Physica 22D (1986) 247–259
North-Holland, Amsterdam

A SELF-OPTIMIZING, NONSYMMETRICAL NEURAL NET FOR CONTENT ADDRESSABLE MEMORY AND PATTERN RECOGNITION

Alan LAPEDES and Robert FARBER
Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

A natural, collective neural model for Content Addressable Memory (CAM) and pattern recognition is described. The model uses nonsymmetrical, bounded synaptic connection matrices and continuous valued neurons. The problem of specifying a synaptic connection matrix suitable for CAM is formulated as an optimization problem, and recent techniques of Hopfield are used to perform the optimization. This treatment naturally leads to two interacting neural nets. The first net is a symmetrically connected net (master net) containing information about the desired fixed points or memory vectors. The second net is, in general, a nonsymmetric net (slave net), whose synapse values are determined by the master net, and is the net that actually performs the CAM task. The two nets acting together are an example of neural self-organization. Many advantages of this master/slave approach are described, one of which is that nonsymmetric synaptic matrices offer a greater potential for relating formal neural modeling to neurophysiology. In addition, it seems that this approach offers advantages in application to pattern recognition problems due to the new ability to sculpt basins of attraction. The simple structure of the master net connections indicates that this approach presents no additional problems in reduction to hardware when compared to single net implementations.

1. Introduction

There is a long history [1] of attempts to model certain high level cognitive functions such as Content Addressable Memory (associative recall), and pattern recognition, by the collective properties [2] of formal neural networks. Fixed points of the neural dynamics play a key role. Essential to the mathematics of most of these models is the physiologically unreasonable assumption that the synaptic connection matrix is a symmetric matrix.

In this paper we consider the general problem of finding bounded synaptic connection matrices, T_{ij}, that produce fixed points of neural dynamics at specified memory vectors, without imposing symmetry. For the CAM problem these fixed points correspond to the memories one is attempting to recall, while the associated basins of attraction correspond to the partial information used to evoke the full memory. In pattern recognition problems, the fixed points represent the objects to be identified, while the basins of attraction represent partial information, or distortions of the object.

In general, a given arbitrary T_{ij} will not produce fixed-point behavior (of the dynamics described below) and our problem is to find those that do. Even if one limited the elements of an $N \times N$ T_{ij} matrix to just two values, ± 1 (representing excitation and inhibition) then a brute force search over all 2^{N^2} possible matrices would clearly be impractical (also inelegant!). In the following, we show how this exponential search problem can be formulated as an optimization problem that is solved by another neural net consisting of N^2 neurons. Our technique is related to recent work of Hopfield [3], where he shows how a good optimum to the NP-complete Traveling Salesman Problem (which on the face of it requires $\mathcal{O}(N!)$ searches for the N city problem) can be found by the collective action of a neural net consisting of N^2 neurons. In a sense, we take the "optimum finding" properties of a neural net and instead of applying it to the external world (e.g. the Traveling Salesman Problem), apply it to optimizing a second neural net. Our formulation thus naturally involves two nets – the first net optimizes, or "programs", the

0167-2789/86/$03.50 © Elsevier Science Publishers B.V.
(North-Holland Physics Publishing Division)

second net and the second net then does the job. We refer to this relation between the two nets as a master/slave relationship. This optimization method of neural programming provides for merging two basins of attraction together (and separating them again), weighting certain components of a fixed point so that it attracts more strongly ("sculpting" the basins), and the division of neurons into inhibitory and excitatory subpopulations. It also offers a higher potential for relating formal neural net models to neurophysiology by virtue of the nonsymmetric T_{ij}'s and the ability to separate neurons into excitatory/inhibitory neuron classes.

2. Theory

We shall consider a formal neural network that encapsulates in a schematic form certain properties of real neurons [4]. We assume that on a suitable timescale a neuron's output can be described as a firing rate of action potentials and we concentrate on the rate of firing, $g(U_i)$, not individual potentials. The firing rate of the ith neuron, $g(U_i)$, is a function of the membrane potential, U_i, across the neural body and it rises from 0 to a maximum of $1/r_c$, where r_c is the absolute refractory period. It is thus a sigmoid shaped curve (fig. 1) which for concreteness we take to be

$$g(U_i) = \frac{1}{2r_c}(1 + \tanh \beta U_i) \quad (\beta \text{ determines the slope}). \qquad (1)$$

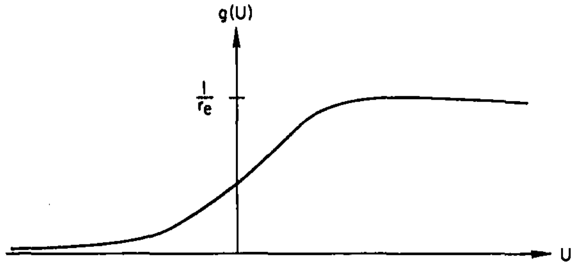

Fig. 1. Sigmoidal firing rate curve as a function of membrane potential.

