Running Head: Connectionist Models and Bayesian Inference

Connectionist Models and Bayesian Inference

James L. McClelland
Department of Psychology, Carnegie Mellon University
and the Center for the Neural Basis of Cognition

Send Correspondence to:

James L. McClelland
Center for the Neural Basis of Cognition
115 Mellon Institute
4400 Fifth Avenue
Pittsburgh, PA 15213
jlm@cnbc.cmu.edu
(412)-268-3157 (Voice) / (412)-268-5060 (Fax)

Abstract

In this article the intimate relationship between connectionist models and Bayesian models of optimal inference is explored. First it will be shown how connectionist networks can implement certain classes of optimal perceptual interencing systems. Second, it will be observed how the structure and connectivity of a connecitonist net can represent the statistical structure of the items occurring in a cognitive domain. Third, methods for adaptive shaping of general forms of connectionist architecture into specific structured networks will be discussed. In all it will be argued that connectionist models can capture many of the key properties desired of rational models, although limited training data and the need to simplify certain computations for implementability and learnability typically means that the the statistical models formed in the networks are approximate at best and that the likelihoods that they use to choose an optimal behavior are necessarily estimated likelihoods rather than true likelihoods, and so they are not in fact truly optimal in fact, even though they would be if they had perfect estimates. Such models are called 'quasi-optimal', and it is suggested that this sort of quasi-optimality may be the best that can be hoped for in a cognitive system, and that connectionist models may prove fruitful in the further search for an understanding of the particular ways in which aspects of human cognition deviate from true optimality.

The study of what might be the rational response to a situation, given an intrinsically uncertain world, owes a great deal to the Reverend Bayes, and in contemporory times to a growing band of information processing theorists, cognitive scientists, and artificial intelligence researchers who have pressed extensions and applications of Bayesian ideas. Our most widely used optimal model within psychology may be the theory of the ideal observer within signal detection theory Green and Swets (1966). Within machine vision, the problem of seeing is often framed as one of selecting the most probable or "optimal" interpreation of an input array, based on Bayean calculations (e.g., Bülthoff & Yuille, 1996). Within cognitive psychology, models that address optimality spring in large measure from the recent work of John Anderson, who has used them to address a range of aspects of cognition, from perception to categorization to memory retrieval.

The main point of the present paper is to call attention to another framework in which considerations of optimality in the face of uncertain data have arisen. This framework is the connectionist framework. Several major papers, including those of Hinton and Sejnowski (1983), inciteAckleyHintonSejnowski85, Dayan, Hinton, Neal, and Zemel (1995), Smolensky (1986), Geman and Geman (1984), MacKay (1992a, 1992b), and Rumelhart, Durbin, Golden, and Chauvin (1995) have established and explicated these links, and books covering these matters in great detail have now becoming available e.g. MacKay (1997). These links between connectionist models and Bayesian probability estimation do not appear to have been much appreciated in psychological circles. The present paper endeavors to remedy this situation.

I will begin by reviewing certain very basic relationships between connectionist models and Bayesian approaches to estimating posterior probabilities and formulating sensible choice policies—including optimal ones—in the face of uncertainly. These relationships have been known and appreciated in the connectionist literature since the mid 1980's. This section will include a discussion of a specific content domain—identification of visually presented letters and words—in which it is fairly easy to see explicitly how a connectionist network model—a stochastic version of the interactive activation model (McClelland & Rumelhart, 1981)—actually implements a statistical model of a domain, and how processing in this model can implement Bayesian calculations, yielding outcomes that would be truly optimal under certain limiting conditions. The next section of the paper will consider the problem of learning, here taken to be the problem of determining how appropriate network models for particular content domains can emerge through an interplay of prior network architecture constraints in concert with experience. In discussion I will consider the relevance of these ideas to our efforts to understand the extent of human rationality. I will note that the true optimality is an interesting ideal against which to compare actual performance models—including connectionist ones, but that real performance (and learning) will tend to be "quasi-optimal", at best, and might in fact be decidedly non-optimal under some circumstances. Connectionist models may be helpful for understanding these varieties of real performance.

Connectionist Models as Optimal Inferencing Systems

How individual connectionist units can calculate a posterior probability

In this section, I will begin by considering how individual connectionist units can calculate posterior probabilities of hypotheses, given knowledge of certain probabilitistic quantities and some evidence, in accordance with Bayes' Law, and I will consider how these units can use these posterior probabilities to behave in accordance with either a probability matching or an optimization policy. The treatment begins with a review of Bayes' Law itself and gradually transforms the fundamental

equation into one that is used to govern the updating of activations of units in connectionist networks. The ideas here are due to Hinton and Sejnowski (1983), and I follow many of the details of their exposition, which unfortunately is not published in a very accessible outlet. I have chosen to present this material in full so as to make as explicit as possible the close correspondence between the activation of connectionist units and the computation of posterior probabilities.

According to Bayes Law, the posterior probability of some hypothesis h_i, which may be true or false, given some evidence e, is equal to the prior probability of the hypothesis being true, $p(h_i)$, times the probability of the evidence, given the hypothesis, $p(e|h_i)$, divided by the probability of the evidence, $p(e)$.

$$p(h_i|e) = \frac{p(h_i)p(e|h_i)}{p(e)} \ . \tag{1}$$

The overall probability of the evidence can be partitioned into the probability of the evidence given that the hypothesis is true, times the probability that the hypothesis is true; and the probability of the evidence given that the hypothesis is false, times the probability that the hypothesis is false. These cases exhaust the situations in which the evidence may be obtained, so the sum of their probabilities is the probability of the evidence. Thus

$$p(h_i|e) = \frac{p(h_i)p(e|h_i)}{p(h_i)p(e|h_i) + p(\overline{h_i})p(e|\overline{h_i})} \ . \tag{2}$$

Now we consider the evidence itself. We take the evidence to consist of a set of assertions a_j over a set of elements which we will index with j. These assertions have value 1 (when the evidence is present) and 0 (absent). We use e_j as shorthand for $a_j = 1$ and $\overline{e_j}$ for $a_j = 0$.

If the values of the elements of the evidence are conditionally independent given that the hypothesis is true, then

$$p(e|h_i) = \Pi_j p(e_j|h_i)^{a_j} \ . \tag{3}$$

The term $p(e_j|h_i)$ is the conditional probability that element j of the evidence will be present given that hypothesis i is true. The exponentiation by a_j has the effect of allowing only those terms for which element j is actually present to influence the value of the product. Thus, in words, the probability of the evidence given the hypothesis is equal to the product, over all of the elements of evidence present, of the probability of the element being present, given the hypothesis. Again, this will be true if the elements of the evidence and conditionally independent given the hypothesis. The meaning and importance of this condition is discussed below.

If the values of the elements of the evidence are conditionally independent under the condition that the hypothesis is false, then

$$p(e|\overline{h_i}) = \Pi_j p(e_j|\overline{h_i})^{a_j} . \tag{4}$$

Substituting the above expressions into Equation 2, dividing numerator and denominator by the first, and rearranging a bit, we obtain:

$$p(h_i|e) = \frac{\frac{p(h_i)}{p(\overline{h_i})}\Pi_j \frac{p(e_j|h_i)}{p(e_j|\overline{h_i})}^{a_j}}{1 + \frac{p(h_i)}{p(\overline{h_i})}\Pi_j \frac{p(e_j|h_i)}{p(e_j|\overline{h_i})}^{a_j}} \tag{5}$$

The term appearing in the numerator and denominator is the posterior odds of the hypothesis, i.e., the ratio of the posterior probability of the hypothesis to its negation. This odds term includes the prior odds, $\frac{p(h_i)}{p(\overline{h_i})}$, times the likelihood ratio $\Pi_j \frac{p(e_j|h_i)}{p(e_j|\overline{h_i})}$.

Often, the absence of evidence is important, as readers of Sherlock Holmes stories know. In a very famous Holmes case, an interpretation of the evidence was rejected upon noticing that a particular dog did not bark on the night of the crime. To deal with such cases, we can introduce specific elements of the evidence vector corresponding to such things as "the dog did not bark". Another way of building this information into the weights and biases without explicit elements for missing evidence was suggested by Hinton and Sejnowski (1983), and we will introduce this because it will be useful at a later point in our analysis.

