ANALYZING COOPERATIVE COMPUTATION

Geoffrey E. Hinton
Computer Science Department
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

Terrence J. Sejnowski
Biophysics Department
The Johns Hopkins University

ABSTRACT

Making a perceptual interpretation can be viewed as a computational process in which a plausible combination is chosen from among a large set of interdependent hypotheses. In a cooperative computation the hypotheses are implemented by units that interact non-linearly and in parallel via excitatory and inhibitory links (Julesz, 1971; Marr & Poggio, 1976; Sejnowski, 1976). A particular perceptual task is specified by external inputs to some of the units and the whole system must then discover a stable state of activity in which the active units represent the hypotheses that are taken as true. We describe a search procedure based on statistical mechanics that finds near optimal combinations of hypotheses with high probability, and we show that the hardware units required for its efficient implementation are similar to neurons. Even though the individual units are non-linear, there is a linear relationship between the synaptic weights and the logarithms of the probabilities of global states into which the system settles. This makes it possible to implement a convergent learning procedure which specifies just how the synaptic weights need to be changed in order to learn the constraints in a given domain.

Introduction

Consider the problem of making a 3-D interpretation of a 2-D line drawing. Each line in the picture, considered in isolation, could depict any one of a large set of 3-D edges. People resolve this local ambiguity by using assumptions about the ways in which edges go together in the 3-D world. These assumptions make some combinations of edges far more plausible than others. There are two roughly separable problems in understanding the use of assumptions in perception. The first is to specify clearly what the assumptions are, and the second is to find a search procedure that can discover interpretations which optimally fit the input data and the assumptions, even when some of the assumptions conflict with one another (Attneave 1982). Our concern here is with the second problem: How can we discover interpretations that optimally fit a large set of plausible assumptions?

Attneave (1982) and others (Hinton 1977) have proposed cooperative models in which neuron-like hardware units represent particular 3-D edges and the rules are implemented by excitatory and inhibitory interactions between these units. Each line in the drawing provides input to the whole set of 3-D edges which are consistent with it, and under the influence of this input the whole system settles into a stable state of activity which represents the interpretation. It is not obvious that such a search process can be made to work. The apparent difficulty of analyzing the behaviour of cross-coupled, non-linear systems makes it tempting to believe that the *only* way to make progress is through computer simulation. In this paper we attempt to show that mathematical analysis is possible and illuminating.

Most of the existing proposals for cooperative search mechanisms assume that there are real-valued activity levels which change smoothly during the search (Rosenfeld, Hummel & Zucker, 1976). These activity levels are often associated with the firing rates of neurons, and they are normally used to represent the value of a physical parameter such as slope in depth, or the current probability that a hypothesis is correct. The method we shall describe uses a very different representation. The units that stand for hypotheses only have two states, true and false. However, the decision rule which determines which state they enter is probabilistic, so they can change their state even if they are receiving constant input. The use of a probabilistic decision rule makes the cooperative search *easier* to analyze than with a deterministic rule because it makes it possible to apply methods from statistical mechanics. Instead of being a drawback, the non-determinism has the advantage of allowing the system to escape from sub-optimal states. We start by describing a system in which there is a deterministic decision rule that is applied at random moments and then we generalize this case to a non-deterministic rule.

Cooperative search with deterministic binary units

Hopfield (1982) postulates a system with a large number of binary units. The units are *reciprocally* connected, with the

strength of the connection being the same in both directions. Given the current inputs from outside the system, any particular state of the system has an associated "energy" and the whole system behaves in such a way as to minimize its energy. The energy of a state can be interpreted as the extent to which it violates a set of plausible constraints, so in minimizing its energy it is maximizing the extent to which it satisfies the constraints.

The total energy of the system is defined as

$$E = -1/2 \sum_{ij} w_{ij} s_i s_j - \sum_i (\eta_i - \theta_i) s_i \qquad (1)$$

where η_i is the external input to the i^{th} unit, w_{ij} is the strength of connection (synaptic weight) from the j^{th} to the i^{th} unit, s_i is a boolean truth value (0 or 1), and θ_i is a threshold.

A simple algorithm for finding a combination of truth values that is a *local* minimum is to switch each hypothesis into whichever of its two states yields the lower total energy given the current states of the other hypotheses. If hardware units make their decisions asynchronously, and if transmission times are negligible, then the system always settles into a local energy minimum. Because the connections are symmetrical, the difference between the energy of the whole system with the k^{th} hypothesis false and its energy with the k^{th} hypothesis true can be determined locally by the k^{th} unit (Hopfield, 1982), and is just

$$\Delta E_K = \sum_i (w_{ki} s_i) + \eta_k - \theta_k \qquad (2)$$

Therefore, the rule for minimizing the energy contributed by a unit is to adopt the true state if its total input exceeds its threshold, which is the familiar rule for binary threshold units (Minsky & Papert, 1968).

