Journal of Statistical Physics, Vol. 41, Nos. 1/2, 1985

Parallel Computing Structures Capable of Flexible Associations and Recognition of Fuzzy Inputs

T. Hogg[1] and B. A. Huberman[1]

Received January 17, 1985; final May 3, 1985

We experimentally show that computing with attractors leads to fast adaptive behavior in which dynamical associations can be made between different inputs which initially produce sharply distinct outputs. We do so by first defining a set of simple local procedures which allow a computing array to change its state in time so as to produce classical Pavlovian conditioning. We then examine the dynamics of coalescence and dissociation of attractors with a number of quantitative experiments. We also show how such arrays exhibit generalization and differentiation of inputs in their behavior.

KEY WORDS: Collective computation.

What is the range of behavioral functions of the brain that can be reproduced by the collective behavior of arrays of simple, locally connected, computing elements? Moreover, to what extent can this repertoire be encompassed by the dynamics of a single architecture? Examples of these possibilities include self-organization in the presence of time vaying inputs, their recognition even when distorted, and the ability to establish flexible asociations between them. Answers to these questions[1-3] are important in understanding the emergence of complex behavior out of a collection of simple units, in determining to what extent VLSI structures can be made to behave in adaptive fashion, and more generally, in elucidating the global dynamics of systems made up of elementary computational cells.

Our approach to these issues considers arrays of simple local units that exhibit some interesting property. For various values of the array

[1] Xerox Palo Alto Research Center, Palo Alto, California 94304.

0022-4715/85/1000-0115$04.50/0

parameters, we then quantitatively examine the emergence of global computational behavior as a function of time. Within this context, we have recently shown that there is a class of architectures that can be made to compute in a distributed, self-repairing fashion, by exploiting the existence of attractors in their phase spaces.[4] Such a mechanism leads to computing structures which are able to reliably learn several inputs and to recognize them even when slightly distorted.

The architecture that we have now investigated consists of a rectangular array of simple processors each of which operates on integer data received locally from its neighbors (see Fig. 1a). Overall input and output to the machine takes place only along the edges. Each processor has an internal state or bias, represented by an integer B, which can only take on

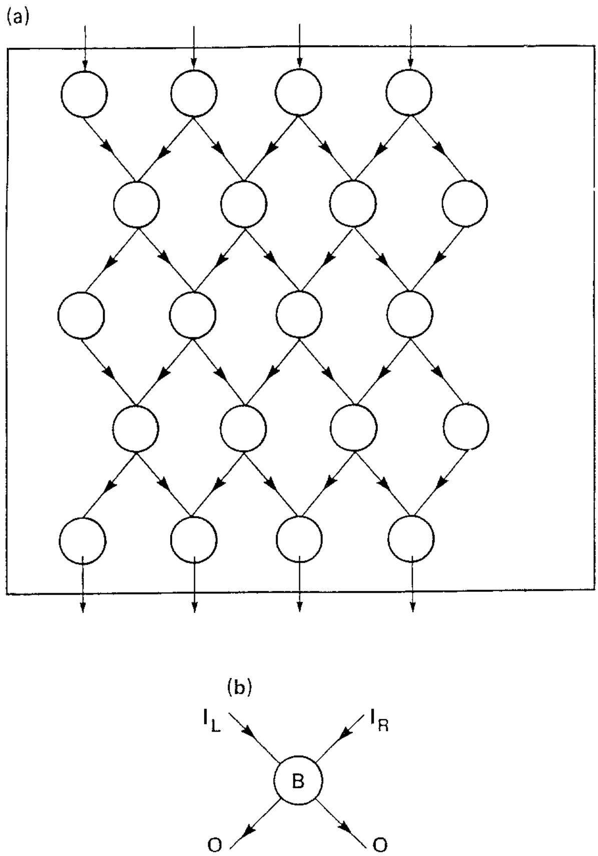

Fig. 1. (a) Schematic diagram of the arrays used in these experiments. (b) Local rules obeyed by each cell. The output from each unit is sent to the lower left and right units.

a small set of values. The unit determines it local output based on its inputs and its internal state. Figure 1b shows how such outputs are computed. At each time step, every element receives data values from the units to its upper left and right, I_L and I_R, respectively, and computes its output O in the following way:

$$O = \max(S_{\min}, \min(S_{\max}, s(I_L, I_R) \cdot (|I_R| + |I_L|) + B)) \tag{1}$$

where for even rows, if a is zero $s(a, b)$ is the sign of b, otherwise the sign of a; and for odd rows the roles of a and b are reversed. $S_{\min}$ and $S_{\max}$ constrain the data to lie in a specific interval, thereby creating a nonlinear computation. Essentially, Eq. (1) amounts to summing the magnitudes of inputs to a given cell, choosing an appropiate sign, adding the bias B, and restricting the result to the proper range, $[S_{\min}, S_{\max}]$. Those output values equal to the extrema of this range are said to be *saturated.*

This rule leads in turn to a contraction mechanism whereby many inputs are mapped into the same output. In the language of dynamical systems, this corresponds to the appearance of a fixed point in the phase space of the system. Furthermore, the contraction of volumes in phase space makes these fixed points attractive in the sense that perturbations quickly relax back to the original values. The set of inputs which map into a given output defines the basin of attraction for that output, as illustrated in Fig. 2a.