Consider a particular potential element of the evidence relevant to a given hypothesis. When the evidence is absent, we want the posterior odds of the hypothesis to reflect the odds that the element of the evidence will be absent when the hypothesis is true. We can achieve this by simply multiplying the prior odds times that quantity, $\frac{p(\overline{e_j}|h_i)}{p(\overline{e_j}|\overline{h_i})}$. Having done this, however, we must find a way to cancel this term back out when the evidence is present. This is done by including the reciprocal of these odds, or $\frac{p(\overline{e_j}|\overline{h_i})}{p(\overline{e_j}|h_i)}$ into the corresponding term of the products over j in Equation 5. Assuming the various elements of the evidence are conditionally independent, the same argument applies equivalently to all of the potential elements of evidence. The resulting new expression for the posterior odds becomes

$$q_i \frac{p(h_i)}{p(\overline{h_i})} \Pi_j \frac{p(e_j|h_i)p(\overline{e_j}|\overline{h_i})}{p(e_j|\overline{h_i})p(\overline{e_j}|\overline{h_i})}^{a_j} \tag{6}$$

The first fraction $\frac{p(h_i)}{p(\overline{h_i})}$ in this expression is the prior odds of the hypothesis, which we will write o_i. The elements of the product now have an interesting property, which is that they can be thought of as expressing the odds that the evidence and the hypothesis agree (both true or both false). This is a measure of the degree of contingency between these two, and will henceforth be designated c_{ij}. The remaining term q_i represents $\Pi_j \frac{p(\overline{e_j}|h_i)}{p\overline{e_j}|\overline{h_j}}$, the product of the factors that correct the posterior odds for the absence of each element of the evidence. We can substitute back into Equation 5, to obtain an updated version of that equation in terms of our summary quantities:

$$p(h_i|e) = \frac{q_i o_i \Pi_j c_{ij}^{a_j}}{1 + q_i o_i \Pi_j c_{ij}^{a_j}} \tag{7}$$

Thus far we have carried out a computation based entirely on a Bayesian analysis. We now examine how this result relates to the computations standardly performed by connectionist units. What we will do is translate the above expression into the standard *logistic* network activation function. The first step is to divide the numerator and denominator by the numerator, to obtain:

$$p(h_i|e) = \frac{1}{\frac{1}{q_i o_i \Pi_j c_{ij}^{a_j}} + 1} \tag{8}$$

Commuting the terms in the denominator, and using the identities $x = e^{\log x}$, $\log x^z = z \log x$, and $\log(1/x) = -\log(x)$, we obtain:

$$p(h_i|e) = \frac{1}{1 + e^{-\log(q_i o_i) + \Sigma_j a_j \log c_{ij}}} \tag{9}$$

This expression is identical to the logistic activation function, usually written:

$$logistic(net_i) = \frac{1}{1 + e^{-net_i}} \tag{10}$$

where $net_i = bias_i + \Sigma_j a_j w_{ij}$, $bias_i = \log(q_i o_i)$, and $w_{ij} = \log c_{ij}$.

What the above derivation establishes is that the logistic activation function often used with connectionist units can compute the posterior probability of a hypothesis for which the unit stands, if (a) the probabilities of the elements of the evidence obey the conditional independence conditions stipulated, and if (b) the bias term in the net input equals the log of the prior odds of the hypothesis (corrected for possibly absent evidence by q_i) and the weights to a unit equal the log of the contingency between the hypothesis and the evidence, as defined above.

The logistic function may be used to set the activation of the unit to the value computed by this function, in which case the activation becomes equal to the posterior probability if a and b apply. Alternatively, the logistic function may be used to determine the probability that the activation of the unit will be set to 1. In this case the unit's activation can be thought of as an assertion about the truth of the hypothesis. If a and b apply in this case then the assertation that the hypthesis is true will be made with a probability equal to the probability that it actually is true. This behavior is often called *probability matching.*

Generally speaking, optimal behavior is thought to be behavior that maximizes the probability of being correct. Even though it is not optimal in this sense, probability matching characterizes actual human performance in many task situations, and has been extensively studied by operant conditioners (Herrnstein, 1970; Staddon, 1982). Truly optimal behavior, or maximizing the probability of being correct, can be achieved by the use of an even simpler activation function that the logistic function, namely the *linear threshold* function, in which activation of the unit is set to 1 if the net input as defined above is greater than 0, and is set to 0 otherwise. If a and b apply, then the linear threshold function maximizes the likelihood of choosing the correct truth value for the hypothesis, given the evidence.

One can move continuously between probability matching and optimal choice behavior by parameterizing the logistic function with a scalar $T > 0$, inspired by the thermodynamical quantitity temperature:

$$logistic(net_i/T) = \frac{1}{1 + e^{-net_i/T}} \tag{11}$$

For $T > 1$, the values of this function are less extremal (nearer to the neutral value 0.5 than to either 0 or 1, the two extremes) than probability matching; for $T < 1$ the values of this function are more extremal than probability matching. In the limit as $T \to 0$, the logistic function converges to the optimal behavior of the linear threshold function.

Conditional Independence. Crucial conditions in the above analysis was the condition that the elements of the evidence were *conditionally independent* given the truth or falsity of the hypothesis. This condition may or may not actually apply in fact, and the computation performed by a particular logistic connectionist unit will not in general compute the posterior probability of an hypothesis unless this assumption is correct. Formally, conditional independence of a and b, given x holds if and only if $p(a|x) = p(a|x,b)$ and $p(b|x) = p(b|x,a)$. Such a situation can arise in many situations. For example, consider the hypothesis that the sender of a sequence of seven bits had intended to transmit a particular ascii character (i.e., a particular seven-element sequence of 1's and 0's), but that each bit was subject to probabilistic flipping due to noise. If, after the correct bits to transmit are chosen, each bit is probabilistically flipped independently of the others, then the values of the bits are conditionally independent given that the intended bit sequence was the one corresponding to the chosen character. That is $p(b_i|c_k) = p(b_i|c_k, b_j)$, and $p(b_j|c_k) = p(b_j|c_k, b_i)$, where b_i (b_j) is the means that bit i (j) equals 1, and c_k means that the intended bit sequence is

the one for character k.

Note that conditional independence does not imply complete independence, defined as $p(a) = p(a|b)$, $p(b) = p(b|a)$. In particular, consider the two possible ascii characters consisting of "0000000" and "1111111". If each bit is purturbed independently with probability x, $p(b_i) = 0.5$ regardless of x, but $p(b_i|b_j) = 1 - 2x_2x^2$, which is greater than 0.5 whenever x is less than 0.5, and approaches 1 as x approaches 0.

The point of stressing the concept of conditional independence is fundamentally to broaden the audience of psychologists who understand and use the concept, since it appears to be highly relevant to understanding the circumstances in which relatively simple computations (whether conceived as Bayesian or connectionist) can validly compute posterior probabilities. Such assumptions are often applied in pure Bayesian contexts (note that we introduced them before making the transistion to the connectionist activation function), the reason being that they may the computation of probabilistic terms intractable when they are not introduced. But the fact that they are not likely to be true in general means that models that employ these assumptions are not necessarily guaranteed to compute true Bayesian quantities. Rather, these quantities might best be treated as hypothetical psychological estimates of these quantities, which may not be exactly valid but could be valid under some circumstances.

A similar conclusion comes the more basic observation that the computations described only compute true probabilities when the correct prior and conditional probabilities are used in the computations. In general it seems likely that any psychological model will be based on the use of estimates of these quantities, and so, once again, the quantities computed will themselves not be true posterior probabilities but only estimates of these probabilities.

We now go on to consider an extension of the above analysis. The same caveats all apply as above; as before, they seem to apply to non-connectionist as well as connectionist approaches.

