THE UNIVERSITY OF OTTAWA
Department of Electrical Engineering
OTTAWA, CANADA

PROPERTIES OF A TRAINABLE LINEAR THRESHOLD LOGIC UNIT

by

I. J. Zawicki

Submitted in partial fulfillment of the requirements for the degree of Master of Science.

June, 1965.

Approved for the Department of
Electrical Engineering

_____

Supervisor

_____

Chairman of the Examining Committee

_____

Chairman of the Department

ABSTRACT

Many tasks can be reduced to the problem of pattern recognition and the vast majority of applications of learning machines is concerned with such problems. The examples of pattern recognition are : speach recognition, handprinted characters recognition, weather forecasting, automatic control of technological processes, etc. .

The subject matter of this work is the detailed analysis of the basic element of the neuron-net-like learning systems - the Linear Threshold Logic Unit.

As a mathematical model a many-dimensional vector space is used. This approach gives clear insight into the properties of the element and is particularly fruitful in the analysis of process of training.

In the first part of the work, the general problem of the pattern recognition is presented and some properties of the basic element of the learning machine are discussed.

The second part is concerned with the training procedures for the LTLU. Both, geometrical and analytical treatments of the Error Correction Procedures are discussed in details.

Other learning procedures are also surveyed.

ACKNOWLEDGMENTS

The author wishes to express his sincere thanks to his supervisor Professor G. S. Glinski for introducing the author to learning machines, providing facilities, valuable advices and encouragement throughout the course of this work.

The author also acknowledges Dr. J. Brzozowski's critical comments and suggestions.

TABLE OF CONTENTS

INTRODUCTION

In the last decade there has been a great interest in searching for some answers to such questions as: Can a machine think? What are the functions of the brain and how does the brain work? Is it possible to build machines (computers) which will accept hand written or spoken instructions of humans? How can the abilities of generalization, learning, pattern recognition - be implemented in a machine? Can a machine translate from one language to another, make inductive inference, predict weather, prescribe medical treatment?

The field of scientific activity which deals with this kind of problems is sometimes called bionics. Many other names are also used.

Strict definitions of "thinking", "generalization" and of other mentioned above terms has not been established, but nevertheless an intuitive meaning of these notions is commonly known. Satisfactory answers for these questions would have both philosophical and practical significance.

Since such terms like pattern recognition, learning, learning machine, generalization will be used in this work, some descriptive explanation of the meaning of these terms is given below.

Pattern recognition is a process of decision making in which input event is recognized as a member of a given class by a comparison of its attributes with the already known pattern of common attributes of members of that class [1].

Learning is a process in which a performance of the system is being improved, on the base of past experience by proper adjustments of some parameters of the system.

Some authors [11] make distinction between causal learning (or self-organizing) systems and 'teleological' learning systems.

The causal learning system changes its structure in response to the changes in environment (i.e., in the input events) without an aid of any source from outside (i.e., without a teacher).

In the teleological learning system changes of its structure take place in the attempt to achieve some objective. Some outside source selects examples on which to learn and indicates what changes should be made in the system in order to improve its performance.

Examples of causal learning systems are some complex models of the perceptrons [2]. The simple perceptron or the single Linear Threshold Logic Unit are the examples of the teleological learning systems.

The distinction between causal and teleological systems (called also supervised and unsupervised [10]) does not seem to be essential since any teleological system can be considered to be a causal one if the information (signal) about the goal of the system is thought of as an element of the input event.

A learning system is said to possess the ability of <u>generalization</u> if, having been trained, it can give a correct response to input events which were not shown during the training phase.

2. <u>Pattern Classification</u> [1], [10]

Let the set of patterns or input events to the system be called Ω. The set Ω is to be divided into subsets $\Omega_1, \Omega_2, \ldots, \Omega_m$. Any member of the set of input patterns (element of Ω) will be labelled ω (i.e., $\omega \in \Omega_i \subset \Omega$, for some i).

Examples

1. In weather forecasting all possible sets of input data about atmospheric conditions from the weather map are considered as a set of input pattern Ω. The distinct responses required from the forecasting system like: fair, rainy, cloudy, foggy, brisk - correspond to certain subclasses of input data, $\Omega_1, \Omega_2, \ldots, \Omega_m$.

2. In alphanumeric character recognition, there may be as many subclasses Ω_t as there are letters and digits. Particular letters or numbers correspond to ω.

Between a set of original objects to be recognised and classified, and an input to the learning machine there is a device which will transform the original information into signals which can be accepted by the input of the learning machine. This device is called a transducer. In the transducer, on each input signal ω is made a sequence of numerically valued measurements $a_1, a_2, \ldots, a_N$. It is convenient to regard the set of measurements, performed on particular pattern, as a vector in N dimensional vector space R^N.

Thus, with a specified event ω_k is associated a vector

$$a^k = (a_{1k}, a_{2k}, \ldots, a_{Nk})$$

Each of N measurements $a_1, \ldots, a_N$ expresses a property of the event. Numerical value of a_i $1 \leqslant i \leqslant N$ corresponds to the "amount" of i-th property. An appropriate choice of the measurements $a_1 \ldots a_N$ is itself an important and nontrivial problem. The representative attributes should be chosen in such a way that points corresponding to a class of patterns were clustered i.e., distances between members of the same class should be small in

comparison with the distances between elements of different classes.

The results of the measurements performed by the transducer upon ω are now fed into the main part of the system which is called the processor. The processor decides to which class a particular ω belongs. The pictorial illustration of the described process is shown in Fig. 1. In Fig. 1b three clusters represent three classes of input events.

The transducer and the processor can be thought of as transformations which are applied to the original data Ω. The transducer, first, maps each point ω of Ω into a point $\underline{a}$ of the N-dimensional vector space R^N. The processor then partitions R^N into disjoint sets $A_1, A_2, \ldots, A_m$ and if the results of the measurements upon ω_k lie in A_t, i.e., $a^k \in A_t$, the processor makes the decision that $\omega_k \in \Omega_t$. If there are sufficiently many measurements to distinguish between the sets Ω_t, i.e., if $a_k \neq a_i$ when ω_k and ω_i belong to different subsets a processor exists which achieves the proper identification of points in Ω.

In this work we will study a class of partitions of R^N that can be achieved by means of a single adaptive element called a linear threshold logic unit [LTLU].

3. The Linear Threshold Logic Unit.

The LTLU is a device that can realize any one of many different logical functions by adjusting its internal parameter values. Such a unit is shown schematically in the upper part of Fig. 2.

In the lower part of Fig. 2, the adjusting system (adaptor) is shown, which makes, during the training period, a sequence of adjustments which are determined by the performance of the network

for the previous adjustments and by the assumed training procedure (or rules of adaptation). These two devices connected together form a self-organizing logical system (learning machine).

Adjusting system is a fixed logic device and it will not be discussed here.

There have been three main trends in the research work investigating the properties of the threshold logic elements.

The threshold logic approach [6] applies the methods of analysis originated in switching circuits theory and deals with fixed non-adjustable elements operating on two-valued signals.

Equivalence between statistical recognition and threshold functions gives possibilities of probabilistic approach to the problem [18, 19]. Probabilistic methods seem to be very promising when applied to large systems.

The third approach, which is also adopted in this work is based on the utilization of some concepts and methods of linear algebra. This approach allows the mathematical analysis of trainable networks.

Although the linear algebraic methods can be applied in the case of many valued or continuously variable input signals as well, we will confine ourselves to the detailed discussion of the examples of units operating on two-valued signals (0 and 1 or -1 and +1) assuming ideal characteristics of these devices, i.e., no account will be taken of such problems as drifts in components, limits on the values of weights, hysteresis, and others.

Within the LTLU a linear combination of input signals is formed. Input variables are multiplied by weight values w_i

and the resulting products are summed. The analog output

$$\eta = \sum_{i=0}^{N} a_i w_i \qquad (1)$$

is fed into the quantizer to form a quantized output. The relation (1) can be considered as an inner product (a, w) of the input vector $\underline{a}$ with coordinates a_i and the weight vector w with coordinates w_i ($0 \leq i \leq N$).

