A PROBABILISTIC COMPUTATIONAL FRAMEWORK FOR NEURAL NETWORK MODELS

Technical Report AIP - 27

Richard M. Golden

Learning Research and Development Center
and Department of Psychology
University of Pittsburgh
Pittsburgh, PA 15260

The Artificial Intelligence and Psychology Project

Departments of
Computer Science and Psychology
Carnegie Mellon University

Learning Research and Development Center
University of Pittsburgh

A PROBABILISTIC COMPUTATIONAL FRAMEWORK FOR NEURAL NETWORK MODELS

Technical Report AIP - 27

Richard M. Golden

Learning Research and Development Center
and Department of Psychology
University of Pittsburgh
Pittsburgh, PA 15260

29 September 1987

This research was supported by the Computer Sciences Division, Office of Naval Research and DARPA under Contract Numbers N00014-86-K-0678 and N00014-86-K-0107; and in part by the Mellon foundation. I am very grateful to Dean C. Mumme for comments, criticism, and helpful discussions of an earlier version of this manuscript. I would also like to thank David B. Cooper for his suggestion that many neural network models be viewed within a unified probabilistic framework.

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY 2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; Distribution unlimited
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AIP 27	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Carnegie-Mellon University 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Computer Sciences Division Office of Naval Research (Code 1133)
6c. ADDRESS (City, State, and ZIP Code) Department of Psychology Pittsburgh, Pennsylvania 15213	7b. ADDRESS (City, State, and ZIP Code) 800 N. Quincy Street Arlington, Virginia 22217-5000
8a. NAME OF FUNDING / SPONSORING ORGANIZATION Same as Monitoring Organization 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-86-K-0678
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS p400005ub201/7-4-86 PROGRAM ELEMENT NO N/A; PROJECT NO N/A; TASK NO N/A; WORK UNIT ACCESSION NO N/A

11. TITLE (Include Security Classification)
A Probabilistic Computational Framework for Neural Network Models

12. PERSONAL AUTHOR(S)
R.M. Golden

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM 86Sept15 TO 91Sept14	87 September 29	38

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Artificial Intelligence, connectionism, non-linear associator

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

Information retrieval in a "connectionist" or neural network is viewed as computing the *most probable* value of the information to be retrieved with respect to a probability density function, P. With a minimal number of assumptions, the "energy" function that a neural network minimizes during information retrieval is shown to uniquely specify P. Inspection of the form of P indicates the class of probabilistic environments that can be learned. Learning algorithms can be analyzed and designed by using maximum likelihood estimation techniques to estimate the parameters of P. The large class of nonlinear auto-associative networks analyzed by Cohen and Grossberg (1983), nonlinear associative multi-layer back-propagation networks (Rumelhart, Hinton, & Williams, 1986), and certain classes of nonlinear multi-stage networks are analyzed within the proposed computational framework.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. Alan L. Meyrowitz	22b. TELEPHONE (Include Area Code) (202) 696-4302 22c. OFFICE SYMBOL N00014

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

A Probabilistic Computational Framework For Neural Network Models

Richard M. Golden

Learning Research and Development Center
and
Department of Psychology

University of Pittsburgh

Address correspondence before September 1, 1987 to:
Learning Research and Development Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15260

Home Phone: (412) 441-2261
Work Phone: (412) 624-7464

Address correspondence after September 1, 1987 to:
Department of Psychology
Jordan Hall, Bldg 420
Stanford University
Stanford, CA 94305

SUBMITTED
FOR PUBLICATION
6/87

Running head: COMPUTATIONAL FRAMEWORK FOR NEURAL NETWORKS

Abstract

Information retrieval in a "connectionist" or neural network is viewed as computing the most probable value of the information to be retrieved with respect to a probability density function, P. With a minimal number of assumptions, the "energy" function that a neural network minimizes during information retrieval is shown *to uniquely specify P*. Inspection of the *form of P* indicates the class of probabilistic environments that can be learned. Learning algorithms can be analyzed and designed by using maximum likelihood estimation techniques to estimate the parameters of P. The large class of nonlinear auto-associative networks analyzed by Cohen and Grossberg (1983), nonlinear associative multi-layer back-propagation networks (Rumelhart, Hinton, & Williams, 1986), and certain classes of nonlinear multi-stage networks are analyzed within the proposed computational framework.

In this article, a straightforward procedure is proposed for constructing a computational theory for any neural network model (i.e., dynamical system) that is known to be minimizing some function during information retrieval. Within this framework, computation in neural network models is viewed with respect to a MAP estimation (Van Trees, 1968) framework as opposed to the classic Turing machine view of computation. A theory characterizing the information processing computations of a neural network model is useful for several reasons. First, a computational theory allows one to compare and contrast quite different neural network models (algorithms) within the context of a unified theoretical framework. Second, since a computational theory provides independent arguments which specify the unique computational goal of a network model and why that computational goal is optimal, the optimality of a given neural network can be evaluated. Third, a computational theory may provide useful insights into neural network analysis and design problems. And finally, a computational theory provides a convenient and concise language for describing the behavioral goals of a neural network model.

In particular, a MAP (maximum a posteriori) estimation approach (e.g., Van Trees, 1968) to information storage and retrieval forms the foundations of the probabilistic computational framework for neural network models that is proposed in this article. Let $\mathbf{I}$ represent a retrieval cue to some memory system. The goal of information retrieval is to recall a vector $\mathbf{O}^*$ that is a global maximum of the a posteriori density function $p(\mathbf{O}|\mathbf{I};\mathbf{A})$ where $\mathbf{A}$ specifies the density function's parameters. The goal of learning is to find an $\mathbf{A}^*$ that is a global maximum of the a posteriori density function $p(\mathbf{A}|\mathbf{T}_s)$ where $\mathbf{T}_s$ is a set of vectors that were taught to the model. Less formally, the goal of information retrieval is to recall the most probable value of the unknown information, while the goal of learning is to acquire the most probable probabilistic laws of the environment.

No e that a MAP estimation approach to information processing is consistent with basic axioms of rational decision making (Henrion, 1986; Savage, 1971; Van Trees, 1968), with the symbolic logic (Cox, 1946), and yields minimum probability of error decisions (Van Trees, 1968) Thus, in accordance with Marr's (1982) framework, this article provides a computational theory that states the goal of a neural network's computation is to solve the MAP estimation problem, and cites formal arguments that indicate when such a goal is uniquely appropriate.

Some progress towards a computational theory of neural networks has recently been made by several researchers. Smolensky (1986) has formally shown that a small class of stochastic neural network models known as *Boltzmann machines are searching for the most probable* interpretation of some incoming information. Rumelhart, Smolensky, McClelland, and Hinton (1986) have noted that many neural network models can be viewed as maximizing a "goodness" measure but the quality and uniqueness of a given goodness measure were not considered. Golden (1987) (also see Golden, 1986b, and Anderson, Golden, & Murphy, in press) have noted that a class of deterministic auto-associative neural models are also searching for the most probable interpretation of their inputs.

Marroquin (1985) has argued for a description of the computational goals of a large class of algorithms using the probabilistic *framework of Markov random fields*. Such fields have been successfully used in the engineering literature to develop both deterministic and stochastic algorithms which have been applied to a variety of practical problems (e.g., Cohen & Cooper, 1987; Geman & Geman, 1984). Nevertheless, a Markov Random Field (MRF) framework is too restrictive for the issues addressed in this article. The primary orientation of this article is to provide a computational level of description, following Marr (1982), of a broad class of neural network models which includes MRF models as an important subclass.

Marr's (1982) framework for understanding the mind also includes two additional levels of description: the algorithmic level and the implementational level. The algorithmic level of description specifies an algorithm which is designed to solve the problem specified by the computational theory. As Rumelhart and McClelland (1985) have noted, this is the level of description that is most relevant to the perceptual/cognitive psychologist since the behavior of the algorithm and the behavior of people can be qualitatively compared at this level of description. In particular, the failures of a neural network algorithm can be compared to the failures of people in simple information processing tasks (e.g., McClelland & Rumelhart, 1986).

The third level of description of Marr's theory is the implementational level where the specific neural machinery used in the implementation of a given algorithm is described in detail. A neural network model is simply a dynamical system designed to perform some information processing task that possesses a neurally plausible intepretation. Neural networks typically consist of a collection of simple computing elements (suggestively referred to as units or neurons), and a set of connection strengths that indicate how the <u>activity level</u> of one unit in the system can influence the activity level of another unit. Thus, neural models may also serve as a guide for exploring those aspects of the neurophysiology that are *especially relevant* to information processing. The books by Grossberg (1982), Hinton and Anderson (1977), Kohonen (1984), and McClelland and Rumelhart (1986) provide useful introductory reviews of past and current research involving neural network models as information storage and retrieval systems.

