bit should be omitted in the masking algorithm but the amount of overlap of adjacent masks can be varied. It was found that for these random bit pattern sets no mask overlap was required for convergence of states which are one Hamming unit away from an element in the initial pattern set.

10. Discussion

We have presented in this paper some ideas about how to mimick the evolution of an intelligent system. Our approach is based on a generalization of the correlation matrix formalism to higher order. From the point of view of theoretical physics and nonlinear dynamics, our neural network can be considered to be a system of spins interacting with one another through a modifiable high order nonlinearity. The evolution of the nonlinear interaction is, in turn, governed autonomously by another set of nonlinear dynamical equations which operates on a much longer time scale. In a way one can speak of a dynamical system whose equations of motion are slowly changed by another dynamical system in response to changes in the environment, thus one can consider the second level equations to be the "equations of equations". Although simple Hebbian-like learning is usually invoked by neural modellers to allow connection (correlation) matrices to be modified, the equations used by these authors are typically simple linear equations, the major exception being the master-net/slave-net model of Lapedes and Farber [3], to whose work our coupled nonlinear neural network bears strong resemblance.

Multiple associative memory (MAM) has received very little attention among neural researchers, although it seems obvious that MAM should play a major role in the higher level intellectual activities of human beings. The chief reason for this lack of attention appears to be that there does not exist a straightforward way of implementing MAM within the correlation matrix formalism. We have demonstrated that a Bayesian-like inference net can be formulated using multiple associative architecture. It is further shown that universal computing is possible with the addition of a common associative memory storage. The ability to perform arbitrary finite recursive computation is very important since most inference nets studied by AI workers do not have such power, severely limiting their usefulness. Of course most standard computers theoretically have this capability given a large enough disk. However, they have to be programmed with a specific task in mind. Perhaps even more important than being able to perform inference tasks are (a) our neural network potentially can learn to extract environmental invariants directly from observation through what we call "saturation learning" technique, and (b) the nonlinear learning algorithm affords a novel way to evolve the rules in a genetic manner. The latter is made possible by the fact that the rules in general will interact with one another through the nonlinearity in the learning dynamics in a way reminiscent of the chemical interaction of molecules in the immune system. Specifically, since rules are represented in the neural network as a strings of 1's and -1's, they can be thought of as molecules. Within any given rule molecule, there may be subunits which are much more tightly bonded, the likelihood of their existence increases as the rule becomes more and more complex. This means that when rules interact nonlinearly, as long as the interation energy is smaller than the binding energies of the subrules, they tend to recombine in a way that leaves the more tightly bonded subunits intact.

The idea of recombinant rules (borrowing a terminology from genetic engineering) is an attractive one, because it allows an evolution process which is not completely random; i.e., frequently used (and therefore highly successful) subunits are preserved and potentially unprofitable paths are eliminated. To be sure, some of the potentially profitable directions may also be left unexplored. However, they could only be found by an exhaustive search which leads to combinatorial explosion and is therefore unacceptable. Similar consideration also applied to super rules (i.e., a conceptually, contextually, or functionally connected sequences of rules), and to super-super rules, etc. Thus the entire hierarchy of rules and rule chunks are genetically evolved simultaneously.

The genetic evolution we have discussed so far is of the "bottom-up" type, namely, the genetic evolution of the lower levels can affect that of the higher levels but not *vise versa*. Therefore the evolution path still would seem to be random and without direction. Fortunately the same multiple associative architecture which provides the "bottom-up" genetic evolution also provides a natural "top-down" or goal-directed evolutionary pathway. This can be seen from the fact that the nonlinear interaction coefficients, i.e., the laws of physics which govern the interactions of the rules, are themselves evolving in time (in response to the changing domain). The slow evolution of the laws in turn can change the direction and the rate of genetic mutations to favor certain evolution paths. This bi-directional evolution is conjectured to reach criticality when the evolution at all levels become synchronized, i.e., they evolve as one. Even though microscopically the evolution processes would still look random, when taken as a whole, the system would seem to evolve coherently, as if with purpose.

