Revised Paper (April 4, 1994)

Title: Observer-Participant Models of Neural Processing

Author: Robert L. Fry

Organization: Johns Hopkins University/Applied Physics Laboratory

Phone: (410) 792-5945

Email: robert_fry@jhuapl.edu

Author Address:

Johns Hopkins University/Applied Physics Laboratory
Johns Hopkins Road
Laurel, MD 20723

Observer-Participant Models of Neural Processing

Abstract

A model is proposed in which the neuron serves as an information channel. Channel distortion occurs through the channel since the mapping from input boolean codes to output codes are many-to-one in that neuron outputs consist of just two distinguished states. Within the described model, the neuron performs a decision-making function. Decisions are made regarding the validity of a question passively posed by the neuron. This question becomes defined through learning hence learning is viewed as the process of determining an appropriate question based on supplied input ensembles. An application of the Shannon information measures of entropy and mutual information taken together in the context of the proposed model lead to the Hopfield neuron model with conditionalized Hebbian learning rules. Neural decisions are shown to be based on a sigmoidal transfer characteristic or in the limit as computational temperature tends to zero, a maximum likelihood decision rule. The described work is contrasted with the information-theoretic approach of Linsker.

Introduction

Regardless of the model, neurons must be acknowledged as being information receiving and transmitting devices. Synaptic input is selectively transformed into an output. In the model assumed here, each neuron represents a computational element which strives to answer a boolean question which essentially defines its intrinsic role as an observer. Operationally, the neuron generates boolean output decisions (answers) to other elements to which it is connected in response to observable input information. In this sense the neuron participates as well as observes. The learned question which in fact defines the neuron must be formed during learning hence learning here corresponds to the search for the proper question to ask. In the frequently used analogy of the game of *twenty questions*, this corresponds to the search for high entropy questions, i.e. those questions which convey maximal information. Both in the game and in the neuron model described, the maximum question entropy is one bit.

Figure 1 shows the basic neuron model in which an observer q has access to N input sources of boolean information $\{x_1, x_2, x_3, \ldots, x_N\}$. In turn the neuron generates a boolean decision y (also observable by the neuron) regarding the validity of q. The standard boolean values of 0 and 1 are used here to denote the binary distinguished states although nothing in the development precludes the use of 0 and any other value different from 0. By way of analogy to biological neurons, realized neural states consist of the action potential and also the lack thereof. The action potential can be thought of as an indication (answer) by the neuron regarding the truth of the defining question at a given instant in time.

The information processing attributes of a single model neuron are examined in this paper although the expansion of the analysis to multiple neuron models is analytically tractable as well.

Current work is addressing the application of the discussed information-theoretic techniques to multiple neuron models. In such cases the information channel performs a re-coding of the input binary code vector to a new code vector as opposed to a single bit which is the case for an information channel consisting of an individual neuron.

Initially, a maximum entropy (ME) formulation is shown to provide the basic functional form of the model neuron including synaptic weights and a sigmoidal transfer characteristic. This formulation requires the assumption of a set of sampling functions which are in turn a function of both synaptic inputs and neuron output. Furthermore, an ME formulation requires the specification of the statistical moments of the selected sampling functions which must somehow be supplied by a non-specific supervisor. An ME formulation is under-constrained in the sense that the model neuron cannot find a uniquely preferable set of moment constraints on any set of sampling functions which use locally available information. Alternatively, a maximum mutual information (MMI) formulation is shown to be fully constrained in this regard and can make exclusive use of locally available information. Solutions are shown to take the form of the Hopfield neuron model with a requirement for a feedback learning methodology which takes the form of Hebbian learning whereby the synaptic weights are only modified in response to the boolean conjunction of input and output neural events. A conditionalized version of an adaptation equation originally formulated by Oja [11] provides an algorithmic means of implementing the required computations.

This paper focuses on the primary results of the delineated approach and more importantly the approach itself. Lengthy derivations and theorem proofs are contained in the two appendices.

Maximum Entropy Formulation

Designate the model neuron by q. Let $y \in Y$ denote the boolean response of q to an input random boolean state vector $\mathbf{x} \in X$ where $\mathbf{x} = (x_1\ x_2 \bullet\bullet\bullet\ x_M)$ and represents possible input binary code words observable by q. The neuron q is assumed to be capable of forming N different sampling functions $F_n(\mathbf{x},y)$, $n = 1, 2, \bullet\bullet\bullet, N$ which are directly computable from input states and the output state (a concept originating in [7]). For each neural response y and input code vector $\mathbf{x}$, a set of functions $F_n(\mathbf{x},y)$ is computed. The set of sampling functions F_n represent the most elemental aspects of the environment measurable by q. Finally, let the joint probability $P(\mathbf{x},y)$ describe the neuron internal statistical model of the environment which is developed through learning. The environment consists of that which is observable by q and therefore includes both $\mathbf{x}$ and y. As such, the statistical model is actually a state of knowledge possessed by q in so much that probability is the only logically consistent measure of degree of plausible belief [1]. P provides q with the capacity of performing rational decision-making regarding the validity of q given the observable information $\mathbf{x}$. It should be noted that $P(\mathbf{x},y)$ reflects knowledge of the environment state possessed by the subject neuron . Another system or individual will have a differenct probability assignment depending on the respective state of knowledge. Probability is subjective in this sense although maximum entropy formulations provide an objective means of determining P based on available information ([2],[3]).

The neuron q must somehow establish the model P. One means of determining P is through the maximum entropy method [2],[3]. The maximum entropy method [4] assumes the existence of prior information about $P(\mathbf{x}, y)$ in the form of expected values of known functions defined here by F_n thus implying that all

$$<F_n> = \sum_{\mathbf{x}'\in X, y'\in Y} F_n(\mathbf{x}', y')\, P(\mathbf{x}', y') \quad n = 1,2,\ldots,N. \tag{1}$$

are known. The symbol <•> denotes the expected value with respect to P($\mathbf{x}$, y) and serves to simplify mathematical expressions used here. The symbol <•> is also used to denote the fact that the moments realistically must be estimated through an averaging operation. For the generic ME formulation, all $<F_n>$ are assumed to be specified to q by a non-specific supervisory source.

In the ME formalism, the objective is to maximize an entropy functional subject to supplied moment constraints and the required normalization of P such that its sum over all possible states is one. Maximizing entropy subject to posed or known constraints provides the least presumptive distribution that is consistent with known prior information [3]. Furthermore, it can be shown [3] that the resulting probability can be interpreted as the most probable distribution in the infinite function space of all distributions having the prerequisite set of specified moments.

The maximization of entropy subject to posed constraints can be accomplished through the introduction of a set of Lagrange multipliers $\boldsymbol{\lambda} = (\lambda_0\ \lambda_1\ \lambda_2 \cdots \lambda_N)^T$ and the definition of an auxiliary normalization function $F_0(\mathbf{x}) = 1$. The resulting objective function then corresponds to

$$S = -\sum_{\mathbf{x}'\in X, y'\in Y} P(\mathbf{x}', y') \ln P(\mathbf{x}', y') + \sum_{n=0}^{N} \lambda_N \Big[\sum_{\mathbf{x}'\in X, y'\in Y} F_N(\mathbf{x}', y') P(\mathbf{x}', y') - <F_N(\mathbf{x}', y')> \Big] \tag{2}$$

An application of the calculus of variations [4],[5] or the Euler-Lagrange equation can be used to maximize (2) yielding the solution

$$\mathbf{P}(\mathbf{x},y) = \exp\left[- \lambda_0 - \sum_{n=1}^{N} \lambda_n \mathrm{F_n}(\mathbf{x},y) \right] \tag{3}$$

where

$$\begin{aligned} \lambda_0 &= \ln \left\{ \sum_{\mathbf{x}\in X, y \in Y} \exp\left[-\sum_{n=1}^{N} \lambda_n \mathrm{F_n}(\mathbf{x},y\right]\right\} \\ &= \ln \mathrm{Z}. \end{aligned} \tag{4}$$

Equation (3) is a Gibbs distribution since it has the form of an exponentiated energy function. Equation (4) is the corresponding normalization term which in statistical mechanics is called the partition function. The ME computational problem therefore consists of finding the N+1 Lagrange multipliers which maximize (2) and therefore parameterize the distribution P. Finding λ requires the solution of N+1 nonlinear equations. Closed form solutions to this problem are typically intractable although several efficient numerical techniques [5] are known.