This article is organized in the following manner. The first four sections introduce essential notation, provide an overview of the proposed computational framework, and provide examples regarding how the framework can be applied to many of the existing connectionist models in the literature. Following this informal presentation, the probabilistic computational framework is formally presented in section five.

Both the informal and formal presentations of the computational framework are organized into three major sections. First, the important concepts of an environmental and assumed probability density function (PDF) are introduced. The environmental PDF specifies the probability distribution of events in the environment, while the assumed PDF specifies the neural model's assumptions about the environmental PDF. Second, the problem of determining whether a model's assumed PDF can ever be made equivalent to a given environmental PDF is considered. The third section illustrates how maximum likelihood estimation procedures can be used to analyze and design learning algorithms for networks whose assumed PDFs are known.

1 Environmental and Assumed PDFs

1.1 The Environmental PDF

Consider a neural network dynamical system, D_s, whose state is represented by a d-dimensional state vector, $\mathbf{X}$, where the $\underline{i}$th element of $\mathbf{X}$ is the activity level or state of the $\underline{i}$th neural unit in the system. Define a set, S_p, whose elements are d-dimensional vectors. An <u>environmental PDF</u> is used to assign probabilities to subsets of S_p which indicate the relative frequency that a particular set of d neural states is externally imposed upon the d neural units by the environment. Note that the environmental PDF is *completely independent* of the dynamics of the neural network, D_s.

1.2 The Assumed PDF

If it is assumed that a given neural network model is using MAP estimation to recall information, then the a posteriori PDF that is used by the network to compute a MAP estimate is defined as the <u>assumed PDF</u>. Using the set S_p associated with the environmental PDF, the assumed PDF also assigns probabilities to subsets of S_p but is otherwise defined independently of the environment. A neural network's assumed PDF embodies the network's assumptions about the environmental PDF.

1.3 Using the Environmental and Assumed PDFs

Optimal learning and inferencing using environmental and assumed PDFs is now illustrated. To teach a network model a particular environmental PDF, set the network's assumed PDF equal to the environmental PDF. The inferencing problem is now considered. Let the vector $\mathbf{X}_{obs}$ be an event that is generated according to the environmental PDF. Now suppose the network's assumed PDF is equivalent to the environmental PDF, and suppose the network observes some (but not all) of the elements of $\mathbf{X}_{obs}$. Define an error to occur when the model's estimate of the unobservable vector elements are not equal to the actual values of the elements of $\mathbf{X}_{obs}$. With this definition of error, a MAP estimate of the unobservable vector elements is the estimate that minimizes the probability of error. Therefore, the inferences made by the network when computing a MAP estimate of $\mathbf{X}_{obs}$ using its assumed PDF minimize the network's probability of error.

2 Constructing Assumed PDFs

2.1 The Fundamental Theorem

Let $V(\mathbf{X})$ be an "energy" or additive dynamical system summarizing function for a neural network model that decreases as a function of time when the model is retrieving information from memory. Moreover, assume that $V(\mathbf{X})$ provides a sufficient amount of information to uniquely *specify the assumed PDF*. Given these assumptions and a physical constraint regarding how probabilities must be assigned to neural states, a "fundamental theorem" is stated and proved (following arguments by Smolensky, 1986) that says the assumed PDF for the network model is given uniquely by:

$$p(\mathbf{X}) = Z^{-1} \exp[-V(\mathbf{X})] \qquad (1)$$

where Z is a known normalization constant. The notation $\exp[x]$ indicates the exponential function evaluated at x.

2.2 Assumed PDFs for Auto-Associative Neural Networks

Let a system of nonlinear differential equations indicate how the activity level of a particular neuron is modified as a function of the activity levels of the other neurons in the system. If this nonlinear dynamical system maps some subset of points in the state space into either an equilibrium point or a limit cycle, then that dynamical system or auto-associative neural network may be viewed as a *categorization mechanism*. *Cohen and Grossberg (1983)* have shown how additive dynamical system summarizing functions for a large class of deterministic auto-associative neural networks in continuous-time may be constructed. Some popular special cases of their theorem include the continuous-time versions of the BSB model (Anderson et al., 1977), Hopfield's two-state neural model (Hopfield, 1982), Hopfield's (1984) general analysis of auto-associative networks of "semilinear" units, and the interactive activation model (McClelland & Rumelhart, 1981).

Cohen-Grossberg networks are shown to be ascent algorithms that are searching for a global maximum of their assumed PDF given some initial state. Suppose now that the initial activity levels over a subset of the neuronal units in the system are not permitted to change their value. That is, the activation pattern over this subset of units is clamped. Let $x_{m+1} \ldots x_d$ be the activity levels of the clamped units, and let $x_1 \ldots x_m$ be the activity levels of the remaining unclamped units. It is shown that a Cohen-Grossberg auto-associative network is searching for the values $x_1 \ldots x_m$ that maximize the a posteriori density function, $p(x_1 \ldots x_m | x_{m+1} \ldots x_d)$, which is derived from the network's assumed PDF, $p(\mathbf{X})$. That is, the network is searching for the most probable activation pattern over the unclamped units given some known activation pattern over the clamped units.

In particular, for the Hopfield (1982) and the Brain-State-in-a-Box (BSB) neural model (Anderson, Silverstein, Ritz, & Jones, 1977; Golden, 1986a), the assumed PDF is simply

$$p(X) = Z^{-1}\exp[X^T A X] \quad (2)$$

where **X** is a vector whose ith element is the activity level of the ith unit in the system, Z is a known normalization constant, and the ijth element of the **A** matrix is the connection strength between the ith and jth units in the system. Note that (2) is also the assumed PDF for the Boltzmann machine neural network model (Hinton & Sejnowski, 1986) and the Harmony theory neural network model (Smolensky, 1986).

2.3 Assumed PDFs for Back-Propagation Neural Networks

An important algorithm for learning in deterministic neural networks of semilinear units is the back-propagation learning algorithm of Rumelhart, Hinton, and Williams (1986) (also see Parker, 1985, and Le Cun, 1985, for related algorithms). A semilinear deterministic neuron's activation level, $x_i(t+1)$, at time t+1 is simply updated using the following equation:

$$x_i(t+1) = S_i[\sum_j a_{i,j} x_j(t)] \quad (3)$$

where $S_i[\,]$ is a monotonically increasing and differentiable *(i.e., sigmoidal) function*, and $a_{i,j}$ is the connection strength between the ith and jth neurons in the system.

Now consider a set of semilinear neurons that are connected to one another in some arbitrary manner through appropriate selection of the coefficients $a_{i,j}$. Now arrange these coefficients in a parameter vector, **A**. Let **I** be a vector whose ith element is the activation level of the ith input unit. Let **O** be a vector whose ith element is the activation level of the ith output unit. The term visible unit is used to refer to any unit that is either an input or an output unit. The remaining elements in the system are called the hidden units because these units only interact with the input and output units and have no direct interactions with the environment. For convenience, let the complete configuration of the network be specified by some highly

nonlinear associative vector-valued function ϕ_A such that during information retrieval $O = \phi_A(I)$ where the parameter vector, **A**, specifies the connection strength values.

The back-propagation learning algorithm is a gradient descent algorithm that can be used to modify the parameter vector **A** such that a parameter vector A^* is found that minimizes:

$$\sum_i P_e(O_i, I_i)\,|O_i - \phi_A(I_i)|^2 \quad (4)$$

where the pair $[O_i, I_i]$ is the ith association to be learned by the system, the summation is taken over all such pairs, and $P_e(O_i, I_i)$ is the probability that the ith association occurs in the system's environment. Note that an important neurally plausible special case of the back-propagation algorithm is the Widrow-Hoff learning rule. The Widrow-Hoff rule, in turn, is a generalization of the Hebbian learning rule when the vectors in the environmental PDF are orthogonal. Good reviews of these learning rules may be found in Anderson et al. (1977; in press), Kohonen (1984), and Sutton and Barto (1981).

Because the back-propagation learning algorithm is minimizing a mean square error cost function, a natural additive dynamical system summarizing function associated with information retrieval for a constant input vector, **I**, is:

$$V(O) = |O - \phi_A(I)|^2. \quad (5)$$

Using (5) in conjunction with the fundamental theorem, the assumed PDF for an associative back-propagation network is shown to have the following form:

$$p(O|I) = (\exp[-|O - \phi_A(I)|^2]) / \pi^{d/2}. \quad (6)$$

Thus, associative back-propagation networks are algorithms that compute the most probable d-dimensional output vector, **O**, for a given input vector, **I**, where the PDF is given by (6). Or more precisely, these networks retrieve the MAP estimate associated with the a posteriori density in (6).