For fixed-weight settings, to each of the possible input combinations (the number of these combinations is 2^N, for binary inputs), can be assigned either + 1 or - 1 output (assuming that the dead zone - θ - of the quantizer is equal to zero). It means that all possible inputs are classified into two categories. The input-output relationship is defined by setting the weights w_0, w_1, ..., w_N during the training phase. Since to each of the 2^N inputs can be assigned one of two possible outputs, there are 2^{2^N} different possible functions of N binary variables. In general not all of these 2^{2^N} functions can be realized with the single LTLU, i.e., the output cannot be arbitrarily assigned to each input configuration (see Sec. 5).

As an example of application of the Linear Threshold Logic Units simple perceptron can be considered [2] .

The principle of operation of a Simple Perceptron is shown in Fig. 3. Sensory units (S layer) are connected to the inputs of the associative units (A layer). The weights associated with sensory units may be random numbers from the set {0, -1, + 1 }. When the pattern ω_i is presented to the sensory layer ("retina") (0 or 1 input signals) the sum of the weighted input signals is formed

within the A layer units. The output signal from the particular A unit is 0 if the $\eta' < \vartheta$ (ϑ is an arbitrary constant, real number-the threshold). If $\eta' \geq \vartheta$ the output is +1. The equivalent circuit for A-unit is given in Fig. 4.

In the Simple Perceptron connections between sensory units and associators are random and unchangeable i.e., the set of A units activated by a particular input pattern ω_i does not change. One can say that input pattern is transformed ("pre-processed") into another pattern by the system of A units. The new pattern from the outputs of the A units is fed (as binary signals) into the inputs of the output layer R (response layer). The response layer is a LTLU with N inputs and N weights (the threshold is equal to zero). The output from the response unit is +1 if weighted sum of the inputs from associators is greater than $+\theta$ and -1 if this sum is smaller than $-\theta$. θ is a dead zone of the quantizer. Output is equal to zero if $|\eta| \leq \theta$. The weights of the response unit are adaptive. Adaptation is done according to certain rules which guarantee convergence to a solution, if the solution exists at all. A discussion of the adaptation rules is given in Sec. 7.

4. Mathematical Background

In this section the interpretation of the LTLU in terms of the N-dimensional euclidean vector space will be given.

In geometrical interpretation of the relations implemented by the LTLU we will utilize the properties of convex sets and duality in Euclidean space [3, 8] . Therefore, the following definitions and theorems from linear algebra will be useful.

Definition 1

A set C is said to be convex if, whenever a^1 and a^2 are points of C, the entire line segment between a^1 and a^2 also belongs to C. From Fig. 5 it may be seen that an arbitrary point a^i on the segment between a^1 and a^2 can be expressed by formula:

$$a^i = t_i a^1 + (1 - t_i) a^2$$

$$\text{for } 0 \leq t \leq 1$$

Examples of two-dimensional convex and nonconvex sets are shown in Fig. 6.

Definition 2

If a^j is a fixed point (vector) in n-dimensional space and θ is a number, then the truth sets of inequalities of the form $(a^j, x) < \theta$ or $(a^j, x) > \theta$ are called open half-spaces while the truth sets of the inequalities $(a^j, x) \leq \theta$ or $(a^j, x) \geq \theta$ are called closed half-spaces in this space. The truth set of the equation $(a^j, x) = \theta$ is a hyperplane. (The truth set of inequality is a set of elements x for which the inequality holds).

Theorem 1

The intersection of two or more convex sets is a convex set.

Theorem 2

A half-space (closed or open) is convex.

Definition 3

The intersection of a finite number of closed half spaces passing through the origin is called a convex cone. Examples of convex cones in two dimensions are shown in Fig. 7.

Definition 4

The convex hull (or convex cover) C(A), of an arbitrary set A of points is defined as the smallest convex set which contains A. It is a set of all convex combinations $(\lambda_1 a^1 + \lambda_2 a^2 + \ldots + \lambda_q a^q; \sum_{i=1}^{q} \lambda_i = 1)$ of points a^i from A. An example of a convex hull of a set of points $a^1, a^2, \ldots, a^k$ in 2-space is shown in Fig. 8.

Duality in Euclidean Space.

There is no precise definition of duality in Euclidean space [3] and therefore there is a choice of dual spaces available.

A dual space to N-dimensional euclidean space R^N is a space in which the hyperplanes or half-spaces of R^N are represented by points or half-lines.

For our purpose the space of weights will be considered as dual of the space of inputs . Since duality is a symmetric relation, the input space could be considered as a dual of the weight space.

Although both, the input vectors and the weight vectors can also be thought as vectors of a single Euclidean space, the notion of duality yields better insight into some properties of the LTLU.

LTLU and R^N - $(R^N)^*$ duality.

Let us condiser N-dimensional space R^N of input signals of the LTLU and N-dimensional space of corresponding weights $(R^N)^*$. If A is any subset of R^N then the dual of A is that subset W of $(R^N)^*$, whose elements satisfy the inequality.

$$(a, w) \geq \theta$$

where $a \in A \subset R^N$, $w \in W \subset (R^N)^*$

If A is a single point $a^1 = (a_{11}, a_{21}, \ldots a_{N1})$ i.e., in our case one combination of input signals representing pattern ω_1 then W is a closed half-space in the weight space $(R^N)^*$.

If inequality $(a, w) \geq \theta$ has to be satisfied simultaneously for a certain number of sets A_j (or points a^j) i.e., for $U_j A_j$, the dual of $U_j A_j$ is the intersection of the half-spaces which are dual to A_j's. Such an intersection is a convex cone (Definition 3).

The R-unit (Fig. 3) of the Simple Perceptron can be discussed in terms of duality described above. If we assume for a moment that the dead zone of the quantizer is equal to zero i.e., $\theta = 0$, inequalities

$$a_1 w_1 + \ldots + a_N w_N > 0 \qquad (2)$$

$$a_1 w_1 + \ldots + a_N w_N < 0 \qquad (2a)$$

determine two open half-spaces (in $(R^N)^*$ space of weights) which are separated by a hyperplane

$$a_1 w_1 + \ldots + a_N w_N = 0 \qquad (3)$$

In general the number of these hyperplanes is equal to the number of possible input combinations. However, in the case when input combination is zero vector i.e., $a = (0, 0, \ldots 0)$, there is no hyperplane since any vector in weight space satisfies equality (3). Thus we can say that the whole space is divided into certain number of convex sets by $(2^N - 1)$ hyperplanes. If the input combination $a = (0, 0, \ldots 0)$ does not appear (for example in the case where inputs are +1 or -1), the space is divided by 2^N hyperplanes. Each of these hyperplanes has as its normal an input vector and passes through the origin, since the point $w = (0, 0, \ldots 0)$ is a solution to equation (3). Therefore the weight space is divided into certain

number of cones [9] . If two different weight vectors lie on the same side of each one of the $(2^N - 1)$ or 2^N hyperplanes, they belong to a certain cone and they will give identical outputs for any input and therefore they represent the same function.

However, the quantizer of the R-unit has a dead zone , and corresponding inequalities take on the form

$$a_1 w_1 + \ldots + a_N w_N > \theta \qquad (2')$$

$$a_1 w_1 + \ldots + a_N w_N < -\theta \qquad (2'a)$$

thus, instead of one separating hyperplane, corresponding to every input combination, there are two parallel hyperplanes situated symmetrically with respect to origin.

The distance between the two hyperplanes is $2\|d\| = 2\frac{\theta}{\|a\|}$.

A three-dimensional illustration is shown in Fig. 9.

