MASSACHUSETTS INSTITUTE OF TECHNOLOGY
A.I. LABORATORY

July 1976

Artificial Intelligence
Memo No. 376

COMPUTATIONAL GEOMETRY OF LINEAR THRESHOLD FUNCTIONS

Harold Abelson

ABSTRACT

Linear threshold machines are defined to be those whose computations are based on the outputs of a set of linear threshold decision elements. The number of such elements is called the rank of the machine. An analysis of the computational geometry of finite-rank linear threshold machines, analogous to the analysis of finte-order perceptrons given by Minsky and Papert, reveals that the use of such machines as "general purpose pattern recognition systems" is severely limited. For example, these machines cannot recognize any topological invariant, nor can they recognize non-trivial figures "in context".

The work reported in this paper was supported in part by the National Science Foundation under grant number EC40708X and conducted at the Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA. The view and conclusion contained in this paper are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the National Science Foundation or the United States Government.

To appear in INFORMATION AND CONTROL

Computational Geometry of Linear Threshold Functions

Harold Abelson

1. Introduction

This paper is a contribution to "computational geometry" in the spirit of the book Perceptrons by M. Minsky and S. Papert [1]. That is, we seek insights into the amount of computation "inherently needed" to recognize various geometric figures. In doing so, we raise issues about the use of parallel computation, analogue devices, and other pattern recognition techniques. This section briefly reviews the setting given in [1] for such a study and provides an introduction to the remainder of the paper.

By a retina, R, we mean a collection of points, and by a figure on the retina some subset $X \subset R$. The size of the retina, $|R|$ is the number of points in R. In studying pattern recognition we usually imagine R to be a finite set whose points are regarded as the squares in some two-dimensional plane grid and "arbitrary geometric figures" as approximated by some collection of squares. (Figure 1-1.)

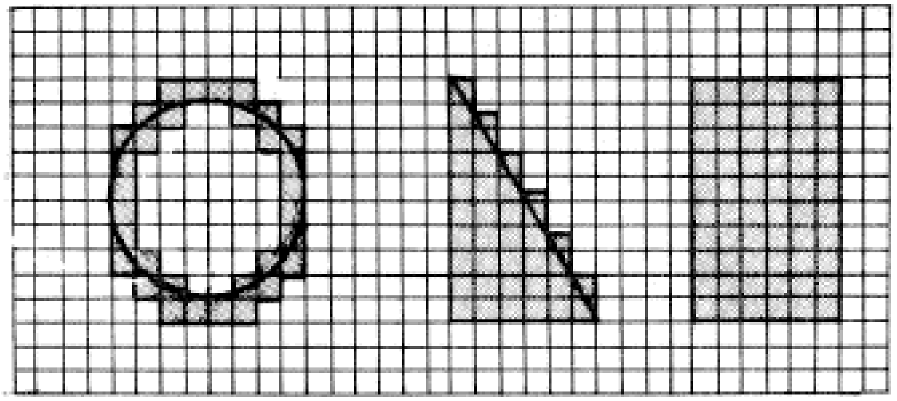

Figure 1

Geometric figures on a grid

A predicate on R is a function ψ defined for figures X on R which can assume only the values 0 and 1. Examples of geometric predicates are:

⌈X is a square⌉

⌈X is convex⌉

⌈X contains more than 47 points⌉ .

(Here, as in [1] we use the notation

$$\lceil \text{some condition} \rceil$$

to mean the value which is 1 if the condition is true and 0 if the condition is false.)

In computational geometry we are interested in synthesizing "complex" predicates out of "simpler" ones. One measure of the simplicity of a predicate is its *order*. A predicate φ is said to be of *order k* if φ makes its decision by examining at most k points of R, i.e., if there exists a set S of k points such that

$$\varphi(X) = \varphi(X \cap S) \qquad \text{for all} \quad X \subset R \ .$$

If $\Phi = \{\varphi_1, \varphi_2, \ldots\}$ is a collection of predicates, then a *perceptron* based on Φ is another predicate ψ which is of the form

$$\psi(X) = \left\lceil \sum_i a_i \varphi_i(X) > \theta \right\rceil \ .$$

where $a_1, a_2, \ldots, a_n, \theta$ are real numbers.

In other words, a perceptron is the result of a linear threshold decision applied to a weighted sum of other predicates. The a_i are the weights and θ is the threshold.

The *order* of the perceptron ψ is the maximum order of any of the predicates in the collection Φ . Notice that a perceptron of order 1 is precisely what is usually called a <u>linear threshold function</u> on R .

In [1] Minsky and Papert consider questions such as "What order perceptrons are necessary in order to compute various geometric predicates?" They show, for example, that

$\ulcorner X$ is locally convex$\urcorner$ can be computed with a perceptron of order 3

and $\ulcorner X$ is a (discrete approximation to a) circle$\urcorner$ can be computed with order 4.

More interesting are the results which illustrate fundamental limitations of perceptrons. One can ask if a predicate is of <u>finite order</u>, i.e., if it can be computed by a perceptron of some fixed order, regardless of the size of the retina. (See §1.6 of [1] for a formal definition.) Minsky and Papert show that such predicates as

$\ulcorner X$ is connected$\urcorner$

$\ulcorner X$ has at least 3 components$\urcorner$

are not of finite order. Indeed, a main theorem of [1] states that the only topologically invariant predicates which can be computed in finite order are those which are functions of the Euler characteristic (see [1] §5.9).

Another kind of simple machine, the Gamba Perceptron, is described in [1] as a kind of "perceptron" in which each of the "simple predicates" φ_i is itself a linear threshold function:

$$\varphi_i(X) = \left\lceil \sum_{x_j \in R} b_{ij} x_j(X) > \theta_i \right\rceil$$

$$\psi(X) \doteq \left\lceil \sum_i a_i \left\lceil \sum_{x_j \in R} b_{ij} x_j(X) > \theta_i \right\rceil > \theta \right\rceil .$$

Here $x_j(X)$ denotes the order 1 predicate

$$x_j(X) = \lceil x_j \in X \rceil .$$

Viewed as a perceptron, the Gamba machine ψ has order equal to the size of the retina $|R|$, since each φ_i looks at the entire retina. Hence the order restriction techniques of [1] do not give much information about the capabilities of this kind of device.

From another point of view, however, the Gamba machine is nowhere nearly as complex as the general order - $|R|$ perceptron. Rather, it is a simple "two-layer" device, in which each layer is made up of linear threshold elements. More generally, one could consider "multilayer machines", in which each layer makes linear threshold decisions based on results of previous layers.

This paper deals with properties of these "multilayer" machines. The computational devices we will be concerned with are called Linear Threshold Machines. A linear threshold machine is a general purpose computer together with a number of linear threshold elements $\varphi_1, \ldots, \varphi_r$. The general purpose computer is allowed to perform any computation whatsoever, with one restriction - computations cannot be based upon "direct observation" of the retina itself, but rather upon the outputs of the threshold functions $\varphi_1, \ldots, \varphi_r$. (Figure 2) The rank of the linear threshold machine is defined to be the number of linear threshold functions $\varphi_1, \ldots, \varphi_r$.

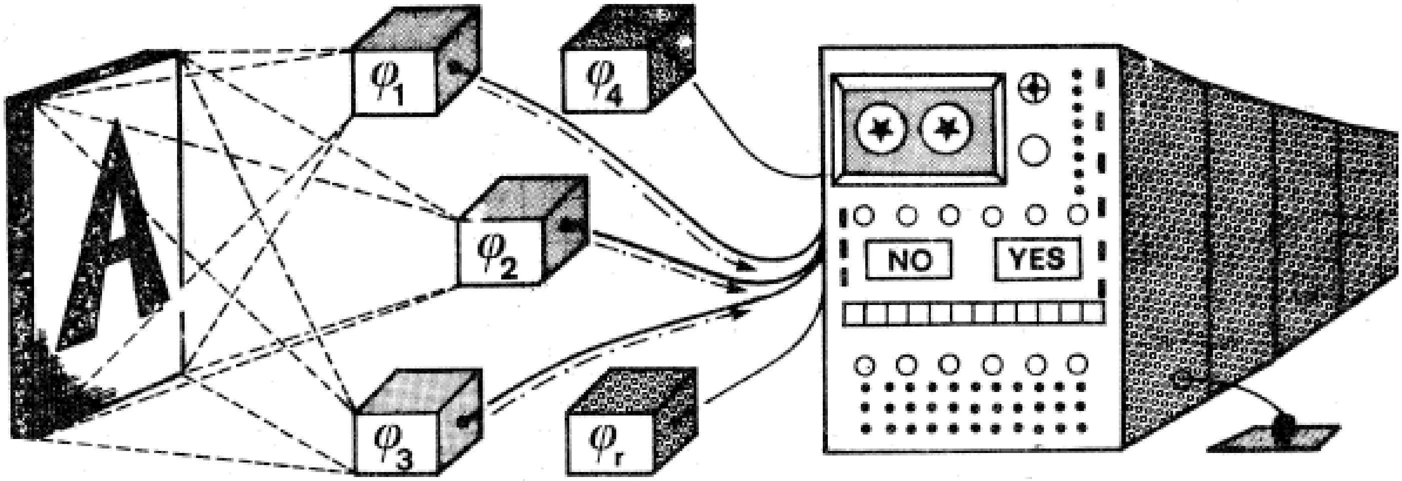

Figure 2

Linear threshold machine of rank r

This class of machines includes the Gamba perceptron, the multilayer machines, and in fact any kind of pattern recognition device that can be constructed out of linear threshold elements so long as the arrangement of interconnections does not include any loops. (Permitting loops would allow one to build a universal computer out of linear threshold elements.)