A neuron is assumed to sum up the impulses coming into it from other neurons in the net, but since the membrane is "leaky" a given potential will decay exponentially in time with a time constant μ. We also assume that the postsynaptic effect on neuron "i" from presynaptic potentials due to other neurons "j" is described by a synaptic connection matrix, T_{ij}, with positive or negative elements describing excitation or inhibition. Our model for neural net dynamics incorporating these assumptions is therefore

$$U_i(t) = \frac{1}{\mu}\int_0^t \left[e^{-(t-t')/\mu} \sum_j T_{ij} g(U_j(t')) + I_i \right] dt' + U_i(0), \qquad (2)$$

or, equivalently,

$$\frac{1}{\mu} U_i + \dot{U}_i = \sum_j T_{ij} g(U_j) + I_i, \qquad (2a)$$

where I_i represents the effect of an external current into neuron "i". These equations also describe an electrical circuit involving operational amplifiers and resistors [5] and can be reduced to a parallel hardware implementation should that prove convenient for practical applications.

Hopfield [4] considered the situation where $T_{ij} = T_{ji}$ and showed that the following is a Lyapunov function for eq. (2a):

$$E = -\tfrac{1}{2}\sum_{ij} T_{ij} V_i V_j - \sum_i I_i V_i + \sum_i \int_0^{V_i} g^{-1}(x)\,dx, \quad V_i = g(U_i), \qquad (3)$$

i.e. (2a) and (3) implies that $dE/dt \leqq 0$,

$$\frac{dE}{dt} = -\tfrac{1}{2}\sum_i g'(U_i) \cdot \dot{U}_i^2 \quad (g' \text{ is positive semi-definite}), \qquad (4)$$

where "prime" represents differentiation with respect to the argument and "dot" denotes time differentiation. Eq. (2a) will therefore decrease E until a minimum of E is reached (E is bounded), at which point from (4) we have $\dot{U}_i = 0 \Rightarrow U_i =$ constant, that is, a fixed-point is obtained. In the high gain limit (large β) the integral makes a negligible contribution to E and the algorithm may be thought of as finding a minima of E_1, where

$$E_1 = -\tfrac{1}{2}\sum_{ij} T_{ij}V_iV_j - \sum_i I_iV_i.$$

A possible choice [3] for E_1 (with a suitable bias chosen for g) is

$$E_1 = -\tfrac{1}{2}\sum_{s=1} \left(V^{(s)}\cdot V\right)^2, \tag{5}$$

where $V^{(s)}$, $s = \{1,2,\dots,m\}$ is a particular choice of m memory states. This choice implies that T_{ij} is

$$T_{ij} = \sum_{s=1}^{m} V_i^{(s)}V_j^{(s)} \tag{6}$$

and thus is equivalent to Hopfield's original formulation [1] of the CAM problem. T_{ij} is obviously symmetric and was determined by the ad hoc choice of E_1. Although this choice works in the CAM problem, subject to limitations, it was chosen with "cleverness aforethought" and not by derivation, and is as we will see, not a general solution to the problem of inserting desired fixed points into the neural dynamics.

We now consider the general problem of finding a T_{ij} that produces fixed points at desired memory states $V_i^{(s)}$, $s = \{1,2,\dots,m\}$. Our concern is more that $V_i = g(U_i)$ should evolve to desired values near 0 or $1/r_e$, and not so much that U_i evolves to a specific value. Due to the form of $g(U_i)$ (fig. 1), there are many U_i values associated with $V_i = g(U_i) \approx 0$ or $1/r_e$, and thus we can make significant "errors" in the exact value of U_i without affecting the results. To focus attention more closely on $g(U_i)$ we now reformulate equation (2a) slightly.

Let

$$V_i(t) = \frac{1}{\mu}\int_0^t e^{-(t-t')/\mu} g(U_i(t'))\,dt' \tag{7}$$

such that

$$V_i + \mu\dot{V}_i = g(U_i(t)). \tag{8}$$

Eq. (2a) becomes

$$V_i + \mu\dot{V}_i = g\Big(\sum_j T_{ij}V_j\Big) \quad \cdots (I_i = 0 \text{ for simplicity}). \tag{9}$$

When $\dot{V}_i = 0$, that is, at a fixed point of (9) we have from (8) that $V_i = g(U_i)$.

Let us now express the requirement that a particular set of m memory vectors $V_i^{(s)}$ for $s = 1$, m are fixed points of (9). Each vector must satisfy (from eq. (9))

$$V_i^{(s)} = g\Big(\sum_j T_{ij}V_j^{(s)}\Big), \tag{10}$$

or, equivalently,

$$g^{-1}\big(V_i^{(s)}\big) = \sum_j T_{ij}V_j^{(s)}, \quad \text{for each } s \text{ and } i. \tag{10a}$$

We may rewrite (10a) as

$$E_1 = \sum_{is}\Big[g^{-1}\big(V_i^{(s)}\big) - \sum_j T_{ij}V_j^{(s)}\Big]^2 \tag{11}$$

with the requirement $E_1 = 0$.