Example

In the case of a two inputs-two weights LTLU similar to the R-unit of the Simple Perceptron, a geometrical representation of the function $f = \bar{a}_1 \bar{a}_2 + \bar{a}_1 a_2 + a_1 a_2 = \bar{a}_1 + a_2$ is shown in Fig. 10. For all weight vectors which lie in the shaded cone in Fig. 10b the LTLU will realize the same function which in input space is represented by Fig. 10a. Indeed, it can be seen, that the scalar product of the input vectors 1, 2 and 4 with arbitrary weight vector w from the shaded area is greater or equal to zero, and for the input vector 3 the scalar product is smaller than zero. (We are assuming in this example that the output of the quantizer is +1 if $(a, w) = 0$).

An example where the dead zone is not equal to zero is shown in Fig. 11. Fig. 11a shows how the input space is partitioned. In Fig. 11b the shaded area represents a set of weight vectors for which a function given in Fig. 11a will be realized. We should notice that in this particular case the classification is not a binary one. Inputs $\bar{a}_1 a_2$ and $a_1 a_2$ are classified as +1, the input $a_1 \bar{a}_2$ as -1, but the input $\bar{a}_1 \bar{a}_2$ will be classified as zero. One could avoid in this case the ternary classification using as the input signals +1 and -1 instead of 0 and 1.

The N inputs -(N+1) weights element, such as that shown in Fig. 2 can also treated in terms of the duality described in the preceding section assuming that there are N+1 inputs, the (N+1) -th input being excited by a constant signal +1. In this case the set of points in input space also consists of 2^N different input combinations (patterns).

The (R^{N+1}) dimensional weight space is divided into a number of cones, each of which corresponds to one of 2^N input vectors. By the same argument as before we can see that each of these cones corresponds to one of the functions which can be realized by the LTLU. Geometrical representation of 1-input - 2-weights unit is shown in Fig. 12.

Number of cones obtained when the (N+1) -dimensional space is partitioned by 2^N hyperplanes is, in general, larger than in the case of partitioning the N-dimensional space by the same number of hyperplanes. Therefore, the number of functions which can be realized by (N+1) variable weights element is larger than in the case of N-weights element.

The problem of how many functions of N-variables can be realized by LTLU or into how many cones the space is divided by 2^N hyperplanes will be discussed in the next section.

5. Linear Separability

As we already know, the possible combinations of binary input signals in N-dimensional space are the vertices of the N-dimensional cube.

Let $A_p = \{a^1, a^2, \ldots, a^p\}$ and $A_{q-p} = \{a^{p+1}, \ldots, a^q\}$

be two finite subsets of R^N. Each of a^j's represents a vector in N-dimensional space (or vertex of the N-cube corresponding to one combination of input signals). In other words

$$a^k = (a_{1k}, a_{2k}, \ldots, a_{Nk})$$

corresponds to a pattern ω_k

The pair (A_p, A_{q-p}) is said to be linearly separable if there exists a vector.

$$w = (w_1, w_2, \ldots, w_N)$$

satisfying the following set of inequalities

$$(a^j, w) > -w_o > (a^l, w) \qquad (4)$$

$$1 \leqslant j \leqslant p \qquad p+1 \leqslant l \leqslant q$$

The condition (4) in input space means that all points a^j $(1 \leqslant j \leqslant p)$ i.e., the set A_p, should lie "above" some hyperplane $aw = -w_o$ and points $a^l (p+1 \leqslant l \leqslant q)$ i.e., the set A_{q-p} should lie "below" the hyperplane. The existence of the separating hyperplane is equivalent to the existence of the solution for the system of inequalities (4).

The necessary and sufficient conditions for the existence of this separating hyperplane are obtained from the theory of convex sets.

There is a theorem in the theory of convex sets [3] which states that two convex hulls (Definition 4) in R^N may be strictly separated by a hyperplane if and only if they are disjoint, i.e., if A_p and A_{q-p} are finite sets in R^N, then (A_p, A_{q-p}) is linearly separable iff $C(Ap) \cap C(A_{q-p}) = \phi$, where ϕ is an empty set. Taking into account the relations (4) and the theorem mentioned above we can say that iff A_p and A_{q-p} are such that

$$C(A_p) \cap C(A_{q-p}) = \phi$$

where ϕ is an empty set, there exists a vector $w \in R^N$ and a number w_o satisfying inequalities (4).

The logical functions which can be realized using a single LTLU are referred to as being "linearly separable" functions. Those which are not realizable are called "non-linearly separable" functions.

As an example of classification which cannot be perfomed by the two inputs LTLU one can give the situation which is shown in Fig. 13a. If we wanted the points (the input combinations) 1 and 3 to be classified, say, as a +1 and the points 2 and 4 as a -1 we could not achieve this. It is apparent that there does not exist, in this case, any linear function whose truth set (a straight line in our case) could yield this particular classification. This is because the convex hulls of our two sets (linear segments 1-3 and 2-4) are not disjoint. They have a common point which is the point of intersection T.

In the Fig. 13b there is shown one of the possible classifications which can be realized by two-inputs three-weights element. The shaded area represents the convex hull C(1, 2, 3) of the input vectors numbered as 1, 2, 3.

The problem of how many functions of N variables can be realized with N-inputs LTLU will be discussed in section 6.

The inequalities (4) can be restated in the form

$$(a^{*j}, w^*) > 0 > (a^{*\ell}, w^*) \tag{5}$$

where $a^{*j} = (1, a_{1j}, a_{2j}, \ldots a_{Nj})$ $w^* = (w_o, w_1, \ldots, w_N)$

Since A_p and A_{q-p} are finite sets, there must exist a $\theta^* > 0$ such that

$$(a^{*j}, w^*) > +\theta^* > 0 > -\theta^* > (a^{*\ell}, w^*) \tag{6}$$

$$1 \leqslant j \leqslant p \qquad\qquad p+1 \leqslant \ell \leqslant q$$

But if (6) has a solution for some $\theta^* > 0$, it has a solution for every $\theta > 0$ for if w^* corresponds to θ^*, then $C_1 w^*$ is a solution for $\theta = C_1 \theta^*$. Thus, the existence of the dead zone of the quantizer does not limit the classification abilities of a system.

Now, let us assume that any one of the input patterns - represented by vectors a^{*k} $(1 \leqslant k \leqslant q)$ - can be assigned to one of the two classes $\{+1, -1\}$. Multiplying each vector a^k by the desired classification ρ_k $(= +1 \text{ or } -1)$, the set of inequalities (6) can be written

$$(b^{*k}, w^*) > 0 \tag{7}$$

where

$$b^{*k} = a^{*k} \rho_k, \quad (1 \leqslant k \leqslant q)$$

The vectors b^{*j} for which the desired output is always positive (+1) will be called the "positive input vectors".

Since the inequalities (4), (5), (6), (7) are all equivalent to each other, the algorithm for determining vector w^* in (7) solves the problem of separation of the sets A_p and A_q.

The geometrical interpretation of the formula (7) is the following [4]. The existence of a vector w^* satisfying (7) means that the angle between arbitrary vector b^{*k} and a vector w^* is smaller than 90^o. The angle between two vectors b^j and w is defined as $\cos\alpha = \frac{(b^{*j}, w)}{\|b^j\| \|w\|}$. This is equivalent to the fact that the polyhedral cone C, defined as a set of vectors of the form $\lambda_1 b^{*1} + \ldots + \lambda_q b^{*q}$; $\lambda_1 \geq 0 \ldots \lambda_q \geq 0$ is a proper cone i.e., that C never contains both b^{*k} and $-b^{*k}$ except for $b^{*k} = 0$. In other words C (apart from the vertex) lies in the interior of a half-space. The cone C^* of all vectors w^* such that

$$(b^{*k}, w^*) > 0$$

for all $b^{*k} \in C$ is called the dual cone to C. The larger C is, the smaller C^* is. When C is a half space, C^* is a half line. Two dimensional example is shown in Fig. 14. Here α_1 corresponds to C and α_2 - to its dual C^*. [1)]

Example

Let us consider the Boolean function

$$f = a_1 a_2 + \bar{a}_1 \bar{a}_2$$

All possible combinations of input signals (or input vectors) are shown in the truth Table 1

Table 1

	a_o	a_1	a_2	f
a^1	+1	-1	-1	+1
a^2	+1	-1	+1	-1
a^3	+1	+1	+1	+1
a^4	+1	+1	-1	-1

Table 2

	b_o	b_1	b_2
$b^1 = a^1$	+1	-1	-1
$b^2 = -a^2$	-1	+1	-1
$b^3 = a^3$	+1	+1	+1
$b^4 = -a^4$	-1	-1	+1

1) In the remaining part of this work, input vectors $\underline{a}$, positive input vectors $\underline{b}$ and weight vectors $\underline{w}$ will be denoted without asterisk.

