Presented at the 1993 Symposium on Nonlinear Theory and Its Applications, (Hawaii, December 5-10, 1993).

FREE ENERGY AS A DRIVING FUNCTION IN NEURAL NETWORKS

Alianna J. Maren
Accurate Automation Corporation
7001 Shallowford Rd., Chattanooga, TN 37421-1716

ABSTRACT

Use of a new type of driving function makes possible a class of neural networks which exhibit a richer range of temporal behaviors than currently are available. Most existent driving functions for neural networks (e.g. the energy function for a Hopfield neural network) drive the network towards one of a set of specific states encoded in the network. In contrast, this note describes a class of free energy functions which drive a neural network towards a state with certain characteristics. However, the energy function is non-specific with regard to the actual way in which the desired state characteristics are achieved. The non-state-specific driving force is used in conjunction with a state-specific driving force, which is encoded as usual into the connections weights and is implicit in the architecture of the system.

The advantage of this approach is that by decoupling the state- specific and non-state-specific driving forces within a neural network, we obtain a general class of networks which is responsive to the inputs, and yet which permits temporal evolution of states and temporal pattern association. The state-specific driving forces are encoded via connection weights, both as weighted sums from the input layer and as (Hebbian) learned lateral connections between certain nodes in the output layer. The non-state-specific driving forces come about through application of a Helmholtz free energy to the output layer. The free energy includes both energy and entropy terms. The entropy is formulated in terms of distribution of local configurations of states, or *micropatterns*. Thus the free energy drives the network towards certain types of configurations but not to any one specific configuration.

1. DECOUPLING THE DRIVING FORCES: A NEW APPROACH IN NEURAL SYSTEMS DESIGN

One of the greatest challenges in neural network design is to create a class of neural networks with richer *temporal processing and association properties* than are currently existant. The temporal properties of a network are related to the way in which driving forces are used. Current driving forces are *state specific*. This paper introduces the novel approach of decoupling the driving forces in a neural network into two forces; one which is state-specific, and another which is non-state-specific. The state-specific driving force is encoded, as usual, within the connection weights of the network. The *non-state-specific driving force is a free energy minimization process* which drives the network towards states with certain *configuration characteristics*, rather than to specific instantiations of such states. The interaction between the two forces produces a class of network which has good pattern response capabilities, but which has a wide range of *temporal properties* which have not hitherto been found in any neural network. Most significantly, this architecture promises a route to more cortical-like behavior of the artificial neuron assemblage. This leads to the

name for the prototype of this new class of neural network, the CORTECON: A COntent-Retentive, TEmporally-CONnected network.

Current neural paradigms allow for two possibilities with regard to temporal processing of information. The first is a feedforward architecture in which the "driving force" for the dynamics is implicit in the structure of the network. The second comprises the class of attractor networks, in which the dynamics is driven by minimization of an energy function. In both cases, the resultant state of the network is related to both the input for a given period of operation, and to the network connection weights. In the first case, the network designs typically do not admit stochastic behavior, and temporally-interesting behaviors are obtained only by feedback of the output values back into the network, along with new inputs. In the second case, stochastic behavior can be introduced by noise, but the network typically follows a path initiated by presentation of the input stimulus. In the second case, the *stable or resultant states* are encoded in the connection weights between the neurons. Thus, once the energy-minimizing driving force is engaged, the network state becomes deterministic. (If noise is introduced, the network is quasi-deterministic. However, this does not address the fundamental issue of how the typical energy function governs processing.) Although temporal correlations can be introduced by certain types of connection weights, the temporal processing of these networks is typically rigid.

A new approach to neural network design is to *decouple* the driving force into two distinct driving forces. The interaction of their effects produces the resultant network states. The first driving force is expressed through the connection weights, as is typical. The prototype two-layer CORTECON design allows for both feedforward and lateral connection weights, each of which "drive" the network configuration towards one related to the input pattern. The second force driving the network is a free energy minimization function which is applied to the second, or "computational" layer of the network. This free energy function is different from the energy functions which have typically been used as driving forces in neural network. It drives the network towards spatial configurations of on/off units which have certain characteristics, but which are not specifically encoded as attractor states. This combination of two distinct driving functions engages the network in a complex set of processes.