How a collection of connectionist units can select the most likely one of several mutually exclusive hypotheses

Consider a set of mutually exclusive hypotheses among which we might wish to choose in the face of some evidence. An example would be the set of possible letter identities that might be assigned to a sequence of bits, where the letter identities are the hypotheses and the values of the bits are the evidence. In Bayesian terms, the posterior probability of each hypothesis, given the evidence, would be:

$$p(h_i|e) = \frac{p(h_i)p(e|h_i)}{\Sigma_{i'}p(h_{i'})p(e|h_{i'})} \tag{12}$$

In this case, the partitioning of cases in the denominator extends over all possible letter identities (indexed here with the subscript i'), rather than over a given hypothesis and its negation, as in our earlier analysis. Assuming that the elements of the evidence are conditionally independent, given a particular intended letter, These probabilities can be calculated by a the use of a generalization of the logistic function:

$$a_i = \frac{e^{net_i}}{\Sigma_{i'}e^{net_{i'}}} , \tag{13}$$

where $net_i = bias_i + \Sigma_j w_{ij}a_j$ as before. In this case, $bias_i = \log p(h_i)q_i$, $q_i = \Pi_j p(\overline{e_j}|h_i)$, and $w_{ij} = \log(\frac{p(e_j|h_i)}{p(\overline{e_j}|h_i)}$. This activation function is often called "softmax".[1]

[1]I describe this function as a generalization of the logistic function because the expression collapses exactly to the logistic function where there are just two mutually exclusive hypotheses, h_1 and h_2. The standard logistic computes

Probability matching can be achieved by selecting a single unit to activate, with the probability of selecting i equal to a_i. This case is equivalent to Luce's choice model where the item strengths are the e^{net_i}. It is likewise equivalent to Morton's Logogen Model (Morton68) and the Fuzzy Logical Model of Perception (MassaroOden78).[2] Optimization, or maximization of the probability of being correct given a set of features, is accomplished simply by selecting the unit with the largest value of net_i. This 'winner take all' computation is exactly what is done in conventional *competitive learning* networks (e.g., Rumelhart & Zipser, 1985).

The exact calculation of the softmax activation can be performed by a feedforward network, including an input layer, a hidden layer, an aggregation layer, and a final output layer (Figure **??**. The hidden layer contains an exponential unit for each alternative, i.e., a unit that calculates $a_i = e^{net_i}$. The aggregation layer calculates the sum of these quantities. The output layer again contains a unit for each alternative, this time calculating the ratio of the activation of the corresponding exponential unit divided by the aggregate quantity calculated at the previous layer.

An alternative, more elegant, implementation uses a recurrent computation. One unit is assigned to each hypothesis, and there is a single aggregation unit, which simply adds the activations of the hypothesis units and feeds this aggregated quantity back to each hypothesis unit. Depending on the choice of various details, such a system can closely approximate the winner-take-all and softmax computations (Grossberg, 1978). In the brain, it appears that a population of inhibitory interneurons is often used to perform the operation of aggregating the excitation and feeding it back to the excitatory neurons (McNaughton, 1989).

As in the single-hypothesis case, parameterizing softmax with T can yield a range of policies from random choice as $T \rightarrow$ inf, to probability matching when $T = 1$ to optimization as $T- > 0$. Another frequent policy is simply to set the activations of the units to the posterior probabilities, as given by the above expression. This is very useful when one wishes to calculate the posterior probability of some larger over-arching hypothesis (e.g., that the stimulus is a particular word) based on the probabilities of various lower-order hypotheses (e.g., that a particular position within the word contains a particular latter), but it runs the risk of introducing averaging over cases that results in violations of some of the contingent relationships among the activations of elements.

Finding a Complete Interpretation of a Scene

Thus far we have seen how connectionist models can implement certain calculations, implementing a form of optimal assessment of the probability that inputs correspond to particular preestablished hypotheses. In this section, we introduce the far more powerful idea that a connectionist network might be able to behave in accordance with subjective probabilities of entire *configurations* of hypotheses which taken together can be seen as an interpretation of an ensemble of inputs, such as a visually-presented scene. For the moment we consider the general case, in which the various hypotheses may all be mutually dependent. In order to begin this analyis, we need to define the subjective probability of an interpretation. Once we define this we can then consider how the selection of an interpretation can depend on the value of this subjective estimate.

We consider some composite assertion h, which can be thought of as a pattern of assertions over

the probability of h contrasted with the mutually exclusive hypothesis $\overline{h}$, whose probability is of course $1 - p(h)$.

[2] In these models, the total item strength is generally expressed as a product of terms, e.g., the product of a bias term which may reflect item frequency, a stimulus term, reflecting the degree of support for the item from the stimulus, and a context term, reflecting the degree of support for the item from the context. The values added together to determine the net input to the unit for an hypothesis can be thought of as the logarithms of these multiplicative terms.

an ensemble of units whose activations represent individual assertions about the truth of particular hypotheses, in the context of some evidence e. The subjective probability $\rho(h)$ of this particular ensemble of assertions is defined as

$$\rho(h) = \frac{(\Pi_i o_i q_i' i_i)^{a_i} \Pi_{j<i} {c_{ij}}^{a_i a_j}}{Z} \tag{14}$$

Interpreting this expression, we note that the numerator contains two products. The first product, which ranges over hypotheses, consists of terms which reflect the odds of each hypothesis, based on the prior odds o_i, the input i_i, and the correction factor q_i'. The term o_i is as previously defined. The term i_i stands for the likelihood of the direct evidence given hypothesis i, and has the value $Pi_k c_{ik}^{a_k}$ where k indexes the elements of the evidence vector. The term q_i' is a correction factor analogous to q_i from before, which now takes into account 0 values of other hypotheses, as well as of the evidence. We discuss how the value of q_i' could be determined later. This first product, by itself, is just the product of the individual posterior odds of the assertions that have value "true" in h, and would, with q_i in place of q_i', describe the odds of the interpretation as a whole under the assumption that the various hypotheses are all mutually independent. The second product, which ranges over the connections among the units, can then be seen as correcting the estimate of the subjective probability for the effects of the contingencies amoung the hypotheses. Here the c_{ij} are defined in analogy to the previous definition, $c_{ij} = \frac{p(h_j|h_i)p(\overline{h_j}|\overline{h_i})}{p(h_j|\overline{h_i})p(\overline{h_j}|\overline{h_i})}$. The notation $\Pi_{j<i}$ refers to a product over all pairs of units; i and j both index the units, but because of the restriction $j < i$, each pair of units is counted only once. This product can be seen as cumulating the contingencies over all pairs of "active" hypotheses i and j.

Note that c_{ij} has the same value for a given pair i, j, regardless of their order. This follows from the definition of conditional probability, $p(x|y) = \frac{p(x\&y)}{p(y)}$. Using this definition, we can replace all four conditionals with the corresponding ratios in the definition of $c_i j$. The denominators of the ratios cancel out, so that $c_i j$ reduces to $\frac{p(h_j\&h_i)p(\overline{h_j}\&\overline{h_i})}{p(h_j\&\overline{h_i})p(\overline{h_j}\&\overline{h_i})}$, which is symmetric in i and j since the logical & operator is commutative.

The final element of Equation 14 is the grand denominator term Z. Z is often called the *partition function*, and, in previous cases, is a sum of terms identical to those in the numerator, where the summation ranges over all possible alternative ensembles of hypotheses h'. Thus the equation can be seen as a "Luce ratio" over ensembles of hypotheses, rather than individual hypotheses.

We now consider the relationship between such expressions and quantities relevant to connectionist networks. The logarithm of $\rho(h)$ bears a close relation to the quantity connectionists have labeled *Goodness* of the state of a network (G_s). For example in Rumelhart et al (1986; see also McClelland & Rumelhart, 1988), G_s is defined as

$$G_s = \Sigma_i a_i (bias_i + input_i) + \Sigma_{j<i} a_i a_j w_{ij} \tag{15}$$

Given this definition, $\log \rho(h) = G_s - \log Z$, or $G_s = \log \rho(h) + \log Z$, if $bias_i = \log(o_i q_i')$, $input_i = logi_i$, and w_{ij} is set equal to $c_i j$. Other connectionists, notably Hopfield (1982) and Hinton and Sejnowski (1983, 1986) use the quantity E_s, in analogy with quantities used in physics. Their E_s is simply equal to the negative of our quantity G_s. Note that, since G_s is strictly monotone increasing with $\rho(h)$, any policy that maximizes G_s also maximizes $\rho(h)$.

The key observation, due initially to Hopfield, is that the net input to a given connectionist unit is equal to the derivative of G_s with respect to the activation of the unit:

$$\frac{\partial G_s}{\partial a_i} = net_i = bias_i + input_i + \Sigma_j a_j w_{ij} \tag{16}$$

Hopfield suggested that a network could maximize G by essentially the following policy: Visit units one at a time, choosing the index i of the unit to visit randomly with replacement, and setting the activation of the unit to 1 if $net_i \geq 0$ or to 0 otherwise. Following this policy G_s may increase or stay the same at each step, and never decreases, due to the symmetry of the weights. Following this policy then the network performs hillclimbing in G_s, eventually reaching a *local maximum* after which the value ceases to change. This policy is equivalent to finding an ensemble of hypotheses h whose subjective probability $\rho(h)$ is a local maximum.