For simplicity let us first consider the case where the synapse values, T_{ij}, are bounded, say between -1 and 1, thus allowing for bounded amounts of inhibition and excitation. Parameterizing T_{ij} as

$$\begin{aligned} &T_{ij} = 2\cdot S_{ij} - 1, \quad \text{where } S_{ij} = g\big(U_{ij}\big), \\ &U_{ij} = (\text{another independent variable}) \end{aligned} \tag{12}$$

and where $g(U_{ij})$ is another sigmoidal curve between 0 and 1 achieves our purpose. No other restrictions, such as symmetry, are placed on T_{ij}. Note that $V_j^{(s)}$ and $g^{-1}(V_j^{(s)})$ are known numbers and that our condition, (11), that eq. (9) have fixed points at the known values $V_i^{(s)}$, becomes a condition that E_1 (below) is at a global minimum

$$E_1 = \sum_{is}\left[g^{-1}(V_i^{(s)}) - \sum_j \left(2g(U_{ij}) - 1\right)V_j^{(s)}\right]^2 \tag{13}$$

$$= \sum_{ijkl} - T_{ijkl}g(U_{ij})g(U_{kl}) - \sum_{ij} I_{ij}g(U_{ij}) + \text{constant}. \tag{13a}$$

The T_{ijkl} and I_{ij} in (13a) are determined by expanding the squares in (13) and are clearly known constants depending on the chosen memories $V_i^{(s)}$. Note that $T_{ijkl} = T_{klij}$. Also note that T_{ijkl} is of a very simple form, i.e.

$$T_{ijkl} = -4\sum_s \delta_{ki}V_j^{(s)}V_l^{(s)}, \tag{13b}$$

$$I_{ij} = +4V_j\left[g^{-1}(V_i) + \sum_l V_l\right]. \tag{13c}$$

Adding a suitable integral places our minimization problem (13a) in exactly the form of eq. (3) if $U_i \to U_{ij}$, $T_{ij} \to T_{ijkl}$ and $I_i \to I_{ij}$. In other words, we can now determine $T_{ij} = 2g(U_{ij}) - 1$ by solving the new set of neural net equations similar to (2a),

$$\frac{1}{\mu}U_{ij} + \dot{U}_{ij} = \sum_{kl} T_{ijkl}g(U_{kl}) + I_{ij}. \tag{14}$$

U_{ij} will evolve under (14) to a fixed point, $\overline{U}_{ij}$ (because $T_{ijkl} = T_{klij}$), and we then take

$$T_{ij} = 2g(\overline{U}_{ij}) - 1 \tag{15}$$

as the solution to our problem. As a check we insert the calculated T_{ij} back into eq. (9) and verify by numerical simulation that the desired fixed-points, $V_i^{(s)}$, are present. Results on the numerical integration of (14) are presented later. We briefly note here, however, that a good minimum of E_1 (eq. (11)) is generally found, that T_{ij} is in general non-symmetric, (although T_{ijkl} is symmetric), and that the check of the desired fixed-point generally succeeds. Also, as was emphasized to us by J. Denker and E. Mjolsness [6], the simplicity of the form of T_{ijkl}, eq. (13b), reduces the double sum in eq. (14) to N operations and eq. (14) can be rewritten in a local form by use of a linear interneuron. This is important for practical hardware implementations of this algorithm.

3. Discussion

Before proceeding to the simulation results, we wish to discuss the interpretation of the procedure outlined above. The equation set (14) describes a set of N^2 neurons, indexed by U_{ij} ($i, j \in [1, N]$), called the master net, that are symmetrically connected ($T_{ijkl} = T_{klij}$) and subject to an external current, I_{ij}. This master net of neurons evolves to its own fixed-point, $\overline{U}_{ij}$, at which point the master neurons are firing at sustained rates $g(\overline{U}_{ij})$. These master neurons are connected to a second net of N neurons, called the slave net, and the master neurons modulate the synaptic connections T_{ij} of the slave net such that $T_{ij} = 2g(\overline{U}_{ij}) - 1$ (see fig. 2). Thus each sustained neuronal firing rate in the master net determines a synaptic connection between two neurons in the slave net. These slave net connections need not be symmetrical. Simulations show that if the slave net is now evolved with the

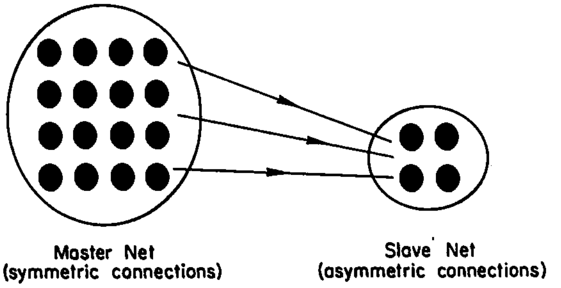

Fig. 2. Symmetric master net modulating synaptic connections of nonsymmetric slave net.

synaptic weights given by $T_{ij} = 2g(\bar{U}_{ij}) - 1$, then the slave net evolves to one of the desired memory values.

This model suggests that if certain cognitive tasks are performed by a neurophysiological implementation of this model for CAM, then one might expect to find a larger net of $\approx N^2$ reciprocally connected inhibitory neurons modulating the synaptic connections of a smaller network of $\approx N$ nonsymmetrically connected neurons. We pursue this suggestion elsewhere. Also we note that if the T_{ij} values were given by ± 1 instead of a continuous interpolation between these values, then this formalism could be loosely interpreted as a search by the master net over the 2^{N^2} configurations of the T_{ij}, much as Hopfield's [3] neural net for the N city Traveling Salesman Problem can be thought of as searching over $\mathcal{O}(N!)$ tours (of course neither net actually performs this search explicitly).