The use of a *free energy function* as a driving force instead of the usual energy function is novel in neural network design. The free energy function contains an entropy term, which combines additively with the energy term to create the free energy. The next section introduces the Helmholtz free energy with unique entropy terms which operate on the *spatial configuration* of neighboring units. The following section discusses how this unique expression for the free energy contributes to driving the network towards states that exhibit certain *spatial configuration characteristics.*

2. HELMHOLTZ FREE ENERGY WITH SPATIAL CONFIGURATION ENTROPY

The "computational layer" of this new class of neural network can be modeled as large 1-D or 2-D "grids" of bistate processing units. (A 1-D grid has been used for prototyping the CORTECON.) This grid can be treated as an ensemble of bistate units, and Ising statistical mechanics model can be applied. The basic formalism for the Helmholtz Free Energy is

$$A = E - TS, \tag{1}$$

where A is the Helmholtz Free Energy, E is the energy, T is the temperature, and S is the entropy. We can express (1) in reduced form, by dividing through by temperature, Boltzmann's constant (k), and the total number of units in the system (N). (Both the latter terms are involved in the expression for entropy.) This yields

$$\underline{A} = \underline{E} - S/(Nk), \tag{2}$$

where $\underline{A}$ and $\underline{E}$ are the reduced Helmholtz free energy and the reduced energy, respectively.

The equilibrium state of a system (pressure and volume fixed) is defined as the minimum in the Helmholtz free energy. Two processes contribute to this free energy minimization; minimizing the total energy of the system (defined in terms of the energies of individual units and their interactions), and maximizing the entropy of the system. The entropy of a system describes the distribution of its components into the different possible states. Usually the states which are considered for entropy are the energy states of individual units.. An alternative is to consider as different "states" the variety of local spatial configurations of processing units in different states. This can be used to construct an entropy function which drives the system towards an spatial configuration characterized by a distribution of certain types of local patterns. These *micropatterns* are composed of nearest-neighbors and next-nearest-neighbors, which provide respectively three and six distinct types of configurations, as is shown in Figure 1, for configurations composed of units in one of two states, A or B. Certain configurations are different when arranged left-to-right vs. right-to-left. They are treated as instantiations of the same type of configuration, but are doubly weighted using the "redundancy" parameters beta_i and gamma_i respectively, as is shown in Figure 1.

The specific Helmholtz free energy equation which is used as a driving function for this class of neural network is given as [Maren et al., 1992: Kikuchi & Brush, 1967]

$$\begin{aligned} \underline{A} = A/NkT &= 2\beta\epsilon(-z_1 + z_3 + z_4 - z_6) \\ &- 2\sum_{i=1}^{3}\beta_i Lf(y_i) + 2\sum_{i=1}^{6}\gamma_i Lf(z_i) \\ &+ \mu\beta\left(1 - \sum_{i=1}^{6} y_i z_i\right) + 4\lambda(z_3 + z_5 - z_2 - z_4) \end{aligned} \tag{3}$$

where

ϵ is the interaction energy between processing units,
β is the Boltzmann's constant,
y_i and z_i are the cluster variables, as illustrated in Figure 1,
β_i and γ_i are cluster variable coefficients that account for redundancy in the way a

given cluster variable can be measured, and
μ and λ are Lagrange coefficients.

The term Lf(x) is given as

$$Lf(x) = x\ln(x) - x. \quad (4)$$

Configuration	Fraction	α_i
A	x_1	1
B	x_2	1

Configuration	Fraction	β_i
A - A	y_1	1
A - B	y_2	2
B - B	y_3	1

Configuration	Fraction	γ_i
A A \ / A	z_1	1
A B \ / A	z_2	2
A A \ / B	z_3	1
B B \ / A	z_4	1
B A \ / B	z_5	2
B B \ / B	z_6	1

Figure 1: The Fraction Variables from Cluster Variation Theory.

The first term on the RHS of Eq. (3) is the interaction energy, which is negative when neighboring units are in the same state. The next two terms are the entropy, and express the distribution of the ensemble into different types of local spatial configurations. The variables used to describe these relationships are the nearest-neighbor configurations variables, y_i, and the "triples," z_i, both shown in Fig. 1. The remaining two terms are Lagrangian multiplier terms.

3. A NEURAL NETWORK DRIVEN BY FREE ENERGY MINIMIZATION

To use this free energy approach in creating a neural network, we define a two-layer neural network consisting of an *input layer* and a *computational layer*, as illustrated in Figure 2. The input layer functions in the usual manner of accepting inputs and propagating them via weighted sums to the computational layer. The computational layer is composed of processing units which receive inputs and Gaussian noise and which also experience activation decay. The *state* of each processing unit is governed by a function of both its activation (due to the previously mentioned factors) and the overall drive to minimize the free energy. The process of minimizing the free energy can alter a unit"s state. To do this, the absolute value of the unit's activation must be less than a predetermined threshold. Units above threshold value are "fixed" on or off, and stay that way until changes in

input or activation decay cause the activation to become smaller than the (positive or negative) threshold value. Once a domain's activation is smaller than threshold, it becomes labile, and the free energy minimization process can change the state of the domain.

The free energy minimization process is conducted in a manner similar to training a Boltzmann machine. Units in the computational layer are selected at random. If a unit has activation smaller than threshold, it's state is changed, and the free energy for the computational (output) layer is redetermined. If the change results in a lower total free energy, the change is kept. Otherwise, the unit is returned to its previous state. This means that even if a domain is receiving positive activation, and would typically be in an "on" state, the free energy minimization process can turn it off, and vice versa.

