# ON THE ORGANIZATION OF MEMORY[⊛]

by

Andrzej Ehrenfeucht*
Jan Mycielski**
* Department of Computer Science
** Department of Mathematics
UNIVERSITY OF COLORADO
Boulder, Colorado 80302

Report #CU-CS-014-73 March, 1973

Abstract. A new learning algorithm is presented which may have applications in the theory of natural and artificial intelligence.

[⊛] This work was supported by N. S. F. Grants GJ-660 and GP-19405.

1. Introduction. In order to construct machines which could display an intelligence somewhat like that of animals or man, or to understand how the brain functions, one must develop a model of the organization of memory. Several models have been proposed [1,4,6,7,11 where other references are given]. Here we propose still another one which is new in that it tends to explain how the brains learns to interpret correctly (or act purposefully upon) the immense amount of information which it obtains continuously from the senses.

For simplicity, we assume at first that the brain serves only to answer questions which admit a "yes" or "no" answer and that the questions are sequences of 0's and 1's of a fixed length m. Our questions depict the totality of data (parameters) which the brain gets at a given time (ours is a discrete time model) and hence m is very large (see section 5, remarks 1 and 2 concerning the possible meanings of m). At the beginning the brain does not know what to say and answers arbitrarily but it gets a "reward" when the answer is right and a "punishment" when the answer is wrong. Thus it gets post facto information what is the right answer and attempts to produce correct answers to the questions which follow. By the problem of organization of memory we understand the problem of defining an algorithm to answer new questions on account of past experience. Various statistical estimation procedures and classical methods of interpolation and approximation of functions seemingly would apply to this problem. But as information theory shows [3], they loose their power when the dimension m is large, say $m \geq 100$. Our algorithm is free from this defect. This is so because we exploit the following natural assumption: Few of the parameters are important for solving any given question (see remarks 6 and 7 below concerning the validity and the interpretation of this assumption), although different parameters are needed to solve different questions. The brain probably develops a method of selecting the important parameters of any given question; so does our algorithm.

We shall not attempt in this paper to define a neural network capable of performing our algorithm nor theorize on the question how the nervous tissue could do it since, although easy, this would seem to us premature (see section 4).

2. The Algorithm. Let X be the space of questions i.e., a set of sequences of 0's and 1's of length m, and let R be the set of all possible responses to a question (from now on more than two responses may exist). We assume that the questions come in a sequence $x_1, x_2, \ldots$ (where $x_t = (x_t(1), \ldots, x_t(m))$ is in X) such that most of the time $x_{t+1}$ differs from $x_t$ by one coordinate only i.e., $x_t(k) \neq x_{t+1}(k)$ for only one k. Of course, this k depends on t. Thus we may think of $x_1, x_2, \ldots$ as of a (random) walk over some vertices of an m-dimensional cube most steps being a shift along an edge. Let now $f : X \to R$ be a function such that f(x) is the right response to the question x. If f is wild then the past experience, i.e., the sequence $x_1, f(x_1), \ldots, x_n, f(x_n)$ gives no information on the values $f(x_t)$ for $t > n$ unless $x_t = x_j$ for some $j \leq n$ (which is a very exceptional event if say $m \geq 200$, $n \leq 10^{10}$ and the $x_t$'s are produced in a random-like way). Thus the problem of organization of memory makes sense only if f is regular enough.

We shall describe the algorithm for estimating f(x) without any suppositions on f. We take the efficiency of this algorithm as the indicator of that regularity of f. An attempt for a more explicit definition of this regularity is mentioned in Section 3.

Algorithm. Given $x_1, f(x_1), \ldots, x_n, f(x_n)$ one builds the following three objects.

(1) A tree T of sequences 0's and 1's, i.e., a nonempty set of sequences (of various lengths) such that whenever $(a_1, \ldots, a_r)$ is in T then the three sequences $(a_1, \ldots, a_i)$, $(a_1, \ldots, a_i, 0)$ and $(a_1, \ldots, a_i, 1)$ are in T for every $i < r$. (Hence the empty sequence $\emptyset$ always belongs to T, it is called the root of the tree). Any sequence $(a_1, \ldots, a_r)$ of T such that $(a_1, \ldots, a_r, a)$ is not in T for $a = 0,1$ is called an end. E denotes the set of ends of T.

(2) A function $K : (T - E) \to \{1, \ldots, m\}$.

(3) A function $F : E_0 \to R$, where $E_0$ is a subset of E.