Since there are many such basins of attraction, a natural question concerns the possibility of changing them at will with local rules. In other words, one is interested in dynamically modifying the basins of attraction in order to include or exclude a particular set of inputs. Figures 2a and 2b show schematically how this adaptive mechanism would work. These new processes of coalescence and dissociation of attractors lead to results analogous to Pavlovian conditioned reflexes.[5,6] They are achieved by changing the internal state of each of the computing elements using adaptive local rules which mimic the global expansion or contraction process one is interested in achieving. That such local computation leads to this global behavior is indeed surprising in view of the nonlinearity of the system.

We now report experimental results that show how this new paradigm leads to fast adaptive behavior in which dynamical associations are made between different inputs which initially produce sharply distinct outputs. We do so by first defining a set of simple local procedures which allow the array to change its state in time so as to produce classical Pavlovian conditioning. We then examine the dynamics of coalescence and dissociation of attractors with a number of quantitative experiments, and also measure how clouds of nearby inputs are affected by these process.

Specifically, consider an array of the type discussed above and two particular inputs which produce outputs. To make the two inputs produce the same output, one sends them through the array following an instruction which sets the adaptive procedure in the local units to the following contracting rule:

> **if** at least one of O_{new} and O_{prev} is not saturated **and** $O_{\text{new}} \cdot O_{\text{prev}} < 0$
> **then** change B by 1, with the sign of the change given by the sign of the output with largest magnitude (or by the sign of O_{new} when both have the same magnitude)
> **else** B is unchanged

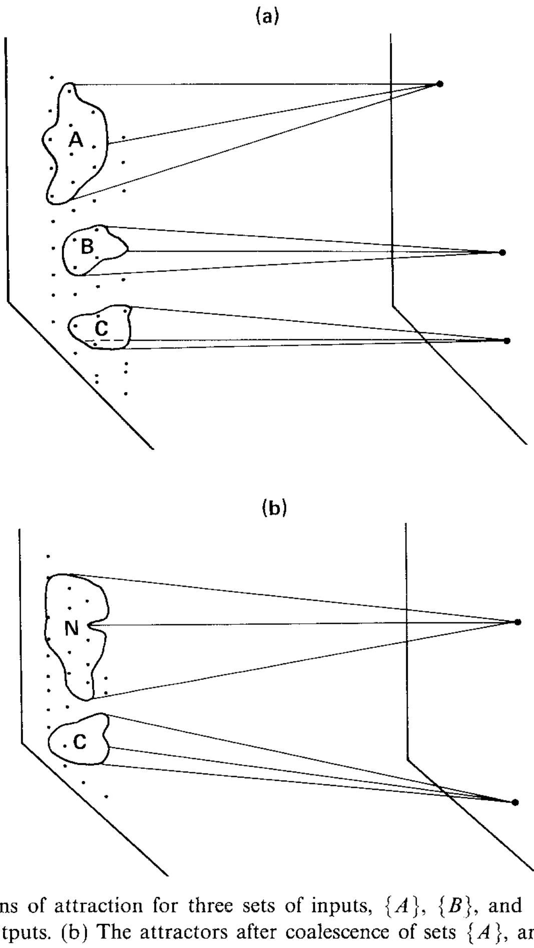

Fig. 2. (a) Basins of attraction for three sets of inputs, $\{A\}$, $\{B\}$, and $\{C\}$ mapping into three different outputs. (b) The attractors after coalescence of sets $\{A\}$, and $\{B\}$ into a new basin of attraction $\{N\}$.

where O_{new} and O_{prev} denote the current and previous outputs of the unit, respectively. As can be seen, this rule simply changes the internal state B by ± 1. Furthermore, to speed convergence one can choose to update the value of O_{new} immediately after changing B.

Likewise, one could start with inputs which initially map into the same output and separate them with the following expanding rule:

> **if** at least one of O_{new} and O_{prev} is not saturated **and** $O_{new} \cdot O_{prev} > 0$
> **then** change B by 1, with the sign of the change opposite of that of either output
> **else** B is unchanged

Table I shows the results of an experiment performed on a ten rows by six columns array of processors obeying the above local rules. They were obtained by sending the inputs through the array which has 64 attractors, out of which three are made to coalesce. At each pass the outputs were compared, and if their difference was not zero, the inputs were sent through the array again. The experiment ended as soon as exact coincidence between outputs was detected. The table also shows the results of an experiment whereby inputs which were initially in the same basin of attraction were separated.