The basic formalism can also be extended in various ways. One such extension is to modify the sum of squares form for E_1 (eq. (13)) by putting constants $C_i^{(s)}$ in front of each term so the E_1 becomes

$$E_1 = \sum_{is} C_i^{(s)} \left[g^{-1}(V_i^{(s)}) - \sum_j \left(2g(U_{ij}) - 1\right) V_j^{(s)} \right]^2 . \tag{16}$$

For simplicity, first consider the coefficients $C_i^{(s)}$ to depend only on (s), i.e. just have a different C for each s. This weights the fixed points that have higher C's more than the others, and has the effect that by adjusting $C^{(s)}$ one can make basins of attraction either fuse, or pull apart, from each other. The more complicated case, where the coefficients depend on i as well as s, allow one to weight certain *components* of a fixed point more than other components in the same, or different, fixed points. In attempting to use CAM's in pattern recognition it is obviously advantageous to weight certain features of a pattern more than others, and thus if the neuronal firing rates are considered to be outputs of feature detectors, then varying $C_i^{(s)}$ helps to accomplish this objective. This weighting is impossible in implementations which use Hamming distance as a measure of the basins of attraction [1].

Finally, we remark that for unsaturated T_{ij} ([number of memories m] $\ll N$) the nature of the eqs. (14) tend to drive $g(\bar{U}_{ij})$ to the limits of 0 or 1 and hence $T_{ij} = 2g(\bar{U}_{ij}) - 1$ to ± 1. We refer to this as the "black/white" case. (Of course by scaling T_{ij} by a constant A, the limits can be changed to $\pm A$.) A "grey level" discretization of T_{ij} between the "black/white" values of $\pm A$ can be achieved by reparameterizing T_{ij} as

$$T_{ij} = A\left\{ (2/2^{\bar{K}} - 1) \cdot \sum_{k=0}^{\bar{K}} 2^k g\left(U_{ij}^{(k)}\right) - 1 \right\}, \tag{17}$$

where $\bar{K}$ is the number of "grey levels". Of course there are now $\bar{K}N^2$ neurons in the master net which would complicate the simulation slightly.

4. Simulations

A complete numerical investigation of the system was not attempted. However, a sufficient number of runs were completed to verify the credibility of the formalism. The reasons for not collecting complete statistics at this stage are twofold. First, it is necessary to integrate $N^2 + N$ differential equations (for the U_{ij} and the U_i) for an N-neuron slave net. This takes a considerable amount of computer time. (Most of the time, however, is spend in mapping the basins of attraction of the N-neuron system. The computation for the N^2-neuron system occurs only once and is relatively quick.) Secondly, there are a large number of parameters, $C_i^{(s)}$, that affect the system, and unless the effects of these are systematically mapped out as well, then statistics degenerate to a test of skill in parameter twiddling. In short, the system behavior is very rich, and we choose to verify only a few of its properties.

The numerical integration routine used was a simple Euler method. For the master net, where a

Lyapunov function E is defined, we scaled $\mathrm{d}t$ by 1.1 if E decreased on the previous time step, and scaled $\mathrm{d}t$ by 0.5 if E increased (and performed that time step again.) This modification was suggested to us by E. Baum [7] and dramatically accelerates convergence to a fixed point. This is not possible for the nonsymmetrically connected slave net and the majority of computer time was used in mapping the basins of attraction in the slave net. It is quite probable that a cleverer algorithm, or a coarser map of the basins of attraction can greatly increase simulation speed when mapping basins.

Initial conditions for the master net were always chosen to be $U_{ij} = 0$, which is a state of no particular bias in any direction, with all master neurons firing at a rate midway between the minimum and maximum rates. The slave net neurons were always evolved from an initial state in which their individual firing rates were either at a maximum (1) or a minimum (0). We chose these initial values for the slave net because the initial state of the slave net in both the CAM and pattern recognition problems contains the initial information to be processed by the net. This information is assumed to be coded into a pattern of 1's and 0's. We have not investigated system performance with indefinite initial information, i.e., with initial slave net rates equal to a decimal value between 0 and 1. The membrane time constants, μ, were chosen to be 5 for both nets and the absolute refractory periods, r_e, were chosen to be 1. The units of time are arbitrary, and if it is chosen to be 1 millisecond, then these time constants are physiologically reasonable. However, these values are not critical to the results. The parameters controlling the slope of firing rate curves, $g(x)$, were both chosen to be 25. This value seemed to work acceptably well and variation by a factor of 10 did not make a significant difference.

The code developed to perform the simulations consists of a menu-driven, highly interactive front-end written in C, and a calculational-backend module written in FORTRAN. Simulations were performed on VAX780's and Crays. It is worth pointing out that machine precision has little effect on final answers, but can radically affect the time taken to converge to a final answer.

The simulations are presented below as six sets of data. Simulation I reports an attempt to insert 5 fixed points into a 10-neuron slave net. Simulation II gives a rough idea of the information packing possible by seeing how performance degrades in an attempt to insert 10 fixed points into a 10-neuron slave net. Simulation III shows how 5 fixed points may be inserted into a 5-neuron slave net by appropriately choosing the $C_i^{(s)}$ parameters (they were previously held fixed at 1.0). Simulation IV further investigates the control that the $C_i^{(s)}$ parameters have over the net by inserting 2 fixed points in a 5-neuron slave net, and then adjusting the $C_i^{(s)}$ to merge, and emerge, basins of attraction. Simulation V gives an example of how the $C_i^{(s)}$ can be used to control not only the size, but also the shape, of the basins of attractions. Finally, Simulation VI shows how additional constraints can easily be imposed on the net such that the slave net can have at least one fixed point when constrained to be made up of purely inhibitory and purely excitatory subpopulations. This division into excitatory/inhibitory subpopulations is a reasonable requirement on any model that purports to have some relation to real neurophysiology.