The triple $\langle T,K,F \rangle$ is stored in the memory. It constitutes a program for estimating f(x) for some x in X. Namely given x, one builds an end $(a_1, \ldots, a_r)$ in E using the following recursive rules

$$a_1 = x(K(\emptyset)), \quad a_{i+1} = x(K(a_1, \ldots, a_i)) \text{ for } i = 0,1,\ldots$$

Then, if $(a_1, \ldots, a_r)$ is in $E_0$, one estimates f(x) as $F(a_1, \ldots, a_r)$.

The construction of $\langle T,K,F \rangle$ from $x_1, f(x_1), \ldots, x_n, f(x_n)$ is the following recursive procedure.

(a) We find, if possible, a k in $\{1, \ldots, m\}$ for which the ratio of the number of pairs $x_t, x_{t+1}$ $(t = 1, \ldots, n-1)$ such that $x_t(k) \neq x_{t+1}(k)$, $x_t(j) = x_{t+1}(j)$ for $j \neq k$ and $f(x_t) \neq f(x_{t+1})$ to the number of all pairs $x_t, x_{t+1}$ such that $x_t(k) \neq x_{t+1}(k)$ and $x_t(j) = x_{t+1}(j)$ for $j \neq k$ is positive and maximal. We put $K(\emptyset) = k$ and we assign the one term sequences (0) and (1) to T. But if

$$f(x_1) = f(x_2) = \ldots = f(x_n)$$

then we put $\emptyset$ into $E_0$ and set $F(\emptyset) = f(x_1)$.

(b) Given a sequence $(a_1,\ldots,a_r)$ which has already been assigned to T we choose the subsequence $x_{t(1)},\ldots,x_{t(s)}$ of $x_1,\ldots,x_n$ consisting of all the terms which satisfy the condition

$$x_{t(i)}(K(a_1,\ldots,a_j)) = a_{j+1} \text{ for all } j < r.$$

Then, if possible, we find a k in $\{1,\ldots,m\}$ such that the number of pairs $x_{t(i)}$, $x_{t(i+1)}$ $(i = 1,\ldots,s-1)$ which differ only at the k-th coordinate (i.e., $x_{t(i)}(j) \neq x_{t(i+1)}(j)$ if and only if $j = k$) and such that $f(x_{t(i)}) \neq f(x_{t(i+1)})$ to the number of all pairs $x_{t(i)}$, $x_{t(i+1)}$ which differ only at the k-th coordinate is positive and maximal. Then we put $K(a_1,\ldots,a_r) = k$ and we assign $(a_1,\ldots,a_r,0)$ and $(a_1,\ldots,a_r,1)$ to T. But if

$$(*) \qquad f(x_{t(1)}) = f(x_{t(2)}) = \ldots = f(x_{t(s)})$$

then we put $(a_1,\ldots,a_r)$ into $E_0$ and set $F(a_1,\ldots,a_r) = f(x_{t(1)})$.

(c) If we cannot perform (a) or (b) (e.g., because (*) fails but there are no pairs $x_{t(i)},x_{t(i+1)}$ differing by only one coordinate with $f(x_{t(i)}) \neq f(x_{t(i+1)})$) then we assign $(a_1,\ldots,a_r)$ to E. (We stop extending this branch and the program remains <u>incomplete</u> for not enough information has been provided by the sequence $x_1,f(x_1),\ldots,x_n,f(x_n)$).

If $E_0 = E$ then the program $\langle T,K,F\rangle$ is <u>complete</u>, otherwise it does not yield any answer to some "difficult" questions x in X.

This ends the description of the algorithm.

<u>3. A formal language and problems of regularity</u>. One can display such $\langle T,K,F\rangle$ using labeled trees, see figure 1. Every branching point and

end point represents a member of T, a step to the left represents a 0 and a step to the right a 1. The labels at the branching points are the values of K, the $r_i$'s at the end are the values of F and the stars indicate ends which are not in $E_0$.

Fig. 1 →

It may be useful to apply the following notation. Let $[a,x,b]$, where $x = 0,1$ but a and b are arbitrary things, be defined by $[a,0,b] = a$ and $[a,1,b] = b$. Now the formula

$$[[[r_3,x(1),[*,x(6),r_4]],x(3),r_3],x(5),[[r_1,x(2),[r_3,x(1),*]],x(3),[r_2,x(4),*]]]$$

represents the $\langle T,K,F\rangle$ of figure 1.

