A TUTORIAL ON THE BOLTZMANN DISTRIBUTION AND ENERGY MINIMIZATION FOR NEURAL NETWORK ENSEMBLES

A. J. Maren

The University of Tennessee
Space Institute
Tullahoma, TN 37388–8897

1.0 INTRODUCTION

Recently, Hopfield [1982, 1984], Kirkpatrick et al., [1983], and Hinton and Sejnowski [1986] have used an analogy between adaptive neural networks and systems of particles in thermodynamic equilibrium to create the "Boltzmann machine" model for learning in neural networks.

One of the major equations describing the behavior of thermodynamic systems is the Boltzmann equation;

$$P_\alpha = \frac{1}{1 + exp(-\Delta E/T)} \ . \qquad [1]$$

This equation has been used by Hopfield and others to model the two-state distribution of neurons in simulated neural nets [Hopfield, 1982 & 1984; Kirkpatrick et al., 1983; and Hinton & Sejnowski, 1986]. Several sources present derivations of this distribution, but cannot be readily understood by anyone who does not have a substantial background in mathematical physics.

This tutorial dervies the Boltzmann distribution from the perspective of statistical thermodynamics, which provides the context in which this function was originated. The emphasis is on providing a complete derivation, and in showing how the results are both intuitively and logically useful for modeling the behavior of other systems, such as ensembles of neurons.

The use of this equation comes about through the analogy between systems of neurons and systems of particles in thermodynamic equilibrium. The points of commonality between the two conceptual systems are:

- Both contain large numbers of units (neurons or particles),
- Both neurons and particles may be in one of a finite number of states, and

- Both systems exhibit interactions among its units; the particles in a thermodynamic system may interact with their neighbors, and neurons interact via synaptic connections.

The analogy has its weak points; e.g. the interactions between nearest-neighbor particles in a thermodynamic system are not very much like the excitatory and inhibatory connections between neurons. Nevertheless, the common use of the Boltzmann machine model of neural networks makes it worthwhile to explore the analogy further, and to present a readily-understood deviation of the Boltzmann distribution equation.

2.0 DERIVATION OF THE BOLTZMANN DISTRIBUTION

We first consider of a fundamental concept in statistical thermodynamics; the *Gibbs free energy* (G) of a system. This energy is characterized by the relationship

$$G = H - TS, \qquad [2]$$

where H is the enthalpy, S is the entropy, and T is the temperature. Enthalpy is a function of pressure and the total internal energy of each of the particles inthe system. Entropy is a measure of the randomness or disorder in the system.

We can think of the system as being composed of a large number of units or particles, each of which can assume any energy ε from a set of possible energy states. The probability with which a unit will be in state j is P_j. The sum of all probabilities P_j is equal to one;

$$\sum_{j=1}^{N} P_j = 1, \qquad [3]$$

where N is the total number of states available.

The values for enthalpy and entropy each depend on the probabilistic distribution of units among the energy states. For enthalpy, we have

$$H = \sum_{j=1}^{N} \varepsilon_j P_j + PV, \qquad [4]$$

where P is pressure, V is volume, and ε_j is the energy associated with state j.

The entropy is defined as

$$S = -k \sum_{j=1}^{N} P_j \; lnP_j. \qquad [5]$$

Thus, inserting Eqns. [4] and [5] into [2], we obtain

$$G = G(P_j) = \sum_j \varepsilon_j P_j + PV + kT \sum_j P_j \; lnP_j \qquad [6]$$

where k is a known constant.

A system has reached equilibrium when no further spontaneous processes are possible. At equilibrium. the free energy G is at a minimum. (This is the defining condition for equilibrium.) The "minimum free energy" can be found by taking the derivative of G

and setting it equal to zero. (This defines either a local minimum or maximum. We can examine the shape of the curves for G to determine which it will be.)

Thus, at equilibrium,

$$\frac{\partial G}{\partial P_j} = 0. \qquad [7]$$

Changes in G are dependent on many variables; the pressure and volume changes in a system, changes in temperature, and changes in the distribution of the particles among the different energy states. We can assume that the pressure and volume of the system are kept constant. This assumption does not affect the relevance of our model.

This leaves for consideration how G, is affected by two factors: temperature (T) and the distribution of the different units in the system among the possible energy states (P_j). Holding T constant and taking the derivative of G with respect to P_j allows us to see how G explicitly depends on P_j. (This is mathematially allowed since T does not depend on P_j.) However the distribution of units among different energy states does depend upon temperature. We will examine this relationship later.