Having shown that such associative behavior can be easily accomplished, one can investigate the behavior of nearby inputs during the merg-

Table I. Results of Two Separate Experiments on an Array with Ten Rows and Six Columns Starting with all of the Bias Values Set to Zero[a]

Input	Original output	Final output
	Coalescence of inputs	
3 3 3 3 3 3	+ + + + + +	+ + + + + +
−2 −2 −2 −2 −2 −2	− − − − − −	+ + + + + +
2 −3 3 −2 1 −1	+ − + − + −	+ + + + + +
	Dissociation of inputs	
3 4 1 4 2 2	+ + + + + +	− − − − − −
2 2 2 5 5 5	+ + + + + +	− − − + + +
4 5 5 3 4 4	+ + + + + +	+ + + − − −

[a] It required four iterations through the array to produce the coalescence shown here, and three iterations to give the dissociation. Reversing the procedure then required 13 iterations to reexpand the coalesced inputs and five to recontract the dissociated ones. For these experiments $S_{max} = -S_{min} = 15$, wheras $-10 \leqslant B \leqslant 10$.

ing and dissociation of the attractors. This amounts to determining how the size and shape of the basins of attraction change when conditioned reflexes are obtained in an adaptive structure. This is important in establishing to what extent this array is capable of both input generalizations and its complement, differentiation.[6]

To answer these questions, we used the following two techniques before and after associating given inputs: (1) determining the distribution of sizes of basins of attraction by sending random inputs through the array and counting how many produced each observed output, and (2) determining the size of the basins of attraction by taking each input and measuring the fraction of nearby inputs that are in the same basin of attraction as a function of distance from the original input. Since the set of inputs form an integer lattice, we measured distances between inputs using a street map metric which sets the distance between two points to be the sum of the magnitudes of the differences in each component. As shown in Fig. 3, the process of association produces a broad aggregation of clouds of inputs surrounding the original inputs. This implies that when associating two specific behavioral responses, the same output can be elicited by operating with inputs which are close to the original ones. We also found similar results in the opposite limit of trying to dissociate inputs which originally produced the same outputs.

Furthermore, although the basins of attraction are initially of equal size, after adaptation the attractors for the learned inputs grew at the expense of others. Specifically, for the case considered here (an array with six columns and ten rows), there are 2^6 attractors, and each one has 1/64th or 1.6% of the inputs in its basin. After the coalescence experiment described above, the attractor containing the three contracted inputs included almost 4% of the inputs in its basin of attraction. Similarly, in the other experiment in which the three inputs were separated, the final basins of atraction contained 4%, 2%, and 2% of the inputs, respectively.

An interesting consequence of this investigation is the correlation between the ability to quickly associate a set of given inputs and the initial state of the array. Generally, merging two basins of attraction when starting with all cells having the same state (i.e., a uniform state) was much simpler (i.e., took fewer passes through the array) than starting with an array which had already adapted to a set of inputs. This became particularly evident in experiments where we started with two separate inputs and a uniform array, associated them together and then tried to separate them. The time required to separate them was much longer than the time that it took to merge them into the same basin. Similar results held when these operations were tried in the opposite order. This effect might provide a concrete mechanism whereby selection from a highly degenerate con-

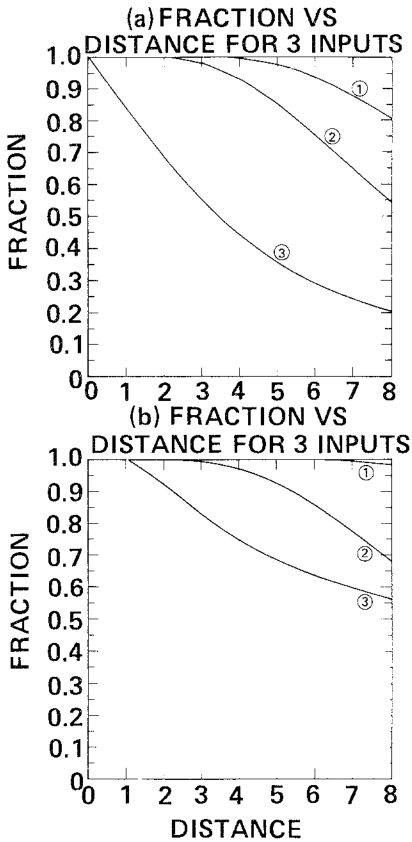

Fig. 3. Fraction of samples that are in each basin of attraction, as a function of the distance between the sample and the learned input. (a) original array (b) after coalescence.

figuration leads to states which are harder to unravel at later times, as has been postulated in the group-selection theory.[7]

Having succesfully tested the ability of such structures to either associate or dissociate inputs, we then proceeded to group many inputs into many different basins of attraction. This was done by sending both instructions for contraction and expansion in the form of input waves which set the array to successively operate according to the expansion and contraction rules given above.