How to measure the regularity of f needed for the success of the algorithm? Let $\mu$ be a probability measure over the space X. Assume that the mechanism generating $x_1,\ldots,x_n$ is a random choice such that if t is not a multiple of 100 then one of the coordinates of $x_t$ is changed to get $x_{t+1}$, the probability of changing $x_t(k)$ being proportional to $\mu\{(x_t(1),\ldots,x_t(k-1), 1-x_t(k), x_t(k+1),\ldots, x_t(m))\}$ while if 100 divides t then $x_{t+1}$ is obtained by a $\mu$-random choice in X independent of $x_t$.

To define our complexity of f let P(f) be the set of all complete (i.e., without starts) programs $\langle T,K,F\rangle$ faithfully representing f. Clearly P(f) is not vacuous. For each $\langle T,K,F\rangle$ in P(f) we define E(T) to be the expected value of the length of the branch of T which we have to scan to evaluate f(x). Now the complexity c(f) of f is defined as the minimum of all the numbers E(T).

It is apparent that if c(f) is small enough and $x_t$ are generated as above, then we can apply successfully our algorithm. But we have no quantitative analysis of the situation.

4. Experiments. There are the following three obvious directions for experimentation: (α) To construct learning machines based on our algorithm; (β) To find out by methods of experimental psychology whether the brain uses some such algorithm; (γ) To find out whether and how the neural network performs such an algorithm.

Concerning (α) all roads seem open. The main difficulty is to find the right set of basic parameters x(1),...,x(m) (and ways for the machine to measure them) so that the desired functions which are to be learned are sufficiently regular in these parameters (see sections 3, and 5 remark 1). Of course, one should begin with an artifically simplified environment and simple enough functions.

Concerning (β) we thought of the following approach. Define an f with c(f) sufficiently small, e.g., the one in figure 2. This f depends on seven variables x(1),...,x(7) and takes on eight values (names of animals). Let the values of the parameters x(k) be displayed by means of seven lights (on or off) arranged like in figure 3. There are $2^7 = 128$ possible configurations of the lights. Let a sequence of configurations $x_1,...,x_n$ be produced by switching at random the lights one at a time and each time the correct animal is shown on an additional display. Let a subject look at those lights and try to recognize the animals which the configurations represent. How soon (that is for what values of n) will he be able to interpret correctly an arbitrary configuration of lights?

ig.2 →
ig.3

We can think of many possible variants of this experiment: (1) The switching of lights may go on fast enough so that a conscious search for an algorithm to interpret them correctly will be impossible. (2) The switching may be slow but still imposed by the experimentator. (3) The switching may be done by the learning subject himself.

Variant (1) is the most interesting perhaps, since it aims at the structure of memory unaltered by the powers of deduction. And this is precisely what we hope our algorithm explains (in a very simplified way). Variants (2) and (3) are perhaps uninteresting unless the values of 7 and 8 above are replaced by some larger $2^h-1$ and $2^h$ where h is the height of the tree. Already with $h = 4$ we have 15 lights, 16 values and $2^{15} = 32768$ configurations of the lights. Can the subject learn to read correctly every configuration after n lessons, n being very much smaller than $2^{15}$ ? (It is important that the lights be arranged so that the location of each be immediately recognizable independently of the on-off configuration. With 15 lights this is still easy.)

Concerning ($\gamma$) we cannot propose a meaningful search; our knowledge of the brain being so inadequate. Considering that our algorithm could be only a drastic simplification of what is really going on in the brain when learning, we are not yet able to state a worthwhile conjecture.

5. Miscellaneous remarks. 1. Many interesting f's cannot be represented by a program $\langle T,K,F\rangle$ of moderate size, say with T having no more than $m^2$ elements, see [4,8,9]. E.g., if f(x) indicates the parity of the number of 1's for every sequence $x = x(1),\ldots,x(m))$ then T must have at least $2^m - 1$ elements. Such difficult or global f's seem unlearnable and have to be built in the organism or in the machine; we can only hope that such

important f's are not too numerous. We think that the senses and the input nerves provide the central nervous system with sophisiticated data or paramters. The brain learns automatically from examples only such functions which are simple enough in those parameters.

However, the f's learned at some time may become parameters (arguments) for the f's to be learned later. Thus f's could be composed (some material on compositions of regular functions is given in [2,8]).

2. Our discrete time t does not have to correspond to the real time in which the organism is living. $x_t$ may be a compilation of all the parameters obtained at present together with many parameters received earlier. E.g., a circular array of memory cells stores information in the following way. A needle moves in the clockwise direction at a constant speed and the cell at which it points changes its content to register an incoming signal. The state of this set of cells is a part of the question $x_t$.