Instead of trying to solve for λ, however, consider making a definitive selection of the sampling functions (in a manner analogous to [6] and [7]) and reviewing the resulting forms of the distribution P, two selected marginal distributions, and two selected conditional distributions all of which are directly obtainable from P. Suppose that it is desirable to select the sampling functions such that they enable the observer q to evaluate and modify its future decisions based on present decisions and inputs. This requires that the selected sampling functions are derived from the input vector $\mathbf{x}$ and the current decision y, a fact reflected by the notation $\mathrm{F_n}(\mathbf{x}, y)$. The following set of N+1 sampling functions are proposed:

$$\mathrm{F}_n = x_n y \quad n = 1,2,\ldots,N \tag{5}$$

and one additional sampling function

$$\mathrm{F}_{\text{✏}} = y. \tag{6}$$

Therefore, the selected sampling functions consist of the simple boolean conjunction of individual inputs x_n with the decision y and also the decision y itself. The corresponding moment constraints consist of $\langle x_n y \rangle$ for $n = 1, 2, \ldots, N$ and also $\langle y \rangle$. The substitution of (5) and (6) into (3) yields the ME distribution

$$\mathrm{P}(\mathbf{x},y) = \frac{\exp\left[-\sum_{n=1}^{N} \lambda_n x_n y - \mu y\right]}{\mathrm{Z}} \tag{7}$$

where Z is the partition function defined by

$$\mathrm{Z} = \sum_{\mathbf{x}'\in X} \sum_{y'\in Y} \exp\left[-\sum_{n=1}^{N} \lambda_n x_n' y' - \mu y'\right]. \tag{8}$$

Selected marginal and conditional distributions immediately follow:

$$\mathrm{P}(\mathbf{x}) = \frac{\sum_{y' \in Y} \exp[-\sum_{n=1}^{N} \lambda_n x_n y' - \mu y']}{Z} \tag{9}$$

$$\mathrm{P}(y) = \frac{\sum_{\mathbf{x}' \in X} \exp[-\sum_{n=1}^{N} \lambda_n x_n' y - \mu y]}{Z} \tag{10}$$

$$\mathrm{P}(\mathbf{x}|y) = \frac{\exp[-\sum_{n=1}^{N} \lambda_n x_n y - \mu y]}{\sum_{\mathbf{x}' \in X} \exp[-\sum_{n=1}^{N} \lambda_n x_n' y - \mu y]} \tag{11}$$

Of special interest is the case of $y = 0$ in equation (11). In this case $\mathrm{P}(\mathbf{x}|y = 0)=2^{-N}$ and y conveys no information about **x** since each binary vector **x** is equally probable given no output is observed. In a neural setting, q would not generate an output action potential. Thus all neurons which are synaptically coupled to y would receive no action potential input. It should be noted, however, that within the decision-making process to be described that the absence of an input can convey as much information as the non-zero presence of an input.

Finally the last conditional distribution to be considered is given by

$$P(y|\mathbf{x}) = \frac{\exp[-\sum_{n=1}^{N} \lambda_n x_n y - \mu y]}{1 + \exp[-\sum_{n=1}^{N} \lambda_n x_n - \mu]} \qquad \textbf{(12)}$$

and hence

$$P(y{=}1|\mathbf{x}) = \frac{1}{1 + \exp[\sum_{n=1}^{N} \lambda_n x_n + \mu]} \qquad \textbf{(13)}$$

which contains the now familiar sigmoidal transfer characteristic used in neural modeling.

Equations (12) and (13) also provide the observer q with a simple maximum likelihood (ML) decision rule which can be found by evaluating the log-likelihood function. The ML decision rule is: Generate an output (choose $y = 1$) if $P(y = 1 \mid \mathbf{x}) > P(y = 0 \mid \mathbf{x})$, otherwise generate no output (choose $y = 0$). Based on the observed information $\mathbf{x}$ and equation (13), the corresponding log-likelihood decision rule of q is to make the decision $y = 1$ if

$$-\mu - \sum_{n=1}^{N} \lambda_n x_n \geq 0 \qquad \textbf{(14)}$$

which is just a deterministic Hopfield [8], [9] decision rule, since the sign appearing in (14) results from the selected sign convention in (1). Since the observed input values are either 0 or 1, the computation in (14) is simplified even more, being the comparison of the sum of all λ_n for which the corresponding x_n is one with a threshold μ. In statistical decision theory, (14) is called the evidence $\zeta(\mathbf{x})$ in favor of y.

The ML decision rule is deterministic. Alternatively, the decision *y* can be made stochastically based on the Boltzmann distribution described by equation (13). The use of (13) entails the computation of the evidence in (14) which is then passed through the sigmoidal function defined in (13). Depending on the application, either decision rule has merits. Biological interpretations and computer simulation favor the stochastic decision rule (13) with the incorporation of a time-varying temperature parameter. Interestingly enough, equation (13) is also the minimum mean-squared error estimator (nonlinear) for *y*.

In summary, an ME formulation yields a Boltzmann distribution (13) for the single model neuron. The maximum likelihood decision rule corresponds to that of a deterministic Hopfield neuron model. A stochastic decision rule is also possible which first requires the computation of an evidence function which can then be passed through a sigmoidal non-linearity. These results require the specification of the moments of a set of *N* sampling functions by a non-specific supervisor. This is considered untenable regarding the development of useful computing structures which are constrained to use locally available information and hence be unsupervised. In the next section, the optimization of the mutual information measure is shown to fully constrain the moments of P and in turn provides a means of determining the parameter set $\boldsymbol{\lambda}$ making strict use of locally available information.

Maximized Mutual Information

Instead of maximizing entropy, consider the maximization of the mutual information between the input **x** and output *y*. The mutual information between two ensembles *X* and *Y* is defined to be

$$I(\mathbf{x};\mathbf{y}) = \sum_{\mathbf{x}'\in X, \mathbf{y}'\in Y} P(\mathbf{x}',\mathbf{y}') \ln \frac{P(\mathbf{x}',\mathbf{y}')}{P(\mathbf{x}')\,P(\mathbf{y}')} \tag{15}$$

If a Gibbs distribution having the same form as (3) is assumed, then all the distributions provided by equations (7) through (12) are defined. The rationale for suggesting that the Gibbs distribution be maintained is as follows. Regardless of the method of moment of specification, the ME formalism will yield (3) as the form of the resulting probability assignment. Therefore, even though the mutual information measure is the suggested means of determination of the subject moments, it is still justified to assume (3) as the ultimate form of the neuron's knowledge state.