The derivative of G with respect to P_j is expressed as follows: (Note that we use the symbol "∂" instead of the more common "d". This is to remind us that we are taking a <u>partial</u> derivative; the derivative of G <u>with respect to</u> P_j, while keeping other variables (e.g., temperature) constant.

$$\begin{aligned}\frac{\partial G}{\partial P_j} &= \frac{\partial}{\partial P_j}\{\sum_j \varepsilon_j P_j + PV + kT\sum_j P_j \; lnP_j\} \\ &= \frac{\partial}{\partial P_j}\sum_j \varepsilon_j P_j + kT\frac{\partial}{\partial P_j}\sum_j P_j \; lnP_j\end{aligned} \qquad [8]$$

This yields (for details, see the appendix):

$$\frac{\partial G}{\partial P_j} = \varepsilon_j + kT[1 + \; lnP_j]. \qquad [9]$$

For equilibrium (Eqn. 7),

$$0 = \varepsilon_j + kT[1 + \; ln \; P_j]. \qquad [10]$$

Rearranging terms gives

$$ln\ P_j = -\left[1 + \frac{\varepsilon_j}{kT}\right] \qquad [11]$$

Taking the exponent of both sides (details in appendix) yields:

$$P_j = \frac{1}{e} \exp\left(\frac{-\varepsilon_j}{kT}\right) \qquad [12]$$

Physically, this means that the probability with which a unit will be in state j depends on the energy of that state (ε_j), and the temperature (T). We can examine the dependence of P_j on each of these two variables. Note, first, if the energy of a state j is low ($\varepsilon_j \approx 0$), then the quantity -$\varepsilon_j/(kT)$ is also close to zero. Then,

$$\exp(\frac{-\varepsilon_j}{kT}) \approx \exp(0) = 1 \qquad [13]$$

This means that the probability that a unit will be in state j is high ($\approx$ 1) if the energy of that state is low ($\varepsilon_j \approx 0$). (Equation [12] was achieved without normalizing values for P_j; thus we are considering only the influence of ε_j and T on the term $\exp(-\varepsilon_j/hT)$.)

If ε_j is high, then ε_j/(kT) is also increased. We are considering the quantity exp ($-\varepsilon_j$ / kT), and the exponent of a large negative number is very small ($\exp(-\infty) \Rightarrow 0$). Thus, if we have a state where the energy is high, the probability that a unit will be in that state is small.

Although our deviation depends on T being constant, the temperature influences the probability distribution among states. ε_j/(kT) can be large, and hence exp ($-\varepsilon_j$/kT) is small. Thus, at low temperatures, the whole probability distribution shifts so that more units are in a low energy state. Likewise for high temperatures, ε_j/(kT) is small, and thus more units can slip into a high energy state.

Suppose there are only two possible energy states in a system, state α and state β. Then

$$P_\alpha + P_\beta = 1, \quad \text{or} \quad P_\alpha = 1 - P_\beta. \qquad [14]$$

For any two energy states, α and β, we can obtain the ratio of the probability with which units will be distributed in those two states:

$$\frac{P_\alpha}{P_\beta} = \frac{\exp\left(\frac{-\varepsilon_\alpha}{kT}\right)}{\exp\left(\frac{-\varepsilon_\beta}{kT}\right)} = \exp[-(E_\alpha - E_\beta)/T] \qquad [15]$$

where

$$E_\alpha = \varepsilon_\alpha/k \quad ; \quad E_\beta = \varepsilon_\beta/k. \qquad [16]$$

For simplicity, let the fraction of units in state α be n. Then the number of units in state β is $(1-n)$. We can rewrite Eqn. [6] to explicitly reflect the composition of a two-state system. (We will neglect pressure and volume considerations, as they do not affect this model.)

$$G = \varepsilon_\alpha n + \varepsilon_\beta(1-n) + kT\{n \; ln \; n + (1-n) \; ln \; (1-n)\}. \qquad [17]$$

Let us rewrite the enthalpy in terms of $\Delta\varepsilon$, where $\Delta\varepsilon = \varepsilon_\beta - \varepsilon_\alpha$:

$$\begin{aligned} G &= -(\varepsilon_\beta - \varepsilon_\alpha)n + \varepsilon_\beta + kT[n \; ln \; n + (1-n) \; ln \; (1-n)] \\ &= \varepsilon_\beta - n\Delta\varepsilon + kT[n \; ln \; n + (1-n) \; ln \; (1-n)] \end{aligned} \qquad [18].$$

Note now that the enthalpy term $(\varepsilon_\beta + n\Delta\varepsilon n)$ depends on n only through $\Delta\varepsilon$, the difference between the energy levels of the two states.

