

XIV. NEUROPHYSIOLOGY*

W. S. McCulloch
J. A. Aldrich
F. S. Axelrod
H. J. C. Berendsen
M. Blum

J. E. Brown
J. D. Cowan
R. C. Gesteland
K. Kornacker
J. Y. Lettvin
Diane Major

R. Melzack
N. Onesto
W. H. Pitts
A. Taub
P. D. Wall

A. COMPUTATION IN THE PRESENCE OF NOISE (RELIABLE MACHINES FROM UNRELIABLE COMPONENTS)

Information theory may be applied to the problems of computation in the presence of noise or reliable processing with unreliable computing elements. It has been demonstrated that redundant computers that achieve arbitrarily low frequencies of error (apart from errors in the final outputs) may be constructed, so that they are not completely redundant but process a finite fraction of information. This depends critically upon the error behavior of components as a function of complexity. If component errors increase with complexity, this reliability can be obtained only by decreasing the fraction of information processed in the computer. However, it is possible to maximize this fraction, for given components and codes. A further result of interest is that such computers need not be precisely connected and, in fact, a certain bounded fraction of errors in connection may be tolerated.

J. D. Cowan, S. Winograd

B. ULTIMATE PERIODICITY OF REAL-TIME COMPUTATION BY A TURING MACHINE WITHOUT OUTPUTS

In McNaughton's survey,¹ I read of a conjecture by Burks on real-time computation by a "Turing Machine with Output." This led me to consider real-time computation by a Turing machine without output. This report is devoted to the simple but interesting result that I obtained to the effect that such computation can only produce ultimately periodic sequences – thus whilst Turing has shown² that π and e are computable, we have now shown that only rational numbers are computable in real time by a Turing machine without output. Our definition of real-time computation is very general; furthermore, we have found that a number of reasonable alterations in this definition has not vitiated the validity of our result.

DEFINITION: A Turing machine prints the sequence $\{a_r\}$ in real time if there exists an integer n such that the machine prints a_r at time nr , and a_r is never printed to the left of a_{r-1} . (The tape may be printed initially on a finite number of squares.) We

*This work was supported in part by Bell Telephone Laboratories, Inc.; the National Institutes of Health; The Teagle Foundation, Inc.; and in part by the U.S. Air Force under WADD Contract AF33(616)-7783.

(XIV. NEUROPHYSIOLOGY)

now introduce an auxiliary concept.

DEFINITION: A finite printer is a quadruple $P = (A, S, s_0, M)$, where

A is a finite set (the Alphabet of P);

S is a finite set (the set of States of P);

s_0 is a member of S (the initial state of P); and

$M: S \rightarrow S \times A$

The sequence $\{a_r\}$ is printed by P if $M(s_{j-1}) = (s_j, a_j)$, $j = 1, 2, \dots$.

LEMMA: Given a Turing machine T and a sequence $\{a_r\}$ that it prints in real time, there exists a finite printer that prints the same sequence.

This lemma, whose proof need not be given in this report, leads immediately to our main theorem.

THEOREM: A sequence printed in real time by a Turing machine is ultimately periodic.

PROOF: Immediately follows from the lemma and the finitude of the set of internal states of a finite printer.

M. A. Arbib

References

1. R. McNaughton, The Theory of Automata – A Survey, to appear in Vol. II of Advances in Computers, F. L. Alt(ed.) (Academic Press, New York, 1961).
2. A. M. Turing, On computable numbers with an application to the entscheidungs problem, Proc. London Math. Soc., Ser. 2, Vol. 42, pp. 230-265, 1936.

C. AN INELEGANT SOLUTION TO THE PROBLEM OF REALIZABILITY OF BOOLEAN FUNCTIONS BY THE FORMAL THRESHOLD TYPE OF NEURON

This report shows that of the 2^{2^n} possible Boolean functions of n variables, not more than $2^{n+1} \binom{2^n}{n}$ are realizable by the formal threshold type neurons. We give a programmable procedure for listing all such realizable functions, and discuss the number of steps involved.

A formal threshold type of neuron N is determined by n "weights" w_1, \dots, w_n , and a "threshold" θ . N fires at time $t + 1$ iff $\sum_{i=1}^n w_i X_i \geq \theta$, where the input to N at time t is given by the binary vector (X_1, \dots, X_n) (where $X_i = 1$ (resp., 0) is to be interpreted as "input line i fires (resp., does not fire)").

We say that N realizes the Boolean function $f(X_1, \dots, X_n)$ if N fires when, and only when, $f(X_1, \dots, X_n) = 1$.

Now let $C^n = \{(X_1, \dots, X_n) \mid X_i = 0 \text{ or } 1\}$, be the set of the 2^n vertices of the hypercube

in Euclidean n -space. A hyperplane $H: \sum_{i=1}^n w_i X_i = \theta$ clearly partitions C^n into 2 subsets H^+ , H^- (i.e., into the set of points for which $\sum_{i=1}^n w_i X_i \geq \theta$, and the set for which $\sum_{i=1}^n w_i X_i < \theta$).

It is now clear that the task of listing realizable Boolean functions reduces to that of listing all partitions of C^n by hyperplanes (note, however, that 2 functions are associated with each such partition).

We observe that any hyperplane H may be moved until it contains n points of C^n ; and that (H^+, H^-) is then determined by these n points, plus the specification of in which of H^+ or H^- each of the n points was. Conversely, it is easy to see that any such specification determines a partition of the hypercube by a hyperplane. Hence, our listing procedure merely consists in looking at all $\binom{2^n}{n}$ sets of n points of C^n ; and then, for each set, listing the 2^n partitions that it determines. This procedure is clearly programmable.

By a complicated argument, which will not be given here, we can show that each hyperplane through n points of C^n determines at least one partition of C^n "unique unto itself." Hence our procedure produces $2n \binom{2^n}{n}$ partitions, of which at least $\binom{2^n}{n}$ are distinct. For large n , $\log \left[2^n \binom{2^n}{n} \right] \sim \log \left[\binom{2^n}{n} \right] \sim n^2 - n \log \left(\frac{n}{e} \right)$, so that our procedure is of the optimal "order of order of (sic) magnitude."

Since there are 2^{2^n} possible Boolean functions, and since $2 \cdot 2^n \binom{2^n}{n} = o \left(2^{2^n} \right)$ as $n \rightarrow \infty$, we see that, as n increases, only a vanishingly small fraction of Boolean functions of n variables can be realized by the formal threshold type of neurons. This would seem to indicate that, at least in the choice of their basic components, computer designers should not strive to mimic the human nervous system too closely.

In closing, we note that our listing procedure provides an algorithm for testing whether or not a particular Boolean function is realizable. But it is inelegant, and the number of steps is discouraging. Manuel Blum has produced several theorems on realizability, and is at present extending them in the hope of finding a satisfyingly elegant algorithm.

M. A. Arbib