Simulation I (100 masters neurons, 10 slave neurons, 5 fixed-points)

This simulation consisted of 100 master neurons and 10 slave neurons. We attempted to have the net self-organize according to the equations in section 2, such that the slave net would have 5 distinct fixed points for the firing rates. These fixed-points were chosen at random as 5 distinct, 10-bit long strings of 1's and 0's representing the final values of the firing rates, $V_i = g(U_i)$, of the slave net. The $C^{(s)}(i)$ parameters were all set to 1. When the master net settled to its own fixed point we than took that computed set of values for the synaptic connections, T_{ij}, of the slave net and ran 5 runs of the slave net, with the initial values equal to the 5 putative fixed points in order to see how many of them actually were present.

Having then determined how many fixed points were present in the slave net, we then investigated their basins of attraction. This was accomplished by choosing 100, distinct, 10-bit long strings of 1's and 0's as 100 initial values for the slave net firing rates and recording the final values to which they evolved in 100 separate runs. The final rates were, with the net parameters described above, always 1 or 0, to within 0.01.

The whole process was repeated six times with six different sets of five fixed points. We thus ended up with statistics based on six attempts to insert five random fixed points into a 10-neuron slave net. These simulations are recorded below in table I as run 1 through run 6. "Strays" refer to how many final values of the slave net (computed from the 100 initial values for each run) that did not end up at one of the five putative fixed-points.

Table I
(100 master neurons, 10 slave neurons, 5 possible fixed points)

	Stability	Basin of attraction
Run 1:	3 of 5 stable	0 strays out of 100 initial values
Run 2:	2 of 5 stable	0 strays out of 100 initial values
Run 3:	4 of 5 stable	0 strays out of 100 initial values
Run 4:	3 of 5 stable	0 strays out of 100 initial values
Run 5:	2 of 5 stable	0 strays out of 100 initial values
Run 6:	2 of 5 stable	27 strays out of 100 initial values

Summary. In an attempt to insert 5 fixed-points into a 10 neuron slave net, with parameter values described above, somewhat more than half are stable with strong basins of attraction.

Simulation II (100 master neurons, 10 slave neurons, 10 fixed-points)

This simulation is similar in all respects to simulation I, except that insertion of 10 fixed points into the slave net was attempted in order to obtain an idea of the information packing that is possible. Two hundred initial values were evolved. Results for 7 runs are in table II.

Table II
(100 master neurons, 10 slave neurons, 10 fixed points)

	Stability	Basin of attraction
Run 1:	4 of 10 stable	0 strays out of 200 initial values
Run 2:	1 of 10 stable	156 strays out of 200 initial values
Run 3:	0 of 10 stable	200 strays out of 200 initial values
Run 4:	3 of 10 stable	200 strays out of 200 initial values
Run 5:	2 of 10 stable	6 strays out of 200 initial values
Run 6:	3 of 10 stable	145 strays out of 200 initial values
Run 7:	3 of 10 stable	182 strays out of 200 initial values

Summary. In an attempt to insert 10 fixed-points into a 10 neuron slave net, about 1/3 were stable with relatively small basins of attraction and additional parasitic fixed-points occurring. This degradation is similar to what happens using the symmetric T_{ij} of Hopfield [1].

In run 2 we also lowered the C parameters of the one stable fixed point by a factor of 10 and found that now 5 fixed points were stable. In run 4 we also increased the C's of the 3 stable fixed points by a factor of 10 and reran the 200 initial values of the slave net, resulting in 143 strays. This indicates that results might be significantly improved by judicious choice of the C parameters. The choice is intuitively clear: the higher the $C^{(s)}$, the stronger the associated fixed point.

Simulation III (25 master neurons, 5 slave neurons, 5 fixed-points)

A further test of information packing was performed by adjusting the $C^{(s)}$ parameters so that 5 random fixed points were inserted into a 5-neuron slave net. With all $C^{(s)}$ parameters initially set to 1.0, only 2 out of the 5 possible fixed points were stable. It was quickly found that if three $C^{(s)}$ parameters remain at 1.0 while the other two were set to 0.1 and 10.0, then all 5 fixed points were stable. Mapping of the 32 possible initial firing rates resulted in 7 strays, however, each stray

Table III
(25 master neurons, 5 slave neurons, 5 possible fixed points)

	Stability	Basin of attraction
Run 1:	5 of 5 stable	7 strays out of 32 initial values

value differed only in one bit from one of the 5 possible fixed points.

Summary. In an attempt to insert 5 fixed points into a 5-neuron slave net, all 5 could be stabilized by judicious choice of the $C^{(s)}$ parameters.

Simulation IV (25 master neurons, 5 slave neurons, 2 fixed points)

Further testing of the control over basins of attraction by the $C^{(s)}$ parameters was accomplished by inserting two fixed points into a 5-neuron slave net. The $C^{(s)}$ were then separately scaled by a factor of 10 resulting in either the first, the second or both fixed points being present in three separate runs.