The concept of local maximum is crucial here and is worth understanding intuitively. In this case, a local maximum is simply a state whose Goodness is greater than or equal to that of any state that can be reached by changing the value of just a single hypothesis. In general, of course, local maxima may not correspond to the best possible state.

Several investigators in the mid-1980's observed that stochastic neural networks could overcome this limitation (Hinton & Sejnowski, 1983; Geman & Geman, 1984; Smolensky, 1986). Hinton and Sejnowski (1983, 1986) observed that if Hopfield's procedure is followed, but the unit chosen for updating choses its activation to be 1 probabilistically, with probability $p(a_i = 1) = logistic(net/T)$, then the network would eventually escape from any local minimum, and, at *thermal equilibrium*, the probability of finding the network in any given state s would be equal to

$$p(s) = \frac{e^{G_s/T}}{\Sigma'_s e^{G_{s'}/T}} \ . \tag{17}$$

In this equation, the quantity T again corresponds to *temperature*, based on the analogy to statistical physics. Two cases are of particular significance: (1) When T equals 1, the above equation reduces exactly to the equation for $\rho(h)$ above. In this case, the network performs probability matching over whole interpretations, selecting states corresponding to interpretations, with the probability of selecting state s equal to the subjective probability of the interpretation. (2) As T approaches 0, the states with greater probability come to dominate more and more, and in the limit the only state with appreciable probability is the one best state, and one can choose a value of T to guaratee that the probability of reaching the optimal state is as close to 1 as desired.

A difficulty arises from the fact that thermal equilibrium is very difficult to reach with very low values of T. Essentially, the higher the value of T the more likely the network is to escape a local minimum in a given number of unit updates, and at low values to T the probability of escape can be vanishingly small. To address this dilemma, some connectionists adopted the policy of simulated annealing, proposed by Kirkpatrick, Gelatt, and Vecchi (1983). According to this policy, one starts with a large value of T and gradually reduces it. While this often works in practice no really good, problem-independent policy for adjusting T has been proposed, and hand tuning seems inappropriate. In general, reaching the optimal solution can only be guaranteed if an infinite time is allowed.

It should also be noted that even the policy of using a fixed value of $T = 1$ only guarantees probability matching if the network is allowed to run to thermal equilibrium. Thermal equilibrium is that time after which the probability distribution of states visited by the network is stationary. The state may change with each update, but the tendency to move into a given state exactly balances the tendency to move out of that state, so that the probabilities of being in particular states remain the same. I know of no procedures for determining when thermal equilibrium has been reached, though in simulations of a couple of particular problems we have found that thermal equilibrium tends to be reached within well under 100 cycles (each unit is updated an average of once per cycle).

In summary, this section has indicated how connectionist models provide procedures for finding good interpretations of inputs, represented as patterns of activations over units. Goodness is defined quantitatively in terms of a function that increases strictly monotonically with the subjective probability of the state, under the definition of subjective probability defined above. As before, several caveats apply.

The most important caveat remains the fact that the probabilities is question are *subjective*. It is important to distance this concept of subjective probability from true probability for two reasons. The first reason is that the probabilistic priors, corrections for 0's, and contingencies that are identified in the equations are likely to be estimates in any psychological model rather than actually correct values. The second, and deeper, reason is that the very framing of the hypothesis space in terms of the particular ensemble of hypotheses instantiated by the units of a network is itself some sort of estimate, or model, of the real causal situation that governs the actual inputs to the network. It is conceivable that a network might encode this causal structure correctly, but whether it really does is a separate question.

An Example Network for Perceptual Inference

Having considered matters rather abstractly thusfar, it may be helpful to consider a specific example. The example I have chosen here is that of the stochastic interactive activation model (McClelland, 1991; Movellan & McClelland, 1995). This model grows out of the original interactive activation model of McClelland and Rumelhart (1981; Rumelhart & McClelland, 1982), and has strong connections with earlier work of Rumelhart (1977; Rumelhart & Siple, 1974).

We will use this example to address a question that is left open by the preceeding analysis. The analysis introduced the concept of a subjective probabilistic model, in which the units and connections represent entities and constraints among them, which together provide an subjective account for the probabilistic structure of obvserved inputs. It was demonstrated that Boltzmann machine networks embodying such models in the units and connections can find the optimal interpretation (through annealing) or (when run at fixed temperature $T = 1$) produce interpretations probabilistically, with the probability of producing the interpretation matching the probability that the interpretation is correct. The mechanism appears to be very general, but there is one important issue that remains: It was not clear to what extent such models could actually capture the true probabilistic structure of experiences. To what extent could the subjective models embodied in these networks be realistic?

The question is difficult, since in the end it becomes a question about the sources of experiences in the world and of the possible projections and distortions of these that make observations only probabilistically related to the underlying sources, and I know of no general characterization of these. In particular cases, however, a generative model of the set of possible exeriences, which amounts to a kind of quasi-plausible cover story, can sometimes be constructed that effectively justifies a particular Bayesean model of the sources of inputs in a particular limited environment. Once such a story exists it can then be translated into a Boltzmann machine. A case in point is a cover story I will now present that can justify a Boltzmann machine version of the interactive activation model, as it applies to inputs arising from words that are four letters long.

In every day experience, such stimuli arise in written text with a probability proportional to their frequency of occurrence. Our model assigns prior probabilities to words based on these frequencies. The probability $p(w_i)$ of a particular four-letter word indexed by i given that the word is one of the four-letter words is $f(w_i)/\Sigma_i' f(w_i')$, where $f(w_i)$ ($f(w_i')$ is the overall frequency of word i (i'), and

i' ranges over words of four letters. Given that a particular word i is intended by the author of the text, the letters of the word are likely to be the ones that typically make up the word in question. However, error is possible in the spellings of words. To capture this in our model, we introduce the notion that the actual letters of will occur with probabilities that depend on the word, but are not absolutely certain. The probability of a particular letter j being generated in position p or word i will be represented $p(wl_ijp)$. The model further assumes that each letter is made up by choosing which of a set of letter-strokes are present in the letter and which are absent. For specificity, we will assume the font used by Rumelhart and Siple (1974) and McClelland and Rumelhart (1981). This font, shown in Figure **??**, was actually used to construct the stimuli used in the Rumelhart and Siple (1974) experiment. We will formulate the problem by considering separate hypotheses for the presence of a particular element and for its absence. These will be designated f_{vep}, where v indexes the value (1 for present, 0 for absent), e indexes the dimension (1-14 for the fourteen elements that may be present or absent) and p indexes the position in the word. Once again, error is possible, perhaps in the set of features actually presented when a particular word is intended or perhaps in the early stages of the perceptual system, so that the set of features detected from the presentation of a word will be probabililistically related to the correct features of the letter but not necessarily exactly correct every time. We denote the probability of detecting a particular value of a particular element in a particular position given an intended letter in that position as $p(lf_{vejp})$. Note that this need not be independent of position. The above model completely determines the probability distribution of features patterns, and allows Bayesian inference to be applied in an effort to address what word might have been intended when a particular pattern of features is detected.