One of the limitations of this strategy appears when trying to simultaneously contract and expand groups of nearby inputs. If inputs $A_1$and A_2 are close together, and so are B_1 and B_2, then an example of this procedure is to attempt to contract A_1 and B_1while simultaneously expanding $A_2$and B_2. In this case, a conflicting set of succesive instructions leads to a frustration effect whereby the outputs never converge to the desired classes. An additional frustration can appear when the two inputs to be contracted have the same magnitude and opposite sign. We should point out however, that a different strategy concerning the sequence of instructions could in principle achieve the desired results.

For instance, one can adopt a less conservative adaptive rule and change the bias values by larger increments. This tends to coalesce or expand individual pairs more rapidly, but also increases the risk of undoing previous instructions. As a specific example, we found that a slight modification of the contraction rule given above removed most of the frustrations associated with equal but opposite inputs. This modified rule is

> **if** at least one of O_{new} and O_{prev} is not saturated **and** $O_{\text{new}} \cdot O_{\text{prev}} < 0$
> **then** change B by 1, with the sign of the change given by the sign of the output with largest magnitude *(or by $+1$ for odd rows and -1 for even rows)*
> **else** B is unchanged

and was used in the statistical experiment described below.

To assess the relevance of this methodology for large-scale applications, such as required for speech or vision processing, we examined the behavior of the array throughout the input space. This was done by attempting to simultaneously coalesce two sets, each containing two inputs, into two separate outputs. To determine the behavior throughout the set of inputs, this process was repeated 1000 times with randomly selected pairs. The results were that 80% of the inputs quickly produced the desired outputs, with only minor modifications to the other 62 attractors.

The success of these experiments opens the possibility of using these computing structures for problems in which recognition of fuzzy inputs should be coupled to a flexible way of programming desired groupings of inputs into specific outputs. This is particularly useful in vision and speech recognition, where a preliminary encoding of either patterns[2] or voice[(9)] would produce sets of inputs which one would then want to put in the same equivalence class.[3] At a more fundamental level, these results provide an experimental basis for the development of a theory of dynamics of dissipative computing structures.[(11)]

[2] As, for example, using textous in preattentive vision (see Ref. 8).

[3] This should be distinguished from both perceptron-like models (Ref. 10a) and statistical computational mechanisms (Ref. 10b). Our array, unlike perceptrons, uses nonlinear rules which allow a much richer behavior such as the associations we have discussed here. Statistical machines, on the other hand, may never converge to a unique equilibrium, and if they do it may take a long time.

ACKNOWLEDGMENTS

We thank W. Keirstedt for his help improving the contraction rule and running the statistical experiments. This research was partially supported by ONR contract N00014-82-0699.

REFERENCES

1. J. S. Albus, *Brains, Behavior, and Robotics* (Byte/McGraw-Hill, Petersborough, New Hampshire, 1981), and references therein.
2. G. M. Edelman and V. B. Mountcastle, in *The Mindful Brain* (MIT Press, Cambridge, Massachusetts, 1978).
3. J. von Neumann, in *Theory of Self-Reproducing Automata*, A. W. Burks, ed. (University of Illinois Press, Urbana, Illinois, 1966).
4. B. A. Huberman and T. Hogg, *Phys. Rev. Lett.* **52**:1048 (1984); T. Hogg and B. A. Huberman, *Proc. Natl. Acad. Sci. (USA)* **81**:6871 (1984).
5. I. P. Pavlov, *Conditioned Reflexes* (Dover, New York 1960).
6. E. R. Hilgard and G. H. Bower, *Theories of Learning*, 3rd ed. (Meredith Publishing Co., New York, 1966), Chap. 3.
7. G. M. Edelman and L. Finkel, in *Dynamic Aspects of Neocortical Function*, G. M. Edelman, W. M. Cowan, and W. E. Gall, eds. (Wiley, New York, 1984).
8. B. Julesz, *Nature (Londen)* **290**:91 (1981).
9. D. R. Reddy, ed., *Speech Recognition: Papers Presented at the 1974 IEEE Symposium* (Academic Press, New York, 1975); G. R. Doddington and T. B. Schalk, *IEEE Spectrum* **18**:9 (1981).
10. (a) M. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry* (MIT Press, Cambridge, Massachusetts, 1969); (b) W. A. Little, *Math. Biol.* **19**:101 (1974) and G. E. Hinton, T. J. Sejnowsky, and D. Ackley, CMU-CS-84-119 Technical Report (1984).
11. T. Hogg and B. A. Huberman, Attractors on finite sets: The dissipative dynamics of computing structures, *Phys. Rev. A* (October 1985), to appear.