The mean square error function in (4) is most appropriate when the output vector, **O**, is a continuous vector-valued variable. When the elements of **O** are binary-valued, Hinton (1987) has suggested an appropriate variant of the back-propagation learning algorithm which searches for a parameter vector $\mathbf{A}^*$ such that the following function of **A** is maximized.

$$\sum_j \sum_i P_e(\mathbf{O}_j,\mathbf{I}_j)\,[o_{j,i} LOG[p_i(\mathbf{A},\mathbf{I}_j)] + (1 - o_{j,i}) LOG[1 - p_i(\mathbf{A},\mathbf{I}_j)]] \qquad (7)$$

where $[\mathbf{O}_j,\mathbf{I}_j]$ is the jth association, $o_{j,i}$ is the ith element of $\mathbf{O}_j$, and $p_i(\mathbf{A},\mathbf{I})$ is the ith element of $\Phi_A(\mathbf{I})$. It is also assumed that the range of the sigmoidal functions associated with the semilinear units in the system is such that $0 \le p_i(\mathbf{A},\mathbf{I}) \le 1$.

The natural additive *dynamical system summarizing function* associated with (7) during *information retrieval* is therefore:

$$V(\mathbf{O}) = -\sum_i [o_i LOG[p_i(\mathbf{A},\mathbf{I})] + (1 - o_i)\,LOG[1 - p_i(\mathbf{A},\mathbf{I})]] \qquad (8)$$

where the ith element of **O**, o_i, can only take on the values of zero or one, and $0 \le p_i(\mathbf{A},\mathbf{I}) \le 1$. Note that a global minimum of $V(\mathbf{O})$ over the discrete state space occurs whenever $o_i = 1$ if $p_i(\mathbf{A},\mathbf{I}) > 0.5$ and $o_i = 0$ if $p_i(\mathbf{A},\mathbf{I}) < 0.5$. The assumed PDF for $V(\mathbf{O})$ in (8) is found using the fundamental theorem to be:

$$P(\mathbf{O}|\mathbf{I}) = \prod_i [o_i p_i(\mathbf{A},\mathbf{I}) + (1 - o_i)\,[1 - p_i(\mathbf{A},\mathbf{I})]] \qquad (9)$$

Finally note that $p_i(\mathbf{A},\mathbf{I})$ may be interpreted as $P(o_i = 1|\mathbf{I})$ if it is assumed that the elements of **O** are statistically independent.

2.4 Assumed PDFs for Multi-Stage Neural Networks

The fundamental theorem is also applied to a large class of serial multiple stage neural networks where a "stage" in this class of networks might be an auto-associative network (e.g., a BSB neural network model) or a multi-layer associative network (e.g., an associative back-propagation neural network model). The concept of a *serial multiple stage network* is introduced, and a *multi-stage network theorem* is presented. The multi-stage network theorem justifies adding the *dynamical system summarizing functions* associated with each stage in the network to form a *dynamical system summarizing function* for the entire multi-stage network.

As an example of a possible application of the multi-stage network theorem, Schneider and his colleagues (Schneider & Detweiler, 1987; Schneider & Mumme, 1987) have been developing a multiple stage neural network architecture for modelling controlled and automatic processing which they refer to as CAP1. This architecture is characterized by a set of auto-associative memory systems whose outputs are channeled through linear associative memory systems. The vector-valued outputs of these associative memory systems are then summed. More formally, the critical dynamics of one version of the CAP1 system during the information retrieval process can be represented by the following system of difference equations:

$$\mathbf{X}_i(t + \Delta t) = S[\mathbf{M}_i\mathbf{X}_i(t)] \qquad (10)$$

$$\mathbf{X}_C(t + \Delta t) = \sum_{i=1}^{C-1} a_i \mathbf{A}_i \mathbf{X}_i(t)$$

where X_j is the state vector associated with the jth dynamical subsystem, S is a vector-valued sigmoidal function, a_j is a scalar, and A_j and M_j are matrices.

An additive dynamical system summarizing function, $V(X)$, for the CAP1 system represented in (10) may be constructed using the multi-stage network theorem. In particular,

$$V(X_1 \ldots X_C) = V_1(X_1) + V_2(X_2) + \ldots + V_{C-1}(X_{C-1}) + V_C(X_1 \ldots X_C) \qquad (11)$$

where $V_i(X_i) = - X_i^T M_i X_i$ for $1 \leq i \leq C\text{-}1$,

and where $V_C(X_1 \ldots X_C) = \sum_{j=1}^{C-1} |X_C - a_j A_j X_j|^2$.

Note that for the multi-stage network theorem to be strictly applicable, dynamical system summarizing functions for the auto-associators and linear associators must be found, and matrices must be constructed that eliminate local minima. Multi-stage networks of the form of (10) can be constructed that meet these requirements. Unfortunately, however, for the multi-stage network theorem to be strictly applicable to the CAP1 system it is also necessary to show that the state of an auto-associative subsystem actually reaches (and not simply "approaches") a global minimum of that subsystem's summarizing function. Such analyses are currently unavailable although extensive experience with simulations of the auto-associative BSB model indicates that the equilibrium points in this model are usually always reached. With this caveat, an assumed PDF for the system can be constructed by applying the fundamental theorem to the dynamical system summarizing function in (11).

3 Compatible Assumed and Environmental PDFs

Can a given neural model whose assumed PDF is a function of some set of parameters ever acquire complete knowledge of its probabilistic environment? To answer this question, simply set the assumed PDF equal to the environmental PDF and "solve" for the parameters of the assumed PDF. If the resulting system of equations does not have a solution, then that implies the neural model can never learn the environmental PDF. If a solution exists, then the assumed and environmental PDFs are compatible. Note the similarity of this type of argument to proofs suggested by Minsky and Papert (1969) or Hinton (1981) that indicate a perceptron can not solve the exclusive-or problem. The arguments in this section, however, are applicable to many nonlinear neural networks (as well as perceptrons) although the resulting conclusions about the performance of these systems are weaker.

The concept of compatible PDFs can be used to construct rigorous arguments that justify specific neural network model learning schemes. For example, a necessary condition for an environmental PDF with K global maxima to be compatible with a particular assumed PDF is that K global maxima of the assumed PDF exist which correspond to the K global maxima of the environmental PDF. The assignment of global minima of an energy function to stimulus set members that are to be learned by Cohen-Grossberg auto-associative neural networks has been suggested by several research groups. Anderson and his colleagues have used this procedure to train the BSB model (Anderson et al., 1977, in press; Golden, 1986a, 1986b, 1987). Hopfield (1982) used this procedure to train his auto-associative network of two-state neurons. Rumelhart et al. (1986) and Plaut, Nowlan, and Hinton (1986) have used this procedure to train auto-associative networks of semilinear units.

A compatibility test for networks of two-state neurons is also derived. The test is based on inspecting the rank of a particular matrix whose construction is dependent upon both the stimulus set and the neural network architecture. The matrix is called the <u>compatibility</u> matrix because it indicates whether the assumed PDF of a specific neural network model is compatible with any environmental PDF defined with respect to a specific stimulus vector set. To illustrate the construction and use of compatibility matrices, consider an environmental PDF that assigns non-zero probabilities to the vectors:

$$X_1 = (1\ 0\ 1) \quad X_2 = (0\ 1\ 1) \quad X_3 = (1\ 1\ 0) \quad X_4 = (0\ 0\ 0)$$

The jth row of the compatibility matrix for a BSB model which stores only second-order correlations is:

$$[x_1 x_2 \qquad x_2 x_3 \qquad x_1 x_3]$$

where x_j is the jth element of X_i. The complete compatibility matrix is therefore:

$$\begin{matrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{matrix}$$

The rank of the compatibility matrix in this case is three which is equal to the number of rows of the matrix, so the assumed PDF is compatible with any environmental PDF defined with respect to the stimulus set.