Before moving on to calculate the derivative of G (and hence distribution among the two states which would minimize G), let us examine the expression for G graphically. Figure 1 (a & b) shows H and TS, (the two components of the free energy G) at low and high temperatures. H depends on n linearly. S is a more complex function of n, but is symmetric about the point $n = .5$. As T increases, the TS curve will increase proportionately. (The enthalpy H may also depend on T, but not to the same extent.)

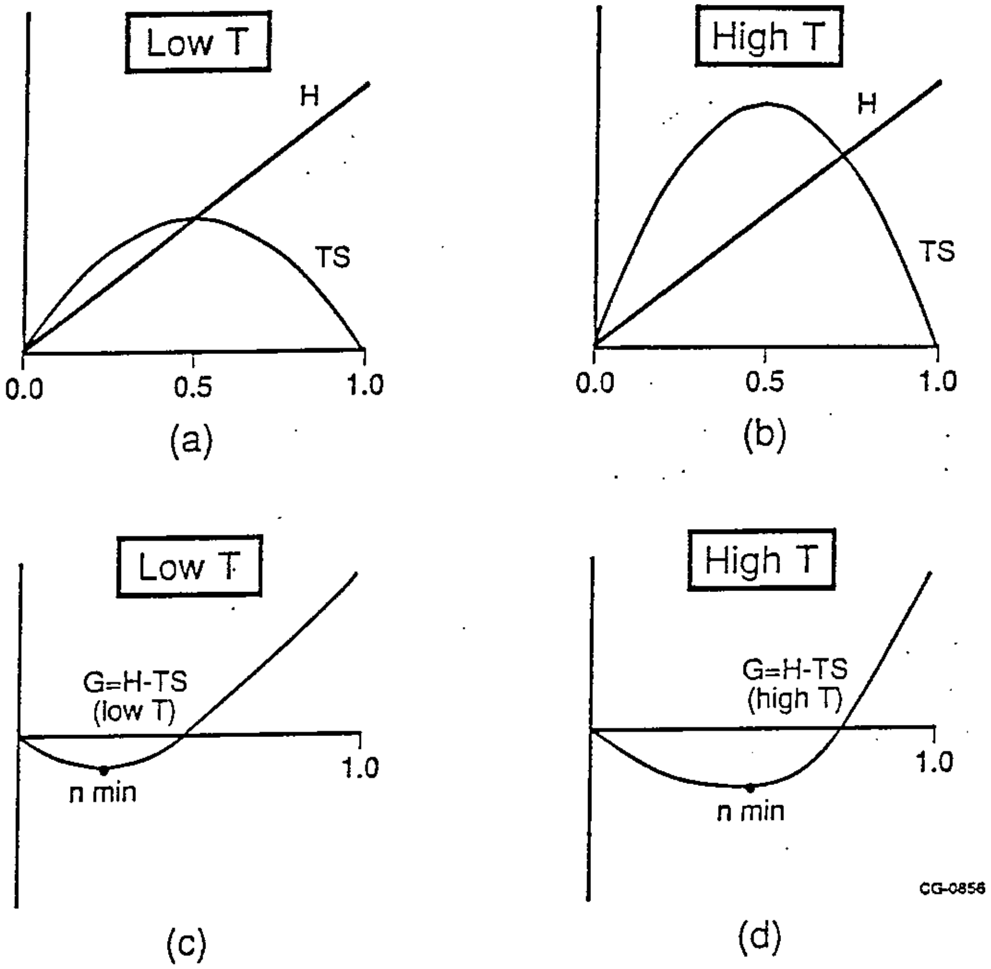

Figure 1. (a & b). Low and high temperature curves for H and TS.
(c & d). Low and high temperature curves for $G = H - TS$.
As T increases, the value for n which minimizes G also increases.

Figure 1 (c & d) shows the curve for G (or, $H - TS$). As T increases, the value for n which minimizes T also increases. Thus, at high values of T, we expect proportionately more of the units in the system to be in the high energy state (state α).

