Generalization of Pattern Recognition in a Self-Organizing System*

W. A. CLARK† AND B. G. FARLEY†

Summary—**A self-organizing system reported upon earlier is briefly described. Two further experiments to determine its properties have been carried out. The first demonstrates that self-organization still takes place even if the input patterns are subjected to considerable random variation. The second experiment indicates that, after organization with the usual fixed patterns, the system classifies other input patterns statistically according to a simple preponderance criterion. Significance of this result as a generalization in pattern recognition is discussed. Some remarks are made on methods of simulation of such systems and their relation to computer design.**

Description of Self-Organizing System

IN A PREVIOUS paper[1] the authors described a system which organized itself from an initially random condition to a state in which discrimination of two different input patterns[2] was accomplished. The behavior of the system was simulated by means of a digital computer—the Memory Test Computer of Lincoln Laboratory.

Briefly, the self-organizing system was composed of two parts. The first part received input patterns and transformed them into outputs, and the second part acted upon parameters of the first so as to modify the input-output transformation according to certain fixed criteria. These parts were termed the transformation and the modifier, respectively.

The transformation is a randomly interconnected network of nonlinear elements, each element having a definite threshold for incoming excitation, below which no action occurs, and above which the element "fires." When an element fires, its threshold immediately rises effectively to infinity (it cannot be fired), and then, after a short fixed delay, falls exponentially back toward its quiescent value. Furthermore, at some short time after firing, an element transmits excitation to all other elements to which it is connected. The effectiveness of the excitation thus transmitted to a succeeding element is determined by a property of the particular connection known as its "weight." In general, there will be several incoming connections at any element, each having its individual weight as shown in Fig. 1. At the instant of transmission (which is the time of impulse arrival at the succeeding element), the appropriate weight is added to any excitation already present at the succeeding cell. Thereafter the excitation decays exponentially to zero. If at any time this excitation exceeds the threshold of the succeeding element, the element performs its firing cycle and transmits its own excitations.

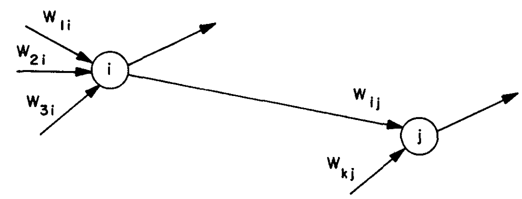


Fig. 1—Typical network elements i and j showing connection weights w.

A network such as the one described is suggestive of networks of the nerve cells, or neurons, of physiology, but since the details of neuron interaction are as yet uncertain, it cannot even be said that the networks are identical without some simplifications which are present.

In the work mentioned, the network was activated and an output obtained in the following way. The net was divided arbitrarily into two groups, designated as input and output groups. The output group was further subdivided in two, and an output was defined at any instant by the difference in the number of elements fired in the two subgroups during the instant. This arrangement might be termed a push-pull output.

The input group was also subdivided into two subgroups, and two fixed input patterns were provided, usually designated as p_1 and p_2. Input p_1 consisted in adding a large excitation into all the input elements of one subgroup simultaneously and repetitively at a constant period, but doing nothing to the other subgroup. Input p_2 was just the reverse. In this way output activity characteristic of the input pattern was obtained.

It was now desired to provide a modifier acting upon parameters of the net so as to gradually reorganize it to obtain output activity of a previously specified characteristic, namely, that patterns p_1 and p_2 would always drive the output in previously specified directions. In our experiments, p_1 was made to drive the output in a negative direction, that is to say, p_1 causes more firing to take place on the average in the first output subgroup than in the second. In the case of p_2, the situation was exactly reversed.

This desired organization of the net was accomplished by means of varying the weights mentioned above in the following way. Examination is made of the change in output at every instant. If a change in a favorable direction occurs (e.g. negative change in case p_1 is the input

* The research reported in this document was supported jointly by the Army, the Navy, and the Air Force under contract with the Massachusetts Institute of Technology.

† Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, Mass.

[1] B. G. Farley and W. A. Clark, "Simulation of self-organizing systems by digital computer," *Trans. IRE*, vol. PGIT-4, pp. 76–84; September, 1954.