The general procedure for constructing a compatibility matrix for a stimulus set of M d-dimensional vectors, $\{X_1 ... X_M\}$, is now described. Define the vector-valued function, $F(C)$, to have the following *row vector form:*

$$F(C) = [c_1, c_2, \ldots c_i, c_1 c_2, \ldots, c_i c_j, \ldots c_1 c_2 c_3, \ldots c_i c_j c_k, \ldots \prod_{i=1}^{d} c_i] \qquad (12)$$

where c_i is the ith non-zero element of C. To find the function $F(C)$ for a given dynamical system, rewrite the network's additive dynamical system summarizing function as a dot product of the parameter vector, A, and $F(X - X_M)$. This can always be done using arguments provided by Besag (1974) (also see Anderson et al., in press). Note that the dimensionality, d_a, of A is less than or equal to 2^{d-1}. For example, $d_a \leq d(d-1)/2$ for the assumed PDF in (2). Then,

$$W^T = [F(X_1 - X_M)^T, F(X_2 - X_M)^T, \ldots F(X_{M-1} - X_M)^T] \qquad (13)$$

where W^T denotes the transpose of the $M - 1$ by d_a dimensional compatibility matrix, W.

4 Design and Analysis of Learning Algorithms using ML Estimation

According to the computational framework presented here, the goal of learning is to compute the <u>most probable</u> values (i.e., MAP estimates) of the parameters of the assumed PDF given a [illegible] of observations of values (i.e., training vectors) of a random variable generated by some stationary environmental PDF. Given negligible prior knowledge about the assumed PDF's parameters relative to the number of environmental observations, the MAP estimation problem reduces to the computationally tractable Maximum Likelihood (ML) estimation problem (e.g., Van Trees, 1968).

Learning in connectionist systems is formulated in terms of ML estimation as follows. An environmental PDF is used to generate N values of some random vector-valued variable. The network is given these N vectors as a training sequence of length N, and then searches for those parameters of the assumed PDF that maximize a <u>likelihood function</u>. A parameter vector of the assumed PDF which is a global maximum of the likelihood function makes the event of observing the training sequence <u>most probable</u>. To compute the likelihood function, the network must assume that the vectors in the training sequence are independent and identically distributed (i.i.d.) according to the assumed PDF. Maximum likelihood estimation yields efficient, unbiased estimates for sufficiently long training sequences (Van Trees, 1968). Finally, note that when the environmental PDF is not stationary, a ML estimation approach is usually not appropriate although analyses of learning in non-stationary environments are still possible (Grossberg, 1987; Macchi & Eweda, 1983).

In most connectionist learning schemes, only a finite number of vectors are taught to the model. This suggests that the environmental PDF that generates the elements of the training sequence may be viewed as a discrete PF. If the length of the training sequence is sufficiently large, then the logarithm of the likelihood function is shown to converge to the asymptotic likelihood function, $E(A)$, when the environmental PF is discrete following informal arguments by Frieden (1983, 1985) and Wise (1986). Thus, optimal (ML) learning algorithms for neural networks whose assumed PDFs are known and which are functioning in environments characterized by discrete PDFs can be designed with standard optimization techniques (e.g., Luenberger, 1984) by maximizing $E(A)$ with respect to A. The asymptotic likelihood function, $E(A)$, is computed using the assumed PDF of the network, $p(X;A)$, and the environmental PF, $P_e(X)$, as follows:

$$E(A) = \sum_i P_e(X_i) \,\mathrm{LOG}\, [p(X_i;A)] \tag{14}$$

where X_i is the *i*th element of the training set which occurs with probability $P_e(X_i)$.

In the limit, gradient ascent upon the logarithm of the likelihood function is shown to be equivalent to gradient descent upon the cross-entropy function (Kullback, 1959; Shannon, 1963) or gradient ascent upon the asymptotic likelihood function. Thus, because the neural network learning algorithms for the Boltzmann machine (Hinton & Sejnowski, 1986) and Harmony theory

(Smolensky, 1986) are gradient descent algorithms that minimize the cross-entropy function, these algorithms are also maximum likelihood estimation algorithms that are estimating the parameters of their assumed PDFs. Moreover, the back-propagation learning algorithm, using either the assumed PDF in (8) or (9) is shown to be a gradient ascent algorithm that maximizes the asymptotic likelihood function as well. Thus, the associative back-propagation learning algorithm is also a maximum likelihood estimation algorithm. Such analyses are illustrative of how learning algorithms for networks whose assumed PDFs are known can be analyzed and designed by simply examining their asymptotic likelihood functions.

5 Formal Presentation of the Computational Theory

The following notation is used to specify probability density functions unless otherwise stated. Let $P(x < X)$ be the probability that the continuous random variable x is less than the constant X. The continuous probability density function, $p(X)$, associated with x is defined as

$$p(X) = dP(x < X)/dX.$$

If x is a discrete random variable whose *i*th value, X_i, is assigned a probability, $P(X_i)$, then the discrete probability density function associated with X can still be expressed using Dirac delta functions as follows:

$$p(X) = \sum_i P(X_i)\, \delta(X - X_i).$$

Note that the function $P(X)$ specifies a probability function, PF, which assigns a probability to a particular value of x, while the function $p(X)$ is the probability density function associated with the random variable x.

5.1 The Fundamental Theorem

In this section, using a series of arguments analogous to those of Smolensky (1986), a *fundamental theorem concerning the uniqueness of the assumed PDF for a given network model* will be proved.

Definition of a dynamical system summarizing function. Let σ denote a type of stochastic or deterministic convergence (e.g., Cauchy, in probability, almost sure). Let D_s denote a stochastic or deterministic dynamical system with state $X(t) \in S_d$ where S_d is a state vector space. Let $V(X)$ be a real scalar-valued function of X. Let $X^* \in S_d$ such that $V^* = V(X^*) \leq V(X)$ *for all* $X \in S_d$. *The function* $V(X(t))$ *is a* dynamical system summarizing function (d.s.s.f.) of type σ *if and only if* $V(X(t)) \rightarrow V^*$ *as* $t \rightarrow \infty$ *in the sense specified by* σ.

Definition of an additive d.s.s.f. Let the jth element of a d-dimensional vector X be the activation level of the jth neuron in some neural network, D_s. Let X be partitioned into two subvectors such that $X = (X_1, X_2)$ where the subnetwork, σ_1, of m neurons whose state is specified by the m-dimensional vector X_1 is physically unconnected with the subnetwork, σ_2, of d-m neurons whose state is specified by the d-m dimensional vector X_2. Let $V_k(X)$ denote a d.s.s.f. that maps a k-dimensional vector into a real number. Then, an additive d.s.s.f. $V_d(X)$ has the property that

$$V_d(X) = V_m(X_1) + V_{d\text{-}m}(X_2) \tag{15}$$

when σ_1 and σ_2 are physically unconnected for at least one value of m.

Sufficient information property. Let $V_d(X)$ be an additive d.s.s.f. for a neural network, D_s. A value of the function V_d provides a sufficient amount of information to specify the unique value

of the network's assumed PDF, p. In particular, $p = G(V_d)$ where G is a continuous and differentiable function. In addition, if D_s consists of two physically unconnected subnetworks with additive d.s.s.f.s $V_m(X_1)$ and $V_{d\text{-}m}(X_2)$ as defined in (15), then $p_m = G(V_m)$ and $p_{d\text{-}m} = G(V_{d\text{-}m})$ where p_m and $p_{d\text{-}m}$ *are the assumed PDFs for the two subnetworks.*

Neural network independence property. Let $V_d(X)$ be an additive d.s.s.f. for a neural network, D_s, with assumed PDF, p. Given that D_s consists of two physically unconnected subnetworks with additive d.s.s.f.s $V_m(X_1)$ and $V_{d\text{-}m}(X_2)$ as defined in (15) whose assumed PDFs, p_m and $p_{d\text{-}m}$, are constructed according to the sufficient information postulate, then $p = p_m p_{d\text{-}m}$.

Definition of an assumed PDF. An assumed PDF, $p(X)$, of a dynamical system, D_s, defined with respect to an additive d.s.s.f., $V(X)$, of type σ has the sufficient information and neural network independence properties. In addition, $-\text{LOG}[p(X)]$ is a d.s.s.f. for D_s of type σ as well.

A Fundamental Uniqueness Theorem for Constructing Assumed PDFs. Given an additive d.s.s.f., $V(X)$, which is defined with respect to some dynamical system, D_s, and state vector space, S_d, *the assumed PDF for* D_s *is uniquely given by:*

$$p(X) = Z^{-1}\exp(-V(X)) \tag{16}$$

$$\text{provided } Z = \int \exp(-V(X))\, dX \text{ is finite,} \tag{17}$$

where the integral in (17) is taken over all elements of S_p which is a subset of the dynamical system state space, S_d.