Note that neither the parameter set $\boldsymbol{\lambda}$ nor the sampling moments $\langle F_n \rangle$ are specified here. In addition, if $\boldsymbol{\lambda}$ somehow becomes specified through the use of (15), then the moments $\langle F_n \rangle$ automatically become constrained to take on specific values since the distribution P is then fully specified and thereby providing the ME basis of acceptance of (3) as the distributional form. The objective here is to find the set $\boldsymbol{\lambda}\in\Lambda$ which maximizes the mutual information between $\mathbf{x}$ and y. This requires finding the extremum of (15) over some permissable $\Lambda\subset\mathbb{R}^N$. One additional constraint $\boldsymbol{\lambda}^T\boldsymbol{\lambda}=\gamma^2$ is imposed yielding an objective function $J(\boldsymbol{\lambda}) = I(\mathbf{x};\mathbf{y};\boldsymbol{\lambda}) + (\alpha/2)\boldsymbol{\lambda}^T\boldsymbol{\lambda}$. Without this constraint, an obvious extremum is $\boldsymbol{\lambda}=0$ in which case $I(\mathbf{x};\mathbf{y};\boldsymbol{\lambda})=0$ since it is easily shown that $I(\mathbf{x};\mathbf{y})\geq 0$. An investigation of local extrema requires attention to the Hessian which has been addressed. This analysis although important, is not included here due to length.

Appendix A derives what is called here the Gibbs Mutual Information Theorem which states that the extrema of $J(\boldsymbol{\lambda})$ can be found by solving the system of equations

$$\frac{\partial J}{\partial \lambda_n} = \sum_{\mathbf{x}'\in X,\mathbf{y}'\in Y} P(\mathbf{x}',\mathbf{y}')\,[<F_n> - F_n(\mathbf{x}',\mathbf{y}')]\,\ln\frac{P(\mathbf{x}'\,|\,\mathbf{y}')}{P(\mathbf{x}')} + \alpha\,\lambda_n = 0 \tag{16}$$

where $n = 1,2,\bullet\bullet\bullet,N$. Equation (16) is exact. Now make the substitutions $\mathbf{x}=(x_1\ x_2\ \cdots x_N\ )^T$ and $\mathbf{y}=y$, and choose the same set of sampling functions $\mathbf{F}(\mathbf{x},\mathbf{y})=(x_1y\ \ x_2y\ \cdots\ x_Ny\ \ y\ )^T$ used earlier. Using Bayes' theorem, the selected sampling functions x_ny, and substituting the distributions (9) and (11) into the argument of the log in equation (16) yields the result that

$$\frac{P(\mathbf{x}\,|\,y)}{P(\mathbf{x})} = \frac{\exp[-\sum_{n=1}^{N}\lambda_n x_n y - \mu y]}{\sum_{\mathbf{x}'\in \mathbf{X}}\exp[-\sum_{n=1}^{N}\lambda_n x_n' y]} \cdot \frac{Z\,e^{\mu y}}{\sum_{y'\in \mathbf{Y}}\exp[-\sum_{n=1}^{N}\lambda_n x_n y' - \mu y']} \tag{17}$$

which can be expressed as

$$\frac{P(\mathbf{x}\,|\,y)}{P(\mathbf{x})} = \frac{\exp[-\sum_{n=1}^{N}\lambda_n x_n y - \mu y]\cdot P(\mathbf{x}=\mathbf{0}\,|\,y)\cdot P(y=0\,|\,\mathbf{x})}{P(\mathbf{x}=\mathbf{0},y)} \tag{18}$$

Since $P(\mathbf{x}=\mathbf{0},y) = P(\mathbf{x}=0|y)P(y)$, it follows that

$$\frac{P(\mathbf{x}|y)}{P(\mathbf{x})} = \frac{\exp\left[-\sum_{n=1}^{N} \lambda_n x_n y - \mu y\right] \cdot P(y=0|\mathbf{x})}{P(y)} \tag{19}$$

Taking the log of equation (19) and substituting the sampling function $F_n(\mathbf{x},y)=x_n y$ into equation (16) yields

$$\begin{aligned} 0 = \; & E\{[\langle x_n y\rangle - x_n y][-(\sum_{i=1}^{N} \lambda_i x_i y) - \mu y]\} \\ & - E\{[\langle x_n y\rangle - x_n y] \ln P(y)\} \\ & + E\{[\langle x_n y\rangle - x_n y]\, g(\mathbf{x})\} \\ & + \alpha \lambda_n \end{aligned} \tag{20}$$

for $n=1$ to N where $g(\mathbf{x}) \triangleq \ln P(y=0|\,\mathbf{x})$. Now for the sampling function $F_0(\mathbf{x},y)= y$, equation (16) becomes

$$\begin{aligned} 0 = \; & E\{[\langle y\rangle - y][-(\sum_{*☞}^{\star} \lambda_{*} x_{*} y) - \mu y]\} \\ & - E\{[\langle y\rangle - y] \ln P(y)\} \\ & + E\{[\langle y\rangle - y]\, g(\mathbf{x})\} \end{aligned} \tag{21}$$

Appendix B shows that the second terms in equations (20) and (21) tend to zero as $P(y=0)$ aproaches $P(y=1)$ or equivalently $P(y)=1/2$. This condition in turn can be guaranteed by ensuring that γ^2 is sufficiently small. The third terms in (20) and (21) are shown in Appendix B to respectively correspond to $<x_n y>(1-<y>)[E\{g(\mathbf{x})|y=0\} - E\{g(\mathbf{x})|y=1\}]$ (which is approximate) and to $<y>(1-<y>)[E\{g(\mathbf{x})|y=0\} - E\{g(\mathbf{x})|y=1\}]$ (which is exact).

Noting that $y^2=y$, equation (21) evaluates to

$$\begin{aligned}&[\lambda^T <\mathbf{x}y> + \mu<y> \\ &+ <y>(E\{\mathbf{g}(\mathbf{x})|y=0\} - E\{\mathbf{g}(\mathbf{x})|y=1\})]\,(1-<y>) \;=\; 0\end{aligned} \tag{22}$$

where $<\mathbf{x}y> = E\{\ (x_1 y\ \ x_2 y\ ,\cdots\ x_N y\ )^T\ \}$ is a column vector. The Lagrange multiplier μ can be solved for explicitly giving

$$\mu \;=\; \frac{-\,\lambda^T <\mathbf{x}\,y>}{<y>} - [\,E\{\mathbf{g}(\mathbf{x})|y=0\} - E\{\mathbf{g}(\mathbf{x})|y=1\}\,] \tag{23}$$

If the term in equation (22) is neglected, then using Bayes' theorem and the facts that $Pr(y=1)=<y>$ and $<\mathbf{x}y>=Pr(\mathbf{x}=1,y=1)$ yield the result that $<\mathbf{x}y>$ equals $P(\mathbf{x=1}\mid y=1)P(y=1)$ where $P(\mathbf{x=1}\mid y=1)=<\mathbf{x}\mid y=1>$ is a column vector. When substituted into (23), the simplified neuron ML decision threshold (see equation (14)) is obtained:

$$\mu \;\approx\; -\,\lambda^T <\mathbf{X}\ |y=1\ > \tag{24}$$

Equation (20) can also be evaluated in the same manner as (21). Its evaluation for $n=1,2,\ldots,N$ yields

$$
\begin{aligned}
&- \boldsymbol{\lambda}^{\mathrm{T}} \langle x_n y \rangle \langle \mathbf{x}\, y \rangle - \mu \langle x_n y \rangle \langle y \rangle \\
&+ \boldsymbol{\lambda}^{\mathrm{T}} \langle \mathbf{x}\, x_n y \rangle + \mu \langle x_n y \rangle + \alpha\, \lambda_n \\
&+ \langle x_n y \rangle\, (1 - \langle y \rangle) [\, \mathrm{E}\{\mathbf{g}(\mathbf{x}) | y{=}0\} - \mathrm{E}\{\mathbf{g}(\mathbf{x}) | y{=}1\}\,] \; = \; 0
\end{aligned} \tag{25}
$$

The substitution of the exact form for μ from equation (23) into (25) serves to conveniently cancel the last term in (25) and yields for n=1,2,...,N the set of equations

$$
[\, \langle \mathbf{x}\, x_n y \rangle^{\mathrm{T}} - \frac{\langle x_n y \rangle \langle \mathbf{x}\, y \rangle^{\mathrm{T}}}{\langle y \rangle} \,]\, \boldsymbol{\lambda} \; = \; -\, \alpha\, \lambda_n \tag{26}
$$

Expressing (26) in matrix form for n=1,2,...,N gives the matrix equation

$$
\mathbf{R}\, \boldsymbol{\lambda} \; = \; -\, \alpha\, \boldsymbol{\lambda} \tag{27}
$$

where

$$
\mathbf{R} \; = \; \langle \mathbf{x}\, \mathbf{x}^{\mathrm{T}} y \rangle \; - \; \frac{\langle \mathbf{x}\, y \rangle \langle \mathbf{x}\, y \rangle^{\mathrm{T}}}{\langle y \rangle} \tag{28}
$$

and $\langle \cdot \rangle$ indicates that the expectation is carried out over the joint distribution P($\mathbf{x}$,y). Hence $\boldsymbol{\lambda}$ is an eigenvector of $\mathbf{R}$ with eigenvalue $-\alpha$.