We now present a connectionist network that provides a possible mechanism for addressing these questions. An input will be a pattern of 1's and 0's specifying activations of input units representing values of the elements of each of four letters. A restriction on the inputs will be that at most one of the values of a given element may be specified. The design goal for the perceptual mechanism will be to construct hypotheses about which word and which letters in each position were intended, by activating units corresponding to these hypotheses. Well-formed hypotheses will be restricted to those involving a single hypothesized word and a single hypothesized letter in each position of each word. The network for this has the exact same units and connections as the interactive activation model introduced by McClelland and Rumelhart (1981). That network had a unit for each word; a unit for each letter in each of the four positions; and a unit for each value of each element of each letter in each position. The diagram in Figure **??**, from McClelland (1985), captures this structure better than the original diagram in McClelland and Rumelhart (1981). Now, we assign the bias terms for the units and the weights on the connections among the units. At the word level, let $b(w_i) = \log p(w_i)$. At the letter level, let $b(l_{jp}) = \log p(l_{jp})$. For the word-letter weights, these will be assumed to be bi-directional, and will be set to $w(wl_{ijp}) = \log p(wl_{ijp})$. The letter-feature weights are be uni-directional, from the feature level to the letter level, so that the feature units serve only to define inputs and are not affected by the computation. These will be set so that $w(lf_{jvep}) = \log p(lf_{jvep})$. Finally, we must assign 'lateral inhibitory' weights, to prevent ill-formed interpretations involving multiple active words or multiple active letters within positions. Since such interpretations are (for the moment at least) considered impossible, these weights "should" be infinitely negative, but we will just assume large enough each weight has a large negative value to make the probability of such interpretations negligable, rather than strictly impossible. We assume the value is the same for all such weights, and reprent the value as g, so that $w(ww_{ii'}) = w(ll)_{jj'p}) = g$ for the connections among all pairs of words ii' and all pairs of letters in the same position $jj'p$.[3]

[3]In the original interactive activation model, letter-letter inhibition was set large to prevent multiple letters from

Consider using this network as a Boltzmann machine. That is, let the units in this network be updated according to the random, asynchonous order proposed by Hopfield, and let the value of the unit chosen for updating at each step be set to 1 with probability $logisticnet$, or to 0 with the complementary probability. Since the weights among units that are subject to updating are symmetric, we can determine that if we clamp the inputs to a particular pattern, then run the network to equilibrium at the fixed temperature $T = 1$, the probability of begin in any state s will be:

$$p(s) = \frac{e^{G(s)}}{\Sigma'_s e^{G(s)}} \tag{18}$$

Where

$$\begin{aligned} G(s) &= \Sigma_i a(w_i) b(w_i) + \Sigma_{jp} a(l_{jp}) \left(b(w_{jp}) + \Sigma_{ve} a(f_{vep}) w(lf_{jvep}\right. \\ &\quad + \Sigma_i \Sigma_{jp} a(w_i) a(l_{jp}) w(wl_{ijp}) \end{aligned} \tag{2}$$

$$+\Sigma_{i<i'} a(w_i) a(w'_i) w(ww_{ii'} + \Sigma_{p,j<j'} a(l_{jp}) a(l_{j'p}) w(ll_{jj'p}) \tag{2}$$

Given that g was chosen large enough so that states with more than one word active or more than one letter active in any position have such negative goodnesses that their probability of occurrence negligable, we need consider only cases in which there is just one active word and one active letter in each position, in which case both of the last two summations are 0. Furthermore the summation over words and letters collapse to those terms for the single word i that is active at the word level and the single letter j active at the letter level in each position p. The activations of these units are 1 and so become implicit in these remaining terms.

$$G(s) = b(w_i) + \Sigma_p w(wl_{ijp}) + \Sigma_p \left(b(l_{jp}) + \Sigma_{ve} a(f_{vep}) w(lf_{jvep})\right) \tag{22}$$

Exponentiating, we obtain:

$$e^{G(s)} = p(w_i) \Pi_p p(wl_{ijp}) \Pi_p \left(p(l_{jp}) \Pi_{ve} p(lf_{jvep})^{a(f_{vep})}\right) \tag{23}$$

Under the model of the inputs descrive above, the value of $e^{G(s)}$ actually corresponds directly to the likelihood that the input was generated in accordance with the corresponding interpretation. Since the true probability that an interpretation is correct is the likelihood of that interpretation divided by the sum of the likelihoods of all the interpretions, the probability that the network will be in a particular state at equilibrium will be equal to the probability that the interpretation that state represents is correct.[4]

being active but word-word inhibition was set relatively small to allow partial activations of multiple words, thereby producing perceptual facilitation for letters in pseudowords as well as words, consistent with a great deal of behavioral data. Such cases can be seen as one of many ways of allowing pseudowords, taken to be strings that, strictly speaking, have no prior frequency but are nevertheless similar to many stimuli that have previously been experienced to be considered as candiate percepts. For further discussion of the relation between this idea and other ways of providing for the possibilty of perceiving non-words, see Movellan and McClelland (1995).

[4]In the actual interactive activation model as amended by McClelland (1985), and in the TRACE model of speech perception (McClelland & Elman, 1986), word-letter (in the latter case, word-phoneme) weights were positive for the letters (phonemes) of a word and 0 otherwise. The positive weights can be seen as representing the log of the ratio of the likelihood of the correct letter to the likelihood of an incorrect letter, $\log(p(cl)/p(il))$, while the 0 weights can be seen as representing the log of the ration of the likelihood of an incorrect letter to itself, $\log(p(il)/p(il)) = \log 1 = 0$. The reader can check that this adjustment of the weights has not net effect on the likelihood of settling to a particular state with a single word active and a single letter active in each position, because adjustment cancels out the ratio that determines the probability of settling to each state. Such factors can influence the relative likelihood of settling to states where 0 words are active or where multiple words are active, for a given value of the inhibitory parameter g, and thus influences the likelihood of blend states. As discussed in the preceding footnote, such states are relevant to the ability of such networks to account for perceptual facilitation of letters in pseudowords.

Now, consider what would happen in the above network if it were not run at fixed temperature, but we gradually annealed, reducing the temperature until ultimately it reaches 0. In the limit of slow annealing,

$$p(s) = \frac{e^{G(s)/T}}{\Sigma'_s e^{G(s)/T}}\,. \tag{24}$$

In this case $p(s) = \frac{l(s)^{1/T}}{\Pi_{s'} l(s')^{1/T}}$, and as T approaches 0 the state with the largest likelihood eventually collects all the probability, so the network will pick the most likely interpretation.

The above example should serve to demonstrate that actual networks can be constructed to carry out veridical probability matching or veridically optimal inference in cases where the probility distributions of inputs can be defined by an appropriate generative model. It is interesting that the generative model is entirely "top-down", in that first a word is specified, then its letters are specified given the word, and then its features are specified given the letters. Yet, even while the generative model is top-down, the activation process is bi-directional. It is worth noting that the generative model in this case strongly adheres to the principle of conditional independence. The value of each element of each letter position is dependent only on the letter in that position, and the identity of the letter in each position is dependent only on the word. These relations are captured by the designated architecture of the model. [5]

Of course, other sorts of cases are possible, and other architectures can be used to capture them. If, for example the features of letters actually varied as a function of which word the letters were in, then the model would be inappropriate. Or if words might appear in several different fonts, but within a word the font was always consistent, a different architecture would be required. It is left as an exercise for the reader to consider how networks embodying such architectures might be constructed.

In general, it is not known what the range of appropriate network architectures is because it is not known what the structure of relevant experiences are. It does appear, however, that generative models of at least some domains of experience can be formulated in explicitly Bayesian terms, and that such models can then be translated into connectionist networks. Such networks are Bayesian inference systems, to the extent that the models they embody are correct.

Bayesean Learning in Connectionist Networks

The fact that we can sometimes write down a generative model of an environment, and then program it into a connectionist network to implement Bayesian inference is important, but it is just the first step in answering the question of how rational responses to an uncertain world might come about. The next step is to address the question of whether, and under what circumstances, it is possible to learn to behave in a rational, or quasi-rational, way. Connectionist models haven't completely solved this problem, but many interesting and useful ideas have been put forward, and they work to an extent in particular cases.

[5] The reader might be puzzled by the fact that the weights and biases in the word perception model actually take simpler values that the ones developed for the general case. Part of the reason for this is that some of the extra complexities tend to cancel out in the conversion from $e^{G(s)}$ to $p(s)$, when we divide through by the sum of $e^(G(s))$ over all possible states. In general, a variety of changes in the details of the models can leave these computations intact. For example, when only one word is active in each interpretation, the $p(w_i)$ terms can be replaced by their raw frequencies by simply neglecting the division by $\Sigma'_i p(w'_i)$, since this factor cancels out of the expression for $p(s)$ in any case (the value of g may have to be adjusted to ensure just only states with one word have non-negligable probabilities in this case).

Learning to Maximize the (Elementwise) Probability of the Output given the Input

A basic discovery about the widely-used back-propagation algorithm is that it can, under certain restrictions, be viewed as learning to produce the output that is most likely given a particular input. Aspects of these ideas are developed in MacKay (1992b) and Rumelhart et al. (1995). We will consider the standard case and then note how the analysis may be generalized.