Proof of the Fundamental Theorem. First note, if an event X is such that $p(X)$ must equal zero, then it is necessary to eliminate $\mathbf{X}$ from the set S_p. Now, consider the case where D_d consists of two physically unconnected subnetworks with additive d.s.s.f.s $V_m(\mathbf{X}_1)$ and $V_{d-m}(\mathbf{X}_2)$ as defined in (15). Let $V_1 = V_m(\mathbf{X}_1)$, and let $V_2 = V_{d-m}(\mathbf{X}_2)$ where $V_k(\mathbf{X})$ maps a k-dimensional vector $\mathbf{X}$ into a scalar. Now by the neural network independence property,

$$p(\mathbf{X}) = p(\mathbf{X}_1,\mathbf{X}_2) = p(\mathbf{X}_1)\,p(\mathbf{X}_2) = G(V_1)\,G(V_2)$$

By the sufficient information property, $p(\mathbf{X}) = G(V_d(\mathbf{X})) = G(V_1 + V_2)$.

$$\text{Thus, } G(V_1 + V_2) = G(V_1)\,G(V_2)$$

$$dG(V_1 + V_2)/dV_1 = G(V_2)\,dG(V_1)/dV_1$$

$$dG(V_1 + V_2)/dV_2 = G(V_1)\,dG(V_2)/dV_2$$

Equating the left hand sides of the above two equations, dividing by the strictly positive $G(V_1)G(V_2)$, and forming an equivalent relationship in the form of an ordinary differential equation with $-1/T$ as the separation constant we obtain:

$$[dG(V_1)/dV_1] \,/\, G(V_1) = -1/T$$

$$dG(V_1)/dV_1 = -\,G(V_1)/T \tag{18}$$

Equation 18 can then be solved to obtain a particular solution as follows.

$$\int dG(V_1)/G(V_1) = \int -dV_1/T + C$$

$$G(V_1) = Z^{-1}\exp[-V_1/T]$$

Because the right hand side of (18) is continuous and differentiable, this solution is unique by Picard's Theorem (Simmons, 1972). Now since $-LOG[p(\mathbf{X})] = -LOG[Z^{-1}\exp[-V(\mathbf{X})/T]$, T must be positive so that as $V(\mathbf{X})$ decreases, $-LOG[p(\mathbf{X})]$ also decreases as required by the definition of an assumed PDF. Also note that $V(\mathbf{X})$ is an additive d.s.s.f. if and only if $V(\mathbf{X})/T$ is an additive d.s.s.f. Thus, the parameter T affects the uniqueness of $p(\mathbf{X})$ in a trivial manner and can be set equal to unity without any loss in generality. Finally, since $\int p(\mathbf{X})\,d\mathbf{X} = 1$, Z is uniquely determined by (17).

Q.E.D.

5.2 Assumed PDFs for Auto-Associative Neural Networks

The following theorem represents a synthesis of some of the results presented in Cohen and Grossberg (1983). Additional results concerning this class of dynamical systems have also been obtained by Cohen and Grossberg (1983).

<u>Cohen and Grossberg Theorem.</u> Consider the large class of *continuous-time neural network* models defined by:

$$dx_i/dt = a_i(x_i)[b_i(x_i) - \sum_{k=1}^{d} a_{i,k} S_k(x_k)] \tag{19}$$

where x_i is the activation level of the ith neuron in the d-neuron system, $a_i(x_i)$ is an arbitrary function of x_i such that $a_i(x_i) > 0$ for all x_i in some set S_d. The function $S_k(x_k)$ is a continuous, differentiable, monotonically increasing function of all x_k in S_d. The function $b_i(x_i)$ is an arbitrary continuous function of x_i for all x_i in S_d. The coefficient $a_{i,k} = a_{k,i}$ for all i and k.

$$\text{Let } V(\mathbf{X}) = -\sum_{i=1}^{d} \int_0^{x_i} b_i(u_i)\,S'_i(u_i)\,du_i + (1/2)\sum_{j=1}^{d}\sum_{k=1}^{d} a_{j,k}\,S_j(x_j)S_k(x_k) \tag{20}$$

where $\mathbf{X}$ is a d-dimensional vector whose ith element is x_i, and $S'_i(u_i)$ is the derivative of $S_i(u)$ with respect to u, and evaluated at u_i.

The function $V(\mathbf{X})$ is an additive d.s.s.f. provided that V is continuous and has continuous first partial derivatives, and an equilibrium point, $\mathbf{X}^*$, exists such that $\mathbf{X}^*$ is a global minimum of $V(\mathbf{X})$. Moreover, $\mathbf{X}^*$ must be a unique global minimum of $V(\mathbf{X})$ with respect to some subset, R^*, of the state vector space, S_d.

Proof. First note that V is additive. Moreover, Cohen and Grossberg (1983) note that $dV(\mathbf{X})/dt \leq 0$. Since V is continuous, has continuous first partial derivatives, and possesses a unique global minimum at $\mathbf{X}^*$ with respect to R^*, V is a Liapunov function (Luenberger, 1979) with respect to R^*. Therefore, for a given $\epsilon > 0$, both an $\mathbf{X}(0) \in R^*$ and a $t^* > 0$ exist such that for all $t > t^*$, $|\mathbf{X}(t) - \mathbf{X}^*| < \epsilon$.

Q.E.D.

Proposition. Let D_e be a Cohen-Grossberg network of the form of (19) when none of the units are clamped which is defined with respect to a dynamical state space, S_d. Let S_p be a subset of S_d. Let $V(\mathbf{X})$ be the d.s.s.f. associated with (19) and defined in (20). If the integral in (17) over S_p is finite, then the assumed PDF, $p(\mathbf{X}) = p(x_1 \ldots x_d)$, of the Cohen-Grossberg network is uniquely given by (16) and (17) with respect to $V(\mathbf{X})$ and S_p. Moreover, an assumed PDF for the network when units m+1...d are clamped is

$$p(x_1 \ldots x_m | x_{m+1} \ldots x_d) = p(x_1 \ldots x_d)/p(x_{m+1} \ldots x_d) \qquad (21)$$

where $p(x_{m+1} \ldots x_d) = \int p(y_1 \ldots y_m, x_{m+1} \ldots x_d)\, dy_1 \ldots dy_m$

Proof. The first part of the proposition follows immediately from direct application of the fundamental uniqueness theorem. The case where units m+1...d are clamped is now considered. In this case, the original system of d differential equations as represented in (19) reduces to a system of m differential equations of the following form because units m+1...d are clamped.

$$dx_i/dt = a_i(x_i)[b_i(x_i) - \sum_{k=1}^{d} a_{i,k} S_k(x_k)] \qquad (22)$$

Separating the clamped terms from the unclamped terms in (22) we have:

$$dx_i/dt = a_i(x_i)[b_i(x_i) - \sum_{k=m+1}^{d} a_{i,k} S_k(x_k) - \sum_{k=1}^{m} a_{i,k} S_k(x_k)]$$

where $x_{m+1} \ldots x_d$ are constants. The d.s.s.f. for (22) is obtained using the Cohen-Grossberg Theorem as follows:

$$V(x_1 \ldots x_m) = -\sum_{i=1}^{m} \int_0^{x_i} [b_i(u_i) - \sum_{k=m+1}^{d} a_{i,k} S_k(x_k)] S'_i(u_i)\, du_i + (1/2) \sum_{j=1}^{m} \sum_{k=1}^{m} a_{j,k} S_j(x_j) S_k(x_k)$$

Now noting that $\sum_{i=1}^{m} \int_0^{x_i} \sum_{k=m+1}^{d} a_{i,k} S_k(x_k) S'_i(u_i)\, du_i = \sum_{i=1}^{m} \sum_{k=m+1}^{d} a_{i,k} S_k(x_k)[S_i(x_i) - S_i(0)]$

where $S_i(0)$ is a constant, the following expression is obtained for $V(x_1 \ldots x_m)$:

$$V(x_1 \ldots x_m) = -\sum_{i=1}^{d} \int_0^{x_i} b_i(u_i) S'_i(u_i)\, du_i + (1/2) \sum_{i=1}^{d} \sum_{k=1}^{d} a_{i,k} S_k(x_k) S_i(x_i) + C$$

$$V(x_1 \ldots x_m) = V(x_1 \ldots x_d) + C = V(\mathbf{X}) + C$$

where C is a constant. The assumed PDF associated with $V(x_1 \ldots x_m)$ is

$$p(x_1 \ldots x_m | x_{m+1} \ldots x_d) = Z^{-1} \exp[-V(x_1 \ldots x_m)] = Z^{-1} \exp[-V(\mathbf{X}) - C] = p(\mathbf{X})/p(x_{m+1} \ldots x_d)$$

where $p(x_{m+1} \ldots x_d)$ is a non-zero normalization constant obtained by integrating over S_p.

Q.E.D.