Geometrical representation of the vectors b^j corresponding to the function f is shown in Fig. 15. It can be seen from this figure that in this case there is no vector w which makes an acute angle with all of the positive input vectors $b^1 \ldots b^4$. This is so, because the function $f = a_1 a_2 + \bar{a}_1 \bar{a}_2$ is not a linearly separable function.

6. Number of Linearly Separable Functions [5,6]

It was mentioned in the preceding section that the number of functions which can be realized by LTLU is equal to the number of cones into which the space of weights can be partitioned by hyperplanes each of which corresponds to one possible combination of input signals.

The general problem of how many linearly separable functions there are for N-variables remains unsolved at present. The calculation of an upper bound for the number of linearly separable functions R(N) can be obtain by considering the problem of the maximum number of regions (cones) into which any number of hyperplanes passing through the origin may divide a space of any dimensions.

Let $C_{m,n}$ mean the number of regions into which m hyperplanes passing through the origin may divide a space of n dimensions. It is obvious that $C_{m,1} = 2$ for each $m > 0$, $C_{m,2} = 2m$, $C_{1,n} = 2$ for any $n > 0$.

The general formula can be derived by the following argument. Suppose that formula has been established for m-1 hyperplanes in n-dimensional space. The m-th hyperplane will be divided by m-1 hyperplanes (along at most m-1 hyperlines) into $C_{m-1,n-1}$ pieces. Each of these hyperplanar pieces divides the region it belongs to into two new regions i.e. we have added to $C_{m-1,n}$ regions at most $C_{m-1,n-1}$ new regions, i.e.,

$$C_{m,n} = C_{m-1,n-1} + C_{m-1,n} .$$

This formula can be given as

$$C_{m,n} = 2 \sum_{k=o}^{n-1} \binom{m-1}{k}$$

where

$$\binom{i}{j} = \frac{i!}{j!\ (i-j)!}$$

Thus, the number of linearly separable functions realized by LTLU of N inputs and N weights

$$R(N) \leqslant 2 \sum_{k=o}^{N-1} \binom{2^N - 1}{k}$$

If there are N+1 weights and N inputs

$$R(N+1) \leqslant 2 \sum_{k=o}^{N} \binom{2^N - 1}{k}$$

Another formula both for lower and upper bound of the number of linearly separable functions is [7] :

$$2^{0.33N^2} < R(N) < 2^{N^2} \ll 2^{2^N} \quad (N > 1)$$

The ratio of R(N) to the total number of functions of N variables decreases as N increases.

7. Training Rules for the LTLU.

7.1. Generalities

During the period of training (or learning) some representative patterns, from the set of all possible input patterns, are presented to the input of the learning machine. If output for a given pattern differs from the desired output, changes in internal parameters of the machine are made. In the case of the simplest learning machine which is the LTLU, these changes are performed on the weights.

The rule according to which the training is made, is called in literature the training procedure, adaptation procedure, reinforcement rule or training algorithm.

The most important criterion by which any adaptation procedure must be evaluated is the convergence to a solution, if the solution exists. It is preferable that a solution be reached in a finite number of corrections of the weight values.

One of the first adaptation procedure for the LTLU was proposed by Mattson [12]. This method was based on the assertion that if a dividing hyperplane is oriented in the input space so that it maps at least one point correctly, then it can be systematically reoriented, by changing weights, one at a time, to map all points correctly. The training rules required a memory of past performance and the ability to alter individual weights. However, it turned out [7], that it was not always possible to realize a given linearly separable function by changing only one weight at a time. Even with a very good initial setting of the weights a combination of changes sometime is needed to obtain convergence of the iterative procedure. In any case, Mattson's method assures a statistical separability of the set of input patterns.

Another iterative procedure for the training of the LTLU was suggested by Widrow and Hoff [13]. A pattern is fed to the input of the LTLU and the desired output (+1, or 0 or -1) is compared with the analog output of the summer. All weights $w_o \ldots w_N$ are to be changed by the same absolute magnitude such that the error Δ is brought to zero.

This is accomplished by varying each gain in the direction which will decrease the error by amount $\frac{\Delta}{N+1}$. After all changes

are made, the error for the present input pattern is zero. The next pattern and its desired output is presented and the error is observed. The same adjustment routine is repeated i.e., the error is again brought to zero.

Above procedure also gives a statistical classification of the set of input patterns and permits occasional incorrect responses. It allows to classify (statistically) the set of patterns into more than two categories using multilevel quantizers and following the same adaptation procedure.

Many other rules for training of the learning machines have been developed. In perceptron, for example, [2], inactive weights are decremented in such a way that the sum of the weights remains constant.

In what follows we will confine ourselves to the discussion of those of the learning procedure, which always guarantee convergence to a solution, if the solution exists. This kind of procedures can be called the deterministic training procedures.

In the Simple Perceptron (or the α-perceptron), Rosenblatt was using a training procedure known as the Error Correction Procedure, [2, 14].

Though the Simple Perceptron is a more complicated device than single LTLU, in both cases the same training procedures may be applied. This is so, because in the Simple Perceptron adaptation is made only in the last layer which is a LTLU. The other components of the Perceptron (the associator units) realize some fixed transformation on input patterns and they have no influence on the training rules.

It is evident from the considerations of Section 5 that any training procedure for the LTLU should result in a construction of the vector w which will be a linear combination of the positive input vectors with positive coefficients $\lambda_1, \ldots, \lambda_q$.

$$w = \lambda_1 b^1 + \lambda_2 b^2 + \ldots, + \lambda_q b^q$$

There have been many different approaches to the problem of the construction of the vector w.

Some of these methods were discovered as a result of investigation of learning machines.

One of the training procedures (the Relaxation Method-see Section 7.3), was originally known as an iterative method for solving linear inequalities.

It has been shown by Mays [15] that some of the learning methods are related to certain differential equations. It is possible to define such a surface that the corresponding adaptation rules move the weight vector along the gradient of the surface in a direction to minimize values by which inequalities (7) are not satisfied. If the time derivative of the weight vector is equal to the negative gradient of the surface then the solution of the differential equation formed in this manner is stable and it is a solution of (7). A learning procedure is an approximate solution of the differential equation.

7.2 The Error Correction Procedure and its Variants.

7.2.1 Description.

A pattern from the training series is shown to the input of the LTLU and the unit gives a response. If this response is correct (i.e., if it conforms to the desired response), then no change is made in the system. If the response is not correct, the weights w_i of the

LTLU are incremented by $a_{ij} \cdot I \cdot \rho_j$. In the case where inputs are 0 or 1, the weights corresponding to inactive inputs ($a_{ij} = 0$) are left alone.

The proportionality coefficient I is bounded in magnitude above i.e., $0 < I \leq I_{max}$ but not necessarily constant from one adaptation to the next.

It is possible to distinguish three modifications of the above procedure:

I. The coefficient I is fixed (fixed increment adaptation [16]). When a pattern is shown to the system and adaptation is needed, the weight vector is changed in a single step by $Ia^j \rho_j$. No matter whether the output is changed or not after the adaptation, next pattern is presented.

II. The coefficient I is fixed but every time a correction is required for the presented pattern, it is repeated by $I a^j \rho_j$ until the system gives a correct response to the pattern. Next pattern is fed to the input.