The preceding discussion is speculative. However, in view of considerable evidence concerning the critical phenomena of complex physical system, this is at least an educated speculation. So far our efforts to simulate the network on a computer have been mostly concentrated on autoassociative memory, where we have demonstrated quite conclusively that the higher order correlation scheme is superior to the correlation matrix scheme in memory capacity and speed of convergence. We have also demonstrated the viability of using sparse connection (nonoverlapping masks) for pattern recall. Although not presented in this paper, we have also done limited simulation runs on heteroassociative memory and multi-associative spatio-temporal memory. In the former case, we have shown that the heteroassociative pattern recognizer can identify shifted one dimensional patterns without error. In the latter we have demonstrated the ability of the spatio-temporal associative storage to store and retrieve the entire set of characters of the English alphabet (in the form of 5 × 7 black on white patterns) sequentially and in the correct order.

Acknowledgments

This work was performed with joint support from the U.S. Department of Energy, the National Science Foundation, and the Air Force Office of Scientific Research. The authors would like to thank Doyne Farmer, Alan Lapedes, Lee Giles, and Parvez Guzdar for extremely valuable conversations.

References

[1] Hopfield, J. J , Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences USA, 79, 2554-2558 (1983).

[1a] Fukushima, K., A self-organizing multilayered neural network, Biol. Cybern. 20, 121-136 (1975).

[2] Kohonen, T., Self-organization and associative memory, Springer-Verlag, N.Y.. (1984); Ackley, D. H., Hinton, G. E., and Sejnowski, T. J., A learning algorithm for Boltzmann machines, Cognitive Science 9, 147-169 (1985).

[3] Lapedes, A. S. and Farber, R. M., A self-optimizing, nonsymmetrical neural net for content addressible memory and pattern recognition, private communications.

[4] Sun, G. Z., Lee, Y. C., Chen, H. H., Memory capacity of a high order correlation memory network, University of Maryland Report (in preparation) (1986).

[5] Duda, R. O., Hart, P. E., and Nilsson, N. J., Subjective Bayesian methods for rule-based inference systems, Proceedings of the 1976 National Computer Conference (AFIPS Conference Proceedings) 45, 1075-1082 (1976); Pearl, J., Reverend Bayes on inference engines: A distributed hierarchical approach, Proceedings of AAAI Conference on Artificial Intelligence, Pittsburgh, Pennsylvania 133-136 (1982); Lowrance, J., Dependency-graph models of evidential support, COINS Technical Report 82-26, Univ. of Massachusetts at Amherst (1982).

[6] Miusky, Marvin, A framework for representing knowledge, The Phychology of Computer Vision, ed. Patrick Winston, McGraw-Hill Book Co., N.Y. (1975).

[7] Holland, J. H., Genetic Algorithm and Adaptation" Technical Report 34, Univ. of Michigan, Cognitive Science Dept. (1981).

[8] Mandelbrot, B., Fractal, W. H. Freeman and Co., San Francisco (1977).

[9] Farmer, J. D., Packard, N., Perelson, A. S., The immune system, adaptation and machine learning, Los Alamos National Laboratory Report LA-UR-86? (1985).

Figure Captions

Fig. 1. The average number of iterations required for a state to converge to a pattern in the initial pattern set. For P = 2 and the number of initial patterns greater than 32, no state converged in the initial set and the number 31 plotted merely indicates the lack of convergence. As the power, P, in the energy exponent increases, the average number of iteation decreases. 100,000 random initial states were processed for each initial pattern set.

Fig. 2. States can converge to patterns outside the initial pattern set. We call these converged states traps and plot the number of traps as a function of the number of initial patterns for four different values of the power, P, in the energy exponent. Again, 100,000 random initial states are introduced for each initial pattern set.

Fig. 3a. When 2048 states are processed, each state being exactly 2 Hamming distances away from a stored pattern, a fraction of the states converge to that pattern. As the number of initial patterns increases, the memory capacity decreases abruptly. The P = 2 curve drops decreases first; the P = 3 curve drops in the neighborhood of 2000 patterns; and the P = 5 curve is still near 1.0 for 10000 patterns.

Fig. 3b. The same as 4a except the initial Hamming distance is 12.

Fig. 3c. The same as 4a except 1.) the initial Hamming distance is 20 and 2.) the state is ony required to converge to some pattern in the initial pattern set (not necessarily the pattern used to create it by modifying 20 of its bits). A P = 9 curve is also included.