Since $\langle \mathbf{x} y \rangle / \langle y \rangle = \langle \mathbf{x} | y{=}1 \rangle$, equation (28) can be re-written in the form

$$\mathbf{R} = \langle(\mathbf{x} - \langle \mathbf{x} \mid y{=}1 \rangle)\, \mathbf{x}^{\mathbf{T}}\, y\rangle \tag{29}$$

Now multiply both sides of (29) by $\boldsymbol{\lambda}^{\mathrm{T}}$ giving

$$< (\boldsymbol{\lambda}^{\mathrm{T}} \mathbf{x} - \boldsymbol{\lambda}^{\mathrm{T}} <\mathbf{x} \mid y=1>) \mathbf{x}^{\mathrm{T}} y> \; = \; \boldsymbol{\lambda}^{\mathrm{T}} \mathbf{R} \\ = \; -\alpha \, \boldsymbol{\lambda}^{\mathrm{T}} \tag{30}$$

since $-\alpha$ is the designated eigenvalue corresponding to $\boldsymbol{\lambda}$. Substituting μ from (24) and transposing both sides of (30) yields

$$< (\boldsymbol{\lambda}^{\mathrm{T}} \mathbf{x} + \mu) \mathbf{x} \, y > \; = \; - \alpha \, \boldsymbol{\lambda} \tag{31}$$

Defining (14) as the evidence $\zeta(\mathbf{x}) = - \mu - \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{x}$ and substituting into (31) yields

$$<\zeta(\mathbf{x}) \, \mathbf{x} \, y > \; = \; \alpha \, \boldsymbol{\lambda} \tag{32}$$

The term $<\zeta(\mathbf{x})\mathbf{x}y>$ can be seen to equal $<\zeta(\mathbf{x})\mathbf{x}|y=1><y>$. Furthermore, from the assumption that $|\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{x}| <<1$ for typical $\mathbf{x} \epsilon X$, it can be seen from equations (8) and (10) that $<y> \simeq 1/2$ thus ensuring that $P(y=0) = P(y=1)$ as stated earlier thus implying that the neuron output has maximum entropy $H(y)$ for a one-bit channel. Substitution of these results into equation (32) yields

$$\begin{pmatrix} <\zeta(\mathbf{x})x_1|y=1> \\ <\zeta(\mathbf{x})x_2|y=1> \\ . \\ . \\ . \\ <\zeta(\mathbf{x})x_N|y=1> \end{pmatrix} \approx 2\,\alpha \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ . \\ . \\ . \\ \lambda_N \end{pmatrix} \tag{33}$$

Equation (33) can be iteratively solved using a modification of the principal component analyzer equation derived by Oja [11]. In particular, let π represent a plasticity coefficient which decreases monotonically and proportionally with time t. Also, assume that the original normalization constraint $\boldsymbol{\lambda}^T\boldsymbol{\lambda}=\gamma^2$ is imposed. Then an approximated solution to (33) can be obtained through the difference equation

$$\boldsymbol{\lambda}(t+\Delta t) = \boldsymbol{\lambda}(t) + \pi\, \zeta(\mathbf{x}(t))\,[\,\mathbf{x}(t) - \gamma^{-2}\zeta(\mathbf{x}(t))\,\boldsymbol{\lambda}(t)] \qquad \textbf{(34)}$$

which is *only* executed upon the Hebbian condition that y=1. Note that the factor of 2 in equation (33) has been absorbed into the γ^2 factor here.

Equation (34) together with the decision function (13) and threshold (24) provide the computational framework necessary for implementing the MMI strategy of the model neuron. The required computations are depicted in Figure 2 and described in the following three steps carried out in response to successive input vectors $\mathbf{x}\in X$ and some presumed non-zero initial conditions on $\boldsymbol{\lambda}$.

(i) First, the inner product of $\boldsymbol{\lambda}$ with the input $\mathbf{x}$ is formed and added to the decision threshold μ as per (24) to form the evidence ζ. Based on the evidence ζ, a decision as induced by (13) is made in which an output either is (y=1) or is not (y=0) stochastically generated. If and only if an output is generated, then y serves to gate equation (34).

(ii) Equation (34) is then carried out in an iterative fashion. Conditional on an affirmative output y=1, individual weight components λ_i are updated in accordance with the evidence $\zeta(\mathbf{x})$ and only when x_i=1 (ζ already being found in step i). It should be noted that regarding the ML decision rule, only the sign and not the magnitude of λ is important. The magnitude does become important in the evaluation of the magnitude of the nuisance terms in equations (20) and (21).

(iii) Finally, the threshold μ in equation (24) must be updated. On first thought, the estimation of N separate entries in the vector $<\mathbf{x}|y=1>$ is required. This is not true though since only the scalar $\lambda^T<\mathbf{x}|y=1>$ must be estimated. Since $\lambda^T\mathbf{x}$ is generated for each input, but only used to update λ in the event y=1, the scalar $\lambda^T\mathbf{x}$ can therefore additionally be used to iteratively obtain an averaged-based estimate of μ as gated by the neuron output y. An auxilliary differential equation can be used to implement this function.

Steps (i)-(iii) provide a simple and extremely efficient strategy regarding the realization of a neural model which attempts to optimize its information throughput. Calculations are simplified even further when it is realized that the actual computation $\lambda^T\mathbf{x}$ only involves a sum off all λ_i such that x_i=1. Simulation results are discussed next.

Simulation Results

The derived single neuron model response was simulated using a prescribed ensemble of 20 training vectors each containing 20 bits. One bit was clamped to zero, one bit was clamped to one, some bits repeated, and some random. It was found that the stochastic decision rule (13) was required as was the incorporation of a monotonically decreasing temperature parameter T which normalized the evidence function. This parameter significantly enhanced the learning rate which typically required under a 100 iterations through the training set. In the limits as $T\rightarrow\infty$ and $T\rightarrow 0$, the decision rule (13) approaches $P(y=1|\mathbf{x})=1/2$ and the ML decision rule of equation (14), respectively.

Figures 3 through 5 describe typical simulation results for 16 Monte Carlo simulation runs using the prescribed training vector set. Figure 3 shows that the algorithm consistently converged to the same eigenvector parameter set. Bit 10 which was clamped to zero conveyed no information and the corresponding eigenvector entry (synaptic efficacy) asymptotically approached zero. Bit 20 which was clamped to one maintained a preferentially high value. Varied training sets consistently gave yielded restricted dynamic ranges for synaptic efficacies which is a desirable characteristic for most hardware implementations.