Using probabilistic decisions to escape from local minima

The deterministic algorithm suffers from the standard weakness of gradient descent methods: It gets stuck at *local* minima that are not globally optimal. This is an inevitable consequence of only allowing jumps to states of lower energy. If, however, jumps to higher energy states occasionally occur, it is possible to break out of local minima. An algorithm with this property was introduced by Metropolis *et. al.* (1953) to study average properties of thermodynamic systems (Binder, 1978) and has recently been applied to problems of constraint satisfaction (Kirkpatrick, Gelatt & Vecci, in press). We adopt a form of the Metropolis algorithm that is suitable for parallel computation: If the energy gap between the true and false states of the k^{th} unit is ΔE_k then regardless of the previous state set $s_k = 1$ with probability

$$p_k = \frac{1}{(1 + e^{-\Delta E_k / T})} \qquad (3)$$

where T is a parameter which acts like temperature (see fig. 1). This parallel algorithm ensures that in thermal equilibrium the relative probability of two global states is determined solely by their energy difference, and follows a Boltzmann distribution.

$$\frac{P_\alpha}{P_\beta} = e^{-(E_\alpha - E_\beta)/T} \qquad (4)$$

At low temperatures there is a strong bias in favor of states with low energy, but the time required to reach equilibrium may be long. At higher temperatures the bias is not so favorable but equilibrium is reached faster.

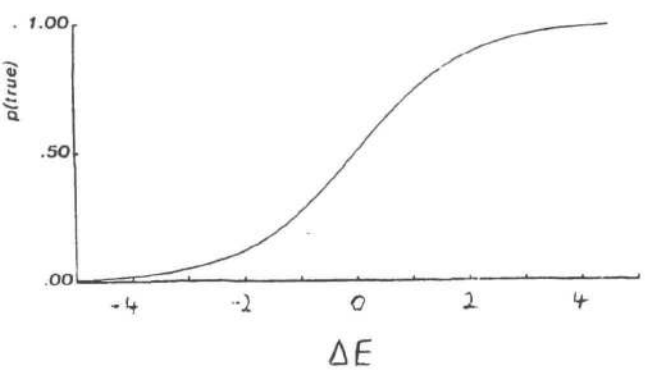

Figure 1

Probability $p(\Delta E)$ that a unit is in its "true" state as a function of its energy gap ΔE plotted for $T = 1$ (Eq. 3). As the temperature is lowered to zero the sigmoid approaches a step function.

Reducing the time to reach equilibrium

One technique that can be used to reach a good equilibrium distribution quickly is to start at a high temperature and then to cool down (Kirkpatrick *et. al.*, in press). This type of search by "simulated annealing" initially finds a large-scale minimum but fluctuates around it because of the high temperature. As the temperature is reduced, a good minimum will be found within the large-scale minimum, and so on. In general, it is impossible to *guarantee* that a global minimum will be found, but a nearly global minimum can be found with high probability.

We are investigating an additional technique which we shall only mention here. Energy barriers are what prevent a system from reaching equilibrium rapidly at low temperature, and if they can be temporarily suppressed, equilibrium can be achieved rapidly at a temperature at which the distribution strongly favors the lower minima. The energy barriers cannot be permanently removed, because they correspond to states that violate the constraints, and the energies of these states must be kept high to prevent the system from settling into them. However, for special cases it is possible to design units which are active during the search process but are quiescent in the final state. When one of these special units is active it lowers the energy of a state that would have been an energy barrier between two local minima. The special units are a way of implementing heuristic knowledge about how to search the space. They have no effect on the energies of final states, and in this respect they are like catalysts.

Learning

So far, we have assumed that the interactions between the units implement the correct constraints, and we have focussed on the search problem. However, in a system where the weights represent many plausible assumptions that interact, it is not obvious how to choose the weights to produce the desired behavior. We will show that, as a consequence of the probabilistic decision rule, it is possible for a cooperative module to internalize the constraints in any domain simply by being told whether the solutions it settles into are right or wrong. When the module settles to the wrong solution, it modifies the weights so as to raise the energy of that state and thus make it less likely to be found in future. Similarly, good solutions that are not found often enough have their energies lowered when they are found. This simple procedure is effective because of the *linear* relationship between the synaptic weights and the logs of probabilities of whole states at thermal equilibrium. If we temporarily ignore the thresholds and the external inputs to the units and assume a temperature of 1, we have:

$$ln(\frac{P_\alpha}{P_\beta}) = -(E_\alpha - E_\beta)$$

$$= \sum_{ij} w_{ij} h_{ij}^{\alpha\beta} \qquad (5)$$

where

$$h_{ij}^{\alpha\beta} = s_i^\alpha s_j^\alpha - s_i^\beta s_j^\beta$$

and s_i^α is the state of the i^{th} unit in the α^{th} global state.