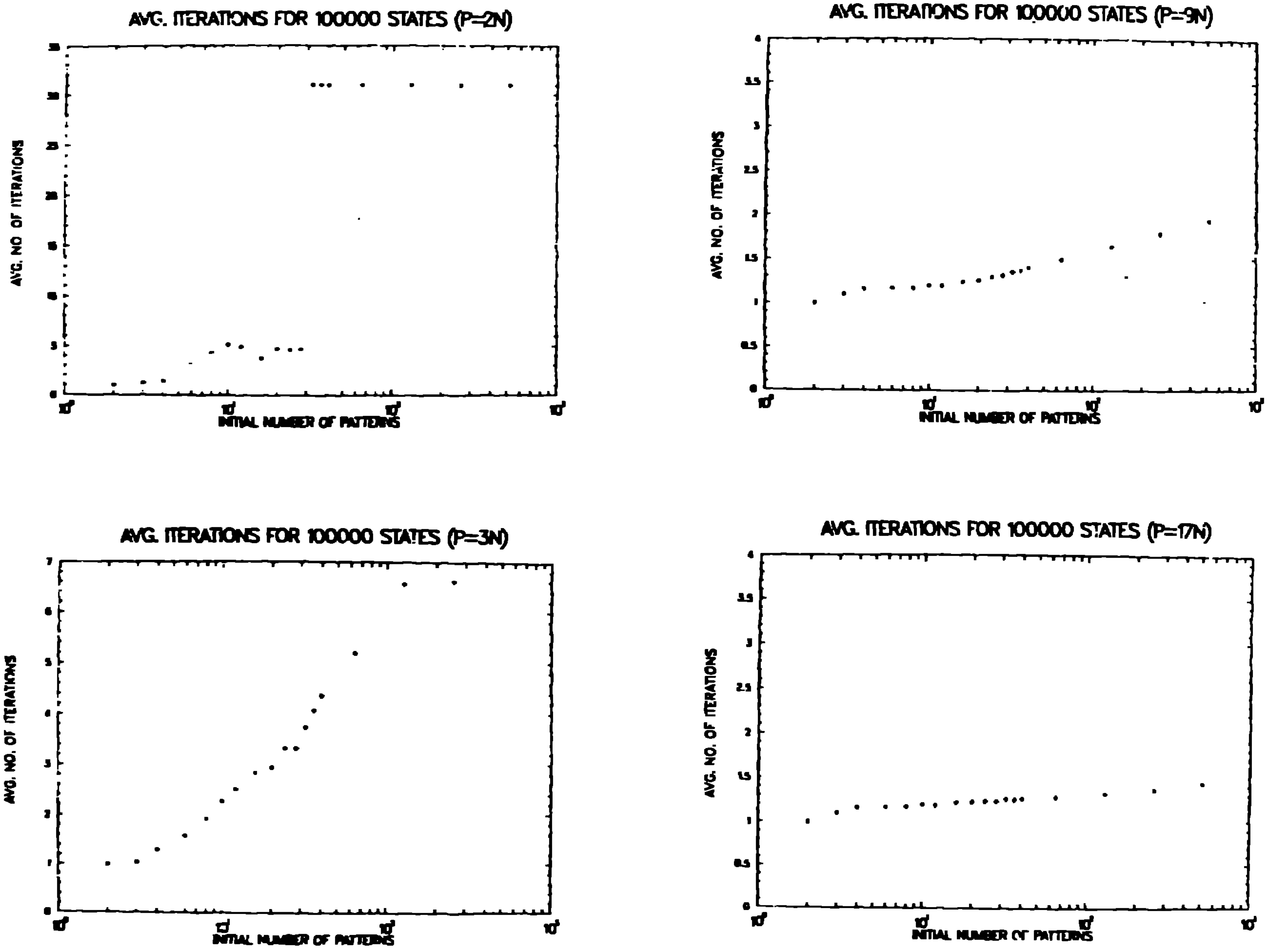
AVG. ITERATIONS FOR 100000 STATES (P=2N)
AVG. NO OF ITERATIONS
0
5
10
15
20
25
30
35
10^0
10^1
10^2
10^3
INITIAL NUMBER OF PATTERNS
AVG. ITERATIONS FOR 100000 STATES (P=9N)
AVG. NO. OF ITERATIONS
0
0.5
1
1.5
2
2.5
3
3.5
4
10^0
10^1
10^2
10^3
INITIAL NUMBER OF PATTERNS
AVG. ITERATIONS FOR 100000 STATES (P=3N)
AVG. NO. OF ITERATIONS
0
1
2
3
4
5
6
7
10^0
10^1
10^2
10^3
INITIAL NUMBER OF PATTERNS
AVG. ITERATIONS FOR 100000 STATES (P=17N)
AVG. NO. OF ITERATIONS
0
0.5
1
1.5
2
2.5
3
3.5
4
10^0
10^1
10^2
10^3
INITIAL NUMBER OF PATTERNS

Fig. 1.