Another possible interpretation is the following. $x_t$ represents a description of the environment and of the state of the organism based on longer observations. $x_t$ is modified accordingly when the senses register a change.

3. The model described in this paper is not intended to explain how man creates various algorithms, it aims at explaining the principle of more basic abilities, e.g., the ability to learn to recognize the identity of the meaning of not quite identical sounds and pictures (automatic learning).

4. The perceptron learning [9,10] and other methods of linear approximation theory and holographic models [6,11] are other learning devices with which the brain could be equipped.

5. One important activity of the brain is to predict or imagine the future (especially at short range). This means to predict a set of coordinates of some $x_t$ 's with $t > t_0$ from $x_{t_0}$ . This involves learning some coordinates of $x_t$ 's as functions of $x_{t_0}$ . Here again our algorithm could be applied. This task suggests a modification of our algorithm in which the decisions of stopping the growth of a branch of T and assigning to its end a value F comes earlier than specified in clauses (a) and (b), namely when the fraction of counterexamples to (*) is small enough.

6. The 0's and 1's of the coordinates of the $x_i$ 's may be produced by some threshold functions applied to some continuous parameters. Our theory would not differ essentially if these parameters took only a small number of values. It may be interesting to generalize our work to the case when the parameter k in $x_t(k)$ varies over a continuum with $x_t$ regular enough.

7. The regular k - continuous functions f introduced and studied in [3] seem to be a natural domain of application of our algorithm. This paper grew out of a problem stated in Remark 10 in [3].

## REFERENCES

[1] G. S. Brindley, Proc. Roy. Soc. Lond. B. 174 (1969), 173-191.

[2] A. Ehrenfeucht, Practical Decidability (to appear).

[3] A. Ehrenfeucht and J. Mycielski, Interpolation of Functions over a Measure Space and Conjectures about Memory (to appear).

[4] L. Hodes, Journal of the ACM 17 (1970), 339-347.

[5] H. C. Longuet-Higgins, D. J. Willshaw, in the collection Associative Information Techniques, ed. E. L. Jacks, 1971, 117-125.

[6] H. C. Longuet-Higgins, D. J. Willshaw and O. P. Buneman, Quart, Rev. of Biophysics 3, 2(1970), 223-244.

[7] D. Marr, Proc. Roy. Soc. Lond. B. 176 (1970), 161-234.

[8] R. McKenzie, J. Mycielski and D. Thompson, Math. Systems Theory 5 (1971) 259-270.

[9] M. Minsky and S. Papert, Perceptrons, and Introduction to Computational Geometry, MIT Press 1969.

[10] J. Mycielski, Bul. Amer. Math. Soc. 78 (1972), 12-15.

[11] D. J. Willshaw, O. P. Buneman and H. C. Longuet-Higgins, Nature 222 (1969), 960-962.