Consider the task of learning to predict the most likely second member of an input-output pair from the first member, given a set of training examples. A wide range of problems are of this type, such as, for example, the problem of predicting where a shell will land given a measurement of its position and velocity. Due to possible inaccuracies in the measurements or purturbations during flight, the mapping from measurements to points of impact is sure to be probabilistic. Suppose the distribution of points of impact, given the measurements, is a two-dimensional Gaussian, i.e., the probability of landing at point (x, y) is proportional to $e^{-\frac{(x-\mu_x)^2+(y-\mu)y)^2}{2\sigma^2}}$, where (μ_x, μ_y) is the mean of the Gaussian and σ is the standard deviation. We wish to adjust the weights of a feed-forward connectionist network with linear output units whose actiations are estimates of μ_x and μ_y to maximize the probability of the actually observed (x, y) values (the hidden units may use non-linear activation functions, such as the logistic function). It turns out that the back-propagation algorithm, which adjusts the connection weights in the network to minimize the sum of the squares deviations between the network's outputs and the observed or "target" values found in training examples is performing just this minimization. Specifically, the squared error measure (in this case $(x - \mu_x)^2 + (y - \mu)y)^2$) has a negative linear relation to the log of the probability of observing (x, y), so minimizing this error measure maximizes the probability of the observed data, given the estimates. The idea can be extended to any number of output elements, yeilding the familiar sum-squared error $\Sigma_i(t_i - a_i)^2$.

This analysis can be generalized to other probability distributions by suitable choices of activation functions and error measures. For example, when the outputs are binary-valued, the correct activation function is the logistic, interpreted as generating an activation between 0 and 1 representing the probability the correct or target value for the corresponsing output will have value 1. In this case the measure is a quantity called the *cross-entropy*, whose value is $\Sigma_i t_i \log(a_i) + (1-t_i)\log(1-a_i)$. This and other cases are explored at length in Rumelhart et al. (1995).

Learning to match the probability distributions over the outputs

The previous section connects network models to probabilistic optimization, but there is an important limitation. The network's output is basically an elementwise expected value. This can fail to capture important characteristics of outputs, such as possible bi-modality or co-variation of elements. Indeed, in many cases, computing an average can be of relatively little use, since the average of the possible outputs may not be one of the possible outputs. The problem arises in motor control. Suppose we desire to learn to set as output the sholder and elbow angles of an arm so that we successfully reach an object whose (x, y) position is given as input, and for training examples we are given a series of examples consisting of (x, y) positions as inputs and (Θ_s, Θ_e) as targets that correctly reach the indicated position. The problem is that there may be more than one adequate (Θ_s, Θ_e) pair for a given (x, y), but their average may well be completely inadeqate (See Figure ??). A similar problem arises in translation between langauges, where a given sentence in one language may be expressible in several different ways in another language, but the average of these different expressions is meaningless.

An solution to this problem exists, and it also addresses the fundamental issue left over from

the first section of this article, in the form of a method for training Boltzmann machines and other stochastic networks, allowing such a network to model the statistical structure of the training environment directly in its own behavior. These ideas were introduced by Ackley, Hinton, and Sejnowski (1985) and extended to symmetric diffusion networks by Movellan and McClelland (1993).

The approach begins by framing the learning problem explicitly in probability matching terms. That is, the goal of learning, under this approach, is to train a network to produce, not just a single output for a given input, but one of a range of possible outputs, where the probability of producing each such output matches the probability of that output actually occurring in the environment. For example, if the English word "olive" is translated into Spanish 50% of the time as "aceituna" and 50% of the time as "oliva", then the network should generate the output pattern representing each of these two alternatives 1/2 of the time.

Information theory provides a measure of the degree to which the distribution of a network's outputs match a distribution provided by a set of training examples.

$$IG = \Sigma_i \Sigma_j P_t(O_j|I_i) \log \frac{P_t(O_j|I_i)}{P_n(O_j|I_i)} \tag{25}$$

This measure is sometimes called *information gain*, hence the abreviation IG, and was suggested for training neural networks by Ackley et al. (1985). The notation $P(O_j|I_i)$ is the probability of output pattern O_j given input pattern I_i. Note that each I_i and each O_j are whole patterns, e.g., I_i may be a phonological pattern representing the English word "olive", with which the two output phonological patterns representing the two possible Spanish translations may be paired. Or I_i could be a position in (x, y) space and the different O_js may be different configurations of joint angles that put the tip of a robot arm at this position. The notation P_t is used to refer to such probabilities as they arise in the training environment, and the notation P_n is used to refer to these probabilities as they are exhibited in the behavior of the network. For O_j given I_i, the expression $\log \frac{P_t(O_j|I_i)}{P_n(O_j|I_i)}$ will have value 0 when $P_t(O_j|I_i) = P_n(O_j|I_i)$, and the sum will be 0 when the distribution of P_t and P_n exactly match. Whenever the distributions do not match the sum will be greater than 0 and so the minimum of this measure corresponds to probility matching.

What Ackley et al. (1985) showed was that a very simple learning procedure can be used to train a Boltzmann machine network to minimize this measure. The network is set up with some connectivity structure, typically consisting of a set of input units, a set of output units, and some hidden units. The network may not be fully connected but wherever there is a connection from one unit to another, there should be a return connection (initialization to the same value is not necessary since the learning algorithm is naturally symmetrizing). The training then proceeds in a series of training trials, structured as follows:

(1) Choose one of the input patterns I_i. Choose an output pattern O_j to pair with it with probability $P_t(O_j|I_i)$. (The thought is that such a sampling of cases would arise naturally from exposure to examples spontaneously generated by a real training environment that conformed to these distributions).

(2) "Clamp" the activations of the input and output units to the values specified in the input and output patterns, and let the network run according to the protocol previously described until equilibrium is reached at a finite temperature (e.g., T=1), then continue running while computing the co-products of the activations of units at equilibrium. Sample the state of the network periodically, n times, and adjust each connection weight according to the following simple, Hebbian, rule:

$$\delta w_{ij} = \epsilon a_i^+ a_j^+ \tag{26}$$

The superscripts on the activations indicate that they come from the "clamped" or "plus" phase of processing.

(3) Run the network again, this time clamping only the input units to the values specified in the input pattern til equilibrium is reached at the same temperature as before, then sample the state of the network n times, this time adjusting the weights according to the oppositive of the Hebbian rule:

$$\delta w_{ij} = -\epsilon a_i a_j \tag{27}$$

The superscripts now indicate the activations come from the "unclamped" or "minus" phase of processing.

The actual order of the phases is irrelavant, and indeed the precedure can be thought of in the other order: This is more like the standard supervised learning paradigm, in which one imagines an input is given, and the network processes it, "anticipating" the output, before the output is then provided by the environment, which the network then processes together with the input already recieved. This makes the procedure relatively naturalistic, although there is a sign reversal in the weight update rule between phases which may seem biologically implausible.

The importance of this procedure is that it provides a solution to the problem of how a system might learn an accurate probabilistic model of an environment. Simulations demonstrating that such networks could learn such models were not actually attempted by Ackley et al. (1985), but such simulations have been done by Movellan and McClelland (1993), both with Boltzmann machines and with symmetric diffusion networks. Their networks learned a bi-directional word translation problem, translating from "English" to "Spanish", with a corpus in which several words in each language had two acceptable translations in the other language. The experience was that the symmetric diffusion networks actually worked better in most cases, but this was not systematically studied.

It should be noted that such a network, once trained, would very likely be capable of producing the most likely output given an input, under a simulated annealing schedule. While to my knowledge this has not been tried, experience with simulated annealing in hand-wired networks suggests that it ought to work as advertised here as well. Interestingly, however, the model, though trained to probability match, could in practice be used to produce the "optimal", or most likely, response.

The model can also be generalized to the unsupervised case, in which the input pattern is left out, and the task is simply to learn to model the distribution of events in the environment, alternatively "hallucinating" such events from no input at all (in the negative phase), and "experiencing" events sampled from some environment in the positive phase. For example, one might imagine training a multi-layer Boltzmann machine with the four-letter words of English. In learning these words, the network should develop a probabilistic model of the corpus, perhaps mimicking the behavior of the interactive activation model.