We begin, in Section 2, with a formal definition of linear threshold machines. Then in Section 3 we show that the parity predicate

$$\psi_{par}(X) = \lceil \text{the number of squares in } X \text{ is odd} \rceil$$

is not of finite rank. This allows us, in Section 4, to apply techniques of [1] to deduce that, as is the case with finite order perceptrons, the only topologically invariant predicates which could be of finite rank are functions of the Euler characteristic.

In Section 5, we begin to consider the problems of "infinite" or "arbitrarily large" retinas. We introduce the notion of uniform linear threshold machine, a linear threshold machine of fixed rank which can make computations which are "independent" of the size of the retina. Section 6 gives some examples of predicates which can be computed, somewhat surprisingly, by uniform linear threshold machines of rank 2.

Section 7 deals with the Saturation Theorem, our main technique for obtaining restrictions on the possible computations of uniform linear threshold machines. Section 8 applies this to show that, as opposed to the finite order perceptrons, which can compute the Euler characteristic, the uniform linear threshold machine cannot compute any non-trivial topological invariants. Section 9 gives further applications of the saturation technique and demonstrates the inability of these machines to recognize figures in context. Section 10 returns to give a more careful version of the Saturation Theorem and shows, for example, that if a linear threshold machine with bounded coefficients is to escape the saturation phenomenon, its rank must grow with the size of the retina, albeit very slowly (as $\log\log|R|$).

2. Linear Threshold Machines

Definition 2.1 A linear threshold function φ on a retina R is a particular kind of predicate computed as follows: For some real-valued function u on R and a real number θ we have

$$\varphi(X) = \left\lceil \sum_{x\in X} u(x) > \theta \right\rceil .$$

Here u is called the measure and θ the threshold associated to φ .

Now we combine these functions into machines. First of all, a Boolean r-tuple is defined to be an r-tuple each of whose elements is 0 or 1.

A rank r decision function Δ is a function defined on Boolean r-tuples and which can assume the values 0 or 1. Finally,

Definition 2.2 A linear threshold machine of rank r

$$M = \Delta\Phi$$

is a predicate consisting of

(i) An r-tuple of linear threshold functions $\Phi = (\varphi_1, \varphi_2, \ldots, \varphi_r)$ and

(ii) A rank r decision function Δ such that

$$M(X) = \Delta(\varphi_1(X), \varphi_2(X), \ldots, \varphi_r(X)) \quad .$$

This is the class of machines with which we will be concerned in this paper. The following observations are clearly true.

1. If the retina has $|R|$ points, then any predicate on R can be computed by a linear threshold machine of rank $|R|$.

2. If $M_1, \ldots, M_k$ are linear threshold machines of rank $r_1, \ldots, r_k$ respectively, then any Boolean function of

the M_i can be computed by a linear threshold machine of rank $r_1 + r_2 + \ldots + r_k$.

The definition (2.1) of linear threshold function is slightly out of line with that used by Minsky and Papert. When computing $\varphi(X)$ we only take the summation over the points of X rather than over the entire retina R . Two alternative definitions we might have used are

<u>Definition 2.3</u> "Order 1 Perceptron"

$$\varphi(X) = \left\lceil \sum_{x \in R} a_x p(x,X) > \theta \right\rceil$$

where $p(x,X)$ is a predicate depending only on whether or not $x \in X$.

Alternatively,

<u>Definition 2.4</u> "(-1,1) threshold function"

$$\varphi(X) = \left\lceil \sum_{x \in R} a_x p(x,X) > \theta \right\rceil$$

where $p(x,X) = 1$ if $x \in X$ and -1 if $x \notin X$.

It is easy to see that all three of the definitions are equivalent so long as we are dealing with a fixed finite retina

R . If, however, we consider *infinite retinas* or sequences of retinas, the different forms (2.1), (2.3), (2.4) make a difference. For example, the predicate

$$\lceil \text{area } X > \frac{1}{2} \text{ area } R \rceil$$

is easily expressed as a (-1,1) threshold function

$$\lceil \sum_{x \in R} p(x,X) > 0 \rceil$$

but a type (2.1) threshold function for this same predicate must involve constants which grow large as the size of the retina becomes large. We have chosen to work with the form (2.1) since we wish to make computations which depend only on the figure X itself, and not explicitly on the retina R .

Finally, there is one more assumption we will make about the threshold functions - that of *finite sensitivity*, that the values of the measure cannot be arbitrarily small in absolute value:

2.5 *Hypothesis of finite sensitivity:* With each threshold function there is associated a sensitivity ε such that, for any $x \in R$ either $\mu(x) = 0$ or $|\mu(x) \geq \varepsilon|$.

This hypothesis will certainly be satisfied for any linear threshold function built out of actual physical components, for example, out of optical filters and photo detectors.

3. Parity

We will be concerned, as in [1], with predicates that can be computed by linear threshold machines which are "independent of the retina". Our first attempt at formalizing this concept is the notion of "finite rank".

Definition 3.1 A predicate ψ is of finite rank r if for any size retina R there is a linear threshold machine of rank r which computes ψ on R.

In this section we exhibit a predicate which is not of finite rank. This is the "parity predicate"

$$\psi_{par}(X) = \lceil X \text{ contains an odd number of points of } R \rceil .$$

We shall show that, for a linear threshold machine to be able to recognize parity, its rank must grow at least logarithmically with the size of the retina. More precisely

3.2 Parity Theorem. Suppose M is a linear threshold machine of rank r which computes parity on a retina R. Then

$$|R| \leq r2^{r-1} \quad .$$

Before proceeding with the proof, we first introduce some notation. Let $M = \Delta\Phi$

$$\Phi = (\varphi_1, \varphi_2, \ldots, \varphi_r)$$

$$\varphi_i(X) = \left\lceil \sum_{x \in X} \mu_i(x) > \theta_i \right\rceil \quad .$$

For any $X \subset R$ let

$$S_i(X) = \sum_{x \in X} \mu_i(x)$$

and let $\sigma_i(X) = \left\lceil S_i(X) \geq 0 \right\rceil$.

Finally, let $\Phi(X)$ be the Boolean r-tuple

$$\Phi(X) = (\varphi_1(X), \ldots, \varphi_n(X))$$

and let $\Sigma(X)$ be the Boolean r-tuple

$$\Sigma(X) = (\sigma_1(X), \sigma_2(X), \ldots, \sigma_r(X)) \quad .$$

Now recall the usual Boolean notion of "implication", i.e., $0 \rightarrow 1$, $1 \rightarrow 1$, $0 \rightarrow 0$ are all valid, but $1 \rightarrow 0$ is not valid. This extends to a partial order on Boolean r-tuples:

3.3 Definition. If a and b are Boolean r-tuples, then we say that $a \leq b$ if $a_i \rightarrow b_i$ for $i = 1,\ldots,r$.

Our first step in proving Theorem 3.2 is to show that any linear threshold machine can be put in "normal form":

3.4 Definition. A linear threshold machine $M = \Delta\Phi$ is said to be normal if each component linear threshold function evaluates to zero on the empty set, i.e.

$$\Phi(\emptyset) = (0,0,\ldots,0) .$$

This is equivalent to saying that each of the thresholds θ_i is positive.

3.5 Normalization Lemma. If M is a linear threshold machine of rank r then there is a normal linear threshold machine, also of rank r , which computes the same predicate as M .