Table IV
(25 master neurons, 5 slave neurons, 2 fixed-points)

	Stability	Basin of attraction
Run 1:	2 of 2 stable	0 strays out of 32 initial values
Run 2:	1 of 2 stable	0 strays out of 32 initial values
Run 3:	1 of 2 stable	0 strays out of 32 initial values

Summary. Changing the $C^{(s)}$ parameters over a range of 10 allows one to have either the first, the second or both fixed points present in the slave net. This exemplifies how one can merge or emerge basins of attraction by manipulating the $C^{(s)}$ parameters in an intuitive way: the higher the $C^{(s)}$, the stronger the associated fixed point.

Simulation V (36 master neurons, 6 slave neurons, 2 fixed points)

This simulation exemplifies the ability to sculpt basins of attraction, i.e., to deform the *shape* of the basin in distinction to the simulation above which changed only the size of the basin. Two fixed points for the slave net firing rates (1 1 0 0 0 0) and (0 0 0 1 1 0), were chosen, and we attempted to adjust the individual $C^{(s)}(i)$ values so that one component of one fixed point attracted more than other components. We arbitrarily selected the second component of the first fixed point (reading from left to right), and proceeded to adjust the

Table VA

Fixed point #1 (+ + − − − −)	Fixed point #2 (− − − + + −)
(+ + + + + +)	
	(− + + + + +)
	(+ − + + + +)
	(− − + + + +)
(+ + − − + +)	
	(− + − + + +)
	(+ − − + + +)
	(− − − + + +)
(+ + + − + +)	
	(− − + − + +)
(+ + − − + +)	
	(− − − − + +)
(+ + + + − +)	
	(− − + + − +)
(+ + − + − +)	
	(− − − + − +)
(+ + + − − +)	
(− + + − − +)	
(+ − + − − +)	
(− − + − − +)	
(+ + − − − +)	
(− + − − − +)	
(+ − − − − +)	
(− − − − − +)	
(+ + + + + −)	
	(− + + + + −)
	(+ − + + + −)
	(− − + + + −)
(+ + − + + −)	
	(− + − + + −)
	(+ − − + + −)
	(− − − + + −)
(+ + + − + −)	
	(− − + − + −)
(+ + − − + −)	
	(− − − − + −)
(+ + + + − −)	
	(− − + + − −)
(+ + − + − −)	
	(− − − + − −)
(+ + + − − −)	
(− + + − − −)	
(+ − + − − −)	
(− − + − − −)	
[+ + − − − −]	
(− + − − − −)	
(+ − − − − −)	
(16 Strays)	

$C^{(s)}(i)$ so that if this component was 1.0 in a set of initial configurations, then those configurations should evolve to the first fixed point. In a pattern recognition problem this would be equivalent to considering the initial firing rates of the slave net as output of feature detectors, and demanding that if the second feature detector is active, then those configurations should evolve to a recognized pattern represented by fixed point number one. In other words, that feature is extremely important in recognizing a distorted pattern as fixed point number one.

Initially we chose all the $C_i^{(s)} = 1.0$, computed the T_{ij} from the master net equations, and then evolved the 2^6 distinct initial configurations of the slave net firing rates. (We again assumed definite initial information with all initial slave net rates at either 1.0 or 0.0). The results are tabulated in table VA, where both fixed points are listed along with those initial configurations that turned out to be in its basin of attraction. Here, "+" denotes 1.0 and "−" denotes 0.0 in the table of initial configurations. For example, the first entry under fixed point #1 indicates that if the initial firing rates

Table VB

Fixed-point #1	Fixed-point #2	Fixed-point #1	Fixed-point #2
(+ + − − − −)	(− − − + + −)	(+ + − − − −)	(− − − + + −)
(+ + + + + +)		(+ + + + + −)	
	(− + + + + +)		(− + + + + −)
	(+ − + + + +)		(+ − + + + −)
	(− − + + + +)		(− − + + + −)
(+ + − − + +)		(+ + − + + −)	
	(− + − + + +)		(− + − + + −)
	(+ − − + + +)		(+ − − + + −)
	(− − − + + +)		[− − − + + −]
(+ + + − + +)		(+ + + − + −)	
(− + + − + +)		(− + + − + −)	
(+ − + − + +)		(+ − + − + −)	
	(− − + − + +)	(− − + − + −)	
(+ + − − + +)		(+ + − − + −)	
(− + − − + +)		(− + − − + −)	
(+ − − − + +)		(+ − − − + −)	
	(− − − − + +)		(− − − − + −)
(+ + + + − +)		(+ + + + − −)	
(− + + + − +)		(− + + + − −)	
(+ − + + − +)		(+ − + + − −)	
	(− − + + − +)		(− − + + − −)
(+ + − + − +)		(+ + − + − −)	
(− + − + − +)		(− + − + − −)	
(+ − − + − +)		(+ − − + − −)	
	(− − − + − +)		(− − − + − −)
(+ + + − − +)		(+ + + − − −)	
(− + + − − +)		(− + + − − −)	
(+ − + − − +)		(+ − + − − −)	
(− − + − − +)		(− − + − − −)	
(+ + − − − +)		[+ + − − − −]	
(− + − − − +)		(− + − − − −)	
(+ − − − − +)		(+ − − − − −)	
(− − − − − +)			

Continued next 2 columns.

for the slave net were all chosen to be 1.0, then that initial configuration evolved to the first fixed point, i.e., it evolved to (1.0 1.0 0.0 0.0 0.0 0.0). At the end of the table we note how many configurations went astray, i.e., did not evolve to a fixed point. Note that no particular bias towards the second component of fixed-point #1 is evident in table VA.