To explain the learning procedure, we invent a hypothetical ideal system which settles into global states with exactly the probabilities required. We then show that if the actual system is told whether its current probabilities for particular states are too high or too low, it can modify its weights so that they more closely resemble the weights in the hypothetical ideal system.

Suppose that under the influence of a constant external input vector, the actual system settles into two different states, S_α, S_β with probability ratio P_α / P_β. Suppose that the probability ratio demanded by the evaluation function (and achieved by the ideal system) is P'_α / P'_β which is higher. The actual system can increase its probability ratio by increasing the energy difference, $E_\beta - E_\alpha$. This can be done by adding δ to each weight between a pair of active units in S_α and subtracting δ from each weight between a pair of active units in S_β. The net change in a weight is then $\delta . h_{ij}^{\alpha\beta}$. We now prove that, provided δ is sufficiently small, each application of this learning procedure is guaranteed to reduce the Euclidean distance, D, between the current set of weights, w_{ij}, and the ideal ones, w'_{ij}. Assume that the actual and ideal systems have the same external inputs and thresholds, and that T = 1. If the error, r, in the probability ratio achieved by the actual system is

$$r = ln(\frac{P'_\alpha}{P'_\beta}) - ln(\frac{P_\alpha}{P_\beta})$$

then from equations 4 and 5, we have:

$$r = -(E'_\alpha - E'_\beta) + (E_\alpha - E_\beta)$$

$$= \sum_{ij} h_{ij}^{\alpha\beta} w'_{ij} - \sum_{ij} h_{ij}^{\alpha\beta} w_{ij}$$

Before applying the learning rule we have

$$D^2_{before} = \sum_{ij} (w_{ij} - w'_{ij})^2$$

and afterwards

$$D^2_{after} = \sum_{ij} (w_{ij} + \delta . h_{ij}^{\alpha\beta} - w'_{ij})^2$$

$$= D^2_{before} - \delta \sum_{ij} (2h_{ij}^{\alpha\beta} w'_{ij} - 2h_{ij}^{\alpha\beta} w_{ij} - \delta (h_{ij}^{\alpha\beta})^2)$$

$$= D^2_{before} - \delta (2r - \delta \sum_{ij} (h_{ij}^{\alpha\beta})^2)$$

$$= D^2_{before} - \delta (2r - \delta . n)$$

So the distance is reduced iff $\delta < 2r/n$

where $n = \sum_{ij} (h_{ij}^{\alpha\beta})^2 =$ the number of weights that are changed.

Having a simple convergent learning procedure for a non-linear system is important because it allows the synaptic weights that implement the energy function to be determined by feedback from the correctness of the interpretation that the system settles into. Thus the constraints implicit in the task can be programmed into the system simply by telling it how well it is doing.

The learning procedure assumes that the system receives feedback from an evaluator that tells it whether the current value of $ln(P_\alpha / P_\beta)$ is greater or less than the ideal value $ln(P'_\alpha / P'_\beta)$. This places a very stringent requirement on the evaluator since it must know about the desired probabilities of whole global states like S_α. To build these desired probabilities into the evaluator, the representations that the system should use must be decided in advance. A less omniscient evaluator would only know what *some* of the units should do for each input vector and would leave the system to decide for itself how to use the remaining, "hidden" units to achieve this. Suppose, for example, that there is a set of global states Ω_a which only differ from one another in the hidden units that the evaluator cannot see. The evaluator specifies required probabilities of the form:

$$P'_{\Omega a} = \sum_{\alpha \in \Omega_a} P_\alpha$$

but it does not specify how the total probability should be distributed over the various states in Ω_a. The different ways of distributing the probability correspond to using different representations in the hidden units.

If there are units that are hidden from the evaluator, it is impossible to define a single hypothetical ideal set of weights. There may be many different complete sets of weights which would yield the required behaviour for the "visible" units, and these sets do not, in general, form a convex set. In travelling towards one suitable set of weights, the system may travel away from other equally suitable sets, so convergence on any one set is not guaranteed. This means we need a different measure of the progress of learning in order to prove convergence. A suitable measure is the information theoretic distance, G, between the actual and required probability distributions over all 2^n states of the n visible units:

$$G = \sum_a P_{\Omega_a} ln(\frac{P_{\Omega a}}{P'_{\Omega a}})$$

The value for G depends implicitly on the w_{ij} and so G can be reduced by changing each weight by an amount that is proportional to the partial derivative of G with respect to that weight. We describe this learning rule further in Hinton and Sejnowski (1983). It is guaranteed to find a minimum of G, but it may only be a local minimum rather than a global one. Local minima occur when the system is doing the best that it can given the representations it has learnt in the hidden units. To do better it has to change these representations which involves a temporary setback in how well it meets the requirements on the probabilities of the states of the visible units. Of course, if the modifications to the weights are probabilistic so that G can sometimes increase, it is possible to escape from local minima and ensure that after enough learning there is a bias in favor of the better local minima.