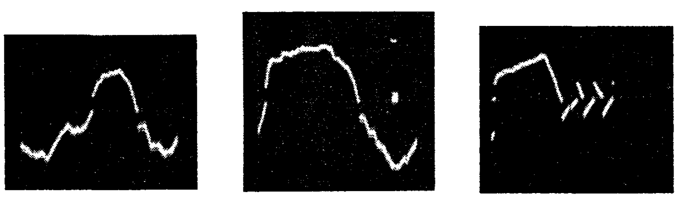
[2] In this paper, the word "pattern" is synonymous with "configuration."

pattern), then all weights which just previously participated in firing an element are increased. If, on the other hand, the change was unfavorable, those weights are decreased. In our experiments the weights have values between 0 and 15 inclusive, and changes are made one unit at a time.

It is important to note that there is no detailed examination of the internal activity of the net. As a result, some of the weights may be altered in the wrong direction at any given time. However, as our results show, in the long run an over-all favorable result occurs, due to what has been termed "statistical cooperation."

Two refinements were added to the system which allow somewhat improved operation. The first is to control the level of activity in the net by manipulating a bias parameter added to all thresholds. Before each input run, the bias is set at a high value, so that no activity occurs, and it then lowers to a point somewhat below that at which activity begins. A second refinement is to add random noise to the weight sums. This tends to break up any short period activity which may occur and which may "stall" the modifier action. In other words, favorable modes of activity are introduced which may not otherwise occur.

In order to automatize the "training" of the net, the system is arranged so that if at any time the output is different from zero, inputs are automatically given as follows: p_1 if the output is positive, and p_2 if it is negative. An exception to the foregoing is made for a small "dead" zone on either side of zero in which no input is given, thus allowing activity to die out on return to zero. "Training" is then given by artificially forcing the output from zero, waiting until its return, and then forcing it to the other side until return, etc. This process is kept up until organization is satisfactory. Fig. 2 shows such a sequence having output plotted vertically with center zero, and time horizontally. It can be seen that return to zero gradually improved until it was almost unhesitating. (Unfortunately, halation due to camera and 'scope face obscures most of the detailed movement.) This net was more difficult to train than most. Occasionally a net is seen which is not successful in training. This is not surprising since the deviation in the random connection set-up is considerable.



(a) (b) (c)

Fig. 2—Output record of 16-element net during organization. Input Patterns p_1 and p_2.

Purpose of Present Experiments

In the earlier work reported, only fixed input patterns p_1 and p_2 were used, as described above. However, if such a system always required precisely the same input pattern, its use would be considerably more restricted than if some variation were allowable. Therefore it is of interest to know whether the same type of organizing response can be obtained in input patterns which are caused to vary, particularly those which may contain a random variation.

Furthermore, consider the set of all possible input patterns to a net such as we have described. If the input period remains constant, there are 2^n such input patterns, each of which may be represented as a binary number of n bits, where n is the number of elements in the input group. For example, the patterns p_1 and p_2 which we have defined may be written as 11110000 and as 00001111, respectively, for a net with 8 input elements. Now, one of our nets will classify the 2^n input patterns into three classes which may be designated (+), (−), and (0), according to whether the input drives the output roughly upward, downward, or neither in a fixed time interval. An unorganized net may be expected to effect this classification at random, but it is of great interest to know what regularities, if any, may exist in the post-organization classification.

Both of the questions discussed above are of interest in practical applications of pattern discriminators or recognizers, and in studies concerning the behavior of living organisms. It is hardly likely, for example, that exactly the same pattern repeats itself very often on exactly the same cells of a retina.

Again, considerable interest attaches in the study of animal behavior to phenomena of the following sort. If a rat is trained to discriminate by suitable behavior between solid vertical and horizontal rectangles, he will, without further training, discriminate by the same behavior between vertical and horizontal rows of dots. This action of classifying patterns which have not previously been seen is called perceptual generalization in psychology.[3] It should be noted that the rules which the generalization follows must be functions of the system under consideration, and could, in principle, be quite arbitrary. In general, for living organisms the rules presumably hold because they have survival value. Naturally the rules of most interest to us are those which we use ourselves. It would be most interesting if rules similar to some of ours could be demonstrated as properties of nonlinear element networks. It is also of interest to note that at least one well-known neurophysiologist feels that the basis of generalization lies in the nervous tissue itself.[4]

For these reasons, then, as well as for their importance in mechanical pattern classification, experiments were carried out to test whether the systems described above would continue to organize themselves subject to

[3] D. O. Hebb, "The Organization of Behavior," John Wiley & Sons, Inc., New York, N. Y., p. 12 ff.; 1949.

[4] K. S. Lashley, "The Problem of Cerebral Organization in Behavior," Vol. VII of Biological Symposia, Cattell, London, England, p. 302; 1942.

randomly varying inputs, and to examine what kind of classifications are made on the input set by nets trained with fixed and varying input patterns.

For these purposes, 16-element nets (8 input and 8 output) were used because it was desired to exhaust all possible input patterns, and we were limited to about 2^8 inputs by available time.

Results of First Experiment

In the first series of experiments "noisy" input patterns were formed from the patterns 11110000 and 00001111 by complementing the pattern digits at random at the instant of input to the net with a fixed probability ranging from about 0.06 to 0.25 in various experiments. This process can be considered as varying each pattern randomly about its mean of 11110000 or 00001111.

These two sets of "noisy" patterns were then used instead of p_1 and p_2 in the same type of experiment as described above. Fig. 3 shows the results of an experiment of this type run with complementation probability of 0.25. The first two trials were run without modifier for comparison. After the modifier was turned on, it can be seen that organization takes place as before. Rather more detailed fluctuation occurs during organization, which is of course due to pattern fluctuation. Such records were found to be typical among the half-dozen or so experiments of this type. Again, one or two failed to organize properly, at least during the time of observation, because of unfavorable special properties of the random initial connections. It is felt that with larger nets, the percentage of failures would be even smaller.

(a) (b) (c)

Fig. 3—Output record of 16-element net during organization. Input patterns p_1 and p_2 with 0.25 pattern digit complementation probability.

Results of Second Experiment

The second series of experiments was carried out to determine input pattern classification after organization with fixed patterns. Fixed inputs 11110000 and 00001111 were used to organize a given net. The modifier was then disabled so that no further changes in the net could occur and all 256 possible input patterns were then presented in turn. The output was set to zero immediately before each pattern was presented for a fixed test period. At the end of the test period, it was determined whether the magnitude of the output was less or greater than a fixed quantity, and, if greater, whether the output was positive or negative. This information serves to classify the input pattern into one of the three groups mentioned above. After recording the class, the next pattern in turn was tested.

Since organization of the net with p_1 and p_2 tends in a general way to connect each input subgroup with its corresponding output subgroup, a plausible type of input classification to be expected would be that input patterns with a preponderance of 1's in a given subgroup might be classified the same. That is, the patterns

1011 0000
1111 0101
1100 0000

might be grouped in the (+) class, whereas

0010 1110
0011 1110
1000 0101

might be grouped in the (−) class, and distributions with a balanced number of 1's would then be expected to fall in the (0) or neutral class.

To test this hypothesis a matrix of order 5 as in (1) was formed for each class according to its count of patterns having a given preponderance of 1's in a subgroup.

$$\begin{array}{cccccc} & 0 & 1 & 2 & 3 & 4 \\ 0 & & & \vdots & & \\ 1 & \cdots & & a_{21} & & \\ 2 & & & & & \\ 3 & & & & & \\ 4 & & & & & \end{array} \tag{1}$$

An element a_{mn} in this matrix then represents "a" patterns having "m" 1's in the first subgroup and "n" 1's in the second. The sum of all the a's in all three matrices must then total 256, the total number of input patterns.

For comparison, all possible patterns are represented in the single matrix:

$$\begin{array}{ccccc} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{array} \tag{2}$$

Thus, of the 256 patterns, 16 have 3 1's in the first subgroup, and a single 1 in the second, etc.

The three classification matrices of a typical net before organization are shown in (3).

$$\begin{array}{ccccc} & & (+) & & \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 3 & 5 & 0 \\ 0 & 3 & 7 & 7 & 2 \\ 1 & 2 & 5 & 10 & 4 \\ 0 & 0 & 2 & 4 & 1 \end{array} \quad \begin{array}{ccccc} & & (-) & & \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 11 & 7 & 2 \\ 0 & 12 & 20 & 9 & 1 \\ 1 & 10 & 9 & 3 & 0 \\ 0 & 2 & 1 & 0 & 0 \end{array} \quad \begin{array}{ccccc} & & (0) & & \\ 1 & 4 & 6 & 2 & 1 \\ 4 & 16 & 10 & 4 & 2 \\ 6 & 9 & 9 & 8 & 3 \\ 2 & 4 & 10 & 3 & 0 \\ 1 & 2 & 3 & 0 & 0 \end{array} \tag{3}$$

Classification matrices of three nets after having been organized are shown in (4).

(+)					(−)					(0)					
0	0	2	0	0	0	0	0	0	1	1	4	4	4	0	
0	8	10	0	0	0	0	5	8	4	4	8	9	8	0	
1	20	16	2	0	0	0	4	1	5	5	4	16	11	1	
2	14	14	3	0	0	0	1	5	2	2	2	9	8	2	
1	4	4	0	0	0	0	0	1	0	0	0	2	3	1	
(+)					(−)					(0)					
0	0	0	0	0	0	0	3	3	1	1	4	3	1	0	
0	2	5	1	0	0	0	11	10	2	4	14	8	5	2	
1	7	14	4	0	0	0	16	8	1	5	17	6	12	5	(4)
2	9	12	2	0	0	0	8	2	0	2	7	4	12	4	
1	4	3	0	0	0	0	1	0	0	0	0	2	4	1	
(+)					(−)					(0)					
0	0	0	0	0	0	0	3	2	1	1	4	3	2	0	
0	4	9	3	0	0	0	7	6	3	4	12	8	7	1	
1	14	23	6	0	0	2	7	8	6	5	8	6	10	0	
3	13	16	3	0	0	1	3	6	4	1	2	5	7	0	
1	4	3	1	0	0	0	1	2	1	0	0	2	1	0	

The last net was operated with considerably more noise than usual added into the excitation at the element thresholds, as an additional experiment.

Upon inspection of the above matrices, it can be seen that the effect of the organization is to cause the entries in a (+) matrix to tend to cluster in the lower left corner, while those in a (−) matrix tend to cluster in the upper right corner. These tendencies are to be expected on the basis of our hypothesis, and perfect classification would make just three classes, namely lower left, upper right corner, and diagonal, corresponding to (+), (−), and (0), respectively. Of course, it might be expected that patterns next to the diagonal may be misclassified, inasmuch as they correspond to a preponderance of only a single 1.

Even in those cases where gross misclassifications occur, it will generally be found that they are a minority of the total possible cases in that cell. However, there are a few exceptions such as the 10 cases in the matrix cell 1, 2 in the first (+) matrix of (4).

When several of the patterns in a cell m, n are misclassified by two or more nets, it would not be expected that these would all be the same pattern, so that if all the nets were consulted for each pattern and majority rule accepted, the results should be improved. This plan was tried for the three nets above. The resulting matrices are shown in (5).

(+)					(−)					(0)					
0	0	0	0	0	0	0	2	2	1	1	4	3	2	0	
0	4	6	0	0	0	0	8	9	4	4	12	7	4	0	
1	14	18	0	0	0	0	7	7	5	6	10	7	8	1	(5)
2	14	13	0	0	0	0	2	2	2	2	3	4	9	2	
1	4	3	0	0	0	0	0	0	0	0	0	1	4	1	

Thirty-one patterns of the 256 were classified differently by each of the three nets, and are not included.

A definite improvement in classification can be seen in (5), especially in the (+) and (−) classes. A large fraction of possible cases are classified according to a preponderance of 1's in most cases. There are still quite a few (0) cases mixed up, but in many of these there is a preponderance of 1's of only one or two. This kind of error is probably accentuated in nets as small as these, because a maximum preponderance of only four is available. It would be expected that in larger nets the results would be improved, since they would have much greater preponderant differences, and should supply better statistics generally.

It should be noted also that no attempt was made to optimize the classification criteria themselves.

Classifications after organizing with "noisy" input patterns were also tried, with much the same results as those described above.

Discussion of Results

We have seen that the system under discussion generalizes from p_1 vs p_2 discrimination to a "1's preponderance" classification. The generalization is statistical in nature, and a number of nets in parallel with majority decision may be needed for more nearly perfect performance.

It is of interest to discuss the applications such a generalization may have to pattern recognition.

We will assume that the principle can be made to apply to systems with many inputs, as there is no known obstacle to using larger nets, or combinations of smaller ones, to achieve this result.

In the first place, generalizations of the horizontal-vertical kind mentioned earlier which are made by the rat, can be effected. Suppose that the whole picture field is made up of a mosaic of input elements, like a retina. Suppose also that the field itself is always mapped onto the same elements, i.e., that some kind of centering scheme is used. Then let the system be trained to distinguish between horizontal and vertical rectangles, as shown in Figs. 4a and 4a′, each rectangle being thought of as made up of a large group of input elements such as we have been considering. There will be an overlap of elements at the center, but total overlap elements are a small percentage of the total in either group and should present no difficulty to acquisition of the discrimination. Now, if preponderance generalization has taken place, discrimination will still take place "correctly" as between vertical and horizontal rows of dots, or other horizontal-vertical figures. In fact, as long as a preponderance of inputs of a test figure lies within the set of inputs of one of the original training patterns, the system will classify them alike. If some of the test figure inputs lie outside both the original patterns, no difficulty would be anticipated, since these inputs should remain neutral or at least inactive. Fig. 4 shows some of the cases in which successful horizontal-vertical discrimination would be expected.

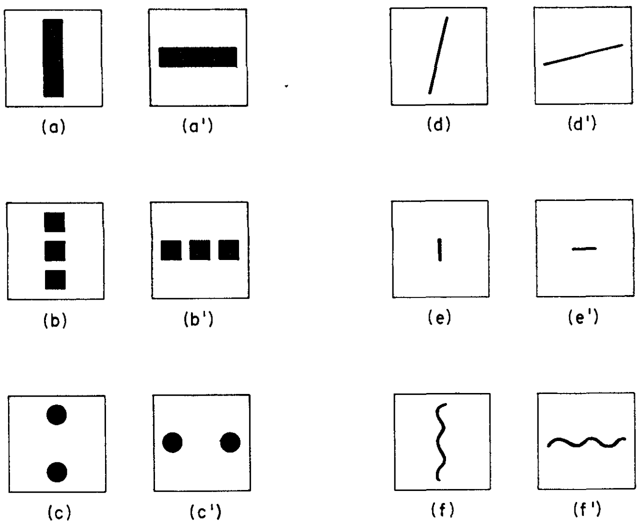


Fig. 4—Horizontal and vertical discrimination patterns.

There is still another possibility in generalization, if the input patterns contained fluctuation, either inherent or artificially introduced. We have seen that discrimination of such patterns can be organized by our system. During such training a "haze" of patterns would be expected to form about the "learned" pattern which will determine an additional set of patterns producing a like response, and hence being classified the same. In this way, a figure will still be classified correctly after having been translated or rotated slightly, for example. This is another instance in which a rat may be used as an example, since he will still recognize a triangle after it has been rotated a few degrees.[5]

These examples serve to show some of the features of pattern recognition which may be effected by a primitive generalization like that found to be a property of the self-organizing nets described. Comparisons to recognition behavior of a rat which have been made are not, of course, meant to imply that it is thought that the same mechanism is in operation, since there are other generalizations shown by the rat not possessed by our nets. However, it is clear that crude but useful generalization properties are possessed even by randomly connected nets of the type described.

Notes on the Simulation Program

The simulation program proceeds from a description of the class of networks under investigation and generates members of the class by selection of particular values of the description parameters at random. These parameters include quiescent thresholds, refractory periods, firing delay periods, input and output group size, threshold and excitation decay constants, and certain parameters which prescribe the manner in which elements are to be interconnected. Element i is connected to element j with a probability P_{ij} which can be made to depend on i and j and on any characteristics of the network as a whole. In the class of networks we have studied, P_{ij} is constant and equal to 0.75, i.e., the networks all have a constant mean connectivity of 0.75.

In carrying out the simulated activity of a net, the time parameter is held fixed while the program scans through a list of the elements calculating current values of threshold and excitation, and carries out the firing cycle changes as required. The time parameter is then advanced one unit and the process is repeated.

For a net of 16 elements, the real time consumed during one simulated time unit is about one-tenth of a second, varying according to the amount of activity in the net. The training of a typical net takes place in about 5 minutes. The subsequent classification of the complete set of 256 input patterns requires about 45 minutes.

The simulation program required about 1,000 16-bit registers of storage. Utility programs such as printouts, displays, and punched tape routines occupied another 1,000 registers. The remainder of the 4,096 registers of the MTC memory was divided between storage of the network structure itself and summary tables for classified patterns.

It may be of interest to note some of the computer design features which are desirable in simulation work of this kind. Most important, of course, is a large random-access memory of high speed. While this is generally taken as a basic requirement of any modern general-purpose computer, it is particularly important in programs which deal with the interaction of many elements with one another. In general, it would be more useful to have this memory in the form of many registers of small word-length rather than few registers of large word-length. In lieu of a "broad, shallow" memory, special instructions which permit the efficient extraction and replacement of parts of a stored word would be very valuable.

In many respects a simulation program of this type is like a large problem in bookkeeping and as such would make efficient use of design features which facilitate the repetition of operations on successive sets of data. For example, an arithmetic element capable of performing parallel operations on several quantities simultaneously would materially reduce the computing time. Similarly, "index registers" which modify the address part of indicated instructions in the program, so that the instructions refer automatically to successive sets of data, might well increase the effective computing speed by a factor of two in this application.

The requirement of randomness is generally met by the use of "pseudo-random" number sequences calcu-

[5] K. S. Lashley, "In Search of the Engram," No. IV of Symposia Soc. for Experimental Biology," Physiological Mechanisms in Animal Behavior," Academic Press, New York, N. Y., p. 473; 1950.

lated by the program. In the event that the simulation experiment makes extensive use of such randomness it would be desirable to incorporate a source of uniformly distributed random numbers as one of the electronic elements of the computer. Such an element would, of course, also be of great value in statistical work and monte-carlo calculations in general.

Finally, it is worth mentioning that simulation experiments involving partially random program behavior, unlike arithmetic computations, generally require the presence of the experimenter at the computer, at least during the program checkout phase and subsequently whenever large changes in operating parameters are made. For this reason any features which assist the experimenter in evaluating the operation of various parts of the program "on the spot" are of great value. In this category one might include programmed cathode-ray tube displays, audio output, and the ability to print out selected memory registers without stopping the computer.

Pattern Recognition and Modern Computers*

O. G. SELFRIDGE†

Introduction

WE CONSIDER the process we call Pattern Recognition. By this we mean the extraction of the significant features of data from a background of irrelevant detail. What we are interested in is simulating this process on digital computers. We give examples on three levels of complexity corresponding to the subjects of the other three speakers here today. We examine in detail the problem on the second level, visual recognition of simple shapes.

Finally, we show how our attack on that problem can be extended so that the computer is essentially performing a learning process and constructing new concepts on the basis of its experience.

Pattern Recognition

By pattern recognition we mean the extraction of the significant features from a background of irrelevant detail. We are interested in simulating this on digital computers for several reasons. First, it is the kind of thing that brains seem to do very well. Secondly, it is the kind of thing that computing machines do not do very well yet. Thirdly, it is a productive problem—it leads naturally to studying other processes, such as learning. And, finally, it has many fascinating applications on its own.

We shall not review here the valuable work that has been done and is being done elsewhere.

Examples of Pattern Recognition

Consider Fig. 1. The horizontal lines on the left differ from those on the right in having vertical spikes mostly at the left end. That is, here there are two patterns: those with a preponderance at the left end and those with a preponderance at the right end. The notion of simple preponderance or elemental discrimination is clearly one of the most primitive sources of patterns.

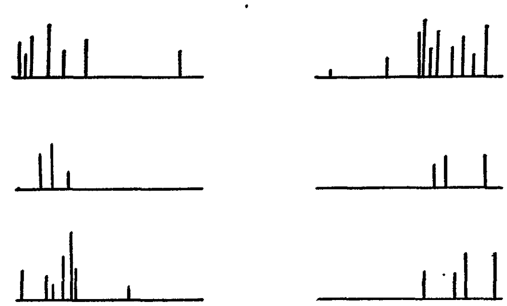


Fig. 1

Here we have filtered each line from perhaps 100 bits down to just one. It is this filtering that is pattern recognition.

Our next example is the visual recognition of simple shapes. This is a two-dimensional problem, of course, while the previous one was merely one-dimensional. Both the shapes in Fig. 2 are clearly squares though (1) they are in different places, (2) they have different sizes, (3) one is hollow, the other not, and (4) they have different orientations.

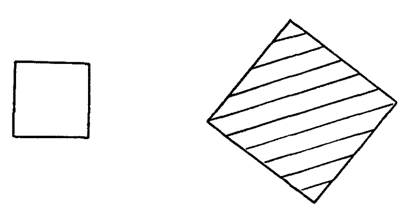


Fig. 2

Our final example, like our first, divides all the configurations of data into two classes. From every chess

* The work in this paper was sponsored jointly by the U. S. Army, U. S. Navy, and U. S. Air Force under contract with M.I.T.

† Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, Mass.