We can take the derivative of G as before, this time with respect to n.

$$\begin{aligned}\frac{\partial G}{\partial n} &= \frac{\partial}{\partial n}\{\varepsilon_\beta - n\Delta\varepsilon + kT[n\ ln\ n + (1-n)\ ln\ (1-n)]\} \\ &= -\Delta\varepsilon + kT\frac{\partial}{\partial n}[n\ ln\ n + (1-n)\ ln\ (1-n)] \\ &= -\Delta\varepsilon + kT[\ ln\ n + 1 - \ ln\ (1-n) - 1] \\ &= -\Delta\varepsilon + kT[\ ln\ n - \ ln\ (1-n)] \qquad [19]\end{aligned}$$

At equilibrium $\partial G/\partial n = 0$, so that

$$\Delta\varepsilon = kT[\ ln\ n - \ ln\ (1-n)]. \qquad [20]$$

This yields

$$ln\ n - \ ln\ (1-n) = \ ln\ \left(\frac{n}{1-n}\right) = \frac{\Delta\varepsilon}{kT}. \qquad [21]$$

Substituting ΔE for $\Delta\varepsilon/k$, we have

$$ln\ \left(\frac{n}{1-n}\right) = \frac{\Delta E}{T} \qquad [22]$$

Taking the exponent of both sides, we have

$$\frac{n}{1-n} = \exp(\Delta E/T) \qquad [23]$$

Solving for n yields

$$\frac{1}{1/n - 1} = \exp(\Delta E/T),$$

or

$$1/n - 1 = \exp(-\Delta E/T),$$

or

$$1/n = 1 + \exp(-\Delta E/T),$$

which yields

$$n = P_\alpha = \frac{1}{1+\exp(-\Delta E/T)}. \qquad [24]$$

The physical meaning of this equation can be interpreted as follows: Suppose ΔE is large ($\varepsilon_\beta \gg \varepsilon_\alpha$, or state β is in a much higher energy than state α). Referring to our previous arguments, we would expect almost all units to be in state α, or that $\eta \approx 1$. This is borne out by equation [24]. If ΔE is large, then exp $(-\Delta E/T)$ is very small, due to taking the exponent of a large negative number. For reference, Figure 2 shows the exponential curve.

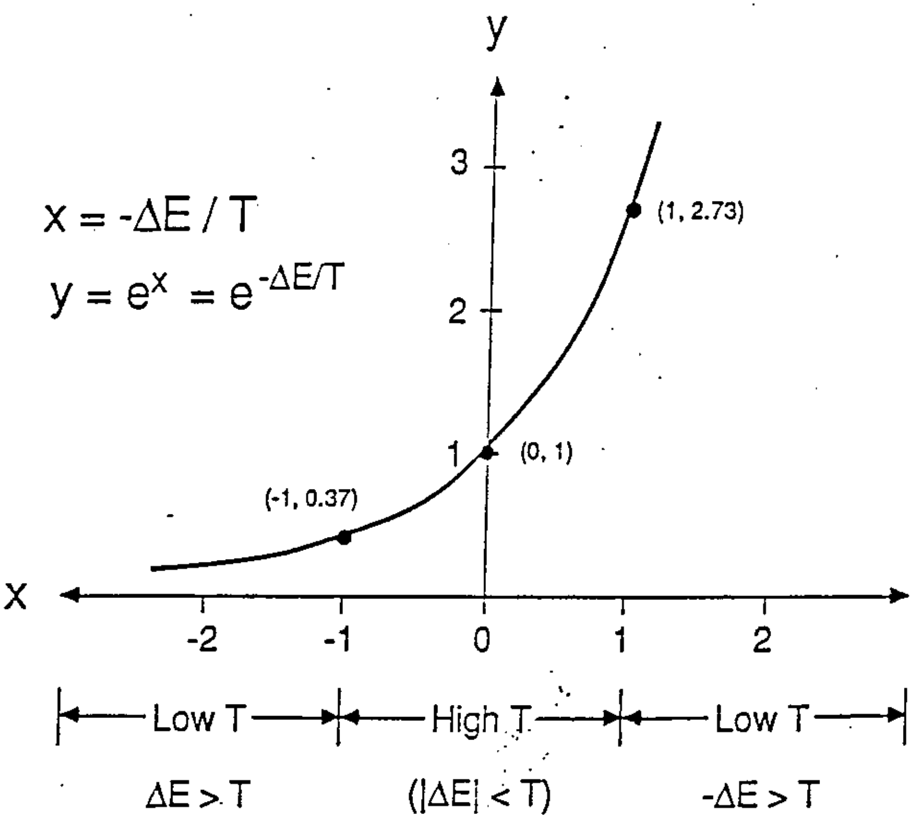

Figure 2. Graph of the exponential function.