Figure 4 describes the corresponding probability assignments $P(y|\mathbf{x})$ for each training vector $\mathbf{x}$ for a selected temperature value T. A key thing to notice in this figure is that approximately half the training vectors have $P(y=1|\mathbf{x})>1/2$ thus implying the condition $P(y=1)\approx 1/2$ (the maximum entropy condition). This was found to be the case for all training ensembles which were simulated. In the limit as $T\rightarrow\infty$, Figure 4 would have asymptotic values of 0 and 1. A review of the training vectors which result in the generation of an action potential show that these vectors show marked similarity so an automatic grouping (decision space) is obtained at the end of training.

Figure 5a through 5d represent a statistical summary of the 16 Monte Carlo runs using the same training data. Figure 5a depicts the estimated threshold μ which can be seen to vary insignificantly. Figure 5b tabulates the cardinality of the subset of training vectors which result in an action potential. Approximately half of the training vectors ultimately invoked action potentials thus satisfying the maximized entropy condition on the neural output y. Figure 5c is an arithmetic average of the computed eigenvectors. Again, bit 0 which was clamped to zero resulting in the corresponding eivenvector entry tending to zero. Finally, Figure 5d is the averaged ML decision rule for each of the input 20 training vectors which shows that a very consistent decision rule is generated.

Conclusions

The use of maximum entropy with the specified neuron model leads to a computing structure which closely resembles the Hopfield neuron model including threshold and decision rules. The ME objective function is inadequate, however, in that the moment constraints on the selected sampling functions must be specified by a supervisory source external to the model neuron. Alternatively, a maximum mutual information (MMI) criterion leads to a fully constrained parameter set for a Gibbs distribution and as a consequence a fully specified set of moments on the selected sampling functions. Furthermore, approximate solutions based on an MMI criterion lead to a computing structure having a simple and extremely efficient implementation and has characteristics which are analogous to several existing biological neuron models. Some of these characteristics include:

- The Hopfield neuron model with corresponding decision threshold
- A Hebbian learning methodology using a conditionalized adaptation of Oja [11]
- Principal component analysis conditional on the neuron "firing"
- A sigmoidal transfer characteristic required for decision making

The work described is contrasted with the work of Linsker ([12]-[16]) in Table I which follows:

Comparison of Linsker Model with the Subject Paper's Model

Basis of Comparison	Linsker	Subject Paper
Signal model	Continuous	Discrete/Binary
Model origins	Gaussian channel	Binary channel
Primary noise and distortion emphasis	Additive input/output noise	Distortion due to many-to-one mapping
Neural parameters	Synaptic efficacies	Synaptic efficacies and decision threshold
Probability model and basis	Gaussian model for noise, sources, and cell output	Determined on basis of sampling functions and ME
Learning strategy	Linear	Nonlinear
Decision rule	Not applicable	Stochastic
Hebbian learning rule	Derived to maximize mutual information	Arises as consequence of assumed model
Sigmoidal nonlinearity	Not required	Arises as consequence of assumed model

Table I

As can be seen from this table, there are major differences between Linsker's model and the model proposed here although there is a concurrence of philosophical approach [15]. The major similarity is that mutual information is used within our respective models and that a feedback mechanism in the form of Hebbian learning provides for the optimization of this information measure.

The most recent work of Linsker [16] perhaps bears the closed resemblance to what is described here. In [16], Linsker explicitly derives local learning rules which serve to maximize mutual information. However, these derivations assume that the input ensemble, the additive input noise ensemble, and the additive output noise random variable are all gaussian processes. This leads to analytically tractable solutions for a linear neuron model. Additive noise sources are not considered here although such an assessment has begun within the scope of the model described. Tentative results suggest that the computational scheme which results remains unaltered for the case of additive input and output gaussian noise. Additive noise serves to increase the conditional entropy of the output given the input by an additive bias in a manner consistent with [16]. In summary of this comparison, it is thought that Linsker obtained different results than stated here which arose from a different set of model assumptions.

The described neural model should be contrasted with conventional applications of information theory to information transmission. In conventional applications, a static channel configuration is presumed and input distribution (code set) is sought which maximizes the information capacity of the specified channel. Alternatively, here the input distribution serves to configure the channel properties through Hebbian feedback so as to maximize the mutual information between channel input and output.

Perhaps more important than the results obtained from the MMI analysis applied to the single neuron is the use of the MMI criterion itself. An important practical extension of the approach described here is the application of the MMI criterion to a parallel assembly of neurons which can describe a parallel information channel. The application of the MMI criterion to such channels suggests that cooperative/competitive interactions among neurons can be used to effect a global MMI strategy between a boolean input set X and output set Y while maintaining a similar computational strategy to that delineated for a single neuron. This requires supplementing the previously selected sampling functions with additional sampling functions which are functions of the neuron input $\mathbf{x}\subset X$, output y, and the outputs of neighboring neurons $\mathbf{y}'\subset Y$. The Gibbs Mutual Information Theorem can then be applied.

Appendix A

Gibbs Mutual Information Theorem

The mutual information between two discrete random ensembles $\mathbf{x} \in X$ and $\mathbf{y} \in Y$ is defined to be

$$I(\mathbf{x};\mathbf{y}) = \sum_{\mathbf{x}' \in X, \mathbf{y}' \in Y} P(\mathbf{x}',\mathbf{y}') \ln \frac{P(\mathbf{x}',\mathbf{y}')}{P(\mathbf{x}')\, P(\mathbf{y}')} \tag{A1}$$

The stated objective is that the channel output code $\mathbf{y}$ convey as much information as possible regarding the channel input $\mathbf{x}$. Assume that a Gibbs distribution describes the modeled state of knowledge which is based on a set of sampling functions $\mathbf{F}=(F_1\ F_2 \cdots F_N)^T$. The ME distribution P is given by

$$P(\mathbf{x},\mathbf{y}) = \frac{1}{Z} \exp[\, - \boldsymbol{\lambda}^T \mathbf{F}(\mathbf{x},\mathbf{y})\,] \tag{A2}$$

with

$$Z = \sum_{\mathbf{x}' \in X, \mathbf{y}' \in Y} \exp[\, - \boldsymbol{\lambda}^T \mathbf{F}(\mathbf{x}',\mathbf{y}')\,] \tag{A3}$$

where $\boldsymbol{\lambda}=(\lambda_1\ \lambda_2 \cdots \lambda_N)^T$ is a column vector. The ME assumption regarding (A2) implies that all $E\{F_n\} = <F_n>$ are known a priori. Here, however, only the functional forms of (A2) and (A3) are assumed.

All the marginals of (A2) are easily found to be

$$P(\mathbf{x}) = \frac{1}{Z} \sum_{\mathbf{y}' \in Y} \exp[- \boldsymbol{\lambda}^T \mathbf{F}(\mathbf{x},\mathbf{y}')] \tag{A4}$$

$$P(\mathbf{y}) = \frac{1}{Z} \sum_{\mathbf{x}' \in X} \exp[- \boldsymbol{\lambda}^T \mathbf{F}(\mathbf{x}',\mathbf{y})] \tag{A5}$$

$$P(\mathbf{x} \mid \mathbf{y}) = \frac{\exp[- \boldsymbol{\lambda}^T \mathbf{F}(\mathbf{x},\mathbf{y})]}{\sum_{\mathbf{x}' \in X} \exp[- \boldsymbol{\lambda}^T \mathbf{F}(\mathbf{x}',\mathbf{y})]} \tag{A6}$$

$$P(\mathbf{y} \mid \mathbf{x}) = \frac{\exp[- \boldsymbol{\lambda}^T \mathbf{F}(\mathbf{x},\mathbf{y})]}{\sum_{\mathbf{y}' \in Y} \exp[- \boldsymbol{\lambda}^T \mathbf{F}(\mathbf{x},\mathbf{y}')]} \tag{A7}$$

The stated objective is to find the vector $\boldsymbol{\lambda}$ which is an extremum of (A1). An auxiliary condition is imposed on the resulting vector that $\boldsymbol{\lambda}^T\boldsymbol{\lambda}=\gamma^2$ where γ^2 is a normalization constant. The objective function becomes

$$J(\boldsymbol{\lambda}) = I(\mathbf{x};\mathbf{y};\boldsymbol{\lambda}) + \frac{\alpha}{2}\boldsymbol{\lambda}^T\boldsymbol{\lambda} \tag{A8}$$

Extrema of (A8) occur at $\partial J/\partial\lambda_n=0$. Since the partial derivative of the second term in (A8) is trivial, it will only be considered at the end of the derivation where it will be re-introduced.