III. The coefficient I is continuously variable in such a way that correct output to each of the presented patterns is enforced in a single step. Next pattern is presented.

It is assumed, in all three cases, that every pattern occurs infinitely many times during the training, if it is necessary.

If I in the II case is sufficiently small, then both procedures II and III are equivalent.

These training rules always adapt a wrong output by adding a correction vector Δw to the actual weight vector w. The i-th component of the correction vector is $\Delta w_i = \rho_i a_{ij} I$ $(0 \leq i \leq N, \ 1 \leq j \leq q)$.

The vector $\Delta w = (\Delta w_o, \Delta w_1, \ldots \Delta w_i, \ldots \Delta w_N)$ is just a multiple of the corresponding positive input vector $b^j = \rho_j a^j$, therefore it is orthogonal to the hyperplane $(a^j, w) = 0$ in N+1 dimensional space.

The rules adapt the actual weight vector in a manner to decrease the distance between it and the set of weight vectors which realize the desired function. This is shown in Fig. 16 for the case of one input element. The crosshatched area shows the set of vectors w which realizes the function $f(a_1) = a_1$ (the identity function)

Table 3

	a_o	a_1	$f(a^j)$
a^1	+1	+1	+1
a^2	+1	-1	-1

a_j	ρ	Sgn Δw
+	+	+
+	-	-
-	+	-
-	-	+

If w_* is the shortest weight vector which realizes the function and $w^o = A$ is the starting point then the adaptation goes in the following way. In the first step, a pattern corresponding to $a^1 = (+1, +1)$ is applied. The actual output is 0 and the desired output is +1 therefore the vector AB, proportional to $b^1 = a^1$ is added to the w^o (i.e., both w_o and w_1 are incremented). In the next step, pattern corresponding to $a^2 = (+1, -1)$ is fed to the input. This time the desired output is -1 (therefore $b^2 = -a^1$), the actual output is +1. The correction vector is proportional to b^2 and it is equal to BC. Now w_o has been decreased and w_1 increased. No more corrections are needed.

The above adaptations are made according to the III variant of the error correction procedure.

The possible input patterns and the required signs of increments of the weights are given in Table 3.

If the length of the correction vector is always larger than $\|I_{min} \cdot b^j\| (b^j \neq 0)$, the smallest value ever applied, then the distance to the hyperplane $b^j \cdot w = \varphi_j$ is reduced during the adaptation and the distance to the point w_* is also reduced by some amount which is larger than a fixed value. Thus, the distance between w and w_* will decrease until the actual weight vector falls within the set of weights that gives the proper responses [9].

7.2.2 Mathematical Analysis of the Fixed Increment Adaptation.

The error correction procedure with fixed increments can be restated as follows.

We are given a set of input vectors $a^1 \ldots a^j \ldots a^q$ (corresponding to the input events $\omega_1, \ldots \omega_j, \ldots \omega_q$) in N + 1 dimensional space. This set of q vectors is the training sequence and each of them, multiplied by the desired classification ρ_i becomes a "positive input vector" b^j.

The adaptation rules for the construction of the vector w can be expressed mathematically as:

$$w^n = \begin{cases} w^{n-1} & \text{if } (b^{jn}, w^{n-1}) > \theta \\ w^{n-1} + Ib^{jn} & \text{if } (b^{jn}, w^{n-1}) \leqslant \theta \end{cases} \qquad (8)$$

where: b^{jn} - positive input vector corresponding to this pattern which is shown as n-th pattern in the training sequence.

In the proof of convergence of the training procedure, the following assumptions are made [4].

1. There exists a vector w such that

$$(b^j, w) > \theta > 0 \tag{9}$$

for all j (i.e., there exists a solution to the problem.)

2. During the training period sequence $b^{j1}, b^{j2}, \ldots b^{jk} \ldots$ $(1 \leqslant j_k \leqslant q,\ k = 1, 2, \ldots)$ is such that each vector b^j occurs infinitely many times, if necessary.

3. Correction takes place every time when a pattern from training sequence is shown to the system. This means that only those patterns are considered for which $w^n \neq w^{n-1}$.

Thus, for each pattern ω_j, $w^n = w^{n-1} + Ib^{jn}$ and for each n:

$$(w^{n-1}, b^{jn}) \leqslant \theta \tag{10}$$

The training process is a construction of the sequence of vectors $w^o, w^1, \ldots w^n, \ldots$ according to rules (8).

If the training procedure converges to a solution, then the equations (9) and (10) cannot hold for arbitrary large n i.e., the number of corrections can only range through finite set of integers. Assuming that w_* is the shortest vector satisfying equation (9) we can find the lower bound on the value of (w^n, w_*).

Let the initial weight vector be w^o. After n corrections we will have

$$w^n = w^o + Ib^{j1} + Ib^{j2} + \ldots + Ib^{jn}$$

The scalar product of the vector and the vector w_* is

$$(w^n, w_*) = (w^o, w_*) + I(b^{j1}, w_*) + I(b^{j2}, w_*) + \ldots + I(b^{jn}, w_*)$$

Since the vector w_* is appropriate for the solution, from (9), we have: $I(b^{jk}, w_*) > \theta$

and

$$(w^n, w_*) > (w^o, w_*) + In\theta \tag{11}$$

The upper bound for the length of w, after n adaptations have taken place, can be found in the following way. The change of the length of the weight vector duing any single correction is:

$$\|w^n\|^2 - \|w^{n-1}\|^2 = \|w^{n-1} + Ib^{jn}\|^2 - \|w^{n-1}\|^2 =$$

$$= (w^{n-1}, w^{n-1}) + 2(w^{n-1}, Ib^{jn}) + (Ib^{jn}, Ib^{jn}) - (w^{n-1}, w^{n-1}) =$$

$$= 2I(w^{n-1}, b^{jn}) + I^2(b^{jn}, b^{jn}) < 2\theta I + I^2 \max(b^{jn}, b^{jn})$$

$(b^{jn}, b^{jn}) = \|b^{jn}\|^2$ achieves its maximum value when all inputs are $|1|$ i.e.,

$$\max (b^{jn}, b^{jn}) = N + 1$$

where (N+1) is the total number of inputs including the threshold input.

Thus,

$$\|w^n\|^2 - \|w^{n-1}\|^2 < 2\theta I + I^2 (N+1)$$

Since it takes n corrections to arrive from w^o to w^n

$$\|w^n\|^2 < \|w^o\|^2 + [2\theta I + I^2(N+1)]n = \alpha_1 + \beta_1 n \qquad (12)$$

On the other hand, from (11) and using Cauchy-Schwartz inequality ($(w^n, w_*)^2 \leq \|w^n\|^2 \, \|w_*\|^2$) one gets

$$\|w^n\|^2 \geq \frac{(w^n, w_*)^2}{\|w_*\|^2} \geq \frac{[(w^o, w_*) + In\theta]^2}{\|w_*\|^2} =$$

$$= \frac{(w^o, w_*)^2 + 2I(w^o, w_*) n\theta + I^2 n^2 \theta^2}{\|w_*\|^2} =$$

$$= \alpha_2 + \beta_2 n + \gamma n^2 \qquad (13)$$

It is obvious from Fig. 17, that the inequalities (12) and (13) will not hold for sufficiently large n. Therefore the solution must be achieved in finite number of steps.

The upper bound for the number of correction steps can be calculated from (12) and (13)

$$[(w^o, w_*)+In\theta]^2 \leq \|w^n\|^2 \|w_*\|^2 < \{\|w^o\|^2+[2\theta I+I^2(N+1)]n\}\|w_*\|^2$$

For the sake of simplicity it can be assumed that $w^o = 0$ and from the above formula we get

$$I^2n^2\theta^2 < [2\theta I+I^2(N+1)]\, n \|w_*\|^2$$

Finally

$$n < \frac{2\theta I+I^2(N+1)}{I^2\theta^2} \|w_*\|^2 = \left(\frac{2}{\theta I} + \frac{N+1}{\theta^2}\right)\|w_*\|^2$$

7.2.3. Example.

As an example of application of the training procedure we will consider the following problem.