Next, we weighted the second feature of fixed-point #1 more heavily by repeating the above procedure with $C_2^{(1)} = 60$, other $C_i^{(s)} = 1.0$. The 16 strays were now pulled into fixed-point #1, however, some configurations with the second feature active were still pulled into fixed-point #2. (See table VB.)

Finally, we increased $C_2^{(1)}$ to 100 with $C_{i\neq 2}^{(1)} = 1.0$, and $C_i^{(2)} = 0.01$. The results are presented in table VC. The second column shows that, as desired, no set of features with the second feature "on" ever evolves to fixed-point #2.

Summary. Adjustment of the $C_i^{(s)}$ parameters allows one to impose new metrics of pattern simi-

Table VC

Fixed-point #1	Fixed-point #2	Fixed-point #1	Fixed-point #2
(+ + − − − −)	(− − − + + −)	(+ + − − − −)	(− − − + + −)
(+ + + + + +)		(− + + + + −)	
		(+ − + + + −)	
(− + + + + +)			(− − + + + −)
(+ − + + + +)		(+ + − + + −)	
	(− − + + + +)	(− + − + + −)	
(+ + − − + +)		(+ − − + + −)	
(− + − + + +)			(− − − + + −)
(+ − − + + +)		(+ + + − + −)	
	(− − − + + +)	(− + + − + −)	
(+ + + − + +)		(+ − + − + −)	
(− + + − + +)			(− − + − + −)
(+ − + − + +)		(+ + − − + −)	
	(− − + − + +)	(− + − − + −)	
(+ + − − + +)		(+ − − − + −)	
(− + − − + +)			(− − − − + −)
(+ − − − + +)		(+ + + + − −)	
	(− − − − + +)	(− + + + − −)	
(+ + + + − +)		(+ − + + − −)	
(− + + + − +)			(− − + + − −)
(+ − + + − +)		(+ + − + − −)	
	(− − + + − +)	(− + − + − −)	
(+ + − + − +)		(+ − − + − −)	
(− + − + − +)			(− − − + + −)
(+ − − + − +)		(+ + − + − −)	
	(− − − + − +)	(− + − + − −)	
(+ + + − − +)		(+ − − + − −)	
(− + + − − +)		(− − − + − −)	
(+ − + − − +)		(+ + + − − −)	
(− − + − − +)		(− + + − − −)	
(+ + − − − +)		(+ − + − − −)	
(− + − − − +)		(− − + − − −)	
(+ − − − − +)		(+ + − − − −)	
(− − − − − +)		(− + − − − −)	
(+ + + + + −)		(+ − − − − −)	

larly, in distinction to previous algorithms using symmetric T_{ij}'s which involve Hamming metrics [1]. In the example above, the feature detected by neuron #2 is a quite important feature, and was successfully weighted more heavily in the recognition process than other features by increasing the numerical value of $C_2^{(1)}$ relative to the others C's. Thus, as desired, no set of features with the second feature "on" ever evolves to fixed point #2. Algorithms involving the usual symmetric T_{ij} matrices use Hamming distance as a measure of pattern similarly. This implies equal weights on all features because Hamming distance is merely the total number of bits that differ in two patterns. The $C^{(s)}(i)$ parameters allow non-Hamming metrics to be created in a natural way. The ability to perform such weighting is an important feature of the present algorithm. Detailed investigation of these non-Hamming metrics in the context of a particular pattern recognition problem will be presented elsewhere [8].

Simulation VI The final simulation is an example of how one can find with the present algorithm, T_{ij}'s that are not only nonsymmetric, but in addition describe a slave network consisting of interacting excitatory and inhibitory subpopulations of neurons. The slave net again evolves to a desired fixed point. This division of neurons into purely inhibitory or purely excitatory types, is an elemental feature of the cortex and is a reasonable requirement of any neural model of Content Addressable Memory that purports to have some relation to real cortex. Symmetric T_{ij}'s and the nonsymmetric ones computed so far, do not have this property. This division is recognized in a T_{ij}, if in each individual column of the synaptic matrix all elements have the same sign. Positive columns are due to excitatory neurons, negative columns are due to inhibitory neurons, while a zero-element denotes no neural connection at all. To incorporate this division into subpopulations, we selected a one-dimensional slave net of 20 neurons, and arbitrarily specified that they were alternating inhibitory or excitatory neurons, starting with an inhibitory neuron on the left (see fig. 3).

i e i e i e i e i e i e i e i e i e i e

(inhibitory/excitatory slave net)

Fig. 3. One-dimensional inhibitory/excitatory slave net. "i" denotes inhibitory neurons, "e" denotes excitatory neurons.

The master/slave algorithm then proceeds exactly as before except that the T_{ij} is now parameterized as:

$$T_{ij} = (-1)^j g(U_{ij}) \tag{18}$$

instead of the previous parameterization

$$T_{ij} = 2g(U_{ij}) - 1. \tag{19}$$

We chose to insert just one fixed point into the slave net, which was selected to be a one-dimensional version of a stripe. This fixed point was chosen to have firing rates at either 0 or 1, as depicted in fig. 4.