With exp $(-\Delta E/T)$ close to zero,

$$\eta = \frac{1}{1 + exp(-\Delta E/T)} \approx \frac{1}{1 + \text{(negligible quantity)}} \approx 1, \qquad [25]$$

or the probability that units will be in state α is close to 1, as expected.

Equation [25] is graphed in Figure 2 as a function of T.

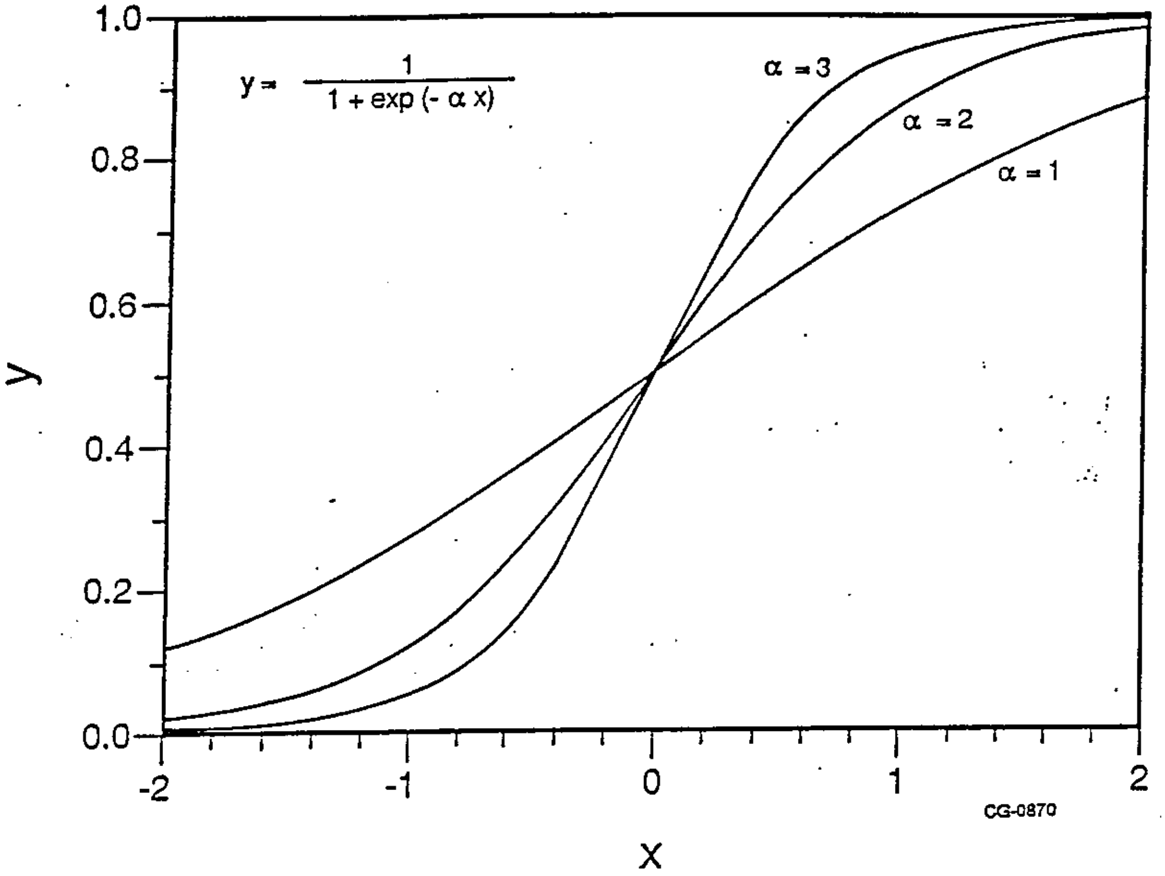

Figure 3. Graph of Equation [25], $y = 1/(1 - exp(-\delta E/T))$.