The partial derivative of $I(\mathbf{x};\mathbf{y};\boldsymbol{\lambda})$ with respect to λ_n is given by

$$\frac{\partial I}{\partial\lambda_n} = \sum_{\mathbf{x}'\epsilon X,\mathbf{y}'\epsilon Y}\frac{\partial P(\mathbf{x}',\mathbf{y}')}{\partial\lambda_n}\ln\frac{P(\mathbf{y}'|\mathbf{x}')}{P(\mathbf{y}')} + \sum_{\mathbf{x}'\epsilon X,\mathbf{y}'\epsilon Y}P(\mathbf{x}',\mathbf{y}')\frac{\partial}{\partial\lambda_n}[\ln P(\mathbf{y}'|\mathbf{x}') - \ln P(\mathbf{y}')] \tag{A9}$$

or

$$\frac{\partial I}{\partial\lambda_n} = Q(\boldsymbol{\lambda}) + R(\boldsymbol{\lambda}) \tag{A10}$$

where Bayes' Theorem has been applied to the argument of the natural logarithm function. The terms $Q(\boldsymbol{\lambda})$ and $R(\boldsymbol{\lambda})$ will be solved for separately.

Evaluating the first term in (A10) and substituting the known distributional expressions yields

$$\frac{\partial \mathrm{P}(\mathbf{x},\mathbf{y})}{\partial \lambda_n} = \frac{\partial}{\partial \lambda_n} \mathrm{Z}^{-1} \exp[\,-\,\boldsymbol{\lambda}^{\mathrm{T}} \mathbf{F}(\mathbf{x},\mathbf{y})\,] \tag{A11}$$

$$= -\,\mathrm{Z}^{-2} \exp[\,-\,\boldsymbol{\lambda}^{\mathrm{T}} \mathbf{F}(\mathbf{x},\mathbf{y})\,] \frac{\partial \mathrm{Z}}{\partial \lambda_n} \;-\; \mathrm{Z}^{-1} \mathbf{F}_n(\mathbf{x},\mathbf{y}) \exp[\,-\,\boldsymbol{\lambda}^{\mathrm{T}} \mathbf{F}(\mathbf{x},\mathbf{y})\,] \tag{A12}$$

The partial of the partition function Z with respect to λ_n implies that

$$\frac{1}{\mathrm{Z}} \frac{\partial \mathrm{Z}}{\partial \lambda_n} = \frac{1}{\mathrm{Z}} \frac{\partial}{\partial \lambda_n} \sum_{\mathbf{x}' \in X, \mathbf{y}' \in Y} \exp[\,-\,\boldsymbol{\lambda}^{\mathrm{T}} \mathbf{F}(\mathbf{x}',\mathbf{y}')\,] \tag{A13}$$

$$= -\,\langle \mathrm{F}_n \rangle \tag{A14}$$

which when substituted back into (A12 yields

$$\mathbf{Q}(\boldsymbol{\lambda}) = \sum_{\mathbf{x}' \in X, \mathbf{y}' \in Y} \mathrm{P}(\mathbf{x}',\mathbf{y}')\,[\,\langle \mathrm{F}_n \rangle \;-\; \mathrm{F}_n(\mathbf{x}',\mathbf{y}')\,] \ln \frac{\mathrm{P}(\mathbf{y}'|\mathbf{x}')}{\mathrm{P}(\mathbf{y}')} \tag{A15}$$

which completes the first term in (A10). The second term in (A10) turns out to be significantly more complicated, but eventually evaluates to a simple result. The determination of $\mathrm{R}(\boldsymbol{\lambda})$ begins by solving for

$$\frac{\partial}{\partial \lambda_n} \ln \mathbf{P}(\mathbf{y}|\mathbf{x}) = \frac{1}{\mathbf{P}(\mathbf{y}|\mathbf{x})} \frac{\partial}{\partial \lambda_n} \mathbf{P}(\mathbf{y}|\mathbf{x}) \qquad \textbf{(A16)}$$

which yields

$$\frac{\partial}{\partial \lambda_n} \mathbf{P}(\mathbf{y}|\mathbf{x}) = \frac{\partial}{\partial \lambda_n} \frac{\exp[\,-\,\boldsymbol{\lambda}^{\mathbf{T}}\,\mathbf{F}(\mathbf{x},\mathbf{y})\,]}{\sum_{\mathbf{y}' \in Y} \exp[\,-\,\boldsymbol{\lambda}^{\mathbf{T}}\,\mathbf{F}(\mathbf{x},\mathbf{y}')\,]} \qquad \textbf{(A17)}$$

This evaluates to

$$\begin{aligned} &\{-F_n(\mathbf{x},\mathbf{y}) \exp[\,-\,\boldsymbol{\lambda}^{\mathbf{T}}\,\mathbf{F}(\mathbf{x},\mathbf{y})\,] \sum_{\mathbf{y}' \in Y} \exp[\,-\,\boldsymbol{\lambda}^{\mathrm{T}}\,\mathbf{F}(\mathbf{x},\mathbf{y}')\,] \\ &+ \Big(\sum_{\mathbf{y}' \in Y} F_n(\mathbf{x},\mathbf{y}') \exp[\,-\,\boldsymbol{\lambda}^{\mathbf{T}}\,\mathbf{F}(\mathbf{x},\mathbf{y}')\,] \Big) \exp[\,-\,\boldsymbol{\lambda}^{\mathbf{T}}\,\mathbf{F}(\mathbf{x},\mathbf{y})\,] \} \\ &\div \Big\{ \sum_{\mathbf{y}' \in Y} \exp[\,-\,\boldsymbol{\lambda}^{\mathbf{T}}\,\mathbf{F}(\mathbf{x},\mathbf{y}')\,] \Big\}^2 \end{aligned} \qquad \textbf{(A18)}$$

Equation (A18) simplifies to

$$\begin{aligned}&\{ - F_n(\mathbf{x},\mathbf{y})\, P(\mathbf{y}|\mathbf{x}) \sum_{\mathbf{y}'\epsilon Y} \exp[- \boldsymbol{\lambda}^T \mathbf{F}(\mathbf{x},\mathbf{y}')] \\ &\quad + (\sum_{\mathbf{y}'\epsilon Y} F_n(\mathbf{x},\mathbf{y}') \exp[- \boldsymbol{\lambda}^T \mathbf{F}(\mathbf{x},\mathbf{y}')]) P(\mathbf{y}|\mathbf{x}) \} \\ &\quad \div \{ \sum_{\mathbf{y}'\epsilon Y} \exp[- \boldsymbol{\lambda}^T \mathbf{F}(\mathbf{x},\mathbf{y}')] \}\end{aligned} \qquad \textbf{(A19)}$$