To my knowledge, application of the Boltzmann machine to such a large unsupervised problem has never been attempted. The conventional wisdom is that Boltzmann machines do not work well in this unsupervised mode. However, Movellan and McClelland (1993) were able to successfully train a Boltzmann machine in a small task with some of the characteristics of the word perception task. The task in question was called "Completion Exclusive-Or" (CXOR). The network consisted of no input units, nine hidden units, and three output units, and the goal was to see if the network could learn to settle only to one of the four patterns "000", "011", "101", and "110". Note that each bit is the XOR of the other two bits. The analogy to words may be clearer with the analogy that MEN, MUD, RED, and RUN are all words, but REN, RUD, MED, and MUN are all nonwords.

In this case, and in general, a "missing" letter in a word is not predictable from a subset of the other letters, but depends on all of then. Twenty randomly initialixed networks were trained by alternating positive and negative phases, as above, until a criterial value of the unconditional IG measure

$$IG = \Sigma_j P_t(O_j) \log \frac{P_t(O_j}{P_n(O_j)} \tag{28}$$

was reached (note that about 100 trials are needed to get an adequate sample value for this measure, due to the stochastic nature of the network's behavior). Once trained, the networks were tested in several different ways. First, their free-running behavior was assessed, allowing them simply to produce outputs without any constraint. In this test every network always generated one of the legal patterns, with no errors. The nets were then tested with one of the input units clamped. In this case, the remaining two bits were filled in with one of the two legal completions (e.g., "1–" was completed as either "110" or "101") 97.5% of the time. Finally, the nets were tested with all possible pairs of inputs. In this case, the remaining bit was completed correctly 96.7% of the time. While some networks had biases toward particular completions, the overall distribution of responses in the test was very close indeed to the conditional probability structure of the domain.

Generative models approaches

While the Movellan and McClelland (1993) simulation was successful, it must be admitted that it did require a great deal of computation; settling in Boltzmann machine requires roughly 100 cycles, and from about 10,000 to 100,000 settles in each of the plus and minus phases were needed to learn the XOR problem to the level just reported. Whether the number of settles needed would increase dramatically for larger problems is not known; but presumably, a substantial number of exposures to each training pattern in the plus phase would be required. Pessimism about such matters has led Hinton and other researchers interested in capturing probabilistic structure in connectionist networks to consider other, more constrained approaches. Two such approaches are represented in Hinton and Zemel (1994; Zemel, 1993) and Dayan et al. (1995). Both of these approaches take seriously the idea that the goal of learning is to discover a generative model that accounts for the ensemble of inputs experienced in the environment.

The approach taken in Zemel (1993; Hinton & Zemel, 1994), which is the one we will discuss here, lays a great deal of stress on the use of constraints over and above those imposed by the training data to find the appropriate generative model. The essential idea is that an unconstrained search for such a model is bound to fail in interestingly complex cases becuase the search space is just too large; additional constraints, perhaps imposed over evolutionary time through natural selection, are necessary to guide the network to a useful domain model. Their main focus was on constraints imposed on internal representations. The goal of the computation is thought of as maximizing the probability of the representations, and the constraints are priors imposed on these representations, which then lead the network to tend to favor particular kinds of representations.

As a concrete example, we will consider one of the cases in Zemel (1993), involving an environment consisting of a set of displays on a 5x5 grid. The displays were actually generated by specifying a subset of the possible horizontal or vertical lines, and activating the units corresponding to the elements of the specified lines. The goal was to have a connectionist network discover, a model corresponding to the procedure whereby the displays were actually generated: That is, they wanted the network to assign a single hidden unit to represent each line that might be in the display; to assign input weights and biases to each unit so that over an ensemble of training cases, it is activated with a probability corresponding to the probability of occurrence of the line

for which it stands in the training data; and to assign weights from these units to output units to capture the relation between the underlying "line" variables and the training data.

In Bayesian terms, the goal of learning can be taken to be to maximize the probability of the representation-to-output weights, and the set of internal representations, contingent on the set of examples in the environment. It follows from Bayes law that

$$p(W, \{A\}|\{X\}) = \frac{1}{p(\{X\})} p(\{X\}|W, \{A\}) p(\{A\}|W) p(W) \ . \tag{29}$$

Defining minus the log of $p(W, \{A\}|\{X\})$ as the "Cost" (C) associated with it, the goal becomes that of minimizing C, which is just

$$C = -\log p(\{X\}|W, \{A\}) - \log p(\{A\}) - \log p(W) \tag{30}$$

(The term $\log(p\{X\})$ has been dropped because it is fixed, and the term $\log p(\{A\}|W)$ has been simplified to $\log p(\{A\})$ because the probability distribution of the activations is assumed independent of the weights).

As previously noted, this approach makes explicit the importance of having a set of priors from which $p(\{X\}|W, \{A\})$, $p(\{A\})$, and $p(W)$ can be calculated. Various assumptions can be used to constrain the definitions of these terms. For example, if the outputs are thought of as purturbed by independent Gaussian noise, the first term becomes the sum squared error. The activations of the representations might be assumed to be independent, so that the probability of a particular pattern becomes the product of the separate probabilities of their activations; further, the representations might be expected to be sparse, so that that each unit would be expected to have a relatively low probability of being active in any given pattern. In this case the second term becomes sum of deviations of the activations of the individual units from their expacted values. An alternative, considered in other cases, is that an input is underlyingly caused by a set of causes each chosen from a set of mutually exclusive possibilities (The set of all possible four-letter strings has this structure). Assumptions about the distribution of weights can also be added, though Zemel (1993) did not do this. With a set of such assumptions in place, one can then use standard connectionist training procedures to minimize the sum of the cost terms. Zemel (1993) showed that this approach can indeed allow a network to assign hidden units to particular input lines. While a conventional "encoder" network trained with back-propagation did learn a set of internal representations that allowed reconstruction of the input patterns, it did not discover the underlying componential structure of the problem.

Zemel (1993; Hinton & Zemel, 1994) studied this and several other relatively transparent problems, and showed that in each case, when the specified a set of priors appropriate to the general characteristics of the domain, the minimization of the relevant cost functions led to representations that transparantly captured the strcuture. In one case, they explicitly chose an example that could not be exactly captured within the set of specified priors, the network nevertheless found a good approximation within this model framework. The priors thus strongly constrained the behavior of the networks, and led them to appropriate solutions when they were consistent with the training examples.

Conclusion: Capturing Irrational as well as Rational Behavior

Our examination of processing in connectionist networks showed how they can perform optimal inference in many cases, and our examination of learning has touched on a couple of cases where

learning in a network can be conceived of as maximizing the probability of something — perhaps the training data, perhaps the internal representations of the training data — and thus of performing some sort of optimization. In other cases the networks perform probability-matching, but this too can be viewed as a kind of optimization (e.g., of the match between the behavior of the network and the structure of its environment). In these cases these models appear to be quite "rational", and indeed their performance can be very good. I do not mean to suggest that they are fully adequate by any means, but at least it should be clear that their development and assessment takes place within a context where the consideration of optimality and optimization plays a crucial role.

However, no model that is ultimately intended to apply to human behavior can really be said to be fully adequate if it only accounts for apparent instances of rational or optimal performance. Many instances exist where human behavior is far from rational or optimal, and a good model of human information processing and learning must ultimately provide a means of addressing such cases, too. In this light, may be interesting that a very basic and simple connectionist learning rule—a rule that has some very interesting properties from the point of view of probabilistic inferences—can also exhibit serious failures of optimality in certain cases.

The rule in question is the Hebbian learning rule, as instantiated in self-organizing, competitive learning networks (Rumelhart & Zipser, 1985; von der Malsburg, 1973; Grossberg, 1976). A competitive learning network consists of an input layer and a representation layer. There are weights from units on the input layer to units on the representation layer. Input patterns arise on the input layer, and these, in turn, give rise to inputs to the units on the representation layer, modulated by the weights. The unit receiving the strongest net input wins, and then the weight to winning unit i from input unit j is adjusted, according to the following simple rule:

$$\delta w_{ij} = \epsilon(a_j - w_{ij}) \tag{31}$$

The rule thus tends to adjust the weights of the winning unit "toward" the pattern of activation on the input. In the case where the input units have binary activations, the weights to a representation unit from an input unit can come to approximate the probability that an input unit is active given that the representation unit is active. Such quantities are, as we have seen, extremely useful in Bayesian models of perceptual inference—for example, these are the very sorts of quantities we stipulated in constructing the optimal stochastic interactive activation model of word perception. Furthermore this model is one that is highly biologically plausible and might very easily be exploited in the brain. If conditions could be arranged so that appropriate representation units were somehow activated under the right circumstances, a multi-later competitive learning network might actually learn to be the interactive activation model of letter perception or something very much like it Grossberg (1976, 1987).