Proof: Suppose, by reordering, that $\varphi_i(\emptyset) = 1$ for $i = 1,\ldots,k$ and $\varphi_i(\emptyset) = 0$ for $k = k+1,\ldots,r$. We will produce a new linear threshold machine by modifying the first k threshold functions. Namely, if

$$\varphi_i(X) = \left\lceil \sum_{x \in X} \mu_i(x) > \theta_i \right\rceil$$

is a linear threshold function, define a new threshold function $\bar{\varphi}_i$ by

$$\bar{\varphi}_i(X) = \left\lceil \sum_{x \in X} (-\mu_i(x)) \geq -\theta_i \right\rceil$$

so that $\bar{\varphi}_i(X) = 1$ if and only if $\varphi_i(X) = 0$. So now let M' be the linear threshold machine

$$M^1 = \Delta\ (1 - \bar{\phi}_1,\ 1 - \bar{\phi}_2,\ \ldots,\ 1 - \bar{\phi}_k,\ \phi_{k+1},\ \ldots,\ \phi_r)$$

The key observation in the proof of theorem 3.2 is the following "regularity condition" for linear threshold machines:

<u>Lemma 3.6</u>. Suppose $M = \Delta\Phi$ is a normal linear threshold machine on R . Suppose X and Y are disjoint subsets of R with $\Phi(X) \leq \Sigma(Y)$. Then

$$\Phi(X) \leq \Phi(X \cup Y) \leq \Sigma(Y) \quad .$$

Proof: We'll verify that $\varphi_i(X) \to \varphi_i(X \cup Y) \to \sigma_i(Y)$ for each i .

Suppose $\sigma_i(Y) = 0$, so that $S_i(Y) \leq 0$. Then, by hypothesis, $\varphi_i(X)$ must also be 0 , so that $S_i(X) \leq \theta_i$. Therefore, since X and Y are disjoint, we have $S_i(X \cup Y) = S_i(X) + S_i(Y) \leq \theta_i$, that is, $\varphi_i(X \cup Y) = 0$. Hence $\varphi_i(X \cup Y) = 0$ whenever $\sigma_i(Y) = 0$, i.e. $\Phi(X \cup Y) \leq \Sigma(Y)$.

Now suppose that $\varphi_i(X \cup Y) = 0$, i.e.,

$$S_i(X) + S_i(Y) \leq \theta_i \quad . \tag{3.5}$$

Then

case 1 If $S_i(Y) \leq 0$ we have $\sigma_i(Y) = 0$, by normality. So the hypothesis implies that $\varphi_i(X)$ must be 0.

case 2 If $S_i(Y) > 0$ then equation 3.5 implies that $S_i(X) \leq \theta_i$, i.e., $\varphi_i(X) = 0$.

So, in either case, $\varphi_i(X) = 0$ whenever $\varphi_i(X \cup Y) = 0$, that is, $\Phi(X) \leq \Phi(X \cup Y)$.

The next lemma applies these "regularity" considerations to the parity predicate. First, if v is any Boolean r-tuple define ones(v) to be the number of ones in v.

<u>Lemma 3.7</u>. Suppose M is a normal linear threshold machine which computes parity on a retina R, and suppose that $x_1, x_2, \ldots, x_m$ are distinct points of R with

$$\Sigma(x_1) \leq \Sigma(x_2) \leq \ldots \leq \Sigma(x_m) .$$

Then ones$(\Sigma(x_m)) \geq m$.

<u>Proof</u>: Define subsets v_i of R to be

$$v_i = x_1 \cup x_2 \cup \ldots \cup x_i$$

$$v_o = \emptyset .$$

Since M is normal we have $\Sigma(v_o) \leq \Sigma(x_1) \leq \Sigma(x_m)$ so

Lemma 3.6 applies to give

$$\Phi(v_o) \leq \Phi(x_1 \cup v_o) \leq \Sigma(x_1)$$

that is, $\Phi(v_o) \leq \Phi(V_1) \leq \Sigma(x_1)$.

This inequality, along with the hypothesis, now implies that $\Phi(S_1) \leq \Sigma(x_2)$ so once again we can apply Lemma 3.6 to obtain

$$\Phi(V_1) \leq \Phi(x_2 \cup V_1) \leq \Sigma(x_2)$$

or, $\Phi(V_1) \leq \Phi(V_2) \leq \Sigma(x_2)$.

Continuing in this manner, we get

$$\Phi(V_1) \leq \Phi(V_2) \leq \ldots \leq \Phi(V_m) \leq \Sigma(x_m) \quad .$$

But V_i and V_{i-1} have opposite parity so $\Phi(V_i) \neq \Phi(V_{i-1})$. Therefore the vector $\Phi(V_i)$ contains at least one more "one" than the vector $\Phi(V_{i-1})$ and so $\Sigma(X_m) \geq \Phi(V_m)$ must contain at least m ones.

<u>Corollary 3.8</u>. Suppose M is a linear threshold machine which computes parity on a retina R . Suppose that all the measure functions μ_i in the threshold functions for M take on only positive values. Then

$$\text{rank } M \geq |R| \quad .$$

Proof: The hypothesis implies that $\Sigma(X) = (1,1,\ldots,1)$ for every subset of R . Therefore we have

$$\Sigma(x_1) = \Sigma(x_2) = \ldots = \Sigma(x_{|R|})$$

and so Lemma 3.7 implies that

$$\text{ones}(\Sigma(x_{|R|})) \geq |R| \quad .$$

But $\text{ones}(\Sigma(x_{|R|})) \leq$ number of elements in $\Sigma(x_{|R|}) = \text{rank } M$.

The same kind of reasoning as in Corollary 3.8 provides the proof of the Parity Theorem:

Proof of Theorem 3.2:

Suppose M computes parity and has rank r . Let

$$B(r) = \sum_{v} \text{ones}(v)$$

where the sum is taken over all distinct Boolean r-tuples v . Then we claim that $|R|$, the size of the retina, must be less than or equal to $B(r)$. For, consider the r-tuples $\Sigma(x_i)$ as x_i runs through the elements of R . If $|R| > B(r)$ then there must be some r-tuple v and points $x_1,\ldots,x_k$ with

$$\Sigma(x_1) = \Sigma(x_2) = \dots = \Sigma(x_k) = v$$

and $k > \text{ones}(v)$. But this is impossible by Lemma 3.7. Thus $|R| \leq B(r)$.

Finally, it remains only to compute $B(r)$:

$$B(r) = \sum_{v} \text{ones}(v) = \sum_{k=0}^{r} k \begin{Bmatrix} \text{the number of } r\text{-tuples} \\ v \text{ with } \text{ones}(v) = k \end{Bmatrix}$$

$$= \sum_{k=0}^{r} k\binom{r}{k} = r2^{k-1} .$$

4. Topological Consequences of the Parity Theorem.

This section follows Minsky and Papert ([1] Chapter 5) very closely in deriving consequences of the fact that the parity predicate is not of finite rank. We deduce that, as parity is not of finite rank, then neither are such predicates as

⌈X is connected⌉

⌈X has two components, one surrounding the other⌉

and so on. We will show that the only topological predicates which could be of finite rank can depend only on the Euler characteristic of X. (In fact, in Section 8, we will show that even these "Euler predicates" can not be of finite rank if we impose certain "uniformity conditions" on our linear threshold machines.)

Following Minsky and Papert, we show that any scheme for computing topological invariants (besides Euler characteristic) on a class of figures $\{X\}$ must also be able to compute parity on a class of "derived figures" $\{\hat{X}\}$. Hence, any machine which is "confused" by parity must necessarily also be confused by topological invariants.

This notion of "predicates on derived figures" is made precise by Minsky and Papert in Section 5.4 of [1]:

Suppose F is a function which associates to any figure X in R a figure $\hat{X} = F(\hat{X})$ in $\hat{R}$. Let $\hat{\psi}$ be a predicate on $\hat{R}$. Then we can define a predicate ψ on R by

$$\psi(X) = \hat{\psi}(F(X)) = \hat{\psi}(\hat{X}) .$$

In this context, Minsky and Papert formulate

<u>Collapsing Theorem for Perceptrons</u> ([1], Theorem 5.4.1): Suppose the function F is such that, <u>each point</u> $\hat{X}$ <u>of</u> $\hat{R}$ <u>depends on at most one point of</u> R, i.e., the points of $\hat{R}$

fall into four categories:

$$\hat{x} \in \hat{X} \quad \text{for all} \quad X$$

or

$$\hat{x} \notin \hat{X} \quad \text{for all} \quad X$$

or there is a point $x \in R$ such that

$$\hat{x} \in \hat{X} \quad \text{iff} \quad x \in X$$

or

$$\hat{x} \in \hat{X} \quad \text{iff} \quad x \notin X \quad .$$

Then order $\psi \leq$ order $\hat{\psi}$. (That is, if $\hat{\psi}$ can be computed by a perceptron of order k , than so can ψ .)