which when substituted back into (A16) gives

$$\begin{aligned}\frac{\partial}{\partial \lambda_n} \ln P(\mathbf{y}|\mathbf{x}) = &\{ \sum_{\mathbf{y}'\epsilon Y} F_n(\mathbf{x},\mathbf{y}') \exp[-\boldsymbol{\lambda}^T \mathbf{F}(\mathbf{x},\mathbf{y}')] \\ &+ - F_n(\mathbf{x},\mathbf{y}) \sum_{\mathbf{y}'\epsilon Y} \exp[-\boldsymbol{\lambda}^T \mathbf{F}(\mathbf{x},\mathbf{y}')] \} \\ &\div \{ \sum_{\mathbf{y}'\epsilon Y} \exp[- \boldsymbol{\lambda}^T \mathbf{F}(\mathbf{x},\mathbf{y}')] \}\end{aligned} \qquad \textbf{(A20)}$$

or

$$\frac{\partial}{\partial \lambda_n} \ln P(\mathbf{y}|\mathbf{x}) = E\{ F_n(\mathbf{x},\mathbf{y}|\mathbf{x}) \} - F_n(\mathbf{x},\mathbf{y}) \qquad \textbf{(A21)}$$

which is the first term of $R(\boldsymbol{\lambda})$. Now the second term in $R(\boldsymbol{\lambda})$ is found by

$$\frac{\partial}{\partial \lambda_n} \ln \mathbf{P}(\mathbf{y}) \;=\; \frac{1}{\mathbf{P}(\mathbf{y})} \frac{\partial \mathbf{P}(\mathbf{y})}{\partial \lambda_n} \tag{A22}$$

Solving as in the prior case

$$\frac{\partial \mathbf{P}(\mathbf{y})}{\partial \lambda_{\blacksquare}} \;=\; \frac{\partial}{\partial \lambda_{\blacksquare}} \mathrm{Z}^{-\blacktriangleright} \sum_{l\,'\in\ast} \mathbf{exp}[\,-\,\lambda^{\ast}\mathbf{F}(\mathbf{x}\,',\mathbf{y})\,] \tag{A23}$$

which equals

$$\begin{aligned}&\{\mathrm{Z}^{-2} \sum_{\mathbf{x}\,'\in X,\mathbf{y}\,'\in Y} \mathrm{F}_{\mathrm{n}}(\mathbf{x}\,',\mathbf{y}\,')\,\mathbf{exp}[\,-\,\lambda^{\mathrm{T}}\,\mathbf{F}(\mathbf{x}\,',\mathbf{y}\,')\,]\}\,\{\sum_{\mathbf{x}\,'\in X} \mathbf{exp}[\,-\,\lambda^{\mathrm{T}}\,\mathbf{F}(\mathbf{x}\,',\mathbf{y})\,]\} \,+ \\ &\{\,-Z^{-1} \sum_{\mathbf{x}\,'\in X} \mathrm{F}_{n}(\mathbf{x}\,',\mathbf{y})\,\mathbf{exp}[\,-\,\lambda^{\mathrm{T}}\,\mathbf{F}(\mathbf{x}\,',\mathbf{y})\,]\}\end{aligned} \tag{A24}$$

Equation (A22) becomes

$$\frac{\partial}{\partial \lambda_n} \ln \mathbf{P}(\mathbf{y}) \;=\; \langle \mathrm{F}_{n} \rangle \;-\; \frac{\sum_{\mathbf{x}\,'\in X} \mathrm{F}_{n}(\mathbf{x}\,',\mathbf{y})\,\mathbf{exp}[\,-\,\lambda^{\mathrm{T}}\,\mathbf{F}(\mathbf{x}\,',\mathbf{y})\,]}{\sum_{\mathbf{x}\,'\in X} \mathbf{exp}[\,-\,\lambda^{\mathrm{T}}\,\mathbf{F}(\mathbf{x}\,',\mathbf{y})\,]} \tag{A25}$$

$$= \langle F_n \rangle - E\{F_n(\mathbf{x},\mathbf{y})\,|\mathbf{y}\} \qquad \textbf{(A26)}$$

Substituting (A21) and (A26) back into R(λ) yields

$$\textbf{(A27)}$$

$$R(\lambda) = \sum_{\mathbf{x}'\in X,\mathbf{y}'\in Y} P(\mathbf{x}',\mathbf{y}')\{E[F_n(\mathbf{x}',\mathbf{y}')\,|\,\mathbf{x}'] - F_n(\mathbf{x}',\mathbf{y}') - \langle F_n \rangle + E[F_n(\mathbf{x}',\mathbf{y}')\,|\,\mathbf{y}']\}$$

Equation (A27) consists of four terms. Each term can be evaluated separately by finding the expected value of each. Respectively, these terms are given by

$$E\{E[F_n(\mathbf{x}',\mathbf{y}')\,|\,\mathbf{x}']\} = \langle F_n \rangle \qquad \textbf{(A28)}$$

$$E\{E[F_n(\mathbf{x}',\mathbf{y}')\,|\,\mathbf{y}']\} = \langle F_n \rangle \qquad \textbf{(A29)}$$

$$E\{F_n(\mathbf{x}',\mathbf{y}')\} = <F_n> \tag{A30}$$

$$E\{<F_n>\} = <F_n> \tag{A31}$$

After substituting (A28) through (A31) back into R(λ), it can be seen that

$$\mathbf{R(\lambda) = 0} \tag{A32}$$

so that $\partial I/\partial\lambda_n = Q(\lambda)$ or

$$\frac{\partial I}{\partial \lambda_n} = \sum_{\mathbf{x}'\in X, \mathbf{y}'\in Y} P(\mathbf{x}',\mathbf{y}')\,[<F_n> - F_n(\mathbf{x}',\mathbf{y}')]\,\ln\frac{P(\mathbf{y}'\,|\,\mathbf{x}')}{P(\mathbf{y}')} \tag{A33}$$

It is also interesting to find the second partial of I with respect to λ_m in order to assess the convexity of I($\mathbf{x}$;$\mathbf{y}$;λ) as a function of λ. A symmetrical matrix results which needs further evaluation regarding its negative definiteness. The Lagrangian multiplier α serves to regularize the Hessian.

The extremum of the original objective function J(λ) defined in equation (A8) can now be found from (A33) and the derivative of the constraint term. The resulting equation is called the Gibb's Mutual Information Theorem and is given by Equation (A34).

$$\frac{\partial \mathrm{J}}{\partial \lambda_n} = \sum_{\mathbf{x}' \in X, \mathbf{y}' \in Y} \mathrm{P}(\mathbf{x}', \mathbf{y}') \, [< \mathrm{F}_n > - \mathrm{F}_n(\mathbf{x}', \mathbf{y}') \,] \ln \frac{\mathrm{P}(\mathbf{x}' \mid \mathbf{y}')}{\mathrm{P}(\mathbf{x}')} + \alpha \, \lambda_n \qquad \textbf{(A34)}$$

Appendix B

Nuisance Term Analysis

The nuisance terms in equations (20) and (21) are given by

$$E\{(\langle x_{\blacksquare} y\rangle - x_{\blacksquare} y) \ln P(y=0|\mathbf{x})\} - E\{(\langle x_{\blacksquare} y\rangle - x_{\blacksquare} y) \ln P(y=0)\} \tag{B1}$$

and

$$E\{(\langle y\rangle - y) \ln P(y=0|\mathbf{x})\} - E\{(\langle y\rangle - y) \ln P(y=0)\} \tag{B2}$$

respectively. Now let these expressions be represented by (A-B) and (C-D), respectively. Now solve for the designated terms individually beginning with A.