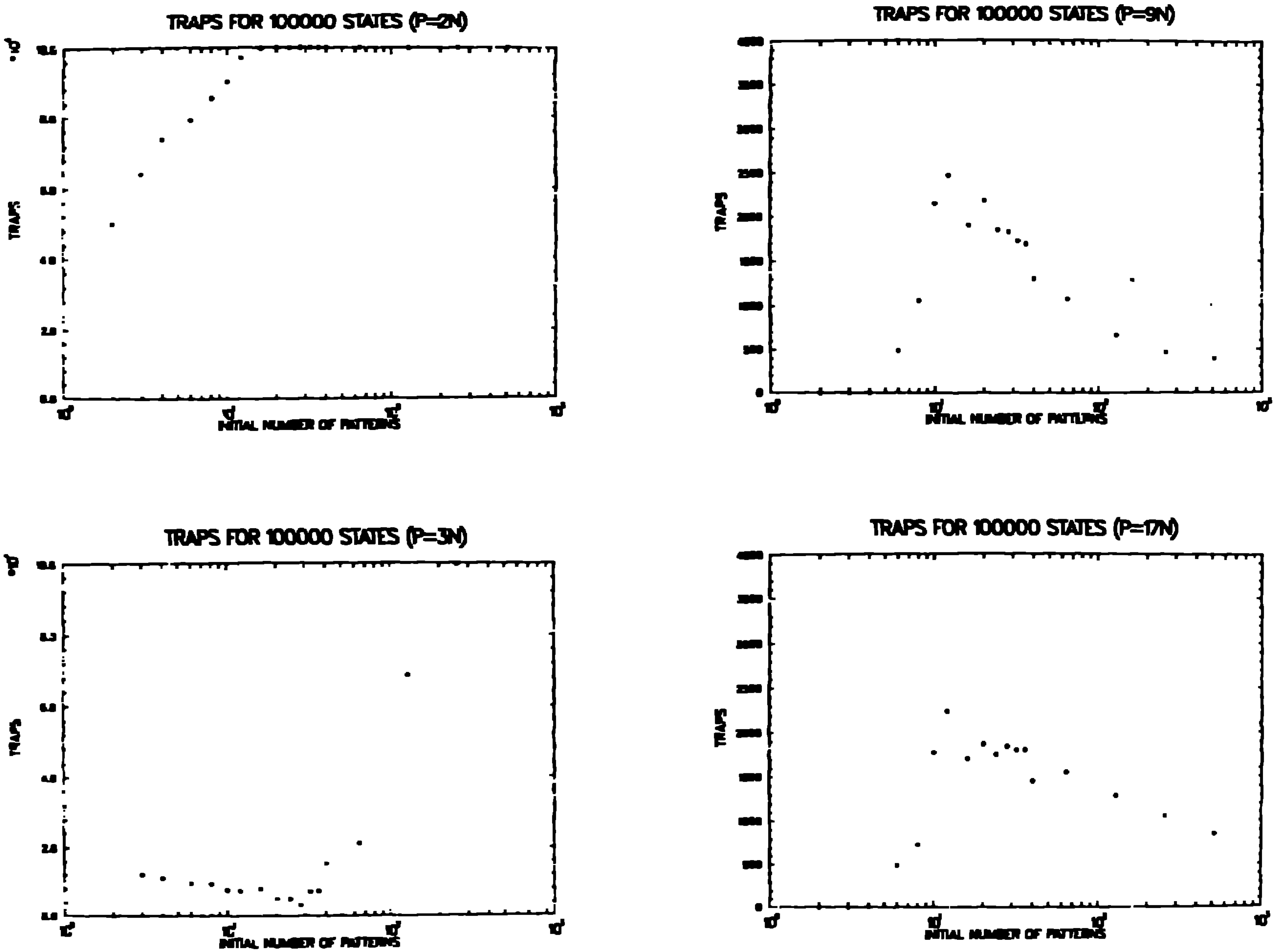
TRAPS FOR 100000 STATES (P=2N)
TRAPS
0.0
2.0
4.0
6.0
8.0
10.0
10^0
10^1
10^2
10^3
INITIAL NUMBER OF PATTERNS
TRAPS FOR 100000 STATES (P=9N)
TRAPS
0
500
1000
1500
2000
2500
3000
3500
4000
10^0
10^1
10^2
10^3
INITIAL NUMBER OF PATTERNS
TRAPS FOR 100000 STATES (P=3N)
TRAPS
0.0
2.0
4.0
6.0
8.0
10.0
10^0
10^1
10^2
10^3
INITIAL NUMBER OF PATTERNS
TRAPS FOR 100000 STATES (P=17N)
TRAPS
0
500
1000
1500
2000
2500
3000
3500
4000
10^0
10^1
10^2
10^3
INITIAL NUMBER OF PATTERNS