3.0 APPLICATION OF FREE ENERGY METHODS AND THE BOLTZMANN DISTRIBUTION TO NEURAL NETWORKS

The key feature of the statistical thermodynamics methods discussed in the previous section is that they model the most probable behavior of a system in thermal equilibrium. More advanced treatments of this subject yield descriptions of the system in terms of the average behavior of units in the system, and in terms of small fluctuations around the average. In this context, we see that there could be some analogy between units in thermal equilibrium and interconnected neurons. In the type of therma system just described, there were two states (α and β). To carry the analogy further, each neuron may be in one of two states; above-threshold ('activated,' or 'firing'), and below threshold ('non-activated').

If we want to express the analogy mathematically, then we need to show that the neural network system may be described by a model similar to that used for thermal systems in the previous section. Hopfield developed this analogy in a landmark paper on neural networks. He defined the input to each neuron as

$$\text{Input}_i = \sum_j T_{ij} V_j \ , \qquad [26]$$

where Input$_i$ is the input to the i^{th} neuron, V_j is the 'state' of each neuron (1 or 0), and T_{ij} is the interaction between neurons, represented as the "effectiveness of a synapse:

For computational simplicity, he further defined a simple function for T_{ij}, and set $T_{ij} = T_{ji}$. He defined the total enthalpy of the system as:

$$E = H = \frac{-1}{2} \sum_{i \neq j} \sum_j T_{ij} V_i V_j \qquad [27]$$

Hopfield used the term 'energy' for 'enthalpy', which is acceptable. However, his term 'energy' does not mean the same thing as 'free energy' which requires consideration of both enthalpy (or energy) and entropy.[1]

[1] Hopfield did his initial derivations using an information-theoretic approach, which is formally isomorphic to the statistical mechanical method, and which yields the same results.

Hopfield gave this change in energy (ΔE) which would be produced by a single unit changing state (ΔV_i) as:

$$\Delta E = -\Delta V_i \sum_{j \neq i} T_{ij} V_j \ , \qquad [28]$$

Note that the use of a minus sign indicates that energy is increased if $\sum_{j \neq i} T_{ij} V_j$ is positive, and we go from $V_i = 0$ to $V_i = 1$ ($\Delta V_i = 0 - 1 = -1$), but decrease energy if we go from $V_i = 1$ to $V_i = 0$.

Hopfield also used an entropic measure similar to that used in the previous section, where

$$S = - \sum_{j} P_j \ ln \ P_j \ , \qquad [29]$$

This allows us to form an expression for the free energy of the "neural system" which considers the change in free energy which would be produced by having the different neurons change their "states".

$$G = \sum_{j} \Delta E P_j + T \sum_{j} P_j \ ln \ P_j . \qquad [30]$$

The form of this equation is similar to that of Eqn [6], and so solutions for the minimum free energy are also similar. Thus, the previous results are covered, and we again have

$$P_\alpha = \frac{1}{1 + \exp(-\Delta E / T)} \ . \qquad [24]$$

4.0 THE SIMULATED ANNEALING METHOD OF NEURAL NETWORK WEIGHT OPTIMIZATION

Although Hopfield [1982] identified both the enthalpic and entropic terms described in the previous section, he did not go so far as to connect them in the form of a free energy equation, and derive the results of eqns [30] and [24]. This was because the approach he used did not identify a need for the "temperature" term.

The combination of these terms into a free energy equation and derivation of the probability distribution equation [Eqn. 24] was actually done by Kirkpatrick et al. [1983]. Kirkpatrick and his colleagues carried the analogy between systems of neurons and systems of units in thermal equilibrium even further, by addressing the problem of learning appropriate weights for neural network systems.

In the Boltzmann machine approach to neural networks developed by Hinton (as an extension from the Hopfield method), there are three types of neurons; input, hidden. and output. These neurons are fully connected, and the weights between the neurons determine the output of the system given an initial input vector.

In order to approach the most effective weight assignments, Kirkpatrick and his colleagues developed the simulated annealing method. Kirkpatrick's approach was to identify yet another analogy between types of systems or types of problems:

> "Finding the low-termperature state of a systems when a prescription for calculating its energy is given is an optimization problem not unlike those encountered in combinatorial optimization. However, the concept of the temperature of a physical system has no obvious equivalent in the system being optimized. We will introduce an effective temperature for optimization, and show how one can carry out a simulated annealing process in order to obtain better heuristic solutions to combinatorial optimization problems.
>
> I~~n~~terative improvement, commonly applied to such problems, is much like the microscopic rearrangement processes modeled by statistical mechanics, with the cost function playing the role of energy."