The function g($\mathbf{x}$)=ln P(y=0|$\mathbf{x}$) is given by

$$\mathbf{g}(\mathbf{x}) = \ln \left\{ \frac{1}{1 + \exp[-\sum_{i=1}^{N} \lambda_i x_i - \mu]} \right\} . \tag{B3}$$

Note that

$$E\{x_n y \mathbf{g}(\mathbf{x})\} = \langle x_n y\rangle E\{\mathbf{g}(\mathbf{x}, x_n=1) | x_n=1, y=1\} \tag{B4}$$

since $<x_n y>=\Pr(x_n=1,y=1)$. Also it can be seen that

$$E\{\mathbf{g}(\mathbf{x})\} = (1-<y>)\, E\{\mathbf{g}(\mathbf{x})|y=0\} + <y>\, E\{\mathbf{g}(\mathbf{x})|y=1\} \qquad \textbf{(B5)}$$

If the approximation is made that $E\{g(\mathbf{x})|y=1\} \approx E\{g(\mathbf{x},x_n=1)|x_n=1,y=1\}$ then (B4) together with (B5) imply that A approximately equals

$$<x_n y>\,(1-<y>)\,[E\{\mathbf{g}(\mathbf{x})\,|\,y=0\} - E\{(\mathbf{g}(\mathbf{x})\,|\,y=1\}] \qquad \textbf{(B6)}$$

The approximation made here becomes more exact as the cardinality of the subset of the elements of X contributing unique information regarding the decision y increases. An analysis and comparison of the distributions $P(\mathbf{x}\backslash x_n|x_n=1,y=1)$ and $P(x|y=1)$ shows that the approximation requires that the relative magnitude of $|\lambda_n|$ is much less than the magnitude of the evidence $|\xi(x)|=|\lambda^T\mathbf{x}+\mu|$ for typical $\mathbf{x}\in X$. This of course requires that the model neuron have many inputs. Biological neurons often have more than 5000 inputs thus ensuring that the evaluation of the expression in (B6) yields a value that is approximately zero provided that γ^2 is sufficiently small.

Finally, regarding the evaluation of (B6),

$$E\{\mathbf{g}(\mathbf{x})\,|\,y=\mathbf{0}\} = \frac{\sum_{\mathbf{x}'\in X} \mathbf{g}(\mathbf{x}')\,\mathbf{exp}\,[\,\mathbf{0}\,]}{\sum_{\mathbf{x}'\in X} \mathbf{exp}\,[\,\mathbf{0}\,]} \tag{B7}$$

and

$$E\{\mathbf{g}(\mathbf{x})\,|\,y=1\} = \frac{\sum_{\mathbf{x}'\in X} \mathbf{g}(\mathbf{x}')\,\mathbf{exp}\,[\,\boldsymbol{\lambda}^{\mathbf{T}}\mathbf{x}'\,]}{\sum_{\mathbf{x}'\in X} \mathbf{exp}\,[\,\boldsymbol{\lambda}^{\mathbf{T}}\mathbf{x}'\,]} \tag{B8}$$

Thus if $|\boldsymbol{\lambda}^{T}\mathbf{x}| << 1$ for typical $\mathbf{x}\in X$, then (B7) and (B8) imply that the third term in μ in equation (23) becomes arbitrarily small under these conditions.

In summary, the second nuisance term in (20) can be evaluated by

(i) Ensuring that $|\Sigma\lambda| << 1$ (which also guarantees $|\boldsymbol{\lambda}^{T}\mathbf{x}|$ is small for typical $\mathbf{x}\in X$) by making γ^{2} is sufficiently small.

(ii) Making the relative magnitude of $|\lambda|$ to the magnitude of the evidence $|\boldsymbol{\lambda}^{T}\mathbf{x}+\mu|$ small for typical $\mathbf{x}\in X$ thereby requiring that many neuronal inputs contribute unique information towards the decision y.

The second nuisance term in (21) can be evaluated in a fashion similar to that done here yielding the somewhat more simplified results reported in the text.

The terms B and D in equations (B1) and (B2) can be evaluated in a straightforward manner. Term B is given by

$$E\{(<x_n y> - x_n y) \ln P(y)\} \quad \textbf{(B9)}$$

which can be expanded into

$$<x_n y>(1 - <y>) \ln P(y=0) + <x_n y><y> \ln P(y=1) - <x_n y> \ln P(y=1) \quad \textbf{(B10)}$$

Equation (B10) can be expressed as

$$<x_n y>(1 - <y>) \ln P(y=0) - <x_n y>(1 - <y>) \ln P(y=1) \quad \textbf{(B11)}$$

From equation (B1) it is clear that if P(y)=1/2, then the term B goes to zero. Term D can also be evaluated:

$$E\{(<y> - y) \ln P(y)\} = <y> \ln P(y=0) + <y> \ln P(y=1) - \ln P(y=1) \quad \textbf{(B12)}$$

Grouping like terms this expression becomes

$$E\{(<y> - y) \ln P(y)\} = <y> \ln P(y=0) - (1 - <y>) \ln P(y=1) \quad \textbf{(B13)}$$

Again, $P(y)=1/2$ will cause this term to be zero.

As a final note, guaranteeing that γ^2 is small seems to be a primary requirement regarding the validity of the developed model. This implies that the model neuron must typical operate near the linear region of the decision-generating sigmoidal curve described by the Boltzmann distribution in equation (13). It should be noted that γ^2 represents the only degree-of-freedom afforded the neural designer within the proposed model. Since the principal component analysis result is exact, the flexibility offered by properly choosing γ^2 can only impact the fidelity of the approximations made in the determination of the simplified expression for the Hopfield threshold μ.

REFERENCES

[1] Cox, R.T. (1961). *The algebra of probable inference*, The Johns Hopkins Press, Baltimore.

[2] Jaynes, E. T. (1957). "How does the brain do plausible reasoning?" reprinted 1988 in *Maximum-entropy and Bayesian methods in science and engineering*, Vol. I, from an internal report of the Microwave Laboratory and Dept. of Physics, Stanford, CA

[3] Jaynes, E. T. (1968). "Prior probabilities," in *IEEE Trans. on Systems Sci. and Cybernetics*,**4**, 227-241.

[4] Mead, L. R. and Papanicolaou, N. (1984). Maximum entropy in the problem of moments," *J.Math.Phys.* **25**, 2404-2417.

[5] Cover, T.M. and Thomas, J.A. (1991). *Elements of Information Theory*, Wiley.

[6] Agmon, N., Alhassid, Y., Levine, R.D. (1979). "An algorithm for finding the distribution of maximal entropy," *J. of Computational Phys.* **30**, 250-258.

[7] Luttrell, S. P. (1988). "The use of Bayesian and entropic methods in neural network theory," in *Maximum entropy and Bayesian methods*, 363-370, Kluwer Academic Publishers, Netherlands.

[8] MacKay, D. J. C. (1991). "Maximum entropy connections: Neural networks," in *Maximum entropy and Bayesian methods*, 237-244, Kluwer Academic Publishers, Netherlands.

[9] Hopfield, J. J. (1982). "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci. USA*, **79**, 2554-2558.

[10] Hopfield, J. J. and Tank, D. W. (1985). "Neural computations of decisions in optimization problems," *Biological Cybernetics*, **52**, 141-152.

[11] Oja, E. (1982). "A simplified neuron model as a principal component analyzer," *J. of Math. Biol.* **15**, 267-273.

[12] Linsker, R. (1986). "From network principles to neural architecture: Emergence of spatial-opponent cells," *Proc. Natl. Acad. Sci. USA*, pp.7508-7512.

[13] Linsker, R. (1988). "Self-organization in a perceptual network," *IEEE COMPUTER* 105-117.

[14] Linsker, R. (1988). "Towards an organizing principle for a layered perceptual network," American Institute of Physics publication, pp. 485-494.

[15] Linsker, R. (1990). "Perceptual neural organization: Some approaches based on network models and information theory," *Annu. Rev. Neurosci.*, pp.257-281.

[16] Linsker, R. (1992). "Local synaptic learning rules suffice to maximize mutual information in a linear network," *Neural Computation*, pp.691-701