Fig. 2.

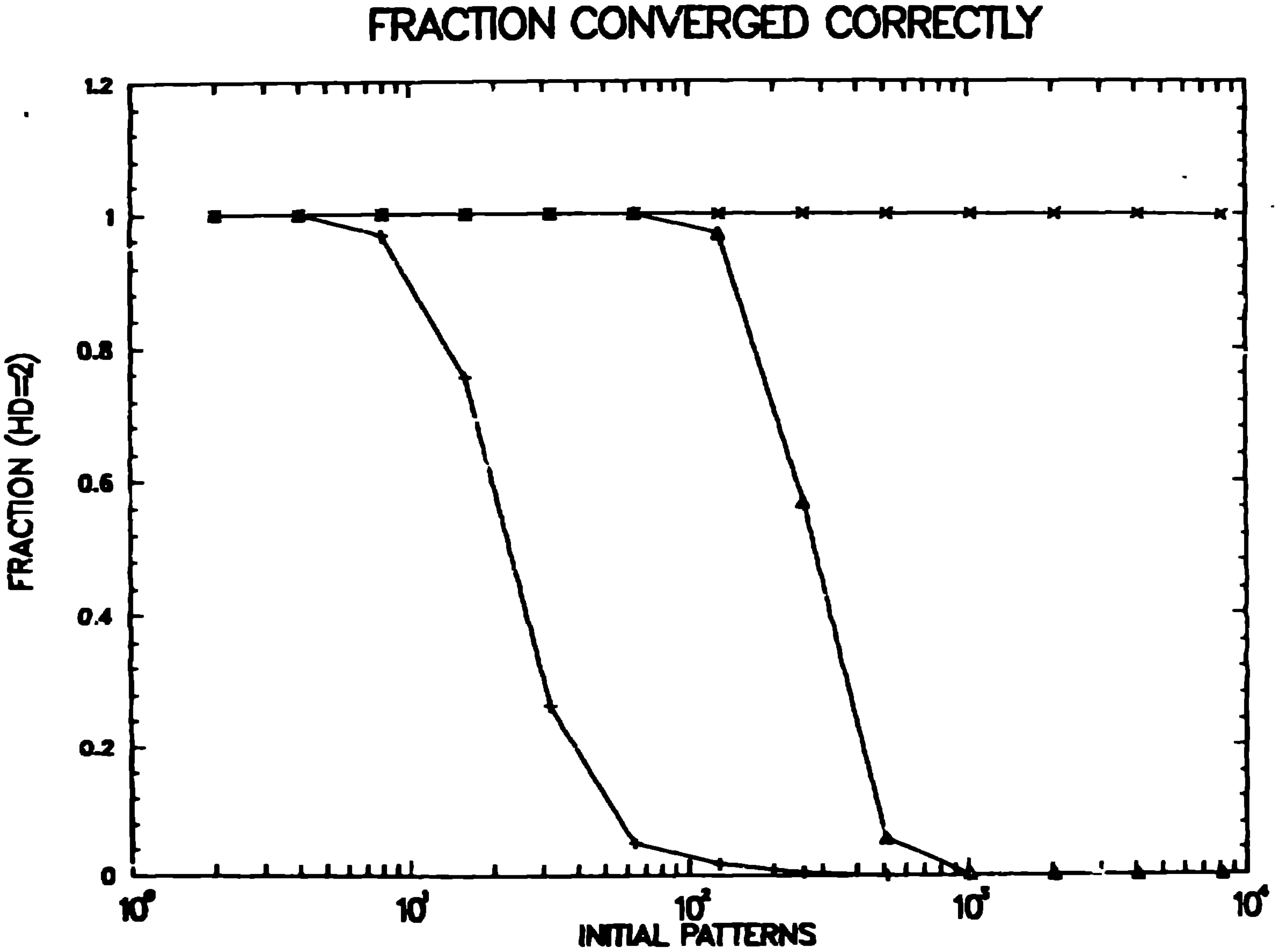
FRACTION CONVERGED CORRECTLY
FRACTION (HD=2)
1.2
1
0.8
0.6
0.4
0.2
0
10^0
10^1
10^2
10^3
10^4
INITIAL PATTERNS

Fig. 3a.