Relation to the brain

There are two different ways to interpret the input-output function that hardware units should have to implement the parallel search (Fig. 1). During a short interval the sigmoid curve describes the probability of a unit being in the true state as a function of the energy gap between the false and true states. For much longer time intervals the curve

describes the proportion of time that the unit is in its true state. If we assume that a hypothesis which is true all the time is represented by a neuron firing at its maximum rate, then the curve in Fig. 1 can be interpreted as the firing rate of a neuron as a function of its average input (Sejnowski, 1977). However, the way in which truth values are represented by action potentials is not the kind of simple encoding in which two different voltage levels stand for the two truth values. Instead, it appears that an action potential only provides a delta-function type of signal that drives integrative processes in the recipient neurons. This amounts to treating a hypothesis as "true" for a whole refractory period after an action potential has been emitted.

The parallel algorithm for cooperative search depends on the computation of energy gaps ΔE_i. In the case of symmetrically connected units the global energy gaps can be computed locally by single units. It seems unlikely that neurons in cerebral cortex are symmetrically connected, but if a neuron receives many inputs it can still estimate what its contribution to the total energy would be if all the connections had been symmetrical. In simulations, asymmetric networks behave like symmetric ones with added noise (Hopfield, 1982), and time delays in transmission have a similar effect. Provided that the task requires symmetric connections, as is the case for problems of constraint satisfaction, an asymmetric network can closely approximate the performance of a symmetric one.

The computational model analyzed in this paper is not a realistic model of processing in cerebral cortex, for it falls far short of explaining the known anatomical and physiological facts. The analysis may, however, provide insight into a class of computational devices that depend on probabilistic parallel processing. Understanding general properties of this class may be a useful first step in understanding particular highly-evolved members of the class. For example, the probabilistic nature of electrical responses of single neurons is well-known, but has generally been regarded as evidence of imprecision. Probability, however, may be a central design principle of cerebral cortex (Sejnowski, 1981). A very close approximation to the function in Fig. 1 can be implemented by simply adding Gaussian noise to a binary threshold unit, with the standard deviation of the noise acting like temperature. We suggest that fluctuations may be deliberately added to neural signals to avoid locking the network into unwanted local optima and to provide the linear conditions needed for efficient learning. The issue of noise in the nervous system deserves renewed experimental investigation and further theoretical analysis.

Acknowledgements

This work was supported by grants from the System Development Foundation and by earlier grants from the Sloan Foundation to Don Norman and to Jerry Feldman. We thank Francis Crick, Scott Fahlman, David Rumelhart, and Paul Smolensky, for helpful discussions.

REFERENCES

Attneave, F. Pragnanz and soap-bubble systems: A theoretical exploration. In J. Beck (Ed.) *Organization and Representation in Perception*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1982.

Binder, K. (Ed.) *The Monte-Carlo Method in Statistical Physics* New York: Springer-Verlag, 1978.

Hinton, G. E. Relaxation and its role in vision. PhD Thesis, University of Edinburgh, 1977; Described in: *Computer Vision*. D. H. Ballard & C. M. Brown (Eds.) Englewood Cliffs, NJ: Prentice-Hall, 1982, pp. 408-430.

Hinton, G. E. & Sejnowski, T. J. Optimal perceptual inference. To appear in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Washington DC, June 1983.

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

Julesz, B. *Foundations of Cyclopean Perception* Chicago: University of Chicago Press, 1971.

Kirkpatrick, S. Gelatt, C. D. & Vecci, M. P. Optimization by simulated annealing. *Science* (in press)

Marr, D. & Poggio, T. Cooperative computation of stereo disparity. *Science*, 1976 *194*, p 283-287.

Metropolis, N. Rosenbluth, A. W. Rosenbluth, M. N. Teller, A. H. Teller, E. *Journal of Chemical Physics*, 1953 *6*, p 1087.

Minsky, M. Papert, S. *Perceptrons* Cambridge, MA: MIT Press, 1968.

Rosenfeld, A. Hummel, R. A. & Zucker, S. W. Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man, & Cybernetics*. SMC-6, 1976, pp 420-433.

Sejnowski, T. J. On global properties of neuronal interaction. *Biological Cybernetics*, 1976, *22*, pp 85-95.

Sejnowski, T. J. Storing covariance with non-linearly interacting neurons. *Journal of Mathematical Biology*, 1977, *4*, pp 303-321.

Sejnowski, T. J. Skeleton filters in the brain. In G. E. Hinton & J. A. Anderson (Eds.) *Parallel Models of Associative Memory*. Hillsdale, NJ: Erlbaum, 1981, pp 189-212.