Implement the function $f(a_1, a_2, a_3) = \bar{a}_1 \bar{a}_2 a_3 + a_1 \bar{a}_2 a_3 + a_1 a_2 a_3$ with core circuit. Input signals are pulses of current. Presence of the pulse in a winding will be symbolized by 1, absence of the pulse is equivalent to 0 signal at the input.

Symbolic diagram for a magnetic core gate with three inputs is shown in Fig. 18.

Inputs a_o, a_1, a_2, a_3 are excited by synchronized current pulses I_i. If a term $x_1 x_2 x_3$ (where $x_i = a_i$ or $\bar{a}_i$) appears in the function expressed in disjunctive normal form, then the core should be set when the combination $x_1 x_2 x_3$ is fed to the input terminals i.e.,

$$1.N_o.I_o + x_1.N_1.I. + x_2.N_2.I_2 + x_3.N_3.I_3 > F_h$$

where F_h is the magnetomotive force (in amper-turns) corresponding to the coercive force of the magnetic material. For all input combinations which are not present in the disjunctive normal for the function

$$\sum_{i=o}^{3} x_i . N_i . I_i < F_h$$

If $I_o = I_1 = I_2 = I_3 = I$ then

$$\sum_{i=0}^{3} x_i . N_i > \frac{F_h}{I} = N_h \qquad \text{for the "true" terms}$$

and

$$\sum_{i=o}^{3} x_i . N_i < \frac{F_h}{I} = N_h \qquad \text{for the "false" terms}$$

During the advance phase of operation the pulse current is sent to the winding N_A. The magnetomotive force $N_A I_A$ works in the direction shown in Fig. 19 and is large enough to change the state of the core from Set to Reset.

Thus, during the Advance phase, at time t+1, an output pulse appears iff the input combination at time t represented a true term of the function realized by the core gate.

It is obvious that, if

$$N_o - N_h = w_o, \quad N_1 = w_1, \quad N_2 = w_2, \quad N_2 = w_3$$

then the core gate can be identified with a LTLU.

The set of inequalities corresponding to function $f(a_1 a_2 a_3) = \bar{a}_1 \bar{a}_2 a_3 + a_1 \bar{a}_2 a_3 + a_1 a_2 a_3$ is the following.

$$1.N_o + 0.N_1 + 0.N_2 + 0.N_3 < N_h$$
$$1.N_o + 0.N_1 + 0.N_2 + 1.N_3 > N_h$$
$$1.N_o + 0.N_1 + 1.N_2 + 0.N_3 < N_h$$
$$1.N_o + 1.N_1 + 0.N_2 + 0.N_3 < N_h$$
$$1.N_o + 0.N_1 + 1.N_2 + 1.N_3 < N_h$$
$$1.N_o + 1.N_1 + 0.N_2 + 1.N_3 > N_h$$
$$1.N_o + 1.N_1 + 1.N_2 + 0.N_3 < N_h$$
$$1.N_o + 1.N_1 + 1.N_2 + 1.N_3 > N_h$$

This system of inequalities can be replaced by

$$\begin{pmatrix} -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ -1 & 0 & -1 & 0 \\ -1 & -1 & 0 & 0 \\ -1 & 0 & -1 & -1 \\ 1 & 1 & 0 & 1 \\ -1 & -1 & -1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} w_o \\ w_1 \\ w_2 \\ w_3 \end{pmatrix} \qquad \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

where $w_o = N_o - N_h$, $w_1 = N_1$ $w_2 = N_2$ $w_2 = N_3$

Now, following the rules of the Fixed Increment Adaptation we shall find the vector w satisfying this et of inequalities.

Let us take, as an initial weight vector $w = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$. As a

first positive input vector b^1 the first row of the matrix is taken. Since $b^1 . w^o < 0$ the weight vector must be changed. $w^1 = w^o + b^{1t} = w^o +$

$$+\begin{pmatrix}-1\\0\\0\\0\end{pmatrix} = \begin{pmatrix}-1\\0\\0\\0\end{pmatrix}$$ (b^{jt} - transpose of the row vector b^j) .

Similarly $b^2 . w^1 < 0$, hence $w^2 = w^1 + b^{2t} = w^1 + \begin{pmatrix}1\\0\\0\\1\end{pmatrix} = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}$.

Following the same procedure one can find

$$w^3 = w^2 + b^{3t} = w^2 + \begin{pmatrix}-1\\0\\-1\\0\end{pmatrix} = \begin{pmatrix}-1\\0\\-1\\1\end{pmatrix}, \quad w^4 = w^3, \quad w^5 = w^4$$

$$w^6 = w^5 + b^{6t} = w^5 + \begin{pmatrix}1\\1\\0\\1\end{pmatrix} = \begin{pmatrix}0\\1\\-1\\2\end{pmatrix}, \quad w^7 = w^6 + b^{7t} = w^6 + \begin{pmatrix}-1\\-1\\-1\\0\end{pmatrix} = \begin{pmatrix}-1\\0\\-2\\2\end{pmatrix}$$

$$w^8 = w^7 + \begin{pmatrix}1\\1\\1\\1\end{pmatrix} = \begin{pmatrix}0\\1\\-1\\3\end{pmatrix}, \quad w^9 = w^8 + \begin{pmatrix}-1\\0\\0\\0\end{pmatrix} = \begin{pmatrix}-1\\1\\-1\\3\end{pmatrix}, \quad w^{10} = w^9, \quad w^{11} = w^{10}$$

$$w^{12} = w^{11} + \begin{pmatrix}-1\\-1\\0\\0\end{pmatrix} = \begin{pmatrix}-2\\0\\-1\\3\end{pmatrix}, \quad w^{13} = w^{12} + \begin{pmatrix}-1\\0\\-1\\-1\end{pmatrix} = \begin{pmatrix}-3\\0\\-2\\2\end{pmatrix}, \quad w^{14} = w^{13} + \begin{pmatrix}1\\1\\0\\1\end{pmatrix} = \begin{pmatrix}-2\\1\\-2\\3\end{pmatrix}$$

$$w^{15} = w^{14}, \quad w^{16} = w^{15} + \begin{pmatrix}1\\1\\1\\1\end{pmatrix} = \begin{pmatrix}-1\\2\\-1\\4\end{pmatrix}, \quad w^{17} = w^{16}, \quad w^{18} = w^{17}, \quad w^{19} = w^{18}$$

$$w^{20} = w^{19} + \begin{pmatrix}-1\\-1\\0\\0\end{pmatrix} = \begin{pmatrix}-2\\1\\-1\\4\end{pmatrix}, \quad w^{21} = w^{20} + \begin{pmatrix}-1\\0\\-1\\-1\end{pmatrix} = \begin{pmatrix}-3\\1\\-2\\3\end{pmatrix}, \quad w^{22} = w^{21}$$

$$w^{23} = w^{22}, \quad w^{24} = w^{23} + \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -2 \\ 2 \\ -1 \\ 4 \end{pmatrix}, \quad w^{25} = w^{24}, \quad w^{26} = w^{25},$$

$$w^{27} = w^{26}, \quad w^{28} = w^{27} + \begin{pmatrix} -1 \\ -1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -3 \\ 1 \\ -1 \\ 4 \end{pmatrix}, \quad w^{29} = w^{28} + \begin{pmatrix} -1 \\ 0 \\ -1 \\ -1 \end{pmatrix} = \begin{pmatrix} -4 \\ 1 \\ -2 \\ 3 \end{pmatrix}$$

$$w^{30} = w^{29} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -3 \\ 2 \\ -2 \\ 4 \end{pmatrix} .$$

Since w^{30} satisfies all the inequalities the problem has been solved. Finaly we have

$$N_0 = w_o + N_h = -3 + N_h, \quad N_1 = w_1 = 2, \quad N_2 = w_2 = -2, \quad N_3 = w_3 = 4$$

The weight vector w^{30} can be expressed in term of positive input vectors in the following manner.

$$w^{30} = (2b^1 + b^2 + b^3 + 3b^4 + 3b^5 + 3b^6 + b^7 + 3b^8)\, t$$

A negative number for N_i means that the direction of the winding is opposite to the direction shown in Fig. 18.