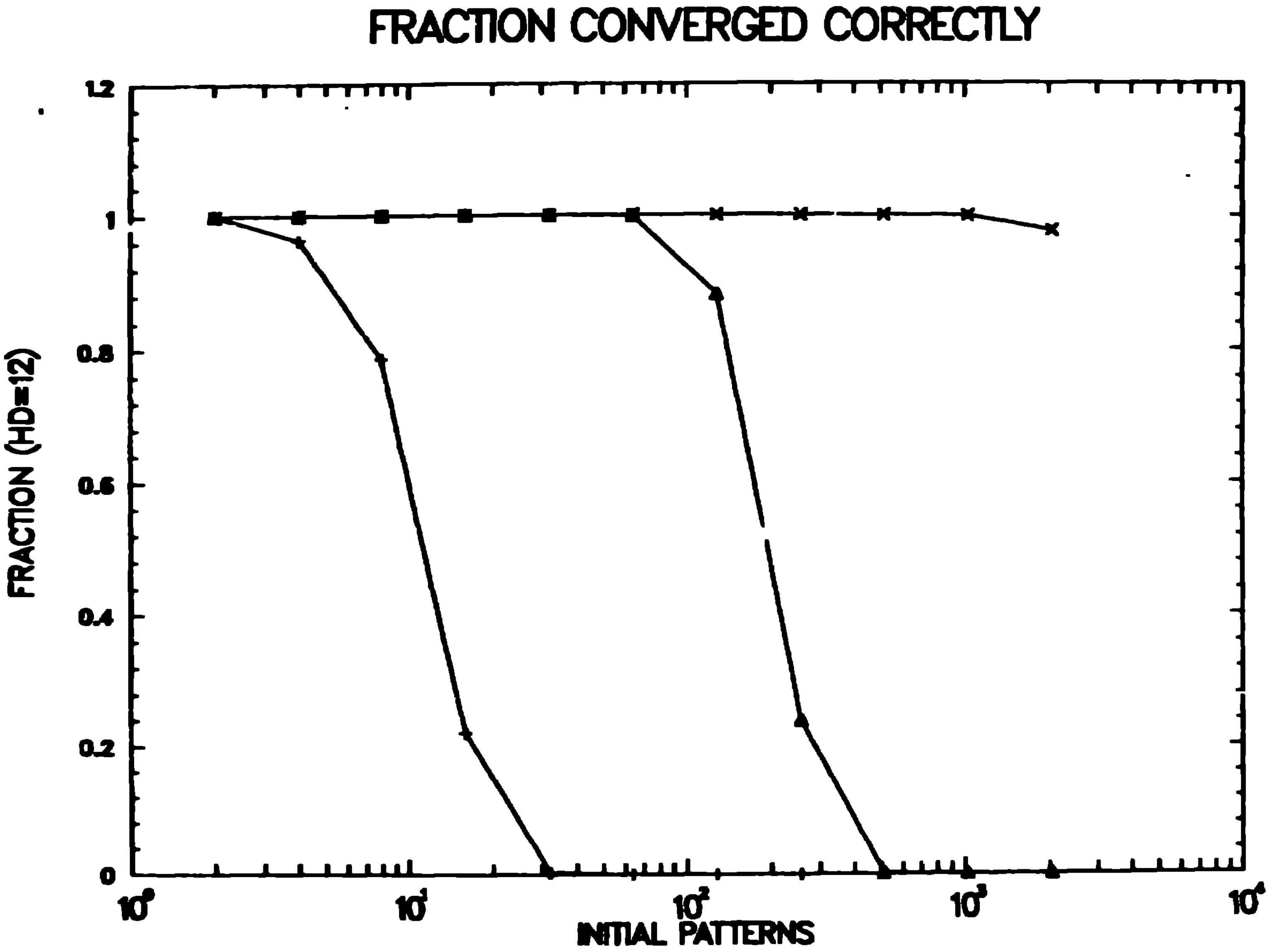
FRACTION CONVERGED CORRECTLY
FRACTION (HD=12)
1.2
1
0.8
0.6
0.4
0.2
0
10^0
10^1
10^2
10^3
10^4
INITIAL PATTERNS

Fig. 3b.