Yet competitive learning models have a serious flaw, which is that they tend to reify whatever model they have of an environment. Once a set of units has learned to partition a set of inputs in a particular way, it tends to persist in doing so, and indeed the Hebbian nature of the learning rule tends to reinforce, rather than weaken such tendencies, even if they may be inappropriate. As an example, McClelland and Thomas (in preparation) applied a competitive learning network to the problem of discovering the set of perceptual categories in a particular input domain, to capture aspects of how a child must discover the set of phonological categories used in the words in its language, and how prior experience in one language environment may actually block successful discovery of the structure in another. They constructed situations in which a network that would learn the categories of a particular environment if it was exposed to that environment from birth, would nevertheless fail to learn these categories if it had previously learned the structure of a

different environment, much as Japanese adults fail to learn to distinguish the English phonemes /r/ and /l/ after growing up in a culture that has only a single /r/-like phoneme. The pathological feature of such networks is that exposure to Enlish actually leads them to reinforce their non-optimal tendency to treat these phonemes as the same. Each time an /r/ or an /l/ is presented, the common representation is activated, and the tendency of the particular /r/ or /l/ input to activate this representation on subsequent equations is reinforced. This behavior is highly non-optimal, in the sense that it produces a persistence of a pre-established response tendency that is highly undesireable in the new environment, and Japanese natives who move to English speaking environments as adults will readily attest. This behavior is also suggestive of other sorts of self-reinforcing reactions to inputs, such as sexist or racist stereotypes. If an input evokes a reaction, the probability of the reaction is increased, whether or not the reaction is appropriate. If experiences as structured so that reactions are generally appropriate, then this tendency to increase the strength of evoked reactions can appear rational or optimal; but in other cases, they can have highly irrational and counterproductive.

The pathological aspects of competitive learning networks may or may not turn out, on further scrutiny, to provide useful insights into cases of non-optimality such as failures to acquire crucial distinctions in non-native languages, or the self-mintenance of prejudicial reactions to individuals of a certain class or race. The point here, as in earlier parts of this article, is not to suggest that the problem has been solved, but only to suggest that connectionist researchers are avid participants in the search for answers to questions about the basis and boundaries of human rationality in the face of an uncertain world, and that connectionist models provide a fertile framework in which to undertake explorations in search of these answers.

References

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

Bülthoff, H. H., & Yuille, A. L. (1996). A Bayesian framework for the integration of visual modules. In T. Inui, & J. L. McClelland (Eds.), *Attention and performance XVI: Information integration in perception and communication* (pp. 49–70). Cambridge, MA: MIT Press.

Dayan, P., Hinton, G. E., Neal, R., & Zemel, R. S. (1995). The Helmholtz Machine. *Neural Computation*, *7*, 889–904.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, *PAMI-6*, 721–741.

Green, D. M., & Swets, J. A. (1966). *Signal detection theory and psychophysics.* New York: Wiley.

Grossberg, S. (1976). On the development of feature detectors in the visual cortex with applications to learning and reaction-diffusion systems. *Biologcial Cybernetics*, *21*, 145–159.

Grossberg, S. (1978). A theory of visual coding, memory, and development. In E. L. J. Leeuwenberg, & H. F. J. M. Buffart (Eds.), *Formal theories of visual perception.* New York: John Wiley & Sons.

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

Herrnstein, R. J. (1970). On the law of effect. *Journal of the Experimental Analysis of Behavior*, *13*, 243–266.

Hinton, G. E., & Sejnowski, T. J. (1983, June). Optimal perceptual inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* Washington, DC.

Hinton, G. E., & Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. In D. E. Rumelhart, & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition*, Vol. 1 (Chap. 7, pp. 282–317). Cambridge, MA: MIT Press.

Hinton, G. E., & Zemel, R. S. (1994). Autoencoders, minimum description length, and Helmholtz free energy. In J. Cowan, G. Tesauro, & J. Alspector (Eds.), *Advances in neural information processing systems* (pp. 3–10). San Mateo: Morgan Kaufman.

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

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*, 671–680.

MacKay, D. J. (1992a). Bayesian interpolation. *Neural Computation*, *4*, 415–447.

MacKay, D. J. (1992b). A practical Bayesian framework for backpropagation networks. *Neural Computation*, *4*, 448–472.

MacKay, D. J. C. (1997). *Information theory, probability, and neural networks.* Cambridge University, Cambridge, England: Draft available at http://wol.ra.phy.cam.ac.uk/mackay/itprnn/#book.

McClelland, J. L. (1985). Putting knowledge in its place: A scheme for programming parallel processing structures on the fly. *Cognitive Science*, *9*, 113–146.

McClelland, J. L. (1991). Stochastic interactive activation and the effect of context on perception. *Cognitive Psychology*, *23*, 1–44.

McClelland, J. L., & Elman, J. L. (1986). The TRACE model of speech perception. *Cognitive Psychology*, *18*, 1–86.

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

McClelland, J. L., & Rumelhart, D. E. (1988). *Explorations in parallel distributed processing: A handbook of models, programs, and exercises.* Cambridge, MA: MIT Press.

McClelland, J. L., & Thomas, A. (in preparation). *Dynamic stability and adaptive intervention: Consequences of Hebbian learning?* manuscript in preparation.

McNaughton, B. L. (1989). Neuronal mechanisms for spatial computation and information storage. In L. Nadel, L. A. Cooper, P. Culicover, & R. M. Harnish (Eds.), *Neural connections, mental computations* (pp. 285–350). Cambridge, MA: MIT Press.

Movellan, J. R., & McClelland, J. L. (1993). Learning continuous probability distributions with symmetric diffusion networks. *Cognitive Science*, *17*, 463–496.

Movellan, J. R., & McClelland, J. L. (1995). *Stochastic interactive activation, Morton's Law, and optimal pattern recognition* (Technical Report PDP.CNS.95.4). Pittsburgh, PA 15213: Department of Psychology, Carnegie Mellon University.

Rumelhart, D. E. (1977). Toward an interactive model of reading. In S. Dornic (Ed.), *Attention and performance VI.* Hillsdale, NJ: Erlbaum.

Rumelhart, D. E., Durbin, R., Golden, R., & Chauvin, Y. (1995). Backpropagation: The basic theory. In Y. Chauvin, & D. E. Rumelhart (Eds.), *Back-propagation: Theory, architectures, and applications* (pp. 1–34). Hillsdale, NJ: Erlbaum.

Rumelhart, D. E., & McClelland, J. L. (1982). An interactive activation model of context effects in letter perception: Part 2. The contextual enhancement effect and some tests and extensions of the model. *Psychological Review*, *89*, 60–94.

Rumelhart, D. E., & Siple, P. (1974). Process of recognizing tachistoscopically presented words. *Psychological Review*, *81*, 99–117.

Rumelhart, D. E., Smolensky, P., McClelland, J. L., & Hinton, G. E. (1986). Schemata and sequential thought processes in PDP models. In J. L. McClelland, D. E. Rumelhart, & the

PDP research group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition*, Vol. 2 (Chap. 14, pp. 7–57). Cambridge, MA: MIT Press.

Rumelhart, D. E., & Zipser, D. (1985). Feature discovery by competitive learning. *Cognitive Science*, *9*, 75–112.

Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhart, & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition*, Vol. 1 (Chap. 6, pp. 194–281). Cambridge, MA: MIT Press.

Staddon, J. E. R. (1982). Behavioral competeition, contrast, and matching. In M. L. Commons, & R. J. Herrnstein (Eds.), *Quantitative analysis of operant behavior: Matching and maximizing accounts, vol 2.* (pp. 243–261). Cambridge, MA: Ballinger.

von der Malsburg, C. (1973). Self-organizing of orientation sensitive cells in the striate cortex. *Kybernetik*, *14*, 85–100.

Zemel, R. S. (1993). *A minimum description length framework for unsupervised learning.* PhD thesis, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada.