

RADC-TDR-63-467

Nils J. Nilsson
Computer Science Department
Stanford University
Stanford, CA. 94305-4110



**DETERMINATION AND DETECTION OF
FEATURES IN PATTERNS**

TECHNICAL DOCUMENTARY REPORT NO. RADC-TDR-63-467

December 1963

**Information Processing Branch
Rome Air Development Center
Research and Technology Division
Air Force Systems Command
Griffiss Air Force Base, New York**

Project No.5581, Task No.558104

**(Prepared under Contract No. AF30(602)-2943 by H. D. Block,
Cornell University, Ithaca, New York, N. J. Nilsson and R. W. Duda,
Stanford Research Institute, Menlo Park, California)**

DDC AVAILABILITY NOTICE

Qualified requesters may obtain copies from the Defense Documentation Center (TISIR), Cameron Station, Alexandria, Va., 22314. Orders will be expedited if placed through the librarian or other person designated to request documents from DDC.

Release to OTS.

LEGAL NOTICE

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

DISPOSITION NOTICE

Do not return this copy. Retain or destroy.

FOREWORD

The research reported in this paper was supported in part by the Office of Naval Research (Contract Nonr-3438(00)), in part by the U. S. Public Health Service (Contract PHT 1-77B-62), in part by the U. S. Air Force (RADC) (Contract AF30(602)-2943), and in part by Stanford Research Institute.

Suggested Keywords: Artificial intelligence, learning machines,
adaptive mechanisms, perceptrons, pattern
recognition, feature detection

ABSTRACT

In this paper feature determination as a method of training the first layer of weights in a two layer learning machine (Perceptron) is investigated. The problem is viewed as one of examining a set of patterns and determining a set of simpler patterns, or features, so that each of the original patterns can be formed by superposing the features. While the general problem of finding a minimal set of features was not solved, two algorithms were given that solve the problem for restricted pattern sets.

PUBLICATION REVIEW

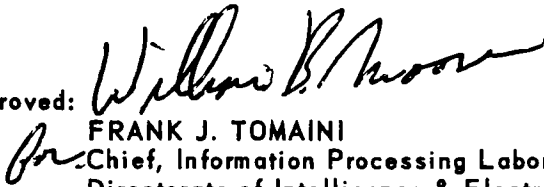
This report has been reviewed and is approved. For further technical information on this project, contact Mr. Fred A. Dion, RAWID, Ext. 5146.

Approved:



FRED A. DION
Project Engineer
Directorate of Intelligence & Electronic Warfare

Approved:



FRANK J. TOMAINI
Chief, Information Processing Laboratory
Directorate of Intelligence & Electronic Warfare

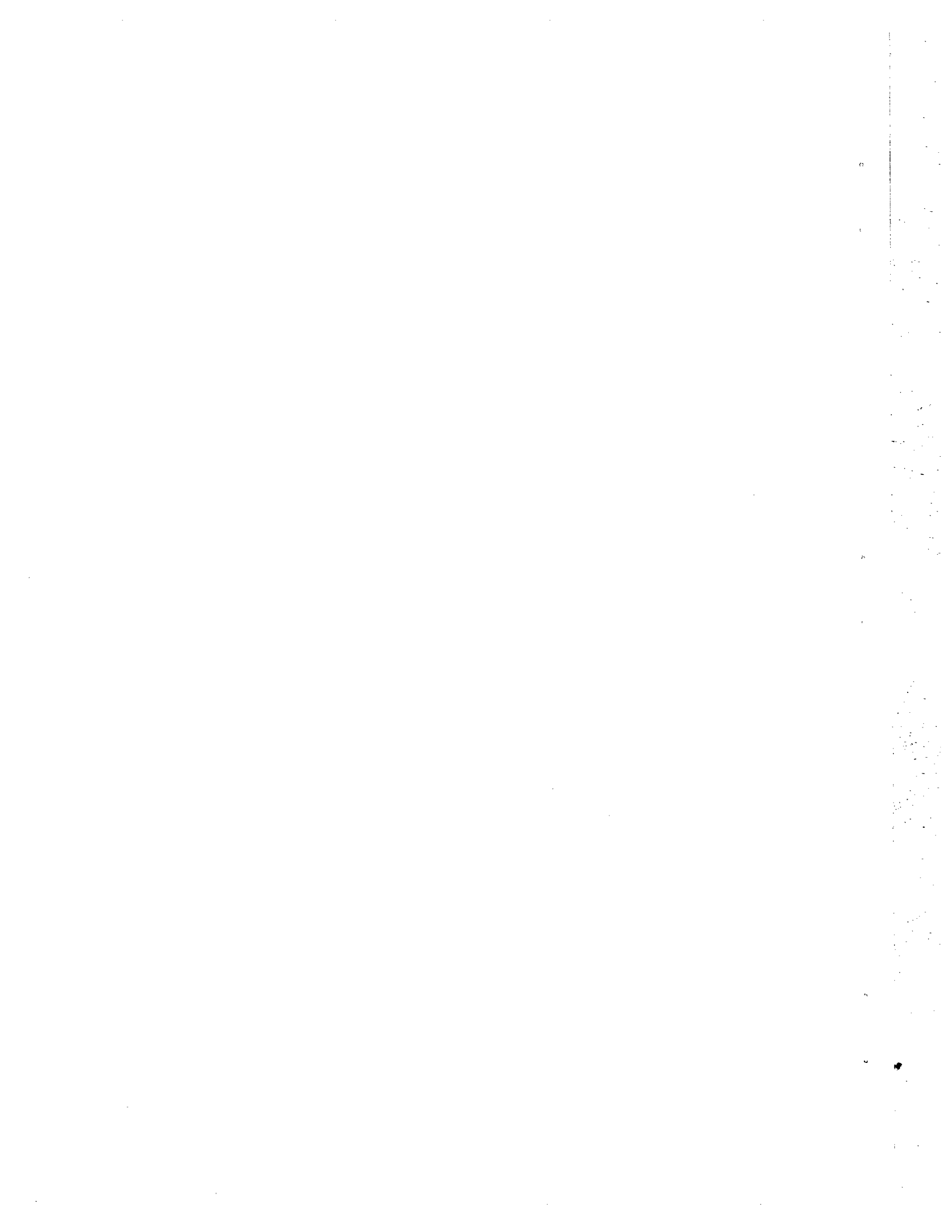


TABLE OF CONTENTS

CONTENTS	PAGE
I INTRODUCTION	1
A. TWO-LAYER LEARNING MACHINES	1
B. REPRESENTATION OF PATTERNS AND WEIGHT SETS	2
C. FEATURE DETERMINATION AS A MEANS TO ORGANIZE THE FIRST LAYER OF WEIGHTS	2
D. STATEMENT OF THE PROBLEM	6
II MATHEMATICAL FORMULATION	8
A. THE GENERAL PROBLEM	8
B. A GENERAL SOLUTION	9
III ALGORITHMS	11
A. THE THRESHOLD CONDITION	11
B. TRAINING ONE ASSOCIATOR	12
C. A SEQUENTIAL ALGORITHM	14
D. A PARALLEL ALGORITHM	16
E. SELECTION OF THE THRESHOLD	18
IV CONCLUDING COMMENTS	19
V ACKNOWLEDGMENT	19
APPENDIX A - FEATURE DETECTION AND EFFICIENT MACHINE ORGANIZATION	20
APPENDIX B - CONVERGENCE PROOF FOR THE SEQUENTIAL ALGORITHM	27
REFERENCES	36



I INTRODUCTION

A. TWO-LAYER LEARNING MACHINES

A typical perceptron-type learning machine consists of layers of *associators* or threshold elements.¹ The inputs to the learning machine are connected, through weights, to the associators in the first layer. The outputs of the first-layer associators are connected, through weights, to the associators in the second layer.

In a two-layer learning machine, the outputs of the associators in the second layer are taken to be the outputs of the machine. Such a two-layer machine is illustrated in Fig. 1.

The ensemble of inputs to the machine at a given instant is called a *pattern*; the ensemble of outputs is called a *response*. A central problem is to find a set of weight values in the first and second layers such that the machine responds to a list of patterns in accordance with some prescribed set of responses. Generally, the task of specifying these weight values as the result of a direct calculation is impractical, and one looks for algorithms by which the weight values can be iteratively modified while the machine is being exposed to a set of representative patterns from the list. The employment of such an algorithm *designs* the machine by a process commonly called *training*.

Algorithms have been proposed that specify how to adjust the values of the weights in one of the layers if those in the other layer remain fixed. Training procedures for the α -perceptron¹ (to adjust the weights in the second layer) and for the Madaline² (to adjust the weights in the first layer) are examples. No universally successful methods, however, are known for adjusting the weights in both layers simultaneously.

This paper presents a procedure for adjusting the weights in the first layer that can be performed without regard to the way in which the second layer is trained. The weights in the first layer are adjusted during training until the first-layer associators *detect* significant *features* in the patterns. (Precisely what constitutes a feature will be described later.) The weights in the second layer can then be adjusted

to obtain the prescribed responses. Thus, the problem of adjusting the weights in the first layer is viewed as a problem of *feature determination*.

B. REPRESENTATION OF PATTERNS AND WEIGHT SETS

For ease of explanation, suppose that the inputs to the learning machine are arranged in a rectangular array or *retina*, so that every possible binary pattern can be represented by a mosaic of black and white cells. Each of these mosaics will be called an *image*. The image illustrated in Fig. 2 represents a pattern on a 5×5 retina. This image can also be represented by the binary 25-tuple

(1,1,1,1,1,0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,)

where a "1" corresponds to a black cell and a "0" corresponds to a white cell. The correspondence between each cell in the retina and each component of the 25-tuple is defined according to the scanning convention indicated by Fig. 3.

We shall restrict the weights in the first layer to the values *one* and *zero*. An image will also be used to represent the values of a set of binary weights incident on each first-layer associator. A black cell will be used to represent a weight value equal to *one*, and a white cell will be used to represent a weight value equal to *zero*. If there are K first-layer associators, then K images completely describe the wiring between the learning-machine input terminals and the first layer of associators. If the word *associator* is taken to include the set of weights as well as the usual summing and threshold devices, then each possible binary-weight associator can be represented by an image together with a threshold value.

An associator will be said to be *matched* to a pattern if its image representation is identical with the image representation of that pattern. Such an associator plays the role of a *template*.

C. FEATURE DETERMINATION AS A MEANS TO ORGANIZE THE FIRST LAYER OF WEIGHTS

Consider the problem of selecting the values of the first-layer weights in an α -perceptron. One well-known solution is to choose these weights randomly.¹ An alternative is to provide a first-layer associator

matched to each pattern (see Fig. 4). Indeed, matching or template techniques are quite usefully employed in pattern recognition when the patterns tend to "cluster" around prototypes. For more diffuse sets of patterns, however, the number of different templates needed becomes prohibitively large.

If it is reasonable to assume that each pattern is composed of simpler patterns or *features*, then a matching scheme can still be used. The features are then building blocks of the complete patterns, and *sub-templates* matched to features can be used (see Fig. 5). The number of features is usually much smaller than the number of patterns that can be composed from them, and, therefore, a sub-template matching scheme could be an economical solution to the problem of specifying the first layer of weights.

The problem of organizing the first layer of weights will be viewed as one of determining such features. Some important aspects of this problem can be illustrated by a simple example in which we construct patterns out of the six features shown in Fig. 6. The patterns will be formed, let us say, by combining any two distinct features from this set of six features. Thus, for example, the pattern of Fig. 2 is formed by the superposition of features F_1 and F_5 of Fig. 6. If we use all of the combinations of two features out of the six, we get $\binom{6}{2} = 15$ patterns. These are shown in Fig. 7.

We now ask the reader to forget, for a moment, the preceding discussion, and to suppose that the patterns of Fig. 7 were presented to him in some sequence, possibly with repetitions allowed. It would not be very long before he would discover that these fifteen patterns were, in fact, constructed out of the six features of Fig. 6. This discovery would enable him to organize a learning machine with six associators in the first layer, one matched to each feature, rather than fifteen associators, one matched to each pattern. Furthermore, these features would also be suitable for other patterns of similar structure, and thus would provide much greater flexibility than that provided by simple template matching. In general, if the environment of presented patterns is actually structured, in the sense that there are a few basic features out of which all the presented patterns are composed, then it is likely that substantial advantages can be realized by making use of this fact in the processing performed by the first layer of associators. The striking

economies that can be obtained by appropriate organization are demonstrated in Appendix A.

While the advantages of determining features are clear, the general problem of finding them is not easily solved. Consider again the set of patterns formed by superposing precisely two of the six features shown in Fig. 6, namely, the set of patterns shown in Fig. 7. Let us renumber the retinal cells by permuting the numbers (1, 2, ..., 25), and then again represent all images according to the scheme of Fig. 3, but with the new numbers assigned to the retinal cells of the image. To be specific, we take (at random) the following permutation:

Old Retinal Cell Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
New Retinal Cell Number	8	15	24	25	14	17	9	1	16	21	2	11	20	22	10	13	6	3	12	7	4	19	23	5	18

Under this permutation the 15 patterns shown in Fig. 7 go over into the 15 patterns shown in Fig. 8. The order of the patterns is (intentionally) not the same in Fig. 8 as in Fig. 7. The actual correspondence is as follows:

Figure 7	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
Figure 8	a	d	j	l	h	m	e	i	o	b	n	f	g	k	c

Since the permutation of the retinal cells is a one-to-one transformation, it is clear that the organizational structure of the 15 patterns of Fig. 8 is the same as that of Fig. 7, namely, there are six basic features and each of the 15 patterns is made up by superposing precisely two of these features.

Now we ask the reader to forget, for a moment, the preceding discussion and to suppose that the patterns of Fig. 8 were presented to him in some sequence, possibly with repetitions allowed. We invite him to try to determine the number of basic features involved and their composition, working under this assumed ignorance. We believe, that he will find, as we did, that the solution is not quite so trivial.* Of course

* This problem can, however, be solved by a simple algorithm, as has been pointed out to us by W. R. Lynn. For a slightly more challenging problem, use as the features all 5 horizontal and all 5 vertical bars. Form the patterns by taking any vertical plus any horizontal bar. The 25 patterns thus obtained are rearranged on the retina as before.

the reader might take the position that each retinal cell is a feature, since each pattern can be constituted by superposing a subset of these. This, however, would require 25 features, whereas we know that 6 suffice. Alternatively, he might take each of the 15 patterns to be a feature. This is open to the same objection. What is desired is a minimal set of features sufficient to account for all the given patterns.

Since the abstract mathematical structures represented by Fig. 7 and Fig. 8 are isomorphic, it is clear that our ease in solving the problem of Fig. 7 as compared to the difficulty of Fig. 8 reflects a psychological or physiological phenomenon, rather than a mathematical distinction between the problems. If the environment actually consists of "bars" as in Fig. 7, and if we know this in advance, then it is clearly advantageous to build "bar detectors" into a recognition system for this environment of patterns. However, if we know nothing about the structure of the patterns in advance, then any algorithm or adaptive process that is to lead to the determination of the features will have to be equally effective when applied to the patterns of Fig. 8 as it is to Fig. 7. Although "bars" may be likely to occur in everyday visual patterns, there are also situations in which nothing may be known in advance about the structure of the patterns. Suppose for example that each "retinal cell" represents, not a cell of a two-dimensional visual image but the presence of a certain feature already abstracted at a previous stage of the process. Thus, for example, the first retinal cell might represent the presence of a horizontal bar, the second might represent a moving spot in the center of the field, the third a shrill sound coming into the system, and so on. In this case there may be no obvious *a priori* features, although the features, or "syndromes," may exist, and we may find it very helpful indeed to know them.

As another example of a situation in which "bars" may not be the natural features, consider the qualitative analysis, say, of the chemical pollutants in a river. On the first day an examination of a sample of the water might reveal the presence of pollutant chemicals *A, B, C* in excess of a normal threshold. On the second day there may be excessive quantities of *C, D, E, F*. On the third day *A, B, C, G, H, I*, and so on. These findings for 15 days might be represented by data such as shown in Fig. 9. Here again we have 25 "retinal cells" and 15 patterns. Now, if it turns out that the occurrence or nonoccurrence of the various chemicals is not really independent, but in fact that daily pollution patterns can

be described in terms of a few features, or syndromes, then one would suspect that there is a reason for this and investigate the cause. For example, if a certain combination of chemicals occurs repeatedly, and if a particular upstream factory is known to have these chemicals as waste products, then one may be led to investigate whether this factory is giving sufficient treatment to its effluent.

D. STATEMENT OF THE PROBLEM

Suppose that matched to each feature occurring in the set of patterns there is an associator in the first layer. By observing the responses of the associators in the first layer, we could *reconstitute* the pattern being presented. This reconstitution would be performed by combining all features whose presence is detected.*

In learning machine pattern classification tasks, it is not necessary to reconstitute the pattern. One desires only to know to which of a relatively small number of categories each pattern belongs. If the first layer of associators is to be trained to detect features, the features can be limited to those that are most helpful in establishing the category of the input pattern. The number of features needed for pattern classification might be substantially smaller than the number needed for pattern reconstitution. Nevertheless, in this paper we shall confine ourselves to the following question: What algorithms can be used to direct the training of a layer of associators such that they eventually become matched to a set of features sufficient to *reconstitute* all of the patterns presented?

We shall begin by presenting a mathematical formulation that includes the following more general problem: Given a set of patterns, determine a set of features, minimal in number, such that each pattern can be formed by the superposition of a subset of these features. Unfortunately, we have not yet been able to find a reasonably brief algorithm to solve the

* The ability to reconstitute patterns from features suggests a means to transmit a set of high-resolution photographs over a low-bandwidth channel. Rather than transmit the black or white information about each cell in the retina (high bandwidth), we transmit only the information regarding which features are present and which are absent for any particular image. At the receiver, the high-resolution image is reconstituted by combining the features that were present. If the total number of features for any set of patterns is less than the total number of retinal cells, then the feature-detection technique can be a means for achieving bandwidth reduction in an image-transmission system.

problem in this degree of generality. However, we shall present two useful algorithms for the case in which the features are large compared to their mutual overlap. Results of experiments by digital computer simulation will be used to show that these algorithms can lead to valuable results, even when this restriction is not met.

II MATHEMATICAL FORMULATION

A. THE GENERAL PROBLEM

The problem of feature determination introduced in Section I can be formulated in set-theoretical terms as follows:

Let S be a given set of points $\{s_n\}$ ($n = 1, \dots, N$). (This represents a retina of N retinal cells.) Any subset of S is called an *image*. Let P_m ($m = 1, \dots, M$) be given images. These are called the *patterns*. This collection of M patterns is denoted by $\mathcal{P} = \{P_m\}$ ($m = 1, \dots, M$). A collection of K images $\mathcal{F} = \{F_k\}$ ($k = 1, \dots, K$) is called a *set of features for* \mathcal{P} if each P_m is the union of some subcollection of the F_k , i.e., if for each m there exists a subset $\sigma(m)$ of the integers $(1, \dots, K)$ such that

$$P_m = \bigcup_{k \in \sigma(m)} F_k \quad (m = 1, \dots, M) \quad (1)$$

Given \mathcal{P} , the general problem is to find a set of features $\{F_k\}$ ($k = 1, \dots, K$) for \mathcal{P} such that K is minimal.

Equation (1) will have no solution if K is too small. On the other hand, if $K \geq \min(N, M)$ a solution will clearly exist: if $M \leq N$, take $K = M$ and $F_k = P_k$ for $k = 1, \dots, M$; if $N < M$, take $K = N$ and $F_k = s_k$ for $k = 1, \dots, N$.

Even if a solution to Eq. (1) exists, it may not be unique, i.e., the features F_k and/or the selection $\sigma(m)$ may not be unique. For example, consider the set of patterns of Fig. 7, with patterns (a), (b), (c), and (d) deleted. If we let \mathcal{P} be the remaining set of eleven patterns, then the top horizontal bar occurs only in the presence of the right vertical bar. Instead of using the top horizontal bar as a feature, we could take the top horizontal bar plus any subset of the right vertical bar. We shall find it convenient to *normalize* to the "maximal size feature" possible. Thus, in this instance we would choose the pattern of Fig. 7(e) rather than the top horizontal bar as our first feature. In general, given a set of features $\mathcal{F} = \{F_k\}$ ($k = 1, \dots, K$), we can find a new set of

features $\bar{\mathcal{F}} = \{\bar{F}_k\}$ ($k = 1, \dots, K$) by taking \bar{F}_k as the intersection of all of the patterns containing F_k . That is,

$$\bar{F}_k = \bigcap_{P_n \supseteq F_k} P_n \quad (k = 1, \dots, K)$$

Clearly, $\bar{F}_k \supseteq F_k$ and $\bar{\mathcal{F}}$ is a set of features for \mathcal{P} . Furthermore, the \bar{F}_k satisfy the equation

$$\bar{F}_k = \bigcap_{P_n \supseteq \bar{F}_k} P_n \quad (k = 1, \dots, K) \quad (2)$$

Thus, we may replace any set of features by their "hulls," which satisfy Eq. (2).

Even with this normalization of the features, however, the selection $\sigma(m)$ may not be unique. For example, in the case just considered, the pattern of Fig. 7(e) can be represented either as the pattern consisting of the feature of Fig. 7(e) alone or of the feature of Fig. 7(e) plus the right vertical bar. Again, we shall find it convenient to *normalize* to the "maximal set." That is, in this instance we would include both features in the right hand side of Eq. (1). In general we shall include in the right hand side of Eq. (1) all features contained in the pattern. With this convention, Eq. (1) takes the form

$$P_n = \bigcup_{F_k \subseteq P_n} F_k \quad (n = 1, \dots, M) \quad (3)$$

and, if the features F_k have been normalized, then

$$F_k = \bigcap_{P_n \supseteq F_k} P_n \quad (k = 1, \dots, K) \quad (4)$$

B. A GENERAL SOLUTION

Equation (3) by itself provides a logically complete algorithm for testing whether a given set of features satisfies Eq. (1). Consequently, one obvious (but impractical) algorithm for solving the general problem of Eq. (1) with minimal K would be to start with $K = 1$, ($K = 0$ for mathematicians) and examine the possibility of a solution of Eq. (3) for all possible choices of the single feature F_1 . There are 2^N such cases.

If no solution exists for $K = 1$, take $K = 2$ and examine whether a solution exists for all possible choices of the pair of distinct features F_1, F_2 . There are $\binom{2^N}{2} = 2^{N-1}(2^N - 1)$ such cases. If there is no solution for $K = 2$, we try $K = 3$, and so on. A solution will be found for $K \leq \min(N, M)$, and the first such solution will clearly have K minimal. Although the number of cases at each step can be substantially reduced by considering only those choices for F_k ($k = 1, \dots, K$) that satisfy Eq. (4), this verification itself will take some computing, and we can hardly claim that this is a practical algorithm.

III ALGORITHMS

A. THE THRESHOLD CONDITION

The general problem we have posed is that of determining a set of features $\{F_k\}$ ($k = 1, \dots, K$), minimal in number, such that

$$P_n = \bigcup_{k: F_k \subseteq P_n} F_k \quad (n = 1, \dots, M) \quad (3)$$

where

$$F_k = \bigcap_{n: P_n \supseteq F_k} P_n \quad (k = 1, \dots, K) \quad (4)$$

In words, Eq. (4) states that the intersection of all patterns sharing a common feature is that feature. A basic problem, then, is that of determining whether or not all the patterns in a given group of patterns share a common feature.

Consider the simpler problem of determining whether or not two given patterns share a common feature. If they do, that feature is included in their intersection; if they do not, their intersection contains merely intersections of distinct features (*i.e.*, overlaps of features). This is illustrated in Fig. 10, where the pertinent features are assumed to be horizontal and vertical bars. Note that in this example the size of the intersection, *i.e.*, the number of retinal points it contains, is large when the patterns share a common feature, and is small when they do not. Henceforth we shall restrict our attention to such situations. More precisely, we shall assume that there exists a *threshold*, θ , such that any two patterns share a common feature if and only if the size of their intersection equals or exceeds this threshold.

An associator can be used to compare the size of the intersection of two patterns with the value of the threshold. Consider the patterns P_1 and P_2 , and the associator shown in Fig. 11. The associator has been matched to P_1 by setting the values of the weights to points in P_1 equal to one, and setting the values of the weights to points not in P_1 equal to zero. If P_2 is presented to the retina, then the sum S gives the size of the intersection $P_2 \cap P_1$. If the threshold of the associator is equal to the threshold θ ,

then the associator is active if and only if P_1 and P_2 share a common feature.

B. TRAINING ONE ASSOCIATOR

When such a threshold, θ , exists, a simple algorithm can be used to train an associator so that it eventually becomes matched to a feature. Consider the set of patterns shown in Fig. 12. Suppose that patterns P_1 , P_2 , and P_6 , and only those patterns, share a common feature, so that the feature is given by their intersection $P_1 \cap P_2 \cap P_6$. The associator can form this joint intersection by forming successive pairs of intersections, viz., $P_6 \cap [P_2 \cap P_1]$.

This result is achieved in the following way. The associator is matched to the first pattern, P_1 (see Fig. 11). The second pattern, P_2 is presented. If the associator becomes active, indicating that P_1 and P_2 share a common feature, then the associator is matched to the intersection $P_2 \cap P_1$; this is easily done by merely setting the value of the weights to points not in P_2 equal to zero. If the associator does not become active, indicating that P_1 and P_2 have no common features, no changes in the weight values are made. This process is now repeated with the other patterns. Whenever a pattern presented shares a feature with all previous patterns that activated the associator, that pattern also activates the associator; setting the values of the weights to points not in that pattern equal to zero results in the associator being matched to the intersection of all of these patterns.

This algorithm for training one associator can be stated mathematically as follows. Let $A(i)$ denote the image representing the weights of the associator after the i -th pattern has been presented, and let $\|I\|$ denote the number of retinal points in any image I . Then

$$A(i+1) = \begin{cases} A(i) \cap P_{i+1} & \text{if } \|A(i) \cap P_{i+1}\| \geq \theta \\ A(i) & \text{otherwise,} \end{cases} \quad (i = 0, \dots, M-1) \quad (5)$$

where

$$A(0) = \bigcup_{i=1}^N s_i \quad (6)$$

It is shown in Appendix B that after the M th pattern has been presented, $A(M)$ is one of the features if the following conditions are satisfied:

$$(1) P_m = \bigcup_{k \ni F_k \subseteq P_m} F_k \quad (m = 1, \dots, M) \quad , \quad (3)$$

$$(2) F_k = \bigcap_{m \ni P_m \supseteq F_k} P_m \quad (k = 1, \dots, K) \quad , \quad (4)$$

and

$$(3) \frac{1}{2} K(K-1)N_{\max} < \theta \leq N_{\min} - (K-1)N_{\max} \quad , \quad (7)$$

where

$$N_{\min} = \min_i \|F_i\| \quad (8)$$

and

$$N_{\max} = \max_{\substack{i,j \\ i \neq j}} \|F_i \cap F_j\| \quad . \quad (9)$$

The first two conditions are merely statements of the basic relations between patterns and normalized features. The third condition, *the threshold condition*, is a more precise statement of the fact that we are concerned with pattern sets for which the size of the intersection of any two patterns indicates whether or not they share a common feature. It is sufficient (although not necessary) to guarantee that the associator will be activated by a pattern if and only if it shares a feature with all previous patterns that activated the associator.*

* The threshold condition has been stated in a form that will be convenient later. If only one associator is to be trained, this condition can be relaxed by replacing the right side of Eq. (7) with N_{\min} .

C. A SEQUENTIAL ALGORITHM

The algorithm just described gives a way of determining one of the features. The remaining features could, in principle, be determined in the same way by permuting the order in which the patterns are presented. (In view of Eq. 4, such an ordering is always possible.) This scheme suffers from the difficulty that the number of permutations of the patterns necessary to guarantee the determination of every feature ($M!$) will usually be prohibitively large. If random permutations are used, it will usually result in several associators being matched to each feature. It seems desirable, in the interest of economy, to prevent more than one associator from becoming matched to the same feature. This can be done by raising the threshold of the associator being trained whenever the pattern presented contains features already determined. The amount that the threshold is raised depends upon the relation of these features to the image of the associator being trained. To be more specific, the threshold is increased by the size of the union of those features of the pattern that (a) have been detected by other associators, and (b) are contained in the image of the associator being trained.

One way to incorporate this mechanism for preventing the multiple determination of features is to begin by training the first associator as before. After one iteration of the patterns, the first associator has been matched to, say, the first feature. Then the second associator is trained, using this threshold-raising mechanism to prevent it from also becoming matched to the first feature. After the second iteration of the patterns, the second associator has been matched to, say, the second feature. This procedure is continued, a new associator being trained after each iteration of the patterns, until, after K iterations, all of the features have been found. This algorithm is referred to as the *sequential algorithm* because features are determined in sequence.

This algorithm can be stated mathematically as follows. Let $A_1(M), \dots, A_j(M)$ denote the images representing the weights of the first j associators after j iterations of the patterns. (These, of course, are supposed to be j different features.) Let $A_{j+1}(i)$ denote the image representing the weights of the $(j + 1)$ -th associator after the presentation of of the i th pattern on the $(j + 1)$ -th iteration. Let $K_{i,j}$ be the set of integers k ($0 < k < j$) defined by

$$K_{i,j} = \{k; 0 < k < j, \|A_k(M) \cap P_i\| \geq \theta, A_k(M) \subseteq A_j(i)\} . \quad (10)$$

Then

$$A_{j+1}(i+1) = \begin{cases} A_{j+1}(i) \cap P_{i+1} & \text{if } \|A_{j+1}(i) \cap P_{i+1}\| \geq \theta + \left\| \bigcup_{k \in K_{i+1,j+1}} A_k(M) \right\| \\ A_{j+1}(i) & \text{otherwise,} \end{cases} \quad (11)$$

$$(i = 0, \dots, M - 1)$$

$$(j = 0, \dots, K - 1)$$

where

$$A_{j+1}(0) = \bigcup_{i=1}^N s_i . \quad (12)$$

It is shown in Appendix B that the conditions given by Eqs. (3), (4), and (7) are sufficient to guarantee that after K iterations of the patterns, the images $A_1(M), \dots, A_K(M)$ will be the K features F_1, \dots, F_K .

It should be emphasized that of these conditions, it is the threshold condition,

$$\frac{1}{2}K(K-1)N_{\max} < \theta \leq N_{\min} - (K-1)N_{\max}, \quad (7)$$

that limits the generality of the results. It should also be noted that this condition was obtained by a "worst-case" analysis; in many situations the algorithm can be used to find useful features even though this condition is not met.

D. A PARALLEL ALGORITHM

Instead of training the associators one after another, we can obtain the features more rapidly by training several of the associators at once. One algorithm for such a training procedure starts with a single associator as before, and introduces new associators whenever they are needed for *reconstitution*. After a pattern has been presented and the weight changes have been made, a test is made to see if the union of the images of the active associators reproduces that pattern. If it does, the next pattern is presented. If it does not, a new associator matched to that pattern is introduced, and then the next pattern is presented.

This procedure for training several associators, which is referred to as the *parallel algorithm*, begins by matching the first associator to the first pattern. Suppose that, at the i th step, j associators are being trained. (At the second step, one associator is being trained.) Let $A_1(i), \dots, A_j(i)$ denote the corresponding images. When pattern P_{i+1} is presented, the new images are determined as follows:

- (1) A_1 is active if and only if $||A_1(i) \cap P_{i+1}|| \geq \theta$. If A_1 is active, $A_1(i+1) = A_1(i) \cap P_{i+1}$; if A_1 is inactive, $A_1(i+1) = A_1(i)$.
- (2) A_2 is active if and only if $||A_2(i) \cap P_{i+1}|| \geq \theta + \theta_2$. Here, $\theta_2 = ||A_1(i+1)||$ if (a) A_1 is active, and (b) $A_1(i+1) \subset A_2(i)$; otherwise, $\theta_2 = 0$. If A_2 is active,

$$A_2(i+1) = A_2(i) \cap P_{i+1} \quad ; \text{ if } A_2 \text{ is inactive,}$$

$$A_2(i+1) = A_2(i).$$

- (3) In general, A_l is active if and only if $||A_l(i) \cap P_{i+1}|| \geq \theta + \theta_l$. Here, $\theta_l = ||\bigcup_{k \in K_l} A_k(i+1)||$, where $k \in K_l$ if and only if (a) A_k is active ($k < l$) and (b) $A_k(i+1) \subseteq A_l(i)$. If A_l is active, $A_l(i+1) = A_l(i) \cap P_{i+1}$; if A_l is inactive, $A_l(i+1) = A_l(i)$.

After the j new images have been found, the union of the images of the active associators is formed. If this union yields pattern P_{i+1} , then the next pattern is presented. If it does not, a new associator matched to this pattern is introduced, and then the next pattern is presented. This procedure is continued for as many iterations of the pattern set as is needed to obtain the features.

The example given in Fig. 13 illustrates the operation of this algorithm. Except for numbering, the six patterns involved are the same as those shown in Fig. 12. The threshold θ is taken to be three. The procedure begins by matching A_1 to P_1 . Next, P_2 is presented, and since $||A_1(1) \cap P_2|| = 10 \geq 3$, A_1 is active, and $A_1(2) = A_1(1) \cap P_2$. Since P_2 can be reconstituted by A_1 , P_3 is presented. P_3 also activates A_1 , and $A_1(3) = A_1(2) \cap P_3$. Now, however, P_3 can not be reconstituted by A_1 alone, and a new associator A_2 matched to P_3 is introduced. The presentation of P_4 activates A_1 , and its image, $A_1(4)$, is reduced to the lower horizontal bar. Since A_1 is active and $A_1(4) \subseteq A_2(3)$, the threshold for A_2 is raised to $3 + ||A_1(4)|| = 8$; this prevents A_2 from becoming active, and thus prevents its image from being reduced to the same lower horizontal bar. The pattern P_4 can not be reconstituted, and a new associator A_3 matched to P_4 is introduced. The remaining steps are performed similarly, and the four features are found in less than two iterations of the pattern set.

The parallel algorithm often yields a set of features in fewer than the K iterations of the patterns required by the sequential algorithm. This speed is gained by training new associators before old associators have determined a feature. This complicates the action of the threshold-raising mechanism, however, and frequently leads to the determination of more features than are needed for reconstitution.

E. SELECTION OF THE THRESHOLD

In order to use either the sequential or the parallel algorithm, we must know the value of the threshold, θ . If the features were known, a threshold for the sequential algorithm could be obtained from the bounds given by Eq. (7). However, the determination of the features is our goal; furthermore, even if Eq. (7) cannot be satisfied by any value of θ , there may well exist a threshold for which the algorithm will disclose a set of useful features. An efficient, generally applicable method of determining such a threshold has not yet been found. However, it is often practical to repeat the procedure for several values of θ , and to select that value that gave the best results.

For example, consider the twenty-four patterns shown in Fig. 14. These patterns were constructed from seven features—horizontal, vertical, and diagonal bars. (For these features, incidentally, Eq. (7) can not be satisfied by any value of θ .) The results of using the parallel algorithm with a threshold of one are shown in Fig. 15. Of the fifteen features found, the seven largest features are sufficient to reconstitute all of the patterns. Similar results were obtained with a threshold of two (see Fig. 16). Higher thresholds led to features with excessive overlap, and poorer results; however, the features obtained using the lower thresholds are clearly useful.

IV CONCLUDING COMMENTS

In this paper we have investigated feature determination as a method of training the first layer of weights in a two-layer learning machine. The problem was viewed as one of examining a set of patterns and determining a set of simpler patterns, or features, so that each of the original patterns can be formed by superposing the features. While the general problem of finding a minimal set of features was not solved, two algorithms were given that solve the problem for restricted pattern sets.

These results suggest several other problems worth further study. On the other hand, one can seek useful algorithms that apply to less restricted or even unrestricted pattern sets. On the other hand, one can seek useful algorithms for pattern sets containing topological constraints characteristic of special patterns, such as visual patterns. In either case, the effects of noise and small distortions must be investigated to ensure the practicability of this approach. Finally, attention should be given to the problem of determining those features most valuable for pattern classification. A good solution to this problem would be a major contribution to the theory of self-organizing systems.

V ACKNOWLEDGMENT

The research reported in this paper was supported in part by the Office of Naval Research (Information Systems Branch) [Contract Nonr-3438(00)], U.S. Public Health Service (Contract PHT 1-77B-62), and in part by the U.S. Air Force (Rome Air Development Center) [Contract AF 30(602)-2943 FSC-A082], and in part by Stanford Research Institute.

APPENDIX A

FEATURE DETECTION AND EFFICIENT MACHINE ORGANIZATION

The purpose of this appendix is to show how the utilization of features may result in a much more efficient use of associators. We shall compare three different organizations of associators which can correctly categorize some simple patterns. The patterns will be formed out of straight-line features, and will be presented to a retina for which the gridwork is very fine.

Consider first the six lines shown in Fig. A-1. We shall call these the "ideal features." We shall form the set of "ideal patterns" by combining any three out of these six ideal features. Thus there are $\binom{6}{3} = 20$ ideal patterns, one of which is illustrated in Fig. A-2.

Now we introduce a complication. Suppose that the artist who is sketching the patterns is somewhat sloppy in positioning the features. For example, he might slightly displace or slightly rotate any given feature. In addition, he might sketch by making more than one try at drawing a given ideal line. Thus, for instance, in attempting to sketch the pattern of Fig. A-2 he might start by sketching the upper horizontal line of Fig. A-1(a). Figure A-3 shows five lines, any one of which might be the outcome of his effort to draw the top horizontal line. Let us call these five lines "equivalent representations" of the ideal line of Fig. A-1(a). Similarly, suppose that each of the six ideal lines of Fig. A-1 has, say, five equivalent representations. Thus a given ideal pattern, such as Fig. A-2, might be represented by any one of $(2^5 - 1)^3$ actual patterns, one of which is shown in Fig. A-4.

We shall also be interested in the case in which the artist is restricted to draw only a single line in his attempt to denote any ideal feature of Fig. A-1 ("drawing" rather than "sketching"). In this case, in attempting to draw Fig. A-1(a), he could use only one of the five lines in Fig. A-3. Then a given ideal pattern, such as Fig. A-3, can be represented by any one of 5^3 actual patterns. For brevity, let us call this *Case (b)*, and the case considered above, in which the artist can use any

one or more lines, *Case (a)*. Thus, although there are only $\binom{6}{3} = 20$ ideal patterns, there are actually $20(2^5 - 1)^3 \approx 6 \cdot 10^5$ distinct patterns that might be presented to the retina in *Case (a)*, and $20 \cdot 5^3 = 2500$ distinct patterns in *Case (b)*.

Suppose that we have available associators having an arbitrary number r of inputs, each with weight +1, and a threshold, θ . If $\theta = r$, the associator responds if and only if *all* of its inputs are active; if $\theta = 1$, it responds if and only if *at least one* of its inputs is active.

We now consider a system having for its input a retina on which the patterns will be projected, and having for its output twenty associators of the type described, corresponding to the twenty ideal patterns. It is desired that when any representative of an ideal pattern is shown to the retina, the corresponding associator should become active; otherwise it should remain inactive. The problem is to form a network of associators to accomplish this task. We shall examine three solutions and compare them as to the number of associators and the number of connections used.

For the purpose of computing the number of connections, we assume for simplicity that the number of retinal points in each ideal feature is a constant, p , and that this is also the number of retinal points in each equivalent representation of any ideal feature. Let $||P_i||$ denote the number of retinal points in the i th pattern. Then, in *Case (a)*, $3p \leq ||P_i|| \leq 15p$, and in *Case (b)*, $||P_i|| \approx 3p$.

For *Case (a)*, one such solution (Solution I) is shown in Fig. A-5, where the i th "internal" associator has a threshold of $||P_i||$ and has connections with unit weights to all of the retinal points of the i th pattern. Thus, each of these associators is activated by a particular pattern on the retina. Each output associator has a threshold equal to one, and connections of unit weights to those internal associators which represent the same ideal pattern. Thus each output associator is active in response to its "equivalence class" of patterns, as required.

This arrangement requires $20(2^5 - 1)^3$ associators (we shall not count the twenty "response" associators, since they must be present in any system, and we are interested in comparing systems). The number of connections is $6 \cdot 5 \cdot \binom{5}{2} p (2^5 - 1)^2 \cdot 2^4 \approx 4.6p \cdot 10^6$ in the first layer, and $20(2^5 - 1)^3$ in the second layer of connections. If we let N_A denote the number of

associators used (not counting response units), and N_C the number of connections, we have for this arrangement

$$N_A = 20(2^5 - 1)^3 \approx 6 \cdot 10^5 \quad (\text{A-1a})$$

$$N_C = 6 \cdot 5 \cdot \binom{5}{2} p(2^5 - 1)^2 \cdot 2^4 + 20(2^5 - 1)^3 \approx 4.6p \cdot 10^6 \quad (\text{A-1b})$$

An economy can be effected by considering the *Case (b)* and using a similar arrangement, where now there are only $\binom{6}{3} 5^3$ patterns to be considered. Here

$$N_A = \binom{6}{3} 5^3 = 2500 \quad (\text{A-2a})$$

and

$$N_C = 2500(3p + 1) \quad (\text{A-2b})$$

It is easy to see that this arrangement also solves *Case (b)*, and we call it *Solution II*.

A much more economical solution (*Solution III*) may be obtained by having each associator of the first layer respond to a particular realization of a line feature (see Fig. A-6). Each associator of the second layer represents the presence of a particular ideal feature, and the associators of the third layer (the responses) are activated by any representative of the corresponding ideal pattern.* The threshold in the first layer is p , the number of retinal points in a feature. There are $6 \cdot 5 = 30$ associators in the first layer and six in the second. Hence

$$N_A = 36 \quad (\text{A-3a})$$

The number of connections in the first set is $30p$, in the second $6 \cdot 5 = 30$, and in the third $3 \cdot 20 = 60$. Hence

$$N_C = 30p + 90 \quad (\text{A-3b})$$

* In the particular problem being discussed, the first layer of associators could be eliminated by connecting each associator of the second layer to that part of the retina covering a feature and its "perturbation." If the feature were less disjoint, however, this method might fail. We shall not discuss it further here.

A comparison of Eqs. (A-1a) and (A-1b) with Eqs. (A-2a) and (A-2b), and (A-3a) and (A-3b) shows the enormous savings that may be realized as a result of appropriate organization. The principle of "early generalization" to effect such economies has been pointed out by Rosenblatt.³

To compare these three types of organization in the general case, let the number of ideal features be F , and let the ideal pattern be composed by the superposition of precisely f ideal features. Let E denote the number of equivalent representations of each feature. Then for the three types of organization discussed above, the results are as follows:*

ORGANIZATION TYPE	N_A	N_C
I	$\binom{F}{f} (2^E - 1)^f$	$pEF \binom{F-1}{f-1} 2^{E-1} (2^E - 1)^{f-1} + \binom{F}{f} (2^E - 1)^f$
II	$\binom{F}{f} E^f$	$(fp + 1) \binom{F}{f} E^f$
III	$F(E + 1)$	$EF(p + 1) + f \binom{F}{f}$

Even more impressive economies are possible if the structural organization of the patterns is hierarchial. To illustrate this, let us elaborate the previous problem. Suppose that the patterns described above represent "letters" in an alphabet of twenty letters. Suppose the retina is extended to four times its original width, so that four-letter words of this alphabet can be placed on it (see Fig. A-7).

Suppose further that each letter can be positioned in each box in, say, seven ways. Although there are now $20^4 = 160,000$ ideal four-letter words, there are actually $20^4 \cdot 7^4 \cdot (2^5 - 1)^{12} \approx 3 \cdot 10^{26}$ distinct patterns that are possible on the enlarged retina in Case (a), and $20^4 \cdot 7^4 \cdot 5^{12} \approx 9 \cdot 10^{16}$ in Case (b). We suppose that we have as response units associators corresponding to each ideal four-letter word on the retina. It is desired to construct a system of associators such that when any particular representation of an ideal word is placed on the retina, the response unit that corresponds to that ideal word becomes active and all others remain inactive. Again we consider three types of solution, analogous to the previous example.

* Note that N_A does not include the number of response units, since these must be used in any case. N_C , however, does include the connections to the response units.

For Solution I we use an internal associator for each distinct pattern and then combine them into words, as shown in Fig. A-8. Here the number of associators used is

$$N_A = 20^4 \cdot 7^4 \cdot (2^5 - 1)^{12} \approx 3 \cdot 10^{26} \quad , \quad (\text{A-4a})$$

and the number of connections is

$$N_C = p \cdot 4 \cdot 5 \cdot 7 \cdot \binom{5}{2} (2^5 - 1)^2 2^4 \left[7 \binom{6}{3} (2^5 - 1)^3 \right]^3 + N_A$$

$$\approx (31p + 1)N_A \approx p \cdot 10^{28} \quad , \quad (\text{A-4b})$$

where, as before, p represents the number of retinal points in an ideal line feature of Fig. A-1. The number of inputs to a first-layer associator is $||P_i||$, where $||P_i||$ is now the number of points in the i th four-letter word. Clearly, in Case (a) $12p \leq ||P_i|| \leq 60p$; in Case (b), $||P_i|| = 12p$.

For Solution II we use a similar arrangement, but start instead with Case (b). It is easy to see that this arrangement will also solve Case (a), but now only $20^4 \cdot 7^4 \cdot 5^{12}$ internal associators are required in the middle layer. There will be $12p \cdot 20^4 \cdot 7^4 \cdot 5^{12}$ connections in the first set, and $20^4 \cdot 7^4 \cdot 5^{12}$ connections in the second. Hence we have in this case

$$N_A = 20^4 \cdot 7^4 \cdot 5^{12} \approx 10^{17} \quad , \quad (\text{A-5a})$$

and

$$N_C = (12p + 1)N_A \approx p \cdot 10^{18} \quad . \quad (\text{A-5b})$$

A much greater saving can be achieved if we organize the system so as to reflect the organization of this particular environment of patterns. This is illustrated in Fig. A-9. Here the first layer consists of associators that respond to a particular representation of a given line feature in a given box in a given position in the box. Thus there are $4 \cdot 6 \cdot 5 \cdot 7 = 840$ such associators. In the second layer, the five equivalent representations of a given line feature in a given box in a given position are combined to represent the ideal line in a given box in a given position in the box. The third combines the ideal lines into

ideal letters in a given box in a given position in the box. The fourth layer combines the seven possible positions of a given ideal letter in a given box. The fifth layer is the response units. The number of internal associators used in this system is therefore

$$N_A = 4 \cdot 6 \cdot 5 \cdot 7 + 4 \cdot 6 \cdot 7 + 4 \cdot 20 \cdot 7 + 4 \cdot 20 = 1648 \quad , \quad (A-6a)$$

and the number of connections is

$$\begin{aligned} N_C &= 840p + 5 \cdot 168 + 3 \cdot 560 + 7 \cdot 80 + 4 \cdot 20^4 \\ &= 840p + 643,080 \quad . \quad (A-6b) \end{aligned}$$

The enormous saving is evident. To compare the three types of organization in general, let L denote the number of letters in a word and P the number of ways in which a given letter can be positioned in a given box. With F , f , and E as defined earlier, the results are as follows:

ORGANIZATION TYPE	N_A	N_C
I	$\left[\binom{F}{f} P(2^E - 1)^f \right]^L$	$pEFPL \binom{F-1}{f-1} (2^E - 1)^{f-1} 2^{E-1} \left[P \binom{F}{f} (2^E - 1)^f \right]^{L-1} + N_A$
II	$\left[\binom{F}{f} PE^f \right]^L$	$(fLp + 1)N_A$
III	$L \left[FP(E+1) + \binom{F}{f} (P+1) \right]$	$L \left[FPE(p+1) + P \binom{F}{f} (f+1) + \binom{F}{f}^L \right]$

Another advantage to the organization of Solution III is that the number of inputs to a given associator is reduced. (Cf. Fig. A-5 with Fig. A-6, and Fig. A-8 with Fig. A-9.) If we think of the model as representing an industrial organization, with associators representing decision-making individuals, then, in view of the natural limitations of human capacity for handling information, such a reduction of input data may be essential for the individual's mental health.

In order to achieve the advantage of matching the organizational structure of the system with that present in the environment of the patterns, it appears that one must be able to either determine the structure

of the environment and design the system accordingly, or else formulate reinforcement rules by means of which the system will adapt its structure to that present in the environment. Since the patterns of active units at any given layer themselves represent input patterns to the subsequent layer, the algorithms developed can be used to find the features of the "patterns" in any given layer.

APPENDIX B

CONVERGENCE PROOF FOR THE SEQUENTIAL ALGORITHM

This appendix contains the convergence proof for the sequential algorithm. As before, we let $S = \{s_n\}$ ($n = 1, \dots, N$) denote the set of retinal points and $\mathcal{P} = \{P_m\}$ ($m = 1, \dots, M$) denote the set of patterns. We assume the existence of a set of features $\mathcal{F} = \{F_k\}$ ($k = 1, \dots, K$) such that

$$P_m = \bigcup_{k \ni F_k \subseteq P_m} F_k \quad (m = 1, \dots, M) \quad , \quad (B-1)$$

and

$$F_k = \bigcap_{m \ni P_m \supseteq F_k} P_m \quad (k = 1, \dots, K) \quad , \quad (B-2)$$

and the existence of a threshold θ such that

$$\frac{1}{2} K(K-1) N_{\max} < \theta \leq N_{\min} - (K-1) N_{\max} \quad , \quad (B-3)$$

where

$$N_{\min} = \min_i \|F_i\| \quad (B-4)$$

and

$$N_{\max} = \max_{\substack{i,j \\ i \neq j}} \|F_i \cap F_j\| \quad . \quad (B-5)$$

The algorithm is most conveniently stated in two parts. Let $A_1(i)$ denote the image representing the weights of the first associator after the i th pattern has been presented. Then the algorithm for changing the weights of the first associator is

$$A_1(i+1) = \begin{cases} A_1(i) \cap P_{i+1} & \text{if } \|A_1(i) \cap P_{i+1}\| \geq \theta \\ A_1(i) & \text{otherwise} \end{cases} \quad , \quad (B-6)$$

($i = 0, \dots, M-1$)

where

$$A_1(0) = \bigcup_{n=1}^N s_n \quad . \quad (B-7)$$

The presentation of all of the M patterns is called an *iteration* of the patterns. Let $A_1(M), \dots, A_j(M)$ denote the images representing the weights of the first j associators after j iterations of the patterns. Let $A_{j+1}(i)$ denote the image representing the weights of the $(j+1)$ -th associator after the presentation of the i th pattern on the $(j+1)$ -th iteration. Let $K_{i,j}$ be the set of integers $k (0 < k < j)$ defined by

$$K_{i,j} = \{k ; 0 < k < j, \|A_k(M) \cap P_i\| \geq \theta, A_k(M) \subset A_j(i)\} \quad . \quad (B-8)$$

Then the algorithm for changing the weights of the $(j+1)$ -th associator is

$$A_{j+1}(i+1) = \begin{cases} A_{j+1}(i) \cap P_{i+1} & \text{if } \|A_{j+1}(i) \cap P_{i+1}\| \geq \theta + \left\| \bigcup_{k \in K_{i+1,j+1}} A_k(M) \right\| \\ A_{j+1}(i) & \text{otherwise} \end{cases} \quad (B-9)$$

($i = 0, \dots, M-1$)
($j = 1, \dots, K-1$)

where

$$A_{j+1}(0) = \bigcup_{n=1}^N s_n \quad . \quad (B-10)$$

We shall show that if conditions (B-1), (B-2), and (B-3) are satisfied, then, after K iterations, the images $A_1(M), \dots, A_k(M)$ are the features F_1, \dots, F_k . We shall first show that after one iteration $A_1(M)$ is a feature, and we shall then show that, after $j+1$ iterations, $A_{j+1}(M)$ is a feature other than $A_1(M), \dots, A_j(M)$.

Part 1. Convergence of A_1

We begin by repeating the algorithm for A_1 .

$$A_1(i+1) = \begin{cases} A_1(i) \cap P_{i+1} & \text{if } \|A_1(i) \cap P_{i+1}\| \geq \theta \\ A_1(i) & \text{otherwise,} \end{cases} \quad (i = 0, \dots, M-1) \quad (B-6)$$

where

$$A_1(0) = \bigcup_{n=1}^N s_n \quad . \quad (B-7)$$

From Eq. (B-3),

$$\|A_1(0) \cap P_1\| = \|P_1\| \geq N_{\min} \geq \theta \quad ,$$

so that

$$A_1(i) = P_1 \quad . \quad (B-11)$$

If at the $(i + 1)$ -th step $\|A_1(i) \cap P_{i+1}\| \geq \theta$, we shall say that P_{i+1} activates A_1 . Thus, at the first step P_1 activates A_1 . Let the first n patterns that activate A_1 be denoted by P_{i_1}, \dots, P_{i_n} . Then after step i_n

$$A_1(i_n) = \bigcap_{j=1}^n P_{i_j} \quad . \quad (B-12)$$

Define $i_{n+1} = i_n + 1$, so that at the next step, pattern $P_{i_{n+1}}$ is encountered. One of two cases can arise:

Case (a): $P_{i_1}, \dots, P_{i_{n+1}}$ have at least one common feature.

Case (b): $P_{i_1}, \dots, P_{i_{n+1}}$ have no common features.

Case (a)

Let F_1 be a feature common to $P_{i_1}, \dots, P_{i_{n+1}}$. Then

$$F_1 \subseteq \bigcap_{j=1}^{n+1} P_{i_j} \quad , \quad (B-13)$$

and, from Eqs. (B-4) and (B-9),

$$\|A_1(i_n) \cap P_{i_{n+1}}\| \geq \|F_1\| \geq N_{\min} \geq \theta \quad . \quad (B-14)$$

Thus, in Case (a), $P_{i_{n+1}}$ activates A_1 .

Case (b)

Lemma 1:

Let I_j be a subset of the set of integers $\{i\}$ ($i = 1, \dots, K$), and let the collection of sets $\{I_j\}$ ($j = 1, \dots, m$) have the property that no integer is in every set of integers. Let $\mathfrak{F} = \{F_k\}$ ($k = 1, \dots, K$) be a collection of point sets. Then

$$S' = \bigcup_{j=1}^m \left(\bigcup_{i \in I_j} F_i \right) \subseteq \bigcup_{i=1}^{K-1} \bigcup_{j>i}^K F_i \cap F_j \quad (\text{B-15})$$

Proof: Let $s \in S'$. Then s cannot be a member of one and only one of the sets of \mathfrak{F} , for, were it so, there would be an I_j such that $s \in \bigcup_{i \in I_j} F_i$ and hence $s \in S'$. Thus $\exists (i, j)$, $i \neq j$, $s \in F_i$ and $s \in F_j$. Thus $s \in F_i \cap F_j$, and, since all possible pairs of intersections appear in the right side of Eq. (B-15), the lemma is proved.

Now suppose that $P_{i_1}, \dots, P_{i_{n+1}}$ have no common features. Then, from Eqs. (B-9) and (B-1) and the lemma,

$$A_1(i_n) \cap P_{i_{n+1}} = \bigcap_{j=1}^{n+1} P_{i_j} = \bigcap_{j=1}^{n+1} \left(\bigcup_{i \in I_j} F_i \right) \subseteq \bigcup_{i=1}^{K-1} \bigcup_{j>i}^K F_i \cap F_j, \quad (\text{B-16})$$

and

$$\begin{aligned} \|A_1(i_n) \cap P_{i_{n+1}}\| &\leq \left\| \bigcup_{i=1}^{K-1} \bigcup_{j>i}^K F_i \cap F_j \right\| \\ &\leq \sum_{i=1}^{K-1} \sum_{j>i}^K \|F_i \cap F_j\| \\ &\leq \sum_{i=1}^{K-1} \sum_{j>i}^K N_{\dots} \\ &= \frac{1}{2} K(K-1) N_{\dots} \\ &< \theta \end{aligned} \quad (\text{B-17})$$

Thus, in Case (b), $P_{i_{n+1}}$ does not activate A_1 .

It follows that $P_{i_{n+1}}$ activates A_1 if and only if the patterns $\{P_{i_1}, \dots, P_{i_{n+1}}\}$ have at least one feature in common. After M steps, let $\{P_{i_1}, \dots, P_{i_{n_0}}\}$ be the set of patterns that activated A_1 , and let F_1 be one feature they share. Then

$$A_1(M) = \bigcap_{j=1}^{n_0} P_{i_j} \quad . \quad (B-18)$$

But $F_1 \subset P_{i_j}$ ($j = 1, \dots, n_0$), and F_1 is not contained in any other pattern. Thus by Eq. (B-2),

$$A_1(M) = F_1 \quad . \quad (B-19)$$

Part 2. Convergence of A_{j+1}

We begin by repeating the algorithm for A_{j+1} .

$$A_{j+1}(i+1) = \begin{cases} A_{j+1}(i) \cap P_{i+1} & \text{if } \|A_{j+1}(i) \cap P_{i+1}\| \geq \theta + \left\| \bigcup_{k \in K_{i+1, j+1}} A_k(M) \right\| \\ A_{j+1}(i) & \text{otherwise} \end{cases} \quad , \quad (B-9)$$

($i = 0, \dots, M-1$)
($j = 0, \dots, K-1$)

where

$$A_{j+1}(0) = \bigcup_{n=1}^N s_n \quad (B-10)$$

and

$$K_{i,j} = \{k ; 0 < k < j , \|A_k(M) \cap P_i\| \geq \theta , A_k(M) \subset A_j(i)\} \quad . \quad (B-8)$$

Suppose that the algorithm has operated successfully for A_1, \dots, A_j and we are starting the $(j+1)$ -th iteration. Then $A_1(M), \dots, A_j(M)$ are j distinct features; for convenience we number them so that

$$A_i(M) = F_i \quad (i = 1, \dots, j) \quad . \quad (B-20)$$

If at the $(i + 1)$ -th step

$$\|A_{j+1}(i) \cap P_{i+1}\| \geq \theta + \left\| \bigcup_{k \in K_{i+1, j+1}} A_k(M) \right\|$$

we shall say that P_{i+1} activates A_{j+1} . Let the first n patterns that activate A_{j+1} be denoted by P_{i_1}, \dots, P_{i_n} .^{*} Then after step i_n

$$A_{j+1}(i_n) = \bigcap_{l=1}^n P_{i_l} \quad . \quad (\text{B-21})$$

Define $i_{n+1} = i_n + 1$, so that at the next step pattern $P_{i_{n+1}}$ is encountered. One of two cases can arise:

Case (a): $P_{i_1}, \dots, P_{i_{n+1}}$ have at least one common feature, besides perhaps some or all of F_1, \dots, F_j .

Case (b): $P_{i_1}, \dots, P_{i_{n+1}}$ have no common features, besides perhaps some or all of F_1, \dots, F_j .

Before considering these cases in detail, we shall establish some useful facts.

Lemma 2:

$$\|F_i \cap P_j\| \geq \theta \text{ if and only if } F_i \subseteq P_j.$$

Sufficiency: If $F_i \subseteq P_j$, then $F_i \cap P_j = F_i$, and

$$\|F_i \cap P_j\| = \|F_i\| \geq N_{\min} \geq \theta \quad .$$

Necessity: If $F_i \not\subseteq P_j$, then $F_i \cap P_j \subseteq F_i \cap \left(\bigcup_{j \neq i} F_j \right)$, and

$$\|F_i \cap P_j\| \leq \left\| \bigcup_{i \neq j} F_i \cap F_j \right\| \leq \sum_{i \neq j} \|F_i \cap F_j\| \leq (K - 1)N_{\max} \quad .$$

If $K = 1$, $\|F_i \cap P_j\| = 0 < \theta$. If $K \leq 2$, $\frac{1}{2} K \geq 1$, and

$$\|F_i \cap P_j\| \leq \frac{1}{2} K(K - 1)N_{\max} \leq \theta \quad . \quad \text{Q.E.D.}$$

^{*} By adopting the convention $P_{i_1} = \bigcup_{n=1}^N P_n$ we can avoid the need for a separate proof that some pattern will activate A_{j+1} .

Consider now the set of integers $K_{i_{n+1}, j+1}$,

$$K_{i_{n+1}, j+1} = \left\{ k ; 0 < k < j + 1 , \|A_k(M) \cap P_{i_{n+1}}\| \geq \theta , A_k(M) \subseteq A_{j+1}(i_{n+1}) \right\} .$$

From Eqs. (B-20) and (B-21), and Lemma 2, we can write this as

$$K_{i_{n+1}, j+1} = \left\{ k ; 0 < k < j + 1 , F_k \subseteq P_{i_{n+1}} , F_k \subseteq \bigcap_{l=1}^n P_{i_l} \right\} . \quad (\text{B-22})$$

Let F^c denote the union of all of the features from F_1, \dots, F_j that are common to $P_{i_1}, \dots, P_{i_{n+1}}$. Clearly

$$K_{i_{n+1}, j+1} = \{k ; F_k \subseteq F^c\} \quad (\text{B-23})$$

and

$$\bigcup_{k \in K_{i_{n+1}, j+1}} A_k(M) = F^c \quad (\text{B-24})$$

Case (a)

In Case (a), $P_{i_1}, \dots, P_{i_{n+1}}$ have at least one common feature, besides perhaps some or all of F_1, \dots, F_j , which we number as F_{j+1} . Then

$$P_{i_l} = F^c \cup F_{j+1} \cup F^{i_l} \quad (l = 1, \dots, n+1) \quad , \quad (\text{B-25})$$

where F^{i_l} is the union of those features in P_{i_l} and not in $F^c \cup F_{j+1}$. Then at step i_{n+1} , it follows from Eqs. (B-21), (B-25), (B-24), and (B-20) that

$$\begin{aligned} \|A_{j+1}(i_n) \cap P_{i_{n+1}}\| &= \left\| \bigcap_{l=1}^{n+1} P_{i_l} \right\| = \left\| (F^c \cup F_{j+1}) \cup \bigcap_{l=1}^{n+1} F^{i_l} \right\| \\ &\geq \|F^c \cup F_{j+1}\| = \|F^c\| + \|F_{j+1}\| - \|F^c \cap F_{j+1}\| \\ &\geq \|F^c\| + N_{\min} - \left\| \bigcup_{k \in K_{i_{n+1}, j+1}} F_k \cap F_{j+1} \right\| \\ &\geq \|F^c\| + N_{\min} - (K - 1)N_{\min} \quad , \quad (\text{B-26}) \end{aligned}$$

and thus, from Eqs. (B-3) and (B-24), that

$$\|A_{j+1}(i_n) \cap P_{i_{n+1}}\| \geq \theta + \left\| \bigcup_{k \in K_{i_{n+1}, j+1}} A_k(M) \right\| . \quad (\text{B-27})$$

Thus, in Case (a), $P_{i_{n+1}}$ activates A_{j+1} .

Case (b)

In Case (b), $P_{i_1}, \dots, P_{i_{n+1}}$ have no common features, besides perhaps some or all of F_1, \dots, F_j . Then

$$P_{i_l} = F^c \cup F^{i_l} \quad (l = 1, \dots, n+1) \quad (\text{B-28})$$

where F^{i_l} is the union of features in P_{i_l} and not in F^c . In particular, the F^{i_l} have no common features. Then at step i_{n+1} , it follows from Eqs. (B-21) and (B-28), and Lemma 1 that

$$\begin{aligned} \|A_{j+1}(i_n) \cap P_{i_{n+1}}\| &= \left\| \bigcap_{l=1}^{n+1} P_{i_l} \right\| \\ &= \left\| F^c \cup \bigcap_{l=1}^{n+1} F^{i_l} \right\| \\ &\leq \|F^c\| + \left\| \bigcap_{l=1}^{n+1} F^{i_l} \right\| \\ &\leq \|F^c\| + \left\| \bigcup_{i=1}^{K-1} \bigcup_{j>i}^K F_i \cap F_j \right\| \\ &\leq \|F^c\| + \sum_{i=1}^{K-1} \sum_{j>i}^K \|F_i \cap F_j\| \\ &\leq \|F^c\| + \frac{1}{2} K(K-1) N_{\max} , \end{aligned}$$

and thus, from Eqs. (B-3) and (B-24), that

$$\|A_{j+1}(i_n) \cap P_{i_{n+1}}\| < \theta + \left\| \bigcup_{k \in K_{i_{n+1}, j+1}} A_k(M) \right\| . \quad (\text{B-29})$$

Thus, in Case (b), $P_{i_{n+1}}$ does not activate A_{j+1} .

It follows that $P_{i_{n+1}}$ activates A_{j+1} if and only if the set of patterns $\{P_{i_1}, \dots, P_{i_{n+1}}\}$ have at least one feature in common, not counting the features that have already been detected by A_1, \dots, A_j . After M steps, let $\{P_{i_1}, \dots, P_{i_{n_0}}\}$ be the set of patterns that activated A_{j+1} , and let F_{j+1} be one feature besides F_1, \dots, F_j that they share. Then

$$A_{j+1}(M) = \bigcap_{l=1}^{l_0} P_{i_l} \quad . \quad (B-30)$$

But $F_{j+1} \subseteq P_{i_l}$ ($l = 1, \dots, l_0$), and F_{j+1} is not contained in any other pattern. Thus by Eq. (B-2),

$$A_{j+1}(M) = F_{j+1} \quad . \quad (B-31)$$

REFERENCES

1. Rosenblatt, F., *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms* (Spartan Books, Washington, D.C., 1961).
2. Widrow, B., "Generalization and Information Storage in Networks of Adaline 'Neurons'," in *Self-Organizing Systems—1962*, pp. 435-461, edited by M. C. Yovits, G. I. Jacoby, and G. D. Goldstein (Spartan Books, Washington, D.C., 1962).
3. Rosenblatt, F., "A Comparison of Several Perceptron Models," in *Self-Organizing Systems—1962*, pp. 435-461, edited by M. C. Yovits, G. I. Jacoby, and G. D. Goldstein (Spartan Books, Washington, D.C., 1962).

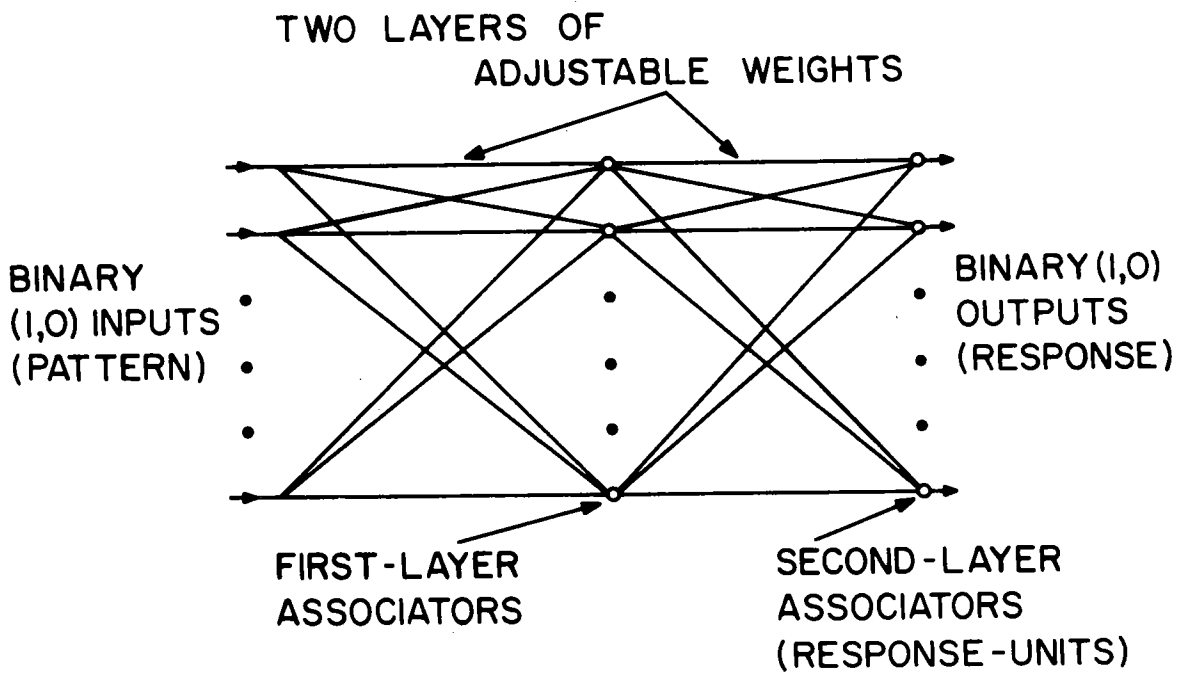


FIG. 1 A TWO-LAYER LEARNING MACHINE

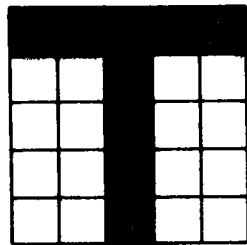


FIG. 2 REPRESENTATION OF A PATTERN BY AN IMAGE ON A RETINA

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

FIG. 3 SCANNING CONVENTION

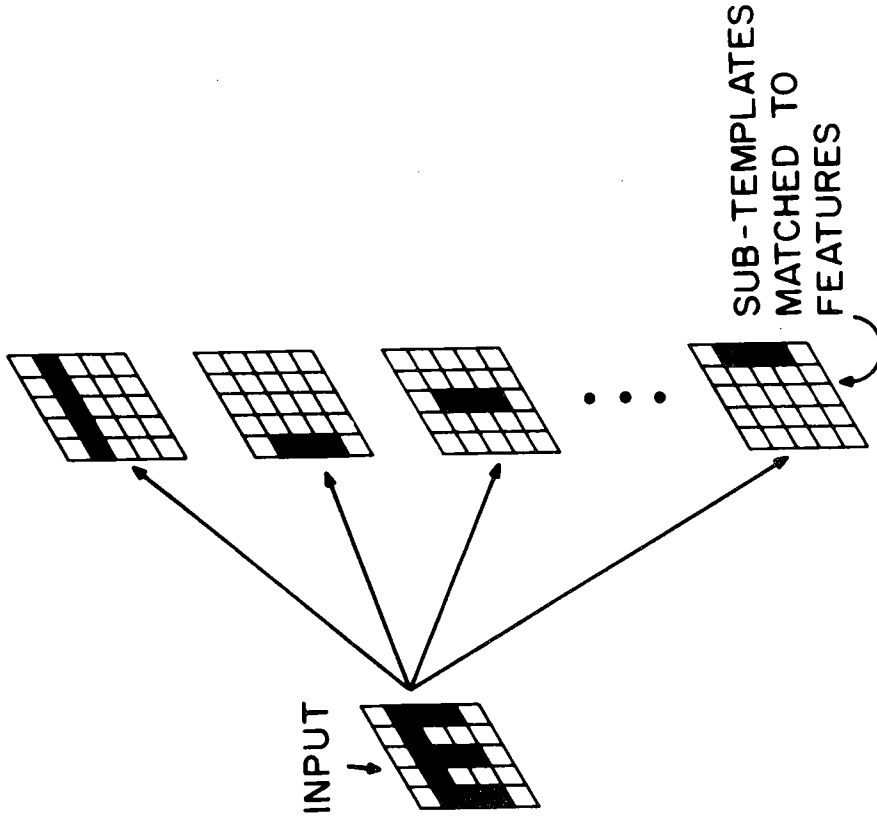


FIG. 5 FEATURE MATCHING

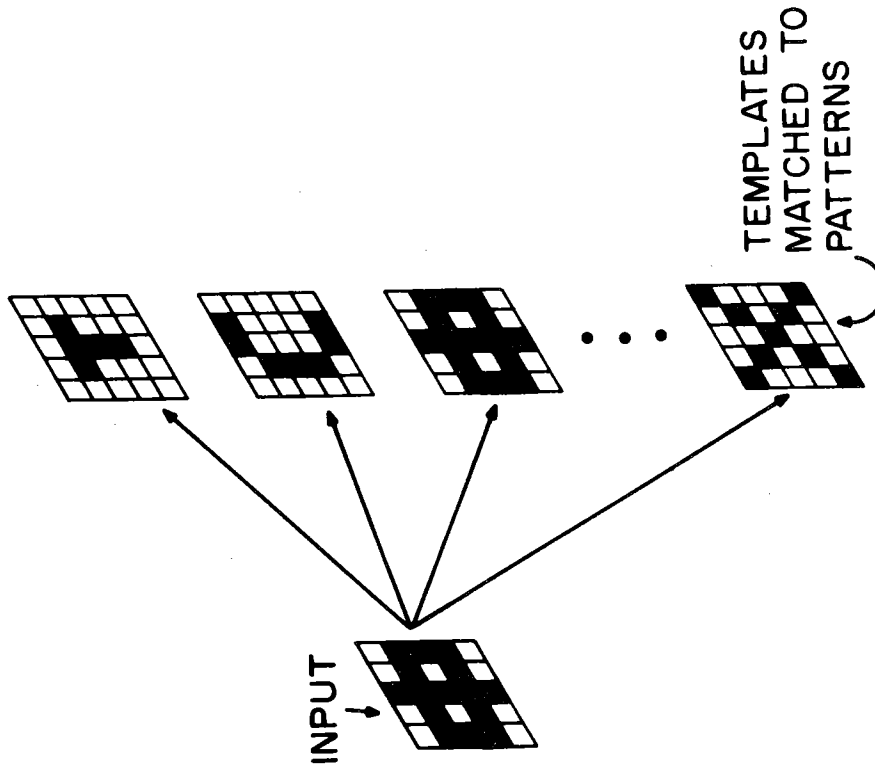


FIG. 4 PATTERN MATCHING

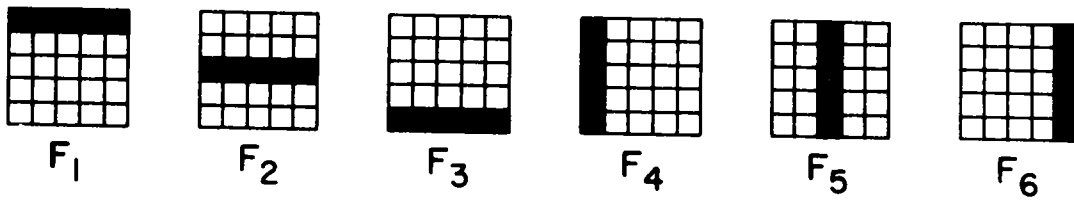


FIG. 6 A SET OF FEATURES

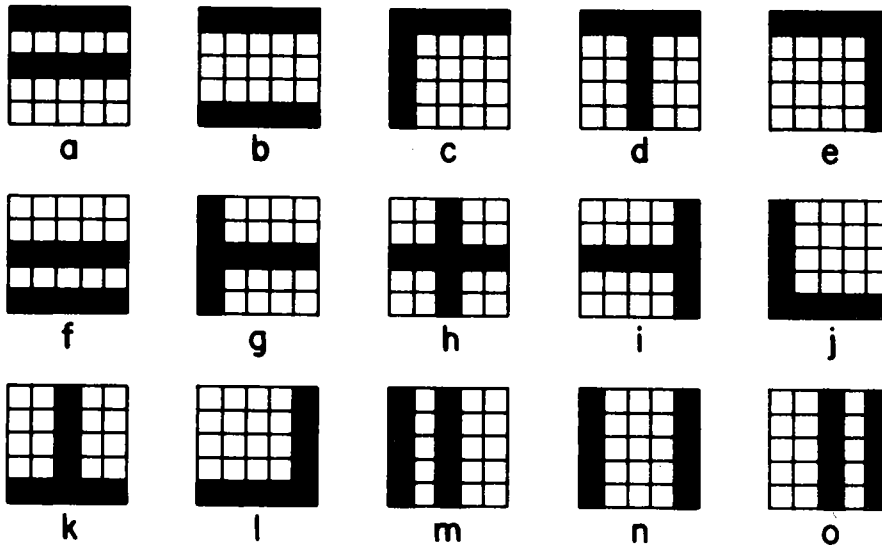


FIG. 7 PATTERNS CONSTRUCTED FROM FEATURES

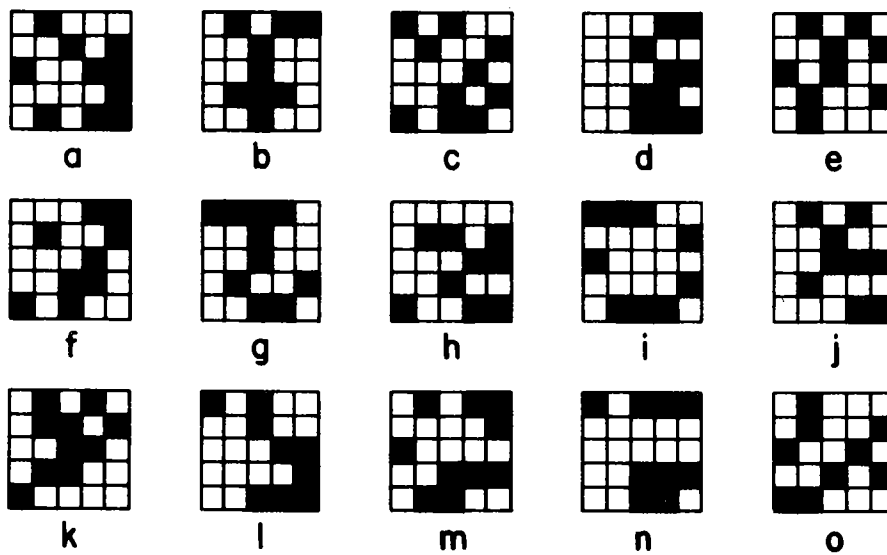


FIG. 8 PATTERNS PRODUCED BY RENUMBERING THE RETINAL POINTS

DATE		CHEMICALS PRESENT IN EXCESS NORMAL THRESHOLD													POSSIBLE COMBINATIONS OF FEATURES														
		N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25		
MON.	APR. 1		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	1 & 2
TUE.	APR. 2			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	2 & 3
WED.	APR. 3				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	3 & 4
THURS.	APR. 4				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	4 & 5
FRI.	APR. 5				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	5 & 6
SAT.	APR. 6				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	1 & 3
SUN.	APR. 7				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	2 & 4
MON.	APR. 8				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	3 & 5
TUE.	APR. 9				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	4 & 6
WED.	APR. 10				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	1 & 4
THURS.	APR. 11				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	2 & 5
FRI.	APR. 12				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	3 & 6
SAT.	APR. 13				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	?
SUN.	APR. 14				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	?
MON.	APR. 15				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	?
POSSIBLE FEATURES		POSSIBLE DISCHARGE FROM LOCAL PLANTS																											
DOMESTIC SEWAGE PLT.		✓													✓														
METAL PLATING PLANT		✓													✓														
MILK PRODUCTS FACTORY		✓													✓														
PULP & PAPER MILL		✓													✓														
SLAUGHTER HOUSE		✓													✓														
TEXTILE MILL		✓													✓														
OTHERS?																													

FIG. 9 AN EXAMPLE PROBLEM IN FEATURE DETERMINATION

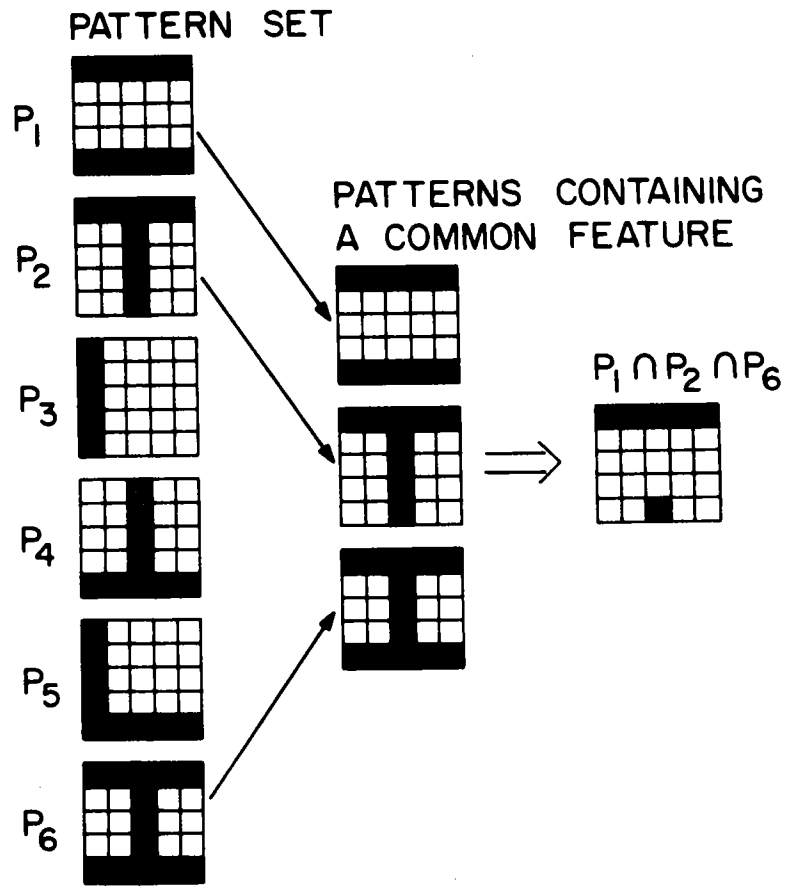


FIG. 12 INTERSECTION OF ALL PATTERNS CONTAINING A COMMON FEATURE

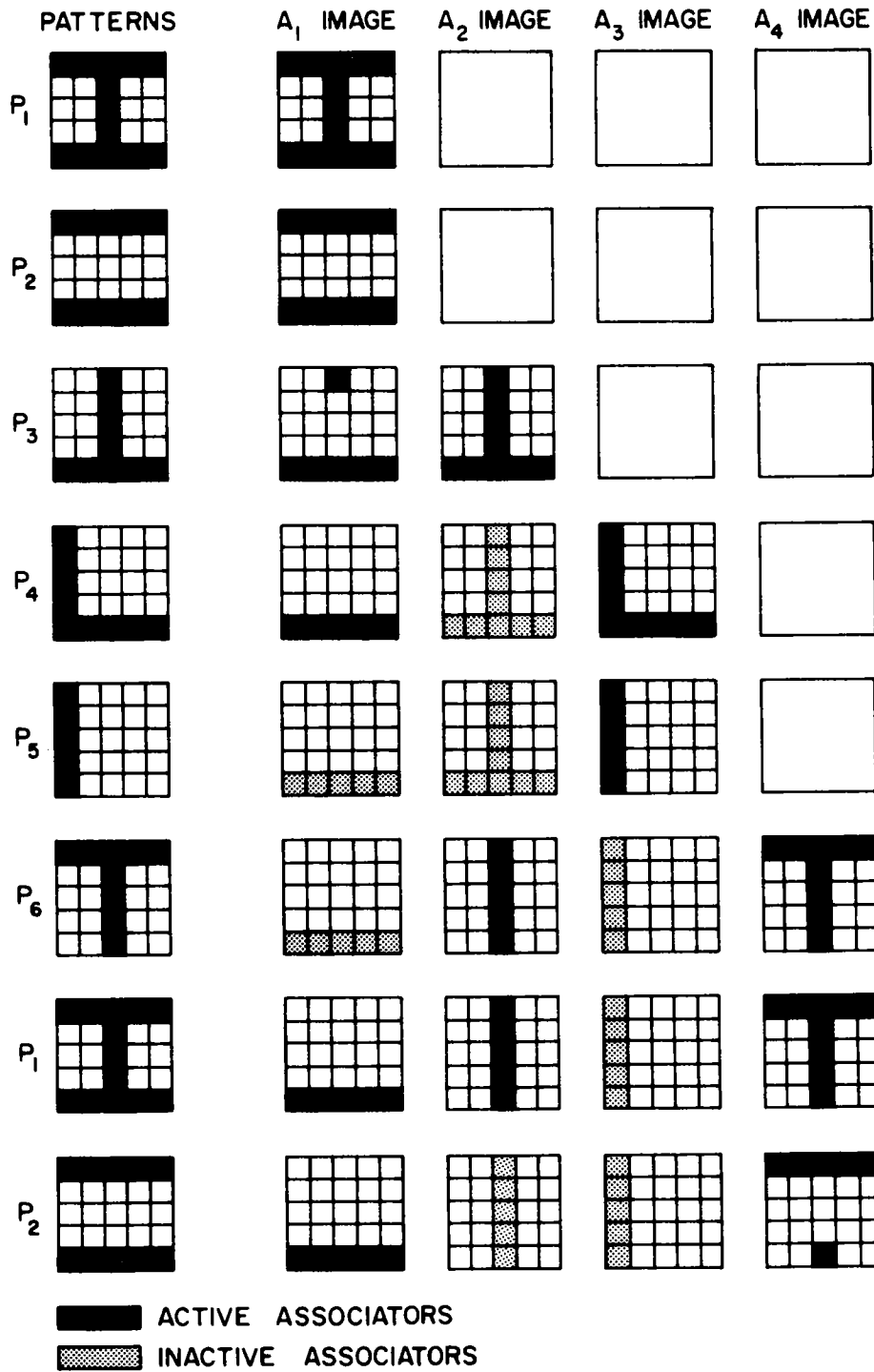


FIG. 13 AN EXAMPLE SHOWING THE OPERATION OF THE PARALLEL ALGORITHM

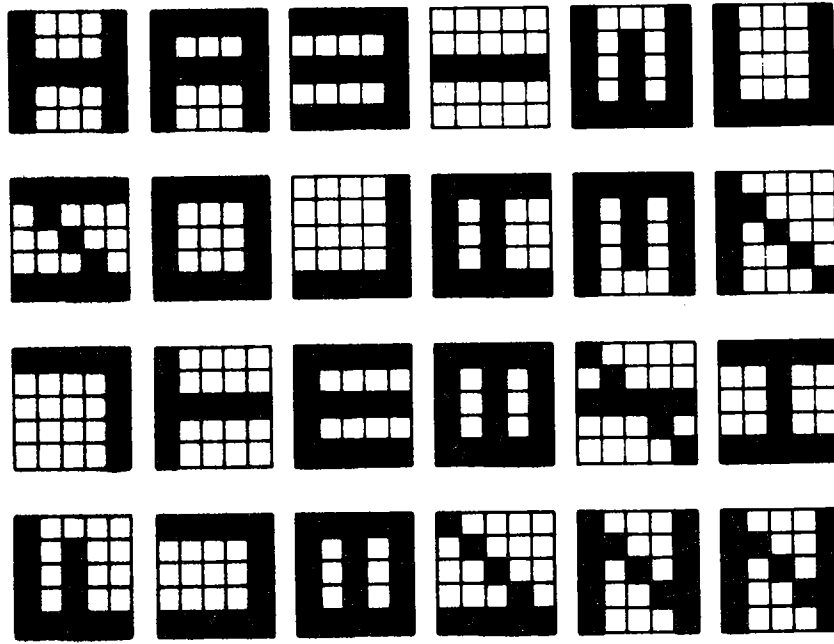


FIG. 14 PATTERNS CONSTRUCTED FROM 7 FEATURES

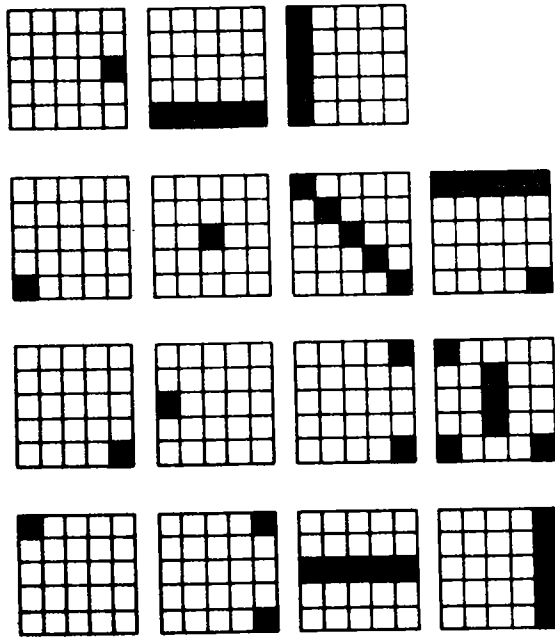


FIG. 15 ASSOCIATORS WITH THRESHOLDS OF ONE

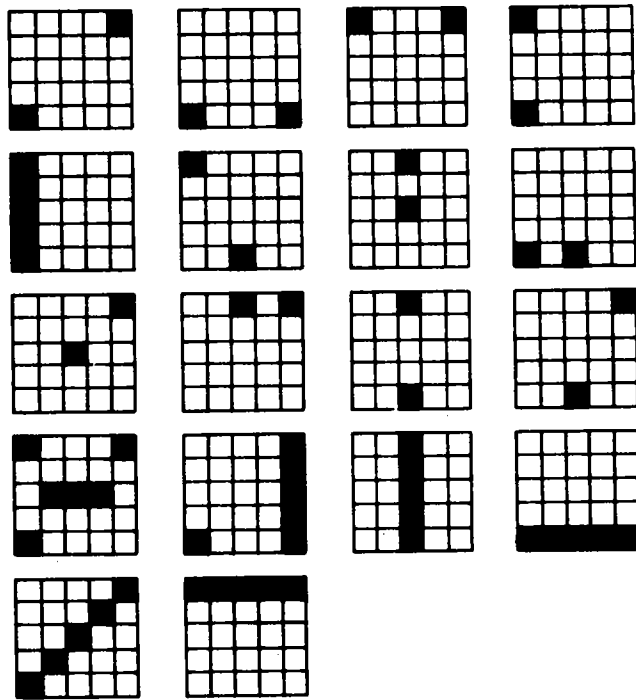


FIG. 16 ASSOCIATORS WITH THRESHOLDS OF TWO

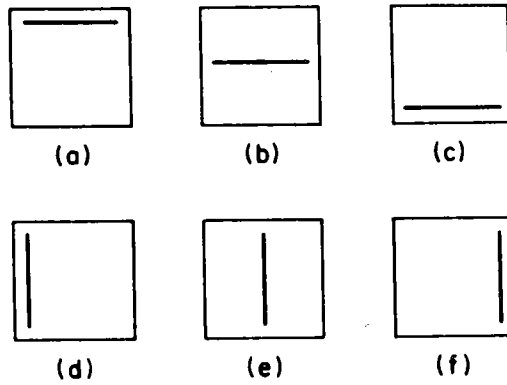


FIG. A-1 IDEAL FEATURES

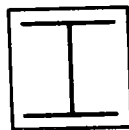


FIG. A-2 AN IDEAL PATTERN



FIG. A-3 EQUIVALENT REPRESENTATIONS OF AN IDEAL FEATURE



FIG. A-4 A SKETCH OF AN IDEAL PATTERN

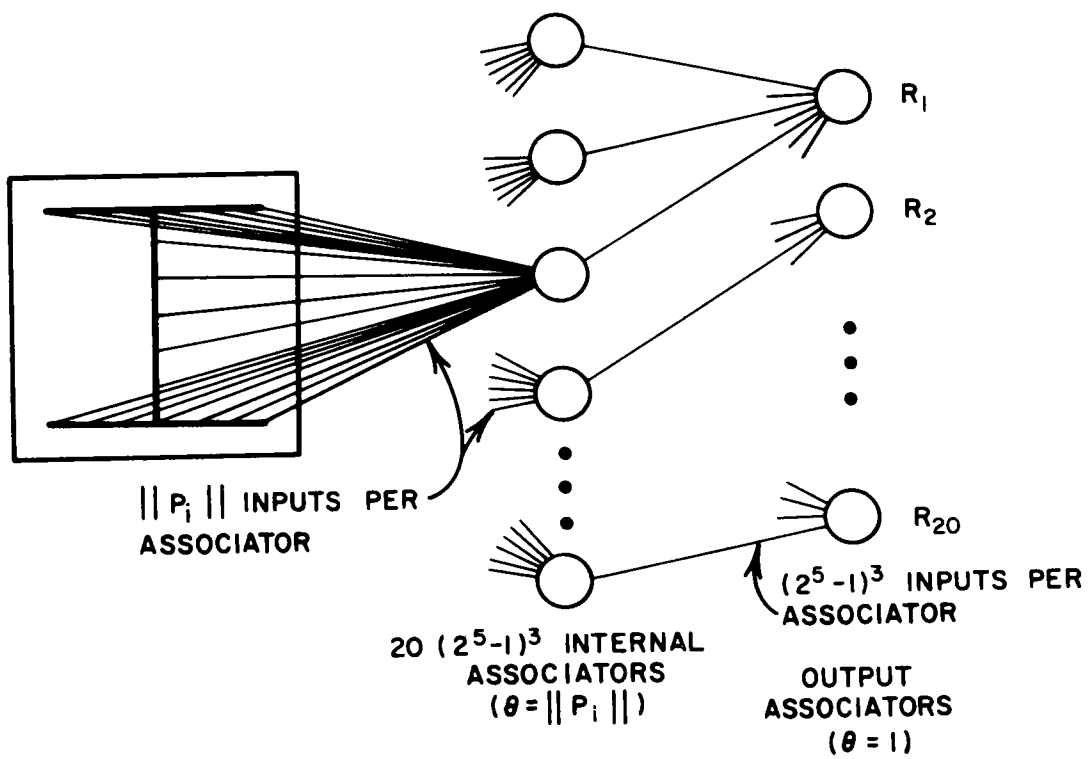


FIG. A-5 SOLUTION I

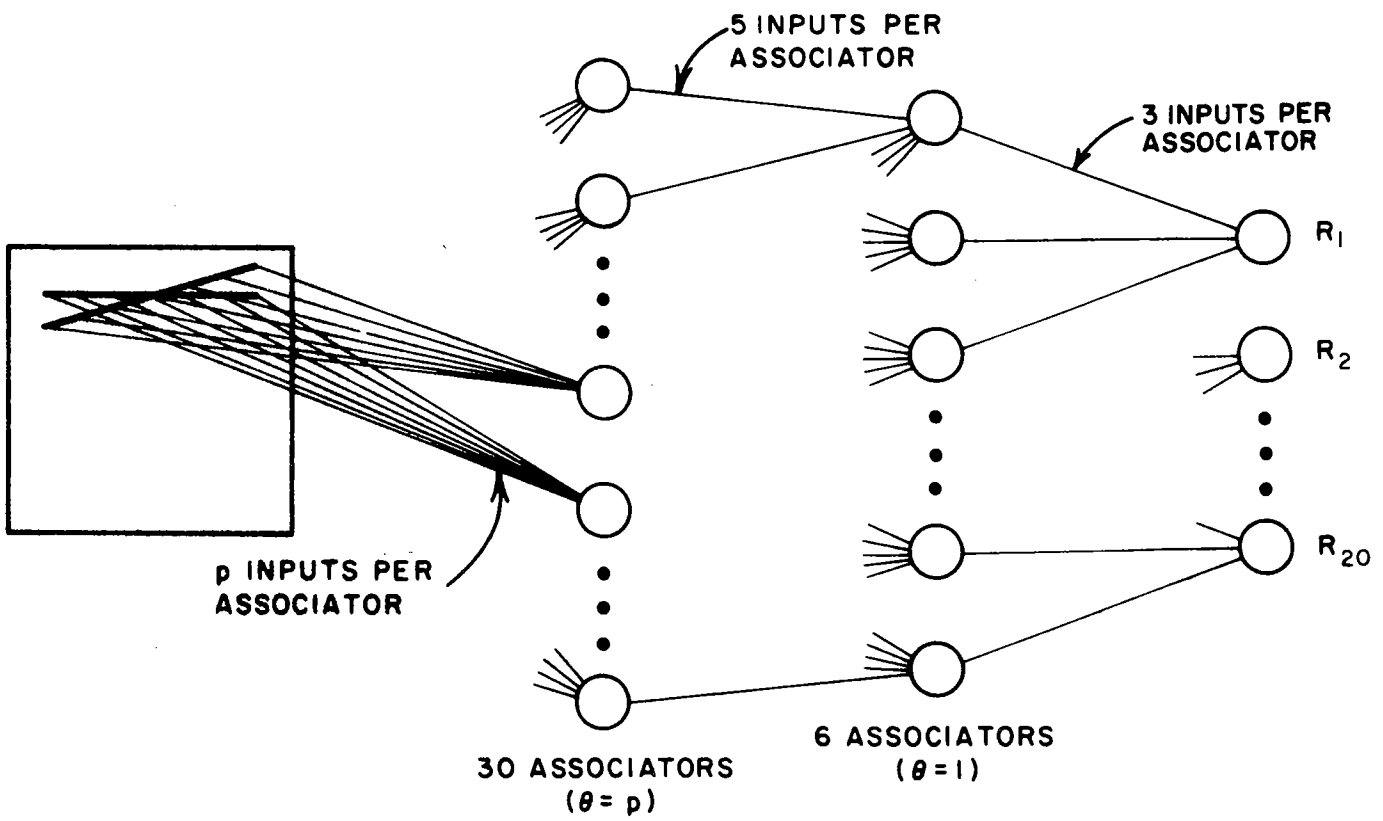
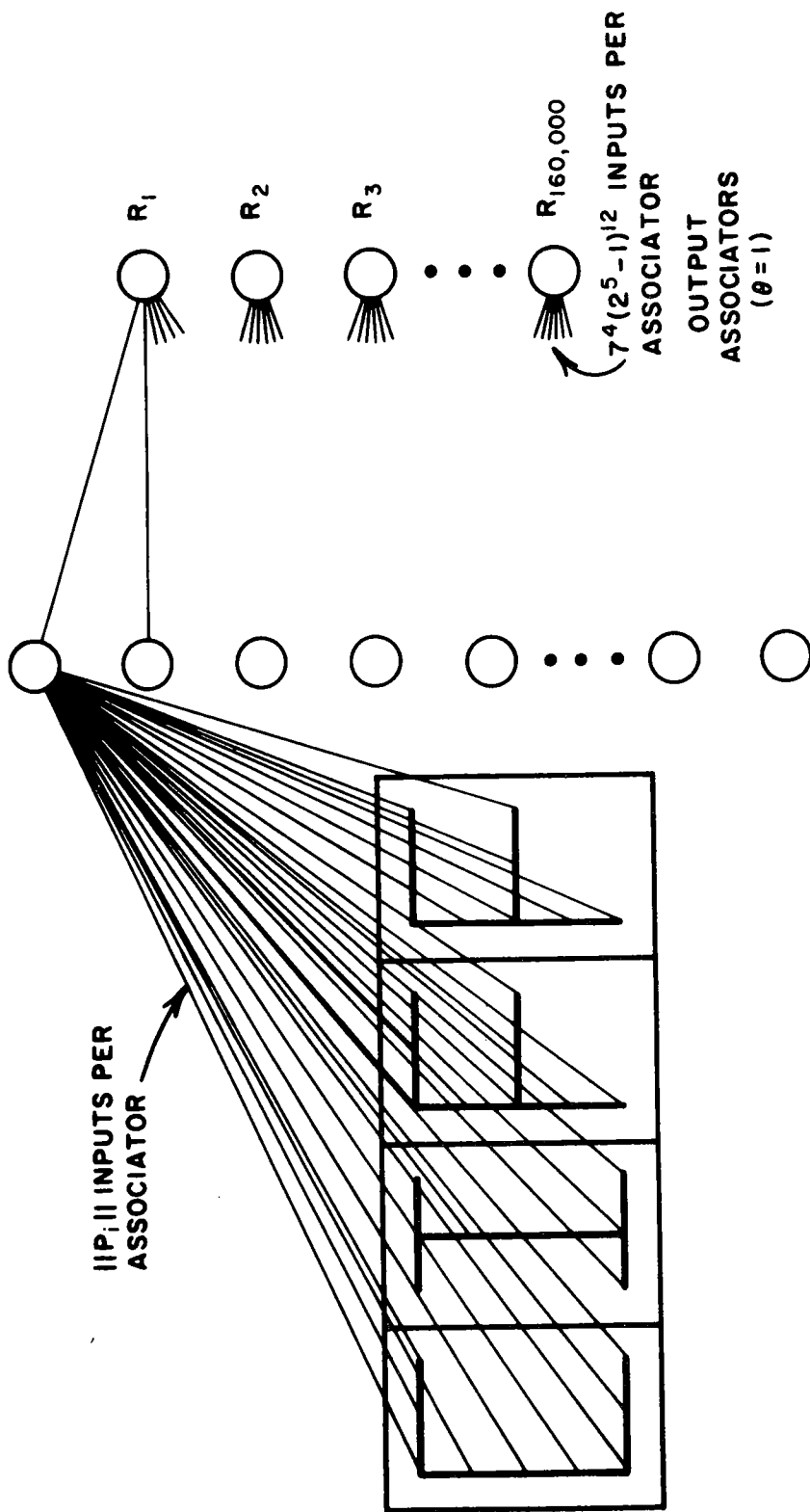


FIG. A-6 SOLUTION III



FIG. A-7 AN IDEAL WORLD



$$20^4 \cdot 7^4 \cdot (2^5 - 1)^2 \cong 3 \cdot 10^{26}$$

ASSOCIATORS
($\theta = || P_i ||$)

FIG. A-8 SOLUTION I

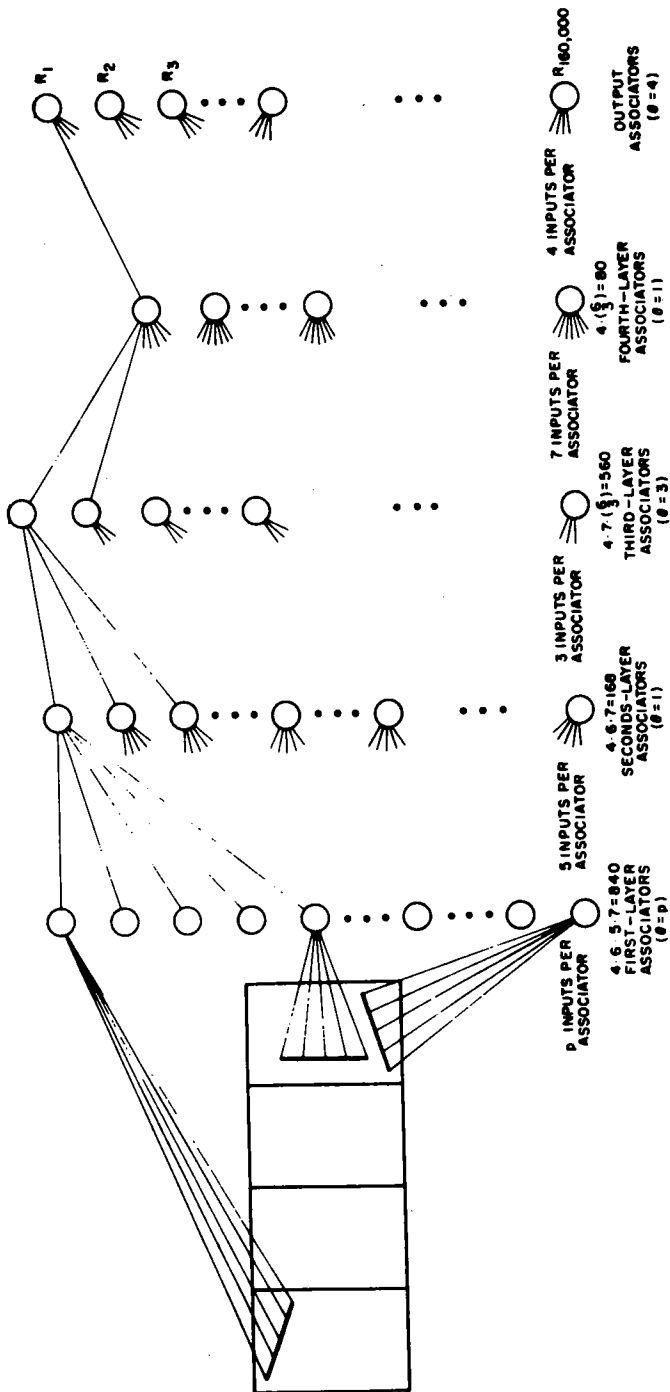


FIG. A-9 SOLUTION III