Analogously, we have

<u>Theorem 4.1 (Collapsing Theorem for Linear Threshold Machines)</u>:

Suppose that, as above, each point $\hat{x}$ of $\hat{R}$ depends on at most one point of R . Then

$$\text{rank } \psi \leq \text{rank } \hat{\psi} \; .$$

That is, if $\hat{\psi}$ can be computed by a linear threshold machine of rank r then so can ψ .

Proof: Suppose $\hat{M} = \hat{\Delta}\hat{\Phi}$ is a linear threshold machine of rank r which computes $\hat{\psi}$ on $\hat{R}$. Let $\hat{\varphi}_i$, $i = 1,\ldots,r$ be the linear threshold functions which comprise $\hat{\Phi}$. Let φ_i be the predicate on R defined by $\varphi_i(X) = \hat{\varphi}_i(F(X))$. Recall (2.3) that as long as we are dealing with a fixed retina (such as R or $\hat{R}$) then "linear threshold functions" are the same as "order 1 perceptrons". Thus we can apply the Collapsing Theorem for Perceptrons to deduce that the φ_i can be computed as linear threshold functions on R . Now define the linear threshold machine M on R by

$$M = \Delta\Phi$$

where $$\Delta = \hat{\Delta}$$

and $$\Phi = (\varphi_1,\ldots,\varphi_r) .$$

Then, for any $X \subset R$

$$M(X) = \Delta\Phi(X) = \Delta\hat{\Phi}(F(X)) = \hat{M}(F(X)) = \hat{\psi}(F(X)) = \psi(X)$$

so that M is a linear threshold machine of rank r which computes ψ .

Corollary 4.2. The predicate

$$\psi_{\text{connected}}(X) = \lceil X \text{ is connected} \rceil$$

is not of finite rank.

Proof: Since we have

1. ψ_{parity} is not of finite rank
2. the Collapsing Theorem is true

the proof is identical to the one given in the context of finite order perceptrons in Sections 5.5-5.7 of [1]. Basically, the idea is to construct a function

$$F\text{: (figures in } R) \longrightarrow \text{(figures in } \hat{R})$$

such that

$$\psi_{\text{parity}}(X) = \psi_{\text{connected}}(F(X))$$

and the Collapsing Theorem implies that

$$\text{rank } \psi_{\text{parity}} \leq \text{rank } \psi_{\text{connected}} .$$

See [1] for details.

The techniques of [1] also allow us to deduce that the only topologically invariant predicates which could be of finite rank must be functions of the Euler characteristic.

Definition 4.3. A predicate ψ is said to be topologically invariant if $\psi(X) = \psi(Y)$ whenever X and Y are topologically equivalent (i.e., X and Y can be "continuously deformed" into one another).

Corollary 4.4. Let ψ be a topologically invariant predicate of finite rank. Suppose X and Y are figures with the same Euler characteristic. Then $\psi(X) = \psi(Y)$.

Proof: The proof exactly follows Theorem 5.9 of [1] which proves the corresponding result for finite order perceptrons. The idea is based on a construction due to Paterson which reduces the computation of ψ modulo Euler characteristic to the computation of the parity of certain derived figures. See [1] for details.

5. Infinite Retinas; Uniform Linear Threshold Machines.

In demonstrating that predicates such as parity and connectedness are not of finite rank, we considered a fixed, finite retina and found lower bounds for the rank of any linear threshold machine which computes these predicates. The lower bound becomes large as the size of the retina becomes large, hence the predicates are not of finite rank.

But the intuitive concept of "finite rank" carries a somewhat stronger connotation. Namely, we would like to think of a "finite rank" predicate as one which can be somewhat computed by a <u>fixed</u> linear threshold machine which works regardless of the size of the retina. We formalize this notion below in the definition of uniform linear threshold machine.

<u>Definition 5.1</u>. By an "infinite retina" $\hat{R}$ we will mean an increasing union of retinas

$$R^1 \subset R^2 \subset R^3 \quad \ldots$$

A <u>uniform linear threshold function</u> $\hat{\varphi}$ on $\hat{R}$ is a compatible collection of linear threshold functions

$$\varphi^i(X) = \left\lceil \sum_{x \in X} \mu^i(X) > \theta^i \right\rceil$$

where μ^i is a measure function on R^i . By "compatible collection" we mean

1) If $R^j \subset R^i$ then μ^i restricted to R^j is the same as μ^j ,
2) all the θ^i are the same .

Thus, it makes good sense, for finite figures X in

$\hat{R}$, to write

$$\hat{\varphi}(X) = \left\lceil \sum_{x \in X} \mu(x) > \theta \right\rceil$$

where μ is a well-defined function on the infinite retina $\hat{R}$.

Definition 5.2. A uniform linear threshold machine of rank r on $\hat{R}$ is a predicate $\hat{M} = \Delta\hat{\Phi}$ where $\hat{\Phi} = (\hat{\varphi}_1, \hat{\varphi}_2, \ldots, \hat{\varphi}_r)$ is an r-tuple of uniform linear threshold functions and Δ is a rank r decision function.

Intuitively, then, we allow our machines to operate on larger and larger retinas by hooking up more and more imputs to the linear threshold functions. The thresholds θ as well as the decision function Δ remain unchanged.

Notice that we make no requirement that the measure functions μ remain bounded as the retina gets large.

Also we could have defined our "uniform threshold functions" based on one of the other definitions of "linear threshold function", 2.3 or 2.4. These would lead to different classes of machines. However, Definition 5.1 seems more natural since, for a fixed figure on an "arbitrarily large" retina, the threshold summations need extend only over the points of the figure. This seems to capture the intuitive notion of "computations which depend only on the figure itself, not on the entire infinite retina".

6. Stratification; Predicates of Rank 2.

Much of this paper is concerned with proving that various geometric predicates are not of finite rank. In this section, by way of contrast, we show how certain "symmetry" predicates can be computed by uniform linear threshold machines of rank 2. These results are reminiscent of the "stratification phenomenon" discussed in Chapter 7 of [1]. This consists, roughly, in using very large coefficients to encode geometric information, thus allowing certain predicates to be computed by simpler machines than might have been thought necessary. The details of this technique for linear threshold machines differ from those given in [1] for perceptrons. However, the results have the same flavor in both cases, and so we retain the name "stratification".

<u>Theorem 6.1</u> (Rank 2 Stratification).

Let $S_1, S_2, \ldots$ be a sequence of disjoint finite subsets of $\hat{R}$. Let ψ_i be the predicate

$$\psi_i(X) = \lceil \text{either } S_i \subset X \text{ or } S_i \cap X = \emptyset \rceil$$

then $\psi = \bigwedge_i \psi_i$ can be computed by a uniform linear threshold machine of rank 2.

(Note: Each S_i must itself be a finite set. But there may be infinitely many distinct S_i's.)

<u>Proof</u>: For each S_i pick a "base point" $b_i \in S_i$. Let n_i = (number of elements in S_i) - 1 . Define the function μ as follows:

$$\text{for } x \in S_1 - b_1 \qquad \mu(x) = m_1 =_{df} 1$$

$$\mu(b_1) = -n_1 m_1$$

and, inductively,

$$\text{for } x \in S_i - b_i \qquad \mu(x) = 1 + \sum_{y \in \bigcup_{j=1}^{i-1} S_j} \operatorname{abs}(u(y)) =_{df} m_i$$

$$\mu(b_i) = -n_i m_i$$

and $\mu(x) = 0$ for x not contained in any S_i .