7.3. The Relaxation Adaptation Procedures

In this procedure the corrections are made proportional to the amount by which analog output of the LTLU is smaller than the dead zone θ.

Mathematically:

$$w^n = w^{n-1} + \mu \left[(\theta - (w^{n-1}, b^j)) + |\theta - (w^{n-1}, b^j)| \right] b^j$$

Whenever $(w^{n-1}, b^j) > \theta$, $w^n = w^{n-1}$ according to this procedure.

Since for very small value of $[\theta - (w^{n-1}, b^j)]$ the weight vector is to be changed by an infinitesimal amount, this procedure may require infinite number of steps. It has been shown [16] that the proportionality constant $\mu=2$ is the only value which will guarantee convergence in a finite number of steps. For $\mu > 2$ the procedure may diverge.

In the Modified Relaxation Adaptation Procedure the lower bound for the size of correction is provided and this procedure will guarantee the convergence in a finite number of steps.

The mathematical formulation of this modified procedure is the following:

$$w^n = \begin{cases} w^{n-1} & \text{for } (w^{n-1}, b^j) \geq \theta \\ w^{n-1} + \mu_1 b^j [\mu_2 - (w^{n-1}, b^j)] & \text{for } (w^{n-1}, b^j) < \theta \end{cases}$$

where $0 < \mu_1 \leq 2 \qquad 0 < \theta < \mu_2$

According to the above rules, the correction vector is proportional to the difference between the quantity μ_2 which is larger than the dead zone θ and the analog output of the unit. The smallest value which can be added to the weight vector is $\mu_1 b^j [\mu_2 - \theta]$. The corrections are being made as if equations $(w, b^j) \geq \mu_2$ were being solved but the decision to make the correction depends on $(w, b^j) > \theta$ [16].

7.4. Correction with Decay Terms

In practice, an upper bound exists on weight sizes and there is no guarantee that a given linearly separable function will be realized by the element if the weight limits are reached.

If a change is needed when one of the components of w, say w_i, has reached its bound, then farther increase is impossible and $\Delta w_i = 0$.

There is, however, a possibility to circumvent this difficulty by combination of error correction with a motion of the vector w toward the origin by a value proportional to the distance away from it [2], [17].

Such a change can be expressed as

$$\Delta w_i = \rho_j a_i I - k w_i \quad \text{if} \quad (b^j, w) < \theta$$

where $0 < k < 1$

Now, the range of changes of w_i is limited. After n adaptations have taken place and supposing that at each step an increase of weight takes place

$$w_i = n\Delta w_i = n\rho_j a_i I - n k w_i$$

or

$$w_i (1 + n k) = n \rho_j a_i I$$

$$\text{if} \quad n \to \infty \quad \text{then} \quad \lim_{n\to\infty} w_i = \lim_{n\to\infty} \frac{n\rho_j a_i I}{1 + nk} = \pm \frac{I}{k}$$

i.e., none of the weights can reach a value larger than $\frac{I}{k}$ during the adaptation.

8. Results of Adaptation of the LTLU on Nonlinearly Separable Function.

When the LTLU is trained to realize some nonlinearly separable function then there is no weight vector such that $(w, b^j) > 0$ for all b^j.

During the training process the actual weight vector will change its position in the weight space moving from one cone, corresponding to a linearly separable function, to another. A linearly separable function realized by the LTLU when the adaptation is stopped depends on patterns which were shown most recently. If adaptation starts again, the weight vector will be changed [9] This vector will be lenghtened if it lies within the dead zone and will be shortened if it lies outside the dead zone while the adaptation takes place.

Any vector which is shorter than $\frac{\theta}{\sqrt{N+1}}$ will always be within a dead zone and therefore an adaptation will always increase its length. On the other hand, if the length of the vector is large, the chances are small that it will lie inside the dead zone. As a result, an adaptation will decrease the length.

Thus, the length of the weight vector will oscillate around a fixed value if a LTLU is adapted on nonlinearly separable function. The value of this mean length r_o around which the vector oscillates is the length at which the vector has 50 percent chance of being increased and 50 percent chance of being decreased.

If one assumes that a weight vector w can fall uniformly on a surface of a hypersphere of radius $\|w\|$ then r_o corresponds to a value $\|w\|$ for which the ratio of the surface within the dead zone to the total surface of the hypersphere is $1/2$.

Assuming that the dead zone is equal to 1, the following values for r_o can be obtained[9].

Inputs (N+1)	r_o
2	0.0
3	1.156
4	1.274
5	1.342
6	1.384
7	1.418
8	1.442
9	1.460
10	1.476
.	.
.	.
.	.
17	1.524
.	.
.	.
.	.
∞	1.56

The assumption of uniform distribution of weight vectors is only partially valid because after adaptation a vector will be on a hyperplane bounding the dead zone. The vectors will not be found uniformly around the sphere but only at points which are on the hyperplanes distance r_o from the origin. These points are symmetrically spaced around the sphere but they are not uniformly distributed.

If there is more than one LTLU being adapted at the same time, all weight vectors will tend to become of the same length. The rate of growing will also be approximately the same when all of them are equal at the beginning of the training process. This fact has a significance for analysis of learning in nets consisting of many LTLU's.

9. Summary and Conclusions.

In this work a unified linear algebraic appoach to the analysis of the properties of the trainable LTLU has been applied.

Known in mathematics idea of duality has been explicitely expressed and applied to different examples of LTLU's. This is the main contribution of this work to the subject of learning machines.

Many results given here were worked out by different researchers independently and one purpose of this work was a systematization of the knowledge on the properties of the trainable LTLU.

Since the capability of learning is the most important property of the LTLU, the training procedures and proofs of their convergence has been surveyed.

Although the examples given in this work are restricted to the cases where input and output signals are binary, this restriction is not imposed by the applied method.

Farther analysis of the LTLU's, working on many-valued or continuous input signals, is needed.

Since the classification capabilities of the single LTLU are limited to those cases where classes of input signals are linearly separable, the systems of interconnected LTLU's need studies.

10. References

1. H.J. Greenberg; A.G. Konheim, Linear and Nonlinear Methods in Pattern Classification. IBM Journal of Research and Development vol. 8, No. 3, July 1964 p. 299.

2. H.D. Block; The Perceptron: A Model for Brain Functioning I. Review of Modern Physics, vol. 34, No 1, January 1962 p. 123.

3. H.G. Eggleston; Convexity. Cambridge University Press 1958.

4. A. Novikoff; On Convergence Proofs for Perceptrons, Proc. of the Symposium on Mathematical Theory of Automata. New York - April 1962, Vol. XII. Polytechnic Press of the Polytechnic Institute of Brooklyn, Brooklyn, N.Y.

5. Scott M. Cameron; An Estimate of the Complexity Requisite in a Universal Decision Network; Bionics Symposium; Living Prototypes - the Key to New Technology. WADD Technical Report 60 - 600 December, 1960.

6. Robert O. Winder; Threshold Logic (A doctoral dissertation) Princeton University, Princeton, N.J., March 1962.

7. Robert O. Winder; Threshold Logic in Artificial Intelligence "Artificial Intelligence", IEEE. Jan 1963, S-192.

8. J.G. Kemeny, H. Mirkil, J.L. Snell, G.L. Thompson, Finite Mathematical Structures. Prentice-Hall, Inc. 1960.

9. W.C. Ridgway III, An Adaptive Logic System with Generalizing Properties. Stanford University Electronics Laboratory Technical Report No. 1556-1., 1962.