0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0

(desired fixed point)

Fig. 4. Desired fixed point of inhibitory/excitatory slave net. The firing rates at the fixed point are set to either 0.0 or 1.0.

Stable stripe patterns of activity in the cortex are of considerable interest and may be the result of destabilization of the cortex by increased disinhibition [9]. The master net was evolved in the usual manner (with all $C_i^{(1)} = 1$) and it was verified that the computed T_{ij} resulted in the desired stripe activity in the slave net. The computed T_{ij}, which is highly structured, is presented in fig. 5.

```
-.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5
-.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5
-.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5
-.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5
-.5  .5  -.5  .5  -0.0  1.0   0.0  1.0  -.5  .5  -.5  .5   0.0  1.0   0.0  1.0  -.5  .5  -.5  .5
-.5  .5  -.5  .5  -0.0  1.0   0.0  1.0  -.5  .5  -.5  .5   0.0  1.0   0.0  1.0  -.5  .5  -.5  .5
-.5  .5  -.5  .5  -0.0  1.0   0.0  1.0  -.5  .5  -.5  .5   0.0  1.0   0.0  1.0  -.5  .5  -.5  .5
-.5  .5  -.5  .5  -0.0  1.0   0.0  1.0  -.5  .5  -.5  .5   0.0  1.0   0.0  1.0  -.5  .5  -.5  .5
-.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5
-.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5
-.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5
-.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5
-.5  .5  -.5  .5  -0.0  1.0   0.0  1.0  -.5  .5  -.5  .5   0.0  1.0   0.0  1.0  -.5  .5  -.5  .5
-.5  .5  -.5  .5  -0.0  1.0   0.0  1.0  -.5  .5  -.5  .5   0.0  1.0   0.0  1.0  -.5  .5  -.5  .5
-.5  .5  -.5  .5  -0.0  1.0   0.0  1.0  -.5  .5  -.5  .5   0.0  1.0   0.0  1.0  -.5  .5  -.5  .5
-.5  .5  -.5  .5  -0.0  1.0   0.0  1.0  -.5  .5  -.5  .5   0.0  1.0   0.0  1.0  -.5  .5  -.5  .5
-.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5
-.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5
-.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5
-.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5  -1.0  0.0  -1.0  0.0  -.5  .5  -.5  .5
```

Fig. 5. Synaptic connection matrix, T_{ij}, for inhibitory/excitatory slave net. Note that each column is consistently of one sign apart from the values 0.0, denoting no neural connections at all. The neural connectivity is a superposition of both local and long-range interactions.

5. Conclusions

The algorithm presented above describes a self-organizing neural net consisting of interacting master/slave subnets. Fixed-point behavior suitable for pattern recognition and content addressable memory is obtained from bounded synaptic connection matrices that are structured and nonsymmetric. (The degree of asymmetry was characterized by evaluating the eigenvalues of the T_{ij}'s for simulations I–VI. In most cases the imaginary part of the complex eigenvalues were of similar magnitude to the real parts, and the magnitudes of real and complex eigenvalues were also similar. Also, a few runs were selected at random and redone using only the symmetric part of T_{ij}, resulting in radically reduced performance. It is therefore clear that the asymmetry is used in an essential way by this algorithm, and that it is not merely decorating symmetric T_{ij}'s with small asymmetries.) Parameters that naturally occur in the formalism allow the net to use differing measures of pattern similarity for pattern recognition problems. The master/slave network structure is relatively simple and presents no additional problems for hardware implementation when compared to single net approaches. The synaptic asymmetry, and division of neurons into purely excitatory or inhibitatory subpopulations, remove two major obstacles in relating such formal neural modeling to real neurophysiology. The ability to use varying metrics of pattern similarity also removes a major obstacle in using artificial implementation of neural nets in practical pattern recognition problems.

Further investigation of these two aspects of the present formulation, in the context of specific problems, will be presented elsewhere.

Acknowledgements

We are grateful to E. Baum, J. Cowan, J. Denker, J. Hopfield, E. Mjolsness, and D. Sharp for helpful discussions. We are grateful to J. Denker for sharing prepublication results on the relation of this model to Adaline models in the special case where synapse values are allowed to be unbounded. We enjoyed partial financial support from the Center For Nonlinear Studies (Los Alamos National Laboratory).

References

[1] For a recent attempt, see e.g.: J.J. Hopfield, P.N.A.S. 79 (1982) 2554.
[2] For early work on collective effects see: H.P. Wilson and J.D. Cowan, Biophys. J. 12 (1972) 1; Kybernetik 13 (1973) 55.
[3] J.J. Hopfield and D.W. Tank, Biol. Cybernetics, to be published.
[4] J. Hopfield, P.N.A.S. 81 (1984) 3088. See also J. Cowan, SIAM J. Appl. Math. 38 (1980).
[5] D.W. Tank and J.J. Hopfield, "Simple Optimization Networks," to be published.
[6] J. Denker, Physica 22D (1986) 216 (these proceedings). E. Mjolsness (private communication).
[7] E. Baum, private communication.
[8] A. Lapedes and R. Farber, work in progress.
[9] J.D. Cowan, Biol. Cybernetics 34 (1979) 137.