Then define

$$\varphi_1(X) = \ulcorner \sum_{x \in X} u(x) \geq 0 \urcorner$$

$$\varphi_2(X) = \ulcorner \sum_{x \in X} (-u(x)) \geq 0 \urcorner \ .$$

We claim that $\Psi(X)$ is true if and only if $\varphi_1(X)$ and $\varphi_2(X)$ are both true, i.e., if and only if

$$\sum_{x \in X} u(x) = 0 \ .$$

To see this, note first that

$$\sum_{x \in S_i} \mu(x) = 0$$

by choice of μ . Now if $\psi_i(X)$ is true we have that either $X \cap S_i = S_i$ or $X \cap S_i = \emptyset$. In either case, then

$$\sum_{x \in S_i \cap X} \mu(x) = 0 \quad .$$

So if $\Psi(X)$ is true, we have

$$\sum_{x \in X} \mu(x) = \sum_{i} \sum_{x \in S_i \cap X} \mu(x) = 0 \quad .$$

Conversely, suppose $\Psi(X)$ is false and let I be the largest value of i for which $\psi_i(X)$ is false. (Recall that we are only concerned with finite figures X , so that I exists.) Then

$$\sum_{x \in X} \mu(X) = \sum_{\substack{\text{all } i \\ \text{such that} \\ \psi_i(X) \text{ is} \\ \text{false}}} \sum_{x \in X \cap S_i} \mu(x) \quad .$$

Let $A_I = \sum_{x \in S_I \cap X} \mu(x)$ and $A_R = \sum_{\substack{x \in X \\ x \notin S_I \cap X}} u(x)$. Then

$\sum_{x \in X} \mu(x) = A_I + A_R$. By construction of μ , we have $\mathrm{abs}(A_I) \geq m_I$ since at least one point of S_I is not in X . Also, by construction, $m_I > \mathrm{abs}(A_R)$ since, if $\psi_i(X)$ is false then $i \leq I$.

$$\text{Thus} \quad |A_I| > |A_R| \quad \text{so} \quad A_I + A_R \neq 0 \ .$$

Finally, notice that this construction will provide uniform linear threshold functions $\hat{\varphi}_1$, $\hat{\varphi}_2$ on a retina sequence $\hat{R}$: $R^1 \subset R^2 \subset \ldots$ We need only make sure that the "higher numbered" sets S_i appear in the higher numbered retinas R^j . The crucial point is that we can "enlarge the retina", add more S_i , without changing the value of μ on the lower numbered S_i .

6.2 Examples.

The following predicates all have rank 2:

(a) Draw a vertical line L down the center of the retina. Define ψ by

$$\psi(X) = \lceil X \text{ is symmetric with respect to } L \rceil \ .$$

Here the sets S_i have two elements consisting of a point x along with its reflection in L . Following through the proof of 6.1 we see that the weight $m_i = 2^i - 1$.

(b) More generally, let G be a finite group acting on R. Then

$$\psi_G(X) = \lceil X \text{ is invariant under } G \rceil$$

has rank 2. Take the S_i to be the orbits of points of R under the G-action, i.e.,

$$S_i = \bigcup_{g \in G} g(x) \qquad \text{for some } x \in R \quad .$$

(c) Pick a point $x_o \in R$. Then

$$\psi(X) = \lceil X \text{ is a bull's-eye centered about } x_o \rceil$$

has rank 2. Take the S_i to be "concentric rings" about x_o.

7. Saturation.

We now turn to some predicates which cannot be computed by uniform linear threshold machines. These include, for example, predicates which recognize <u>any</u> topological invariant and predicates which recognize figures in context.

The main technique for obtaining these results is the Saturation Theorem. This says, roughly, that linear threshold functions will become "overloaded" as the retina becomes large. Consequently, parts of figures may become "invisible" to a linear threshold machine. We formalize this in the notion of "saturation" and "saturation sequence".

Definition 7.1 Suppose that $\hat{M}$ is a uniform linear threshold machine on a retina $\hat{R}: R^1 \subset R^2 \subset \ldots$ and that A and B are subsets of $\hat{R}$ with $A \subset R^a$, $B \subset R^b$, $a < b$. Then we say that B saturates $\hat{M}$ with respect to A on R^a , if for any $S \subset R^a$ we have

$$\hat{M}(B \cup S) = \hat{M}(A \cup B \cup S) \quad .$$

(See Figure 3)

Intuitively, the idea is that B "overwhelms" the decision elements of $\hat{M}$ to such an extent that $\hat{M}$ cannot "see" A .

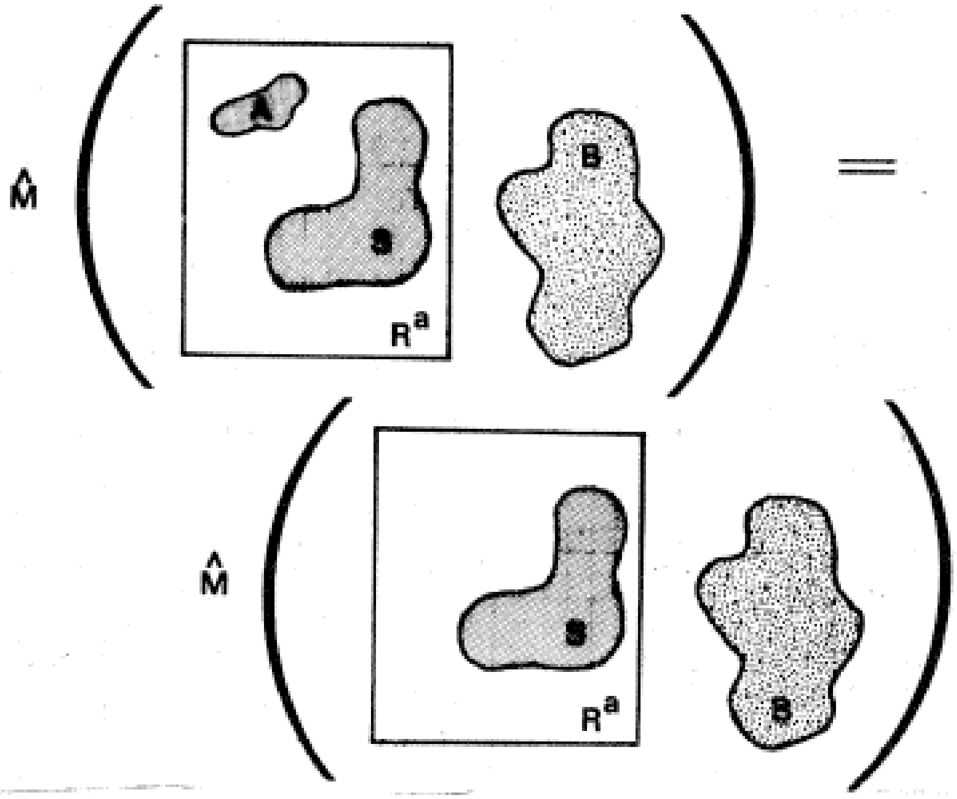

Figure 3

B saturates $\hat{M}$ with respect to A on R^a

<u>Definition 7.2</u> Suppose $\hat{M}$ is a uniform linear threshold machine on $\hat{R}$ and that $\{A_i \subset R^{\ell(i)}\}$ is a sequence of subsets of $\hat{R}$. (Here $\{R^{\ell(i)}\}$ represents an expanding collection of retinas in $\hat{R}$.) Then we say that $\{A_i\}$ is a <u>saturation sequence</u> if there exists an integer N such that

$$A_2 \cup A_3 \cup \ldots \cup A_N$$

saturates $\hat{M}$ with respect to A_1 on $R^{\ell(1)}$.

The main result about saturation is now:

7.3 Saturation Theorem Let $\hat{M}$ be a uniform linear threshold machine on $\hat{R}$ and let $\{A_i \subset R^{\ell(i)}\}$ be any infinite sequence of disjoint sets. Then $\{A_i\}$ contains a subsequence which is a saturation sequence.

Proof: Let $\hat{M} = \Delta\hat{\Phi}$ and $\hat{\Phi} = (\hat{\varphi}_1,\ldots,\hat{\varphi}_r)$

$$\hat{\varphi}_j(X) = \left\lceil \sum_{x\in X} \mu_j(x) > \theta \right\rceil .$$

As in §3 let

$$S_j(X) = \sum_{x\in X} \mu_j(x) .$$

Define the number $\gamma_i(X)$ by

$$\gamma_j(X) = \begin{cases} 0 & \text{if} \quad S_j(x) = 0 \\ 1 & \text{if} \quad S_j(x) > 0 \\ -1 & \text{if} \quad S_j(x) < 0 \end{cases}$$

and let $\Gamma(X)$ be the r-tuple

$$\Gamma(X) = (\gamma_1(X),\gamma_2(X),\ldots,\gamma_r(X)) .$$

Since there are only a finite number of possible values for $\Gamma(X)$, there must be an infinite subsequence of the $\{A_i\}$ for which $\Gamma(A_i)$ takes the same fixed value. We claim that this is the desired saturation sequence.

To prove this, first renumber the A_i's so that $A_1, A_2, \ldots$ is the subsequence picked out above. Also for convenience renumber the the R^i's so that $A_i \subset R^i$. Now let

$$M = \max_{j=1,\ldots,r} \max_{X \subset R^1} \operatorname{abs}(S_j(X)) .$$

Let

$$T = \max_{j=1,\ldots,r} \operatorname{abs} \theta_j$$

and choose $N > \frac{T + M}{\epsilon} + 1$

where ϵ is the minimum sensitivity of the linear threshold functions $\hat{\varphi}_1, \ldots, \hat{\varphi}_r$. (Recall §2.5.)