5.3 Assumed PDFs for Back-Propagation Associative Networks

Proposition. Let the dimensionality of $\mathbf{O}$ be equal to d. Given the additive d.s.s.f. in (5), the corresponding assumed PDF, $p(\mathbf{O}|\mathbf{I})$, is uniquely given by (6) where the set S_p (refer to (17)) is taken as the entire d-dimensional real vector space.

Proof. Direct substitution of (5) into (16) and (17) yields (6). Note that the integral in (17) exists and is equal to $\pi^{d/2}$ because (6) is a multivariate Gaussian density function with mean $\phi_A[\mathbf{I}]$ and covariance matrix equal to the identity matrix multiplied by 1/2.

Q.E.D.

Proposition. Let the dimensionality of $\mathbf{O}$ be equal to d. Given the additive d.s.s.f. in (8), the corresponding assumed PDF, $p(\mathbf{O}|\mathbf{I})$, is uniquely given by (9), with S_p (refer to (17)) consisting of the entire set of d-dimensional vectors whose elements are either equal to zero or one.

Proof. Direct substitution of (8) into (16) and (17) yields (9). Note that Z equals unity for the d.s.s.f. in (8) since the ith element of $\mathbf{O}$ can only take on the values of zero or one, and $0 \leq p_i(\mathbf{A},\mathbf{I}) \leq 1$.

Q.E.D.

5.4 Assumed PDFs for Serial Multi-Stage Neural Networks

Definition of a serial multi-stage network. Let S be a d-dimensional state vector space that is partitioned into C subspaces $S_1 \ldots S_C$ such that if $\mathbf{X} \in S$, then $\mathbf{X}$ can be partitioned into C subvectors such that $\mathbf{X} = (\mathbf{X}_1 \ldots \mathbf{X}_C)$ where the dimensionality of $\mathbf{X}_i \in S_i$ is d_i. Thus, $d = \sum d_i$. A serial multi-stage network defined with respect to S is a set of C deterministic dynamical

systems where the state of the ith system is a d_i-dimensional vector, $\mathbf{X}_i \in S_i$. The state, $\mathbf{X}_i(t)$, of the ith dynamical system at time t is updated according to:

$$\mathbf{X}_i(t + \Delta t) = \mathbf{f}_i(\mathbf{X}_1(t) \ldots \mathbf{X}_i(t)) \tag{23}$$

where $\mathbf{f}_i$ is some vector-valued function.

Definition of a conditionally stable subnetwork. Let D_s be a serial multi-stage network with respect to the state space S which is partitioned into the subspaces $S_1 \ldots S_C$, and subvectors $\mathbf{X}_1 \ldots \mathbf{X}_C$. The ith subnetwork (i.e., dynamical system) is conditionally stable if and only if there exists subvectors $\mathbf{X}_j^* \in S_j$, $j = 1 \ldots i$, a function $V_i(\mathbf{X}_1, \ldots \mathbf{X}_i)$, and an increasing sequence $t^1, t^2, \ldots t^i, \ldots$ such that (a) $V_i(\mathbf{X}_1^*, \ldots \mathbf{X}_i^*) \leq V_i(\mathbf{X}_1, \ldots \mathbf{X}_i)$ for all $\mathbf{X}_j \in S_j$, $j=1 \ldots i$, and (b) if for all $t \geq t^{i-1}$, $\mathbf{X}_j(t) = \mathbf{X}_j^*$ for $j=1 \ldots i-1$, then $\mathbf{X}_i(t) = \mathbf{X}_i^*$ for all $t \geq t^i$.

Multi-Stage Network Theorem. Let D_s be a serial multi-stage network with respect to the state space S which is partitioned into the subspaces $S_1 \ldots S_C$, and subvectors $\mathbf{X}_1 \ldots \mathbf{X}_C$. If all C subnetworks of D_s are conditionally stable with respect to the functions $V_i(\mathbf{X}_1, \ldots \mathbf{X}_i)$ $(i=1 \ldots C)$, then $V(\mathbf{X}) = \sum_{i=1}^{C} V_i(\mathbf{X}_1, \ldots \mathbf{X}_i)$ is an additive d.s.s.f. for D_s.

Proof. Let $\mathbf{X}_j^* \in S_j$, $j=1 \ldots i$ have the property that $V_i(\mathbf{X}_1^*, \ldots \mathbf{X}_i^*) \leq V_i(\mathbf{X}_1, \ldots \mathbf{X}_i)$ for all $\mathbf{X}_j \in S_j$, $j=1 \ldots i$. For subnetwork 1, a t^1 exists such that for all $t > t^1$, $\mathbf{X}_1(t) = \mathbf{X}_1^*$ since $V_1(\mathbf{X}_1)$ is only a function of $\mathbf{X}_1$ by the definition of a serial multi-stage network, and the premise of condition (b) in the definition of conditionally stable is trivially satisfied. For subnetwork i, a $t^i > t^{i-1}$ exists such that if for all $t \geq t^{i-1}$, $\mathbf{X}_j(t) = \mathbf{X}_j^*$ for $j=1 \ldots i-1$, then $\mathbf{X}_i(t) = \mathbf{X}_i^*$ for all $t \geq t^i$ (since subnetwork i is conditionally stable). By induction then, a t^C exists such that for all $t > t^C$, $\mathbf{X}_j(t) = \mathbf{X}_j^*$ for $j=1 \ldots C$ where $\mathbf{X}_1^* \ldots \mathbf{X}_C^*$ is a global minimum of

$V(\mathbf{X}) = \sum_{i=1}^{C} V_i(\mathbf{X}_1, \ldots \mathbf{X}_i)$. To show that $V(\mathbf{X})$ is additive note that if all C subnetworks are independent, then $V(\mathbf{X}) = \sum_{i=1}^{C} V_i(\mathbf{X}_i)$.

Q.E.D.

Corollary. Given the additive d.s.s.f., $V(\mathbf{X})$, constructed using the multi-stage network theorem, the assumed PDF for the multi-stage network is uniquely given by (16) and (17), provided the integral in (17) is finite.

5.5 Compatible Assumed and Environmental PDFs

Definition of Compatible PDFs. Let an environmental PDF, $p_e(\mathbf{X})$, and an assumed PDF, $p_a(\mathbf{X};\mathbf{A})$ be defined over some set of state vectors known as S_p where $\mathbf{A}$ specifies the parameters of $p_a(\mathbf{X};\mathbf{A})$. The PDFs $p_e(\mathbf{X})$ and $p_a(\mathbf{X};\mathbf{A})$ are compatible with respect to S_p if and only if an $\mathbf{A}$ exists such that $p_a(\mathbf{X};\mathbf{A}) = p_e(\mathbf{X})$ for all $\mathbf{X}$ in S_p.

The Compatibility Test for Networks of Two-State Units. Let each member of the set γ of environmental PFs assign non-zero probabilities to each of the M d-dimensional vectors of S_p where each vector $\mathbf{X} \in S_p$ consists of binary-valued elements. Let $P_a(\mathbf{X};\mathbf{A})$ be an assumed PF of a specific neural network model with the parameter vector $\mathbf{A}$. If the rank of the M - 1 by d_a dimensional compatibility matrix (which is defined in (13)) equals M - 1, then any environmental PF, $P_E(\mathbf{X})$, in γ is compatible with $P_a(\mathbf{X};\mathbf{A})$ with respect to S_p.

Derivation of the Test. If $Q_e(\mathbf{X})$ is an arbitrary function, then any environmental PF, $P_E(\mathbf{X})$, in γ can be equivalently expressed by a PF, $P_e(\mathbf{X})$, as:

$$P_e(\mathbf{X}) = P_E(\mathbf{X}_M) \exp[Q_e(\mathbf{X})] \tag{24}$$

where $\mathbf{X}_M \in S_p$, and $Q_e(\mathbf{X}_M) = 0$.

Also any assumed PF, $P_a(\mathbf{X};\mathbf{A})$, may be equivalently expressed as follows (Besag, 1974; Anderson et al., in press) when the elements of $\mathbf{X}$ are binary-valued.

$$P_a(\mathbf{X};\mathbf{A}) = \exp[Q_a(\mathbf{X};\mathbf{A})] / Z_a \tag{25}$$

where $Q_a(\mathbf{X};\mathbf{A}) = \mathbf{F}(\mathbf{X} - \mathbf{X}_M)\mathbf{A}$, the row vector function, $\mathbf{F}(\mathbf{C})$, is defined in (12), and $\mathbf{A}$ is a column vector of dimension d_a.