[Kirkpatrick et al., 1983]

To implement this approach, Kirkpatrick adapted an algorithm developed by Metropolis et al. [1953]. This algorithm provided an efficient simulation of the thermal system (a collection of atoms in equilibrium at a given temperature), and iteratively approached a solution. As described by Kirkpatrick et al. [1983]:

> "Using the cost function in place of the energy and defining configurations by a set of parameters x_i, it is straightforward with the Metropolis procedure to generate a population of configurations of a given optimization problem at some effective temperature. This temperature is simply a control parameter in the same units as the cost funtion. The simulated annealing process consists of first "melting" the system being optimized at a high effective temperature, then lowering the temperature by slow stages until the system "freezes" and no further changes occur. At each temperature, the simulation must proceed long enough for the system to reach a steady state. The sequence of temperatures and the number of rearrangements of the x_i attempted to reach equilibrium at each temperature can be considered an annealing schedule."

The simulated annealing method was first applied by Kirkpatrick et al. [1983] to determine the partitioning and wiring of microprocessor chips. Since then, the simulated annealing method has been adopted as a useful method for learning connection weights in a variety Boltzmann machine applications [E.g. Hinton & Sejnowski, 1986; Sejnowski, 1986].

5.0 ASSESSMENT OF THE BOLTZMANN MACHINE MODEL

The assumption which leads to the comparision of neural networks and a statistical thermodynamic/ model is that of the isomorphism between a collection of neurons and a collection of thermal units. There are several views regarding this comparison.

The most important questions is, "Does it work?" In other words, does treating a collection of neurons as an analog to a thermal system yield a better understanding of or a useful model for a neural network? If so, the answer may be a definate 'yes'. Using an analogy to a thermodynamic system has led to the development of the simulated annealing method for connection weight adaptation, and this has had useful consequences.

At a slightly deeper level, we might ask how far the analogy extends. In a follow-up paper, Hopfield [1984] showed that systems of neuron-like elements which had a graded response (range of activity states between 0 and 1) could be computationally described by the same methods as those used for a collection of binary-state neurons.

Nevertheless, this approach has come under sharp criticism from other researchers in the arena of neural networks. For example, Kohonen [1987] states:

> "At least one should realize that there are certain views frequently held of neural networks in modelling which are completely untenable:
>
> - The states of the neurons are not binary, and thus not describable, e.g., by the spin formalism. Individual neurons are not bistable latches; there is no evidence for them memorizing the active or passive state. This misconception may have resulted from the observation that the *neural impulses* have a constant amplitude ("all-or-none")
> - There is even very little evidence for neurons operating as threshold-logic units. Those thresholds which are encountered in experimental stimulus-response relations are more probably due to *collective feedback effects*, similar to the familiar Schmitt-trigger action in basic electronics, although there is no threshold in the components which make up the Schmitt-trigger."

This point of view may be temporized by realizing that the model proposed by Hopfield and generalized and adapted by Hinton, Sejnowski, Kirkpatrick, and others need not be considered a direct attempt to emulate a system of biological neurons. Instead, by viewing their work as suggestive of a computational methodology, we can revert to our original question of "Does it work?."

~~Nevertheless,~~ there are two important areas to consider for further development and applications. One takes us further into the realm defined by the initial analog between neural networks and thermodynamic systems, and the other takes us further away from that realm.

Moving deeper into the analogy, we find that there has been a recent interest in the ways in which known models can be used to describe the anomalistic behaviors of large scale systems [Huberman & Hogg, 1987, Shraaer, Hogg, & Hukeman, 1987], Specifically the concepts of phase transitions, metastable states, and self organizing systems, *as understood from the perspective of statistical thermodynamics and related fields*, may be even further useful in describing both the behaviors of biological neural and computational neural-like systems. This may be even more true than is currently appreciated, given that biological systems may *as a whole* undergo state changes, inhabit metastabilities, or exhibit complex periodic behaviors.

The types of equations considered in this tutorial are a simplified form of those which caan describe the more complex mentioned above. Inclusion of higher-order interaction terms, or considerations of the behaviors of ensembles of domains (where each domain of units or neurons inhabits a given state) allows this type of method to describe phase transitions between stable and metastable states.