Let $A = A_1$ and let $B = A_2 \cup A_3 \cup \ldots \cup A_N$ and let S be any subset of R^1 . We will show that

$$\hat{\varphi}_j(A \cup B \cup S) = \hat{\varphi}_j(B \cup S)$$

for $j = 1,2,\ldots r$. This will prove the theorem.

Case 1. Suppose we have a j for which $S_j(A) = 0$. Hence

$S_j(B)$ and $S_j(A \cup B)$ are also 0, so

$$S_j(A \cup B \cup S) = S_j(S) = S_j(B \cup S)$$

and therefore $\hat{\varphi}_j(A \cup B \cup S) = \hat{\varphi}_j(B \cup S)$.

Case 2. Suppose $S_j(A) > 0$. Then, by (2.5) we have $S_j(A) \leq \epsilon$ and so

$$S_j(B) \geq (N-1)\epsilon = T + M$$

also, by choice of M we have

$$|S_j(S)| \leq M \qquad \text{so} \qquad S_j(B \cup S) \geq T \geq \theta_j$$

and $S_j(A \cup B \cup S) \geq T + \epsilon > \theta_j$.

Hence $\hat{\varphi}_j(A \cup B \cup S) = \hat{\varphi}_j(B \cup S) = 1$.

Case 3. Suppose $S_j(A) < 0$. Then, as above, we have

$$S_j(A) \leq -\epsilon \qquad \text{and} \qquad S_j(B) \leq -(N-1)\epsilon = -T-M .$$

Also $|S_j(S)| \leq M$, so $S_j(B \cup S) \leq -T \leq \theta_j$

and $$S_j(A \cup B \cup S) \leq -T - \epsilon < \theta_j .$$

Hence $$\hat{\varphi}_j(A \cup B \cup S) = \hat{\varphi}_j(B \cup S) = 0$$

This completes the proof.

To use the Saturation Theorem we proceed as follows: first find a figure X which we would like to make "invisible" to $\hat{M}$, then embed X in a saturation sequence $\{X_i\}$ so that

$$X_2 \cup X_3 \cup \ldots \cup X_N =_{df} \text{sat}(X)$$

saturates $\hat{M}$ with respect to X. The following proposition illustrates the technique:

<u>Proposition 7.4</u> Let ψ_{adj} be the predicate

$$\psi_{adj}(X) = \lceil X \text{ contains at least 2 adjacent points of } R \rceil$$

Then ψ_{adj} cannot be computed by a uniform linear threshold machine.

<u>Proof</u>: Suppose $\hat{M}$ is a uniform linear threshold machine. Let $\{A_i\}$ be a sequence of single points of R, spaced at least 3 apart. By Theorem 7.3 $\{A_i\}$ contains a saturation subsequence $B_1, B_2, \ldots$ Then, as indicated above, let $B = B_1 \subset R^1$ and let $B_2 \cup B_3 \cup \ldots \cup B_N = \text{sat}(B)$ saturate

$\hat{M}$ with respect to B on R^1. Now let S be the figure consisting of a single point of R^1, adjacent to B but not adjacent to any of the other B_i. Then, by saturation,

$$\hat{M}(\mathrm{sat}(B) \cup S) = \hat{M}(B \cup \mathrm{sat}(B) \cup S)$$

but $\psi_{adj}(\mathrm{sat}(B) \cup S)$ is false while $\psi_{adj}(B \cup \mathrm{sat}(B) \cup S)$ is true. (See Figure 4)

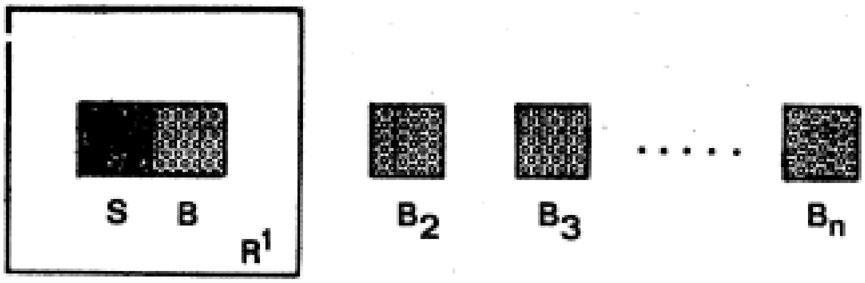

Figure 4

Saturation sequence for ψ_{adj}

Remark: This proposition stands in sharp contrast to the perceptron case, where ψ_{adj} is easily computed by a perceptron of order 2.

9. Topological Invariants.

We have already seen (4.4) that the only topological invariants which could be computed by a finite rank linear threshold machine are those which depend only on the Euler characteristic. Now we apply the Saturation Theorem to conclude that not even these "Euler predicates" are computable in a uniform way, and that consequently uniform linear threshold machines cannot compute any non-trivial topological invariant.

Theorem 8.1 Suppose $\hat{M}$ is a uniform linear threshold machine such that $\hat{M}(X_1) = \hat{M}(X_2)$ whenever X_1 and X_2 are topologicall" equivalent. Then, in fact, $\hat{M}(X) = \hat{M}(Y)$ for any non-empty sets X and Y .

Proof: Let T denote the annulus illustrated in Figure 5. Let X be any non-empty figure. We will show that $\hat{M}(X) = \hat{M}(T)$.

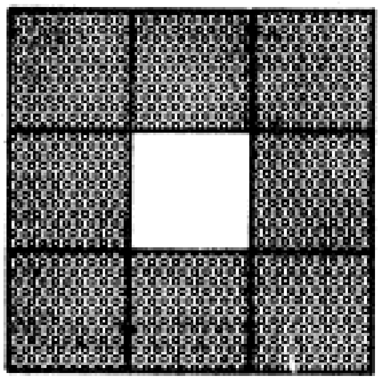

Figure 5

The standard annulus, T

Step 1: Let $e(X)$ be the Euler characteristic of X. By applying 4.4 and choosing the retina large enough we have that $\hat{M}(X)$ is equal to one of the "canonical figures with Euler characteristic $e(X)$", i.e.

$e(X)$ disjoint squares if $e(X) > 0$

or

a $1 - e(X)\cdot$ holed annulus if $e(X) \leq 0$

(see Figure 6). Thus we need only show that $\hat{M}(X) = \hat{M}(T)$ for X equal to any of these canonical forms.

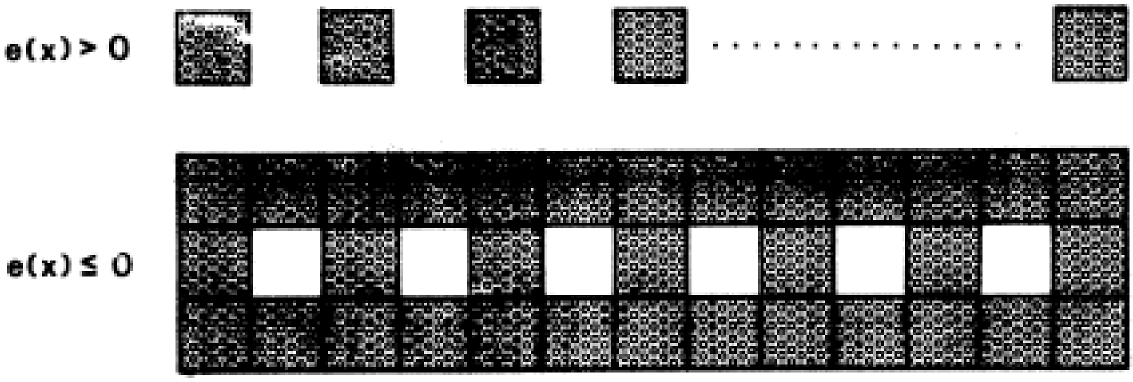

Figure 6

Canonical figures for Euler characteristic

Step 2. In the retina sequence $\hat{R}$ choose a sequence of disjoint copies of T. Now use the Saturation Theorem 7.3 to find a sequence $T_1, T_2, \ldots, T_N$ so that

$$T_2 \cup \ldots \cup T_N = \mathrm{sat}(T_1)$$

saturates $\hat{M}$ with respect to T_1 on R^1. (See Figure 7.) Notice, that by (4.4) we have $\hat{M}(X) = \hat{M}(X \cup \mathrm{sat}(T_1))$ since these sets have the same Euler characteristic.