cah

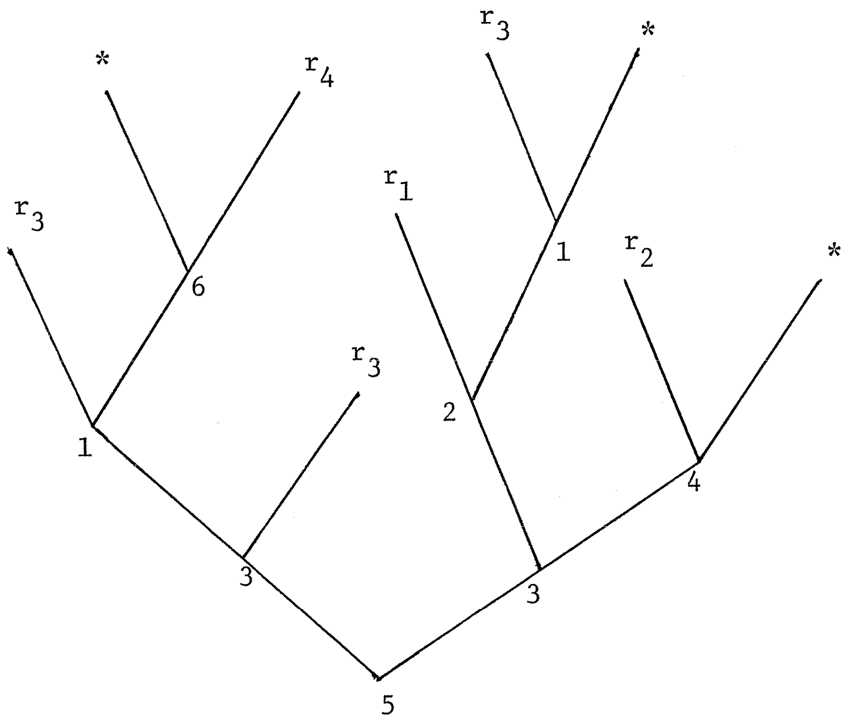

FIGURE 1

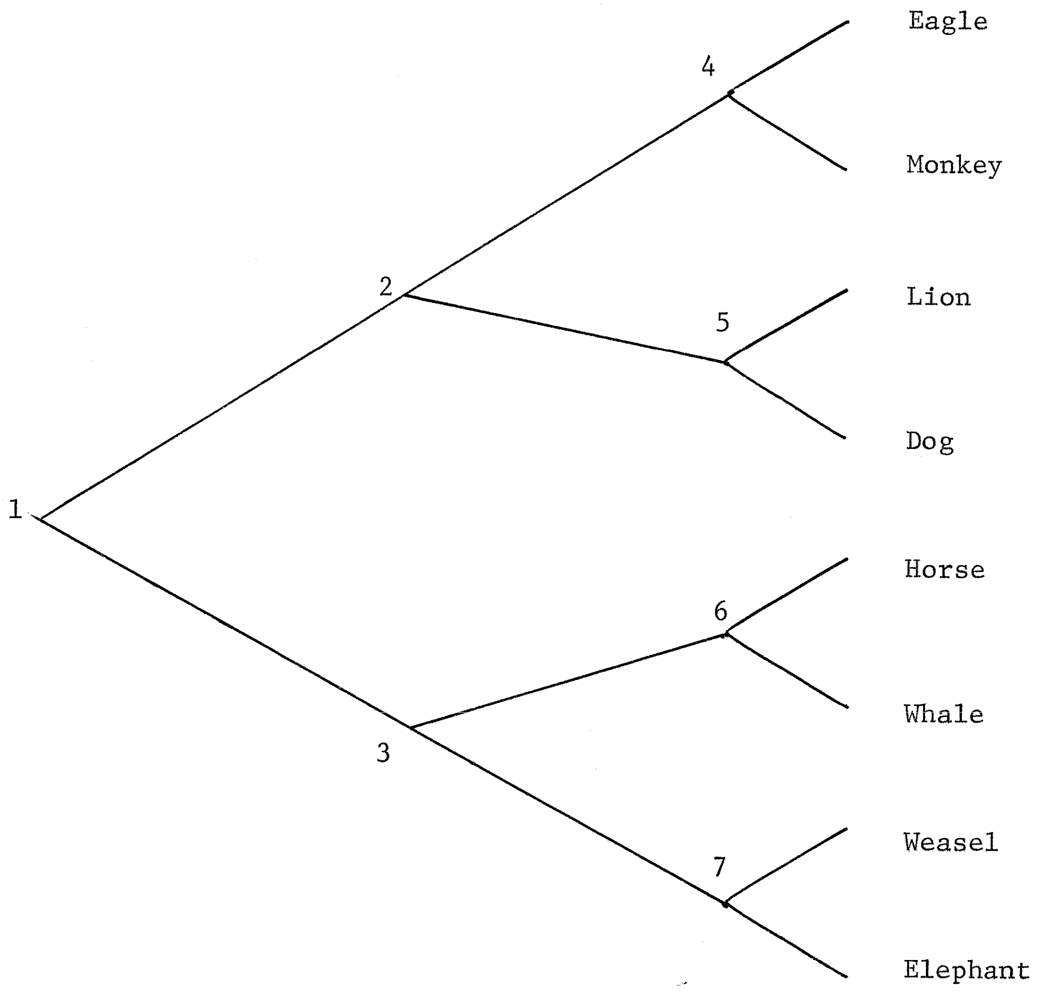

FIGURE 2

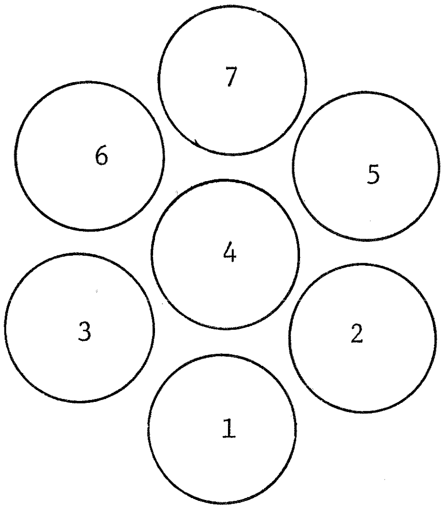

FIGURE 3