Now note if $Q_a(\mathbf{X};\mathbf{A}) = Q_e(\mathbf{X})$ for all $\mathbf{X} \in S_p$, $\mathbf{X} \neq \mathbf{X}_M$, then $P_a(\mathbf{X};\mathbf{A}) = P_e(\mathbf{X})$ for all $\mathbf{X} \in S_p$ since Z_a^{-1} must equal $P_E(\mathbf{X}_M)$ for $\int P_a(\mathbf{X};\mathbf{A})\, d\mathbf{X} = 1$. Therefore, the PF, $P_a(\mathbf{X};\mathbf{A})$, is compatible with $P_e(\mathbf{X})$ if an $\mathbf{A}$ exists such that the system of $n = M - 1$ linear equations:

$$Q_e(\mathbf{X}_i) = Q_a(\mathbf{X}_i;\mathbf{A}) \text{ for } 1 \leq i \leq n \tag{26}$$

is consistent for any $Q_e(\mathbf{X})$ where $\mathbf{X}_i \in S_p$, $\mathbf{X}_i \neq \mathbf{X}_M$. For convenience, (26) can be rewritten as:

$$\mathbf{q} = \mathbf{W}\mathbf{A} \tag{27}$$

where the ith element of $\mathbf{q}$ is $Q_e(\mathbf{X}_i)$, and the n by d_a dimensional compatibility matrix, $\mathbf{W}$, is defined in (13). Let $R(\mathbf{W}) = n$ (thus $n \leq d_a$), and form a new d_a-dimensional square matrix, $\mathbf{Y}$, whose first n rows are $\mathbf{W}$ and whose remaining rows are selected such that $\mathbf{Y}$ has rank d_a. Let $\mathbf{r}$ be a d_a-dimensional vector whose first n elements are $\mathbf{q}$, and whose remaining d_a - n elements are arbitrary. Now since $\mathbf{Y}$ is invertible it is always possible to find at least one parameter vector, $\mathbf{A}$, for a given $\mathbf{r}$ vector using the formula $\mathbf{A} = \mathbf{Y}^{-1}\mathbf{r}$.

Q.E.D.

5.6 ML Estimation Applications to Learning Algorithms

To simplify notation, the function $p(X;A)$ should be considered a PF when x is a discrete random variable and a PDF when x is a continuous random variable in this section of the paper unless otherwise stated.

<u>Definition of a likelihood function.</u> Let a set, T_n, consist of the n state vectors $\{X^1 \ldots X^n\}$. The <u>likelihood function</u>, $L_n(A)$, associated with T_n is defined as:

$$L_n(A) = \prod_{i=1}^{n} p(X^i;A) \tag{28}$$

where $p(X;A)$ is an assumed PDF or PF.

<u>Definition of an ML estimate.</u> If $L_n(A^*) \geq L_n(A)$ for all permissable values of A, then A^* is an <u>ML estimate</u> associated with $L_n(A)$ in (28).

<u>Definition of an asymptotic likelihood function.</u> Let $P_e(X)$ be an environmental PF, and let $p(X;A)$ be an assumed PDF or PF. The <u>asymptotic likelihood function</u>, $E(A)$, is:

$$E(A) = \sum_{i=1}^{M} P_e(X_i)\, \mathrm{LOG}\, [p(X_i;A)]. \tag{29}$$

<u>Definition of a cross-entropy function.</u> The <u>cross-entropy function</u>, $XE(A)$, is:

$$XE(A) = \sum_{i=1}^{M} P_e(X_i)\, \mathrm{LOG}\, [P_e(X_i)/P(X_i;A)] = k - E(A) \tag{30}$$

where $P_e(X)$ is the environmental PF, $P(X;A)$ is the assumed PF, $E(A)$ is the asymptotic likelihood function, and k is not dependent upon A.

<u>Lemma 1.</u> Given $|f(X_i;A)| < K < \infty$ for any X_i. If for any i, and $M < \infty$,

$a_i(n) \to L_i$ as $n \to \infty$, then $\sum_{i=1}^{M} a_i(n)\, f(X_i;A) \to \sum_{i=1}^{M} L_i f(X_i;A)$ uniformly as $n \to \infty$.

<u>Proof.</u> An $n \geq N$ exists such that $|a_i(n) - L_i| < \epsilon/K$.

But $|(a_i(n) - L_i) f(X_i;A)| \leq |a_i(n) - L_i||f(X_i;A)| < |a_n - L_i|K < (\epsilon/K)K = \epsilon$ for $n \geq N$.

Now note that since $a_i(n)\, f(X_i;A) \to L_i f(X_i;A)$ uniformly as $n \to \infty$,

$\sum_{i=1}^{M} a_i(n)\, f(X_i;A) \to \sum_{i=1}^{M} L_i f(X_i;A)$ uniformly as $n \to \infty$.

Q.E.D.

<u>Proposition.</u> Let $p(X;A)$ be either a discrete PF or continuous PDF of a neural network model with parameter vector A. Let $L_n(A)$ be defined in (28) with respect to T_n which is a set of n i.i.d. random vectors associated with PF $P_e(X)$. Define the stochastic sequence of independent random functions, $e_n(A)$, indexed by n such that $e_n(A) = (1/n)\mathrm{LOG}[L_n(A)]$. Let $E(A)$ be defined as in (29). (i) If $|\mathrm{LOG}\, [p(X;A)]| < C < \infty$, as $n \to \infty$, $e_n(A)$ uniformly converges almost surely to $E(A)$. (ii) If $|\nabla\, \mathrm{LOG}\, [p(X;A)]| < C < \infty$, as $n \to \infty$, $\nabla\, e_N(A)$ uniformly converges almost surely to $\nabla\, E(A)$ where all gradients are taken with respect to A.

<u>Proof.</u> First note $e_n(A) = (1/n)\, \mathrm{LOG}\, [L_n(A)] = (1/n)\, \mathrm{LOG}\, [\prod_{j=1}^{n} p(x^j;A)]$

where the random variable $x^j = X_i$ with probability $p(X_i;A)$. Therefore,

$$e_n(A) = (1/n)\, \mathrm{LOG}\, [\prod_{i=1}^{M} p(X_i;A)^{n_i(n)}] = \sum_{i=1}^{M} [n_i(n)/n]\, \mathrm{LOG}\, [p(X_i;A)]$$

where $n_i(n)$, $n = 1, 2, \ldots$ is a stochastic sequence of independent Binomial random variables with mean $nP_e(X_i)$. Because $n_i(n)/n \to P_e(X_i)$ almost surely as $n \to \infty$ by the strong law of large numbers for any X_i, $\sum_{i=1}^{M} (n_i/N)\, \mathrm{LOG}\, [p(X_i;A)]$ uniformly converges to $\sum_{i=1}^{M} P_e(X_i)\, \mathrm{LOG}[p(X_i;A)]$ almost surely by *Lemma 1* since $|\mathrm{LOG}\, [p(X_i;A)]| < C$. The proof of (ii) is based upon a similar argument.

Q.E.D.

<u>Proposition.</u> Let the PDF, $p(O|I)$ defined in (8) be the assumed a posteriori PDF for a given neural network, and the network may have any prior knowledge of the likelihood of I represented by the assumed prior PDF, $p(I)$, which is not a function of the parameter vector A. Then $E(A) = k - \sum_i P_e(O_i,I_i)\, |O_i - \phi_A(I_i)|^2$ where k is not dependent upon A.

<u>Proof.</u> Substituting $p(O_i,I_i) = p(O_i|I_i)p(I_i)$ for $p(X_i;A)$ in (29) yields:

$$E(A) = \sum_i P_e(X_i)\, \mathrm{LOG}[p(I_i)\exp(-|O_i - \phi_A(I_i)|^2) / \pi^{d/2}]$$

$$E(A) = (-d/2)\, \mathrm{LOG}\, [\pi] + \sum_i P_e(O_i,I_i)\, \mathrm{LOG}[p(I)] - \sum_i P_e(O_i,I_i)\, |O_i - \phi_A(I_i)|^2$$

where $P_e(O_i,I_i)$ and $p(I)$ are not functions of A.

Q.E.D.