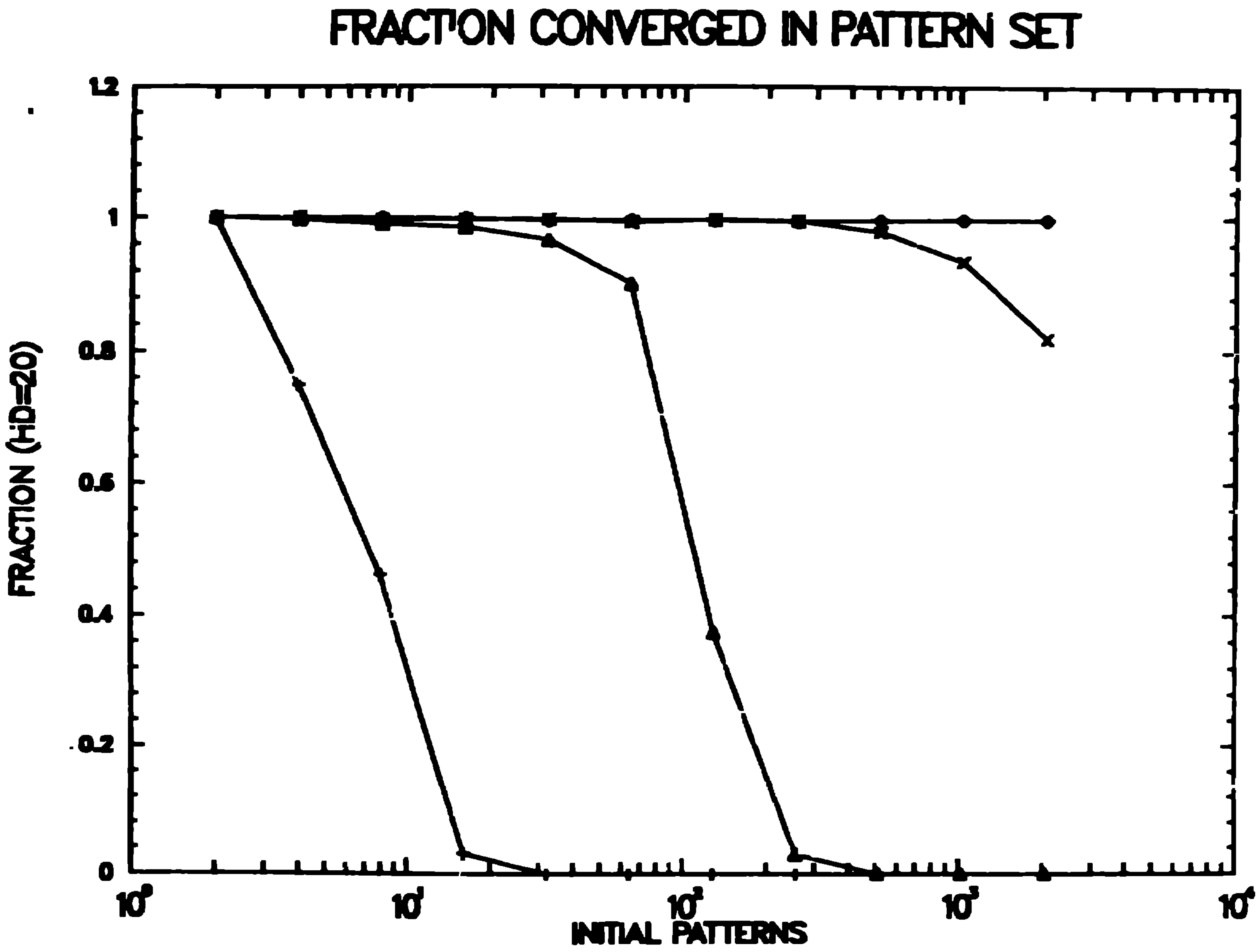
FRACTION CONVERGED IN PATTERN SET
1.2
1
0.8
0.6
0.4
0.2
0
FRACTION (HD=20)
10^0
10^1
10^2
10^3
10^4
INITIAL PATTERNS

Fig. 3c.