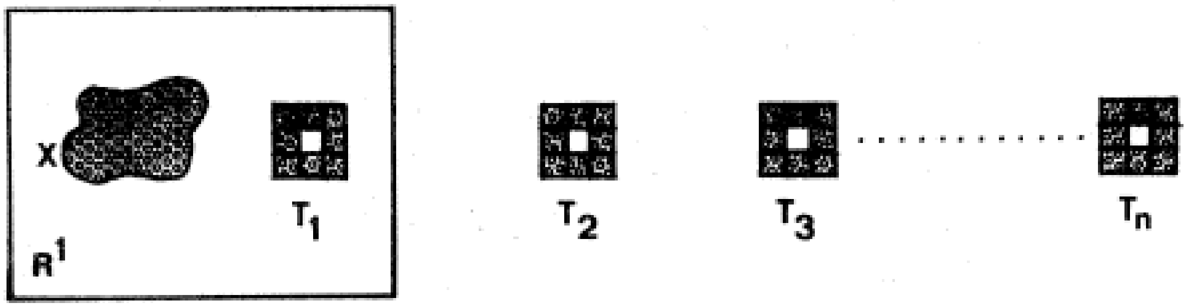

Figure 7

A saturation sequence of annuli

Step 3. Case 1: Suppose $e(X) \leq 0$ so that the canonical form for X is an n-holed annulus. Consider the set $X \cup \mathrm{sat}(T_1)$. By topological invariance we can deform $X \cup \mathrm{sat}(T_1)$ without changing the value of $\hat{M}$ so that the "end-position hole" of X moves over to become T_1 (Figure 8), i.e., $X = \overline{X} \cup T_1$ where $\overline{X}$ has $(n-1)$ holes.

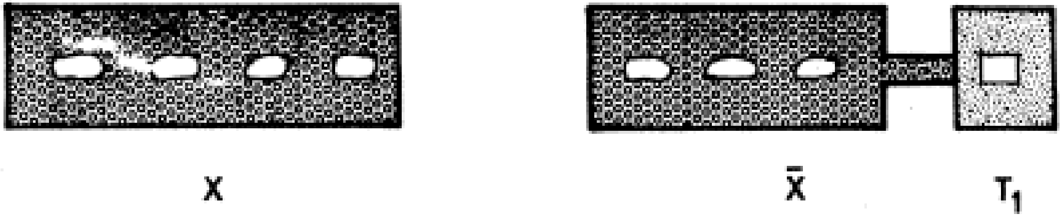

Figure 8

X deforms to $\bar{X} \cup T_1$

Thus we have

$$\hat{M}(X) = \hat{M}(X \cup \mathrm{sat}(T_1)) = \hat{M}(\overline{X} \cup T_1 \cup \mathrm{sat}(T_1))$$

$$= \hat{M}(\overline{X} \cup \mathrm{sat}(T_1)) = \hat{M}(\overline{X}) .$$

Proceeding inductively, we can reduce the number of holes of X one by one until there is only one hole left, i.e., X reduces to an annulus.

Case 2: Suppose that $e(X) > 0$ so that the canonical form for X is n disjoint squares. Let s_n denote the "end most" square and $X = \overline{X} \cup s_n$. Consider again the set $X \cup \mathrm{sat}(T_1)$. Once again the value of $\hat{M}$ is unchanged if we deform this set by moving s_n over to be adjacent to the position occupied on the retina by T_1 . (See Figure 9.)

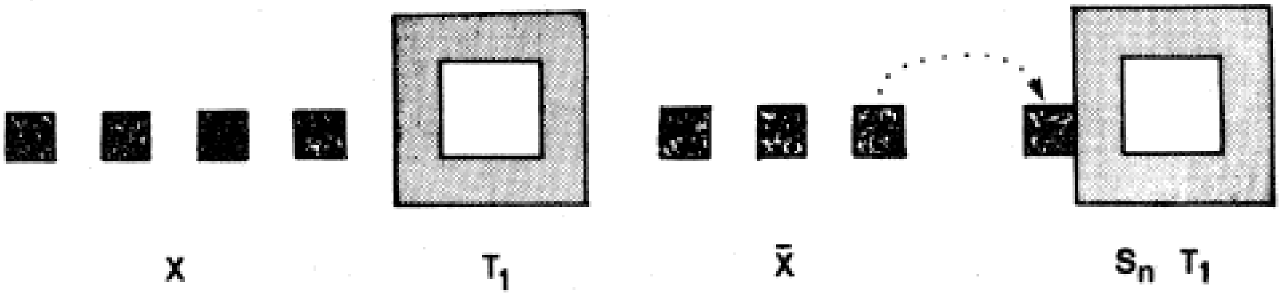

Figure 9

$X \cup T_1$ deforms to $\bar{X} \cup S_n \cup T_1$

Thus

$$\hat{M}(X) = \hat{M}(X \cup \mathrm{sat}(T_1)) = \hat{M}(\overline{X} \cup s_n \cup \mathrm{sat}(T_1))$$

$$= \hat{M}(\overline{X} \cup s_n \cup T_1 \cup \mathrm{sat}(T_1))$$

where the last equality uses saturation to add in T_1. But if s_n is directly adjacent to T_1 then the set $s_n \cup T_1$ is itself topologically an annulus and so has Euler characteristic zero. Thus

$$\hat{M}(\overline{X} \cup s_n \cup T_1 \cup \mathrm{sat}(T_1)) = \hat{M}(\overline{X}) \quad (\text{as long as } \overline{X} \neq 0) .$$

Proceeding in this way, we can eliminate the squares of X one by one.

This completes the proof.

9. Figures in Context.

We recall the following definition from §6.6 of [1].

Definition 9.1 If ψ is a predicate then define a new predicate $\psi_{\text{in context}}$ by

$$\psi_{\text{in context}}(X) = \left\ulcorner \begin{array}{l} \psi(Y) \text{ for some connected} \\ \text{component } Y \text{ of } X \end{array} \right\urcorner$$

Papert and Minsky show that, for such predicates as $\ulcorner X$ is a hollow square$\urcorner$ $\psi_{\text{in context}}$ cannot be computed by a finite order perceptron. In this section we show that uniform linear threshold machines can compute $\psi_{\text{in context}}$ for only the most trivial kind of predicate ψ .

Definition 9.2 We say that a predicate ψ is divisible if ψ satisfies the following condition: For every connected set X on which ψ is true, if we divide X into two disjoint connected sets $X = A \cup B$, then $\psi(A)$ is true or $\psi(B)$ is true.

We can see that most "interesting" geometric predicates are not divisible. For example, if $\psi(X)$ is true and ψ is divisible, then by continual subdividing we see that ψ must be true on the set consisting of one single square of X . Consequently, any predicate which is both divisible and translation invariant must be true on the figure consisting of a single square. Not all predicates which are true on single squares are divisible. Figure 10, for example, illustrates that $\ulcorner X$ is a square$\urcorner$ is not divisible.

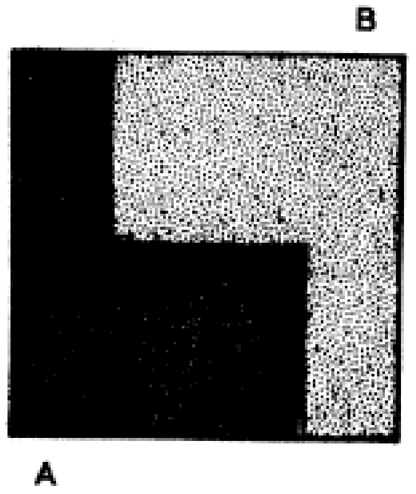

Figure 10

$\lceil$**X is a square**$\rceil$ **is not divisible**

Theorem 9.3 Suppose $\hat{M}$ is a uniform linear threshold machine which computes $\psi_{\text{in context}}$ for some translation invariant predicate ψ. Then ψ must be divisible.

Proof: Suppose ψ is not divisible. Then there exist connected figures A and B such that $X = A \cup B$ is connected, $\psi(A) = \psi(B) = 0$ and $\psi(X) = 1$. Choose a saturation sequence $\{B_i\}$ of sets which are all congruent to B and translate A so that $A \cup B_1$ is congruent to X. (See Figure 11.)