<u>Proposition.</u> Let the PDF, $p(O|I)$ defined in (9) be the assumed a posteriori PDF for a given neural network, and the network may have any prior knowledge of the likelihood of I represented by the assumed prior PDF, $p(I)$, which is not a function of the parameter vector A. Then

$$E(A) = k + \sum_j \sum_i P_e(O_j,I_j)\, [o_{j,i}\mathrm{LOG}[p_i(A,I_j)] + (1 - o_{j,i})\, \mathrm{LOG}[1 - p_i(A,I_j)]]$$

where k is not dependent upon A, and $o_{j,i}$ is the ith element of the jth output vector, O_j

<u>Proof.</u> Substituting $p(O_j,I_j) = p(O_j|I_j)p(I_j)$ for $p(X_j;A)$ in (29) yields:

$$E(A) = k + \sum_j P_e(O_j,I_j) \sum_i \mathrm{LOG}\, [o_{j,i}\, p_i(A,I_j) + (1 - o_{j,i})(1 - p_i(A,I_j))].$$

where k is a constant. Also note that since $o_{j,i} = 0$ or $o_{j,i} = 1$,

$$\mathrm{LOG}\, [o_{j,i}\, p_i(A,I_j) + (1 - o_{j,i})(1 - p_i(A,I_j))] = o_{j,i}\, \mathrm{LOG}\, [p_i(A,I_j)] + (1 - o_{j,i})\, \mathrm{LOG}\, [1 - p_i(A,I_j)].$$

Q.E.D.

References

Ackley, D. A., Hinton, G. E., & Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. Cognitive Science, 9, 147-169.

Anderson, J. A., Golden, R. M., & Murphy, G. L. (in press). Concepts in distributed systems. In H. Szu (Ed.), S.P.I.E. Advanced institute series hybrid and optical computers. Bellingham, Washington: S. P. I. E.

Anderson, J. A., Silverstein, J. W., Ritz, S. A., & Jones, R. S. (1977). Distinctive features, categorical perception, and probability learning: Some applications of a neural model. Psychological Review, 84, 413-451.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. Journal of the Royal Statistical Society, Series B, 36, 192-236.

Cohen, F. S., & Cooper, D. B. (1987). Simple parallel hierarchical and relaxation algorithms for segmenting noncausal Markovian random fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, 9, 195-219.

Cohen, M. A., & Grossberg, S. (1983). Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. IEEE Transactions on Systems, Man, and Cybernetics, SMC-13, 815-825.

Cox, R. T. (1946). Probability, frequency, and reasonable expectation. American Journal of Statistical Physics, 14,1 - 13.

Duda, R. O., & Hart, P. E. (1973). Pattern classification and scene analysis. New York: Wiley.

Frieden, B. R. (1983). Unified theory for estimating frequency-of-occurrence laws and optical objects. Journal of the Optical Society of America, 73, 927-938.

Frieden, B. R. (1985). Estimating occurrence laws with maximum probability, and the transition to entropic estimators. In C. R. Smith and W. T. Grandy, J. (Eds.), Maximum-entropy and Bayesian methods in inverse problems, 133-169. Boston: Reidel.

Gallager, R. G. (1968). Information theory and reliable communication. New York: Wiley.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian

restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 6, 721-741.

Golden, R. M. (1986a). The "brain-state-in-a-box" neural model is a gradient descent algorithm. Journal of Mathematical Psychology, 30, 73-80.

Golden, R. M. (1986b). Representing causal schemata in connectionist systems. In Proceedings of the Eighth Annual Conference of the Cognitive Science Society. Hillsdale, NJ:Erlbaum, 13-22.

Golden, R. M. (1987). Modelling causal schemata in human memory: A connectionist approach. Ph.D. Thesis. Brown University, Providence, RI.

Grossberg, S. (1982). Studies of mind and brain: Neural principles of learning, perception, development, cognition, and motor control. Boston: Reidel Press.

Grossberg, S. (1987). Competitive learning: From interactive activation to adaptive resonance. Cognitive Science, 11, 23-63.

Henrion, M. (1986). Should we use probability in uncertain inference systems? In Proceedings of the Eighth Annual Conference of the Cognitive Science Society, NJ:Erlbaum.

Hinton, G. E. (1981). Implementing semantic networks in parallel hardware. In G. E. Hinton, & J. A. Anderson (Eds.), Parallel models of associative memory, NJ: Erlbaum.

Hinton, G. E. (1987). Connectionist learning procedures (CMU-CS-87-115). Department of Computer Science Technical report. Carnegie-Mellon University.

Hinton, G. E., & Anderson, J. A. (1981). Parallel models of associative memory. Hillsdale, NJ Erlbaum.

Hinton, G. E., & Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. In D. E. Rumelhart, J. L. McClelland and the PDP Research Group (Eds.), Parallel distributed processing: Explorations in the microstructure of cognition. Volume 1: Foundations. Cambridge, MA: MIT Press.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences, USA, 79, 2554-2558.

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. Proceedings of the National Academy of Sciences, USA, 81, 3088-3092.

Kohonen, T. (1984). Self-organization and associative memory. New York: Springer-Verlag.

Kullback, S. (1959). Information theory and statistics. New York: Wiley.

Le Cun, Y. (1985). Une procedure d'apprentissage pour reseau a seuil assymetrique [A learning procedure for assymetric threshold network]. Proceedings of Cognitiva 85, 599-604. Paris.

Luenberger, D. G. (1979). Introduction to dynamic systems: Theory, models, and applications. New York: Wiley.

Luenberger, D. G. (1984). Linear and nonlinear programming. Reading, MA:Addison-Wesley.

Macchi, O., and Eweda, E. (1983). Second-order convergence analysis of stochastic adaptive linear filtering. IEEE Transactions on Automatic Control, 28, 76-85.

Marr, D. (1982). Vision. San Francisco: Freeman.

Marroquin, J. L. (1985). Probabilistic solution of inverse problems. A.I. Memo 860, MIT Press.

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

McClelland, J. L., Rumelhart, D. E., and the PDP Research Group (1986). Parallel distributed processing: Explorations in the microstructure of cognition. Volume 2: Psychological and biological models. Cambridge, MA: MIT Press.

Minsky, M., & Papert, S. (1969). Perceptrons. Cambridge, MA: MIT Press.

Noble, B. (1977). Applied linear algebra. NJ: Prentice-Hall.

Parker, D. B. (1985). Learning-logic (TR-47). Cambridge, MA: Massachusetts Institute of Technology, Center for Computational Research in Economics and Management Science.

Plaut, D. C., Nowlan, S. J., & Hinton, G. E. (1986). Experiments on learning by back propagation (CMU-CS-86-126). Department of Computer Science Technical Report. Carnegie-Mellon University.

Rumelhart and McClelland (1985). Levels indeed! A response to Broadbent. Journal of Experimental Psychology: General, 114, 193-197.

Rumelhart, D. E., McClelland, J. L., and the PDP Research Group (1986). Parallel distributed processing: Explorations in the microstructure of cognition. Volume 1: Foundations. Cambridge, MA: MIT Press.

Rumelhart, D. E., Smolensky, P., McClelland, J. L., & Hinton, G. E. (1986). Models of schemata and sequential thought processes. In J. L. McClelland, D. E. Rumelhart, and the PDP Research Group (Eds.), Parallel distributed processing: Explorations in the microstructure of cognition. Volume II: Applications. Cambridge, MA: Bradford Books.

Savage, L. J. (1971). The foundations of statistics. Canada: Wiley.

Schneider, W., & Detweiler, M. (1987). A connectionist/control architecture for working memory. Unpublished manuscript.

Schneider, W., & Mumme, D. (1987). Attention, automaticity and the capturing of knowledge: A two-level cognitive architecture. Unpublished manuscript.

Shannon, C. E. (1963). The mathematical theory of communication. In C. E. Shannon & W. Weaver (Eds.), The mathematical theory of communication (pp. 29-125). Urbana University of Illinois Press. (Reprinted from Bell System Technical Journal, 1948, July and October).

Simmons, G. F. (1972). Differential equations with applications and historical notes. New York McGraw-Hill.

Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhart, J. L. McClelland and the PDP Research Group (Eds.), Parallel distributed processing: Explorations in the microstructure of cognition. Volume 1: Foundations. Cambridge, MA: MIT Press.

Sutton, R. S., & Barto, A. G. (1981). Toward a modern theory of adaptive networks: Expectation and prediction. Psychological Review, 88, 135-171.

Van Trees, H. L. (1968). Detection, estimation, and modulation theory. New York: Wiley.

Wise, B. P. (1986). An experimental comparison of uncertain inference systems. Ph.D. thesis. Carnegie-Mellon University, Pittsburgh, PA.

Author Notes

This research was supported in part by the Mellon foundation while the author was an Andrew Mellon Fellow in the Psychology Department at the University of Pittsburgh, and partly by the Office of Naval Research under Contract No. N-0014-86-K-0107 to Walter Schneider.

I am very grateful to Dean C. Mumme for comments, criticisms, and helpful discussions of an earlier version of this manuscript. I would also like to thank David B. Cooper for his suggestion that many neural network models might be viewed within a unified probabilistic framework.

Requests for reprints should be addressed to the author who is currently at the Department of Psychology, Jordan Hall, Bldg 420, Stanford University, Stanford, California 94305. USA.