Use of Helmholtz free energy minimization for this network is analogous to using a Lyapunov function. The free energy is not a time-dependent function, nor is it a potential energy function in the sense used for most neural network Lyapunov functions. However, it is used in analogy to the free energy minimization process observed in many natural systems. Use of a free energy function of the type described here implies that the network exists at or near an equilibrium state, and that inputs to individual processing units in the network are treated as perturbations on the overall network state. When a perturbing input is received by the network, the network adapts its overall spatial configuration so that it accomodates both the inputs to each processing unit and overall free energy minimization.

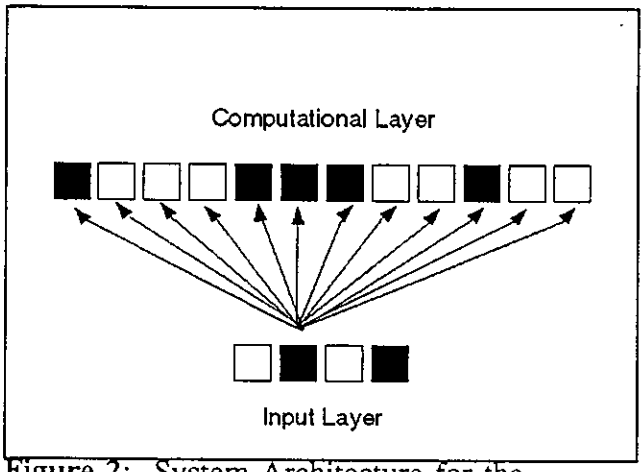

Figure 2: System Architecture for the Computational Engine.

4. RESULTS OF NETWORK OPERATIONS

Early studies with the CORTECON (Maren et al., 1992) confirmed that the free energy minimization process, as carried out via the stepwise process of "flipping" unit states and testing the new free energy, produced results which met theoretical predictions for the free energy of the computational layer. Recent work has focused on identifying the pattern association abilities of the network, and on introducing design features which give the network unique and interesting temporal capabilities. These design features include additive noise in the computational layer units, exponential activation decay of patterns once input stimulus is removed, and use of *interneurons* to strengthen the activation of units in response to a present pattern, or to enhance temporal association with a succeeding pattern.

Our pattern association studies have confirmed that once the input-to-computational layer connection weights have been briefly trained using a variation of Hebbian learning, it is possible to gain recognizable recall of "prototype" output patterns associated with a given

input pattern. Prototype output patterns were identified for each of the randomly established, stored input patterns used in training and testing the network. They were obtained by randomly presenting the different input patterns at least 50 times after network training. The resultant output patterns for each distinct input were stored and averaged. For testing, the inputs were again presented a similar number of times, and the normalized Hamming distance between the resultant output pattern and the corresponding prototype was found. Hamming distances between each of the prototypes (in pairwise manner) was also found. We found that the Hamming distance between the prototypes was typically 3-4 times as large as the Hamming distance between a resultant pattern and its associated prototype. This gives confidence that the unit clustering caused by action of the free energy driving force does not too greatly distort the heteroassociative capabilities of this network.

The combination of free energy (which causes clustering of like units) with learned and sparse lateral connections or *interneurons*, becomes valuable in maintaining output pattern stability during the activation decay which follows when the input pattern is removed. When an input pattern is presented, clusters of like units will develop in the output layer as a result of the free energy minimization. The "core" units of such clusters typically have high activation values, and so are impervious to the random "flipping" of units with less activation. When the input pattern is removed and activation decay begins, the interactions between like nearest neighbors in the clusters help to "persist" the cluster for a longer time than would be so if the free energy minimization process were not present. Further, the interneurons established between the strongest units (whether "on" or "off") are designed to persist the state of the receiving units. This helps them maintain their original state. Interneurons have also been designed to facilitate association and stabilization of temporally-paired patterns.

ACKNOWLEDGEMENTS

This work was supported by a grant from the Thomas F. Jeffress and Kate Miller Jeffress Memorial Trust and by Accurate Automation Corporation IR&D. All network coding and experimentation was performed by Mr. Eyal Schwartz, who also provided significant input into the use of *interneurons* and other aspects of the network design.

REFERENCES

Maren, A.J., E. Schwartz, & J. Seyfried (1992). "Configurational entropy stabilizes pattern formation in a hetero-associative neural network," *Proc. 1992 IEEE Int'l. Conf. on Systems, Man, & Cybernetics* (Chicago, IL; Oct. 18-21, 1992), 89-93.

Kikuchi, R., & Brush, S.G. (1967). "Improvement of the cluster-variation method," *J. Chem. Phys.*, **47**, 195-203.