In the other direction, moving *away* from the analogy between statistical thermodynamics and neural systems, we find that research in the more complex neural systems is moving towards modeling their time-dependent behaviors. Work by Kohonen [1987, and references therein] and Carpenter, Cohen, and Grossberg [1987, and references contained therein] exemplify the leading-edge research in modeling neural nets. In general, the time-dependent behaviors of then neural-network system which they study may be modeled in terms of Liapunov functions, which are a class of continuous, and monotonically- decreasing functions [Halanay, 1971]. Selection of these functions, and application to appropriate classes of neural nets, is a challenging research problem today.

ACKNOWLEDGEMENTS

I would like to acknowledge Rosie Dupless, whose suggestions and comments helped turn this tutorial from a rough draft into a finished product. She also provided the graphs used in Figure 3. I would also like to thank Lisa Blanks, who did the graphics, and Becky Stines, Sue Crawford, and Melissa Weatherford, who typed and set the manuscript. Questions raised by Dr. Moonis Ali about mathematical models of neural nets, as well as questions asked by students in my Pattern Recognition course (Tim Choate, Rossie Duless, and Veronica Minsky) provided the inspiration and incentive to write this tutorial. I would also like to thank Drs. Sheng H. Lin & Dr. Edward J. O'Reilly for their reviews of this manuscript.

REFERENCES

Carpenter, G. A., Cohen, M. A., & Grossberg, S. "Computing with neural nets," *Science*, 235 (1987), 1226–1227.

Halanay, A. "For and against the Liupunov function," in *Symposia Mathematica*, V1 (1971). (London: Academic Press), 167–175.

Hill, T. L. *An Introduction to Statistical Thermodynamics*, (Reading, MA: Addison-Wesley, 1960).

Hinton, G. E. & Sejnowski, T. J. "Learning and relearning in Boltzmann machines." In D. E. Rumelhart and J. L. McClelland (Eds.), *Parallel Distributed Processing, Vol. 1* (1986) (Cambridge, MA: MIT Press).

Hopfield, J. J. "Neural networks and physical systems with emergent collective computational abilities," *Prof. of the Nat'l Acad. Sciences, USA*, 79 (1982), 2554-2558.

Hopfield, J. J. "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. of the Nat'l. Acad. Sciences, USA*, 81 (1984), 3088–3092.

Kirkpatrick, S., Gellatt, C. D., & Veechi, M. D., "Optimization by simulated annealing," *Science*, 220 (1983), 671-680.

Kohonen, T. "Self-organizing maps," Tutorial presented at the *IEEE First Annual International Conference on Neural Networks* (San Diego, CA., June 21–24, 1987).

Metropolis, N., Rosenbluth, A., Rosenbluth, M. N., & Teller, A. "Equation of state calculations by fast computing machines" *J. Chem. Phys.*, 21 (1953), 1087–1092.

Sejnowski, T. "Learning symmetry groups with hidden units: beyond the perception," *Physica 22* D. (1986), 260–275.

Shrager, J., Hogg, T., & Huberman, B. A. "Observation of phase transitions in spreading activation netwoks," *Science*, 236 (1987), 1092–1093.

APPENDIX: MATHEMATICAL FORMULAS

The following relationships may be useful in obtaining some of the results given in the paper:

Logarithmic/Exponential Relationships:

1. $ln\ a + \ ln\ b = \ ln\ (ab);\ ln\ a - \ ln\ b = \ ln\ \left(\frac{a}{b}\right)$
2. $\exp(a+b) = \exp(a)\exp(b).$
3. $ln\ (1) = 0;\ \exp(1) = e,\ \exp(-1) = 1/e$
4. $\exp(\ ln\ (a)) = \ ln\ (\exp(a)) = a.$

Relationships from Differential Calculus:

5. $\frac{d}{dx}[f(x)g(x)] = f(x)\frac{dg(x)}{dx} + g(x)\frac{df(x)}{dx}$
6. $\frac{d}{dx}f[g(x)] = \frac{df(g(x))}{dg(x)}\frac{dg(x)}{d(x)}$
7. $\frac{d}{dx}\exp(x) = \exp(x);\ \frac{d}{dx}\ ln\ (x) = \frac{1}{x}$
8. $\frac{d}{dx}x^k = kx^{k-1}$

Applied to Eqn [9] these relationships give:

$$\frac{d}{dP_j}[P_j\ ln\ (P_j)] = P_j\frac{d}{dP_j}[\ ln\ (P_j)] + \ ln\ P_j\frac{d}{dP_j}[P_j]$$

$$= P_j(\frac{1}{P_j}) + \ ln\ P_j$$

$$= 1 + \ ln\ P_j$$