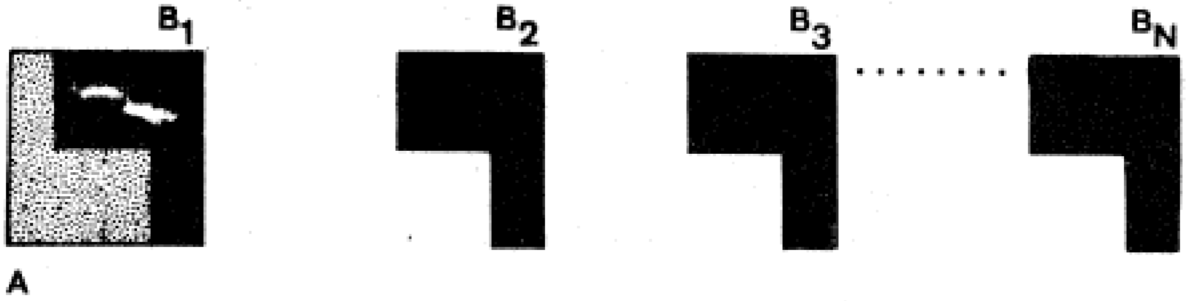

Figure 11

Saturation sequence for $\psi_{\text{in context}}$

Then we have $\psi(B_i) = 0$ for $i = 1,\ldots,N$. Letting $B_2 \cup \ldots \cup B_N = \text{sat}(B_1)$ we have

$$\psi_{\text{in context}}(A \cup \text{sat}(B_1)) = 0$$

and

$$\psi_{\text{in context}}(A \cup B_1 \cup \text{sat}(B_1)) = \psi_{\text{in context}}(X \cup \text{sat}(B_1)) = 1 .$$

On the other hand, we must have

$$\hat{M}(A \cup \text{sat}(B_1)) = \hat{M}(A \cup B_1 \cup \text{sat}(B_1)) .$$

The "negation" of $\psi_{\text{in context}}$ is given by

$$\psi_{\text{all}}(X) = \lceil \psi(Y) \text{ for all connected components } Y \text{ of } X \rceil$$

we leave it to the reader to formulate and prove the corresponding theorem for ψ_{all}, e.g.,

$$\eta(X) = \lceil \text{every component of } X \text{ is a rectangle} \rceil$$

cannot be computed by a uniform linear threshold machine.

10. Bounds for Saturation.

We have shown that uniform linear threshold machines which purport to recognize even very simple predicates must eventually fail on arbitrarily large retinas. But how large is arbitrarily large? This section provides a bound, albeit a rather weak one, in terms of constants associated with the machines.

Definition 10.1 Let $\hat{M} = \Delta\hat{\Phi}$ be a uniform linear threshold machine of rank r on $\hat{R}$:

$$\hat{\Phi} = (\hat{\varphi}_1, \hat{\varphi}_2, \ldots, \hat{\varphi}_r)$$

$$\hat{\varphi}_i(X) = \left\lceil \sum_{x \in X} \mu_i(x) > \theta_i \right\rceil$$

Let $S_i(X) = \sum_{x \in X} \mu_i(x)$, let $S(X) = \max_{i=1,\ldots,r} |S_i(X)|$

let $T = \max_{i=1,\ldots,r} |\theta_i|$, and let ϵ be the minimum sensitivity (2.5) of the $\hat{\varphi}_i$. Now define sequences $m(i)$ and $b(i)$ by

$$m(0) = \max_{X \subset R^1} S(X)$$

$$b(1) = \frac{m(0) + T}{\epsilon} + 1$$

and, inductively,

$$m(n) = \max_{X \subset R^{b(n)}} S(X)$$

$$b(n+1) = b(n) + \frac{n+1}{\epsilon}(m(n) + T) \quad .$$

Finally, let $N(\hat{M}) = b(3^r)$.

<u>Theorem 10.2</u> Let $\hat{M}$ be as above and let $\{A_i \subset R^i\}$ be any sequence of disjoint sets. Then there exist, among the first

$N(\hat{M})$ terms of the sequence, a set $A =$ some $A_n \subset R^n$ and a set $B =$ (union of A_i) such that B saturates $\hat{M}$ with respect to A on R^n .

In other words, for the purposes of applying the Saturation Theorem, $R^{N(\hat{M})}$ is "arbitrarily large".

Proof: If we examine the proof of the Saturation Theorem (7.3) we see that we can saturate $\hat{M}$ if we can find some $A_n \subset R^n$ and N more A's $A_{i(1)}, A_{i(2)}, \ldots, A_{i(N)}$ such that

1) All the A_i have the same vector $\Gamma(A_i)$

2) $N > \frac{M + T}{\epsilon}$ where $M = \max\limits_{X \subset R^n} |S(X)|$.

Since there are 3^r possible values for $\Gamma(A_i)$ the theorem will follow at once from the Lemma below, if we choose $p = 3^k$ and $f(X) = \Gamma(X)$.

Lemma 10.3 Let f be a function from subsets of $\hat{R}$ to the set of integers $\{1,2,\ldots,p\}$. Then from any sequence of $b(p)$ subsets of $\hat{R}$ we can extract sets $A \subset R^n$ and $A_{i(1)}, A_{i(2)}, \ldots, A_{i(N)}$ with

1) $f(A) = f(A_{i(1)}) = \ldots = f(A_{i(N)})$

2) $N > \frac{M + T}{\epsilon}$ $M = \max\limits_{X \subset R^n} |S(X)|$.

Proof: By induction on p.

First, if $p = 1$, then all A's in the sequence automatically have the same value $f(A)$, so choose $A = A_1 \subset R^1$ and condition (2) is fulfilled since $b(1) > \frac{m(0) + T}{\epsilon}$.

Now we assume that the lemma is true for p and prove it for $p + 1$. Suppose we have a sequence of $b(p+1)$ elements, with f taking values in the set $\{1,2,\ldots,p+1\}$. Break the sequence into two pieces:

the first $b(p)$ elements

and

the remaining $\frac{p+1}{\epsilon}(m(p) + T)$ elements.

If f applied to the first piece takes on at most p distinct values then the lemma follows by induction.

Otherwise f takes on all $p+1$ possible values among the first $b(p)$ elements. Now among the second group of elements there is some value which f assumes at least $\frac{m(p)+T}{\epsilon}$ times. But there is also some set A among the first $b(p)$ elements on which f assumes this value. So let the desired sequence be $A \subset R^{b(p)}$ followed by the remaining $\frac{m(p)+T}{\epsilon}$ elements selected from the second group.

Example 10.4 Suppose that the measure functions μ_i used by

$\hat{M}$ are bounded, i.e., $|\mu_i(x)| \leq k$ for $x \in \hat{R}$. Suppose also that the size of the retinas R^n grows linearly with n , $|R^n| = cn$. (This is sufficient for all applications of the Saturation Theorem in Sections 8 and 9.) Then we can estimate $N(\hat{M})$:

$$m(n) = kc\, b(n)$$

$$b(n+1) = b(n) + \frac{(n+1)}{\varepsilon}[kc\, b(n) + T]$$

or $$b(n+1) - b(n) = \frac{n+1}{\varepsilon}[kc\, b(n) + T]$$

and we can estimate the growth of $b(n)$ by considering the differential equation

$$\frac{dy}{dx} = C_1 xy + C_2 \quad .$$

This has the solution

$$y = AC_2\, e^{C_1 x^2/2} - \frac{C_2}{C_1}$$

and so we find

$$\log b(n) \sim n^2 \quad .$$

Finally $N(\hat{M}) = b(3^r)$ so we get

$$\log \log N(\hat{M}) \sim r \quad .$$

Corollary 10.5: For a bounded linear threshold machine to avoid being saturated on large retinas, the rank must grow at least as fast as $\log \log |R|$.

11. Conclusion.

It is instructive to compare the results of this paper with those of [1]. Minsky and Papert demonstrated limitations of perceptrons of small order, providing mathematical justification for the intuition that these computational schemes are somehow too "local" to deal with such "global" predicates as connectivity. Here we have taken a complementary point of view, investigating the limitations of the linear threshold element itself as a decision element.

Like Minsky and Papert we believe that the value of this work lies in the general phenomena that it illuminates rather than in the precise statements of the theorems. In our case, we have shown that Minsky and Papert's "stratification phenomenon" appears in the class of linear threshold machines as well as in perceptrons. We have also indicated the importance of saturation as a potential pitfall for any machine attempting to recognize patterns using only a small number of threshold elements.

Hopefully, all of these results will someday be subsumed by a general mathematical theory of pattern recognition, a theory which will clarify the intuitive guess that any system for "general purpose" pattern recognition must have the ability to "focus in on" local features and also the ability to combine this local data in flexible "global" ways.

Reference

M. Minsky and S. Papert, Perceptrons: An Introduction to Computational Geometry, Cambridge, M.I.T. Press, 1969.