10. G.S. Sebestyen; Decision-Making Processes in Pattern Recognition The MacMillan Co., New York 1962.

11. M. D. Mesarovic; On Self Organizational Systems,
[Self-Organizing Systems 1962, Spartan Books, Washington 1962].

12. R. L. Mattson; The Design and Analysis of an Adaptive Systems for Statistical Classification. M. Sc. Thesis, M. I. T., Course VI, June, 1959.

13 B. Widrow and M. E. Hoff, Jr; Adaptive Switching Circuits, TR No 1553-1, Stanford Electronics Lab., Stanford, California, June 1960.

14. F. Rosenblatt; The Perceptron- A Theory of Statistical Separability in Cognitive Systems, Rept. VG-1196-G-1; Cornel Aeronautical Labs., Buffalo, N. Y., Jan. 1958.

15. C. H. Mays; The Relation of Algorithms Used with Adjustable Threshold Elements to Differential Equations.
IEEE. Transactions on EC, February 1965 No. 1 p 62.

16. Research in Systems Theory, Devices, and Physical Phenomena for Microsystem Electronics. Final Technical Documentary Report No AL-TDR-64-81, June 1964. Stanford Electronics Labs., Stanford, California.

17. J. K. Hawkins, C. J. Munsey; A Magnetic Integrator for the Perceptron Program. Annual Summary Report, AERONUTRONIC a Division of Ford Motor Co., 30 July 1960.

18. C. K. Chow; Statistical Independence and Threshold Functions, IEEE. Transactions on Electronic Computers February 1965 vol EC 14 No 1.

19. Keinosuke Fukunaga and Takayasu Ito ; A Design Theory of Recognition Functions in Self-Organizing Systems.
Mitsubishi Denki., Laboratory Reports Vol 5, Jan. 1964 No 1.

VITA

Name: Ignacy J. ZAWICKI

Born: 14 June 1936, POLAND

Educated:

Secondary: Technical High School (Electrical Engg), Warsaw, Poland.

University: Warsaw Technical University(Politechnika Warszawska), Warsaw, Poland.

Degree: Master of Science in Electrical Engineering.

Course: Electrical Measurements.

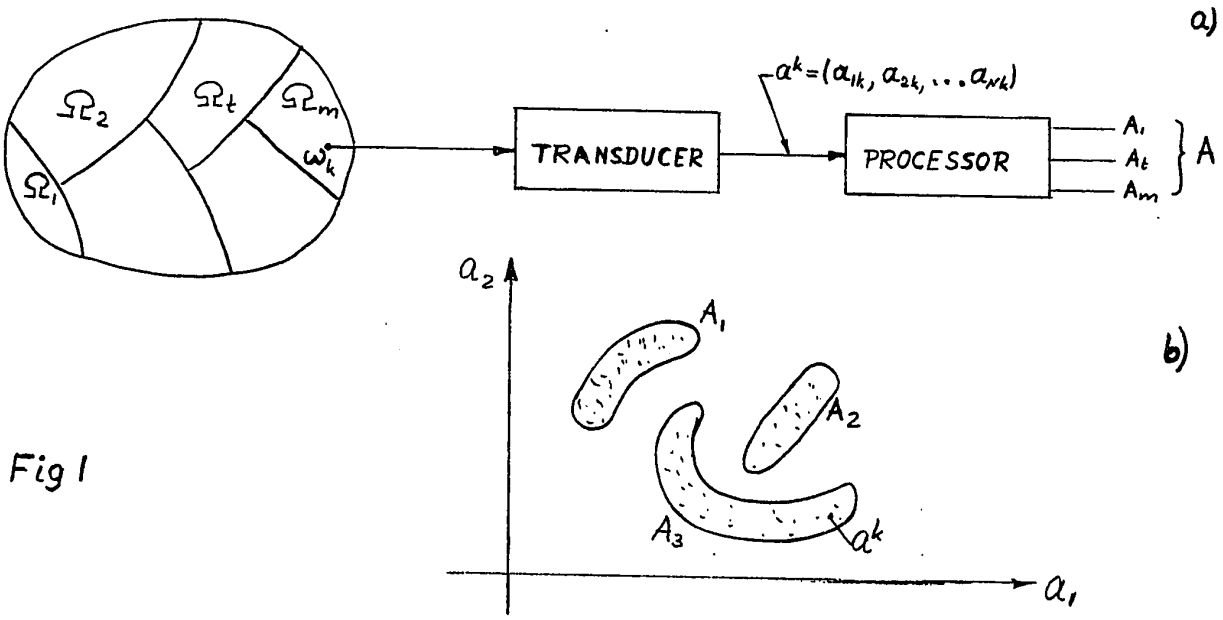

a)
Ω_1
Ω_2
Ω_t
Ω_m
ω_k
TRANSDUCER
$a^k=(a_{1k}, a_{2k}, \ldots a_{Nk})$
PROCESSOR
A_1
A_t
A_m
A
b)
a_2
A_1
A_2
A_3
a^k
a_1

Fig 1

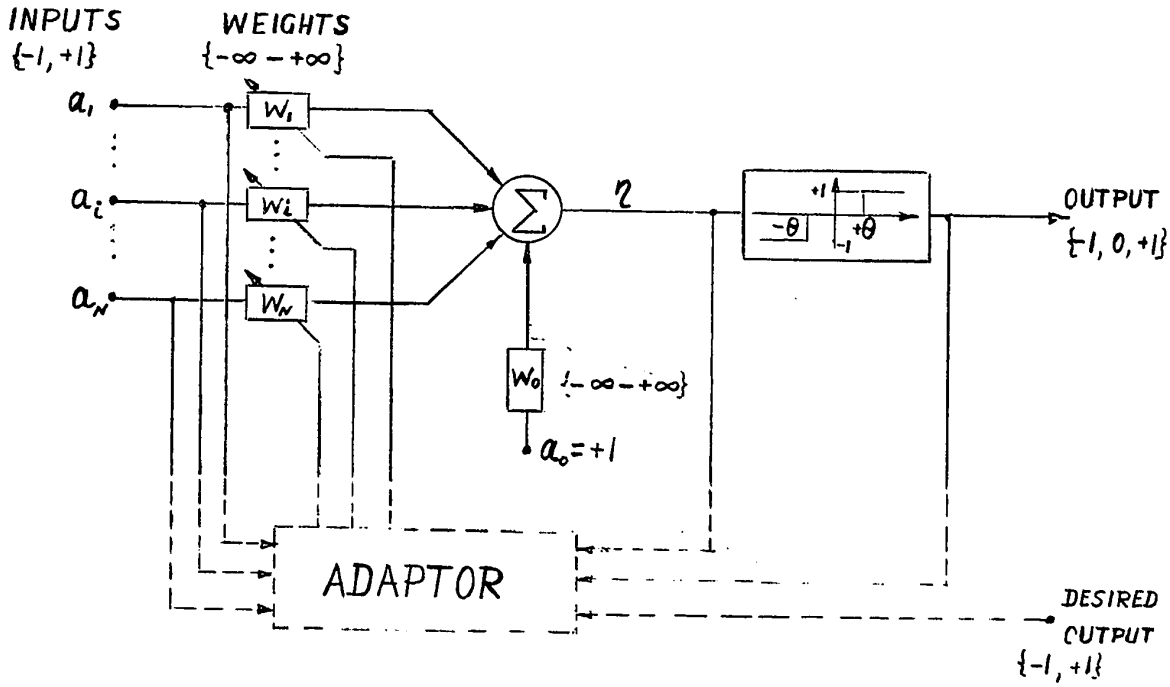

INPUTS
{-1, +1}
WEIGHTS
{-∞ - +∞}
a_1
a_i
a_N
W_1
W_i
W_N
W_0 {-∞ - +∞}
$a_0=+1$
η
+1
-θ
+θ
-1
OUTPUT
{-1, 0, +1}
ADAPTOR
DESIRED
OUTPUT
{-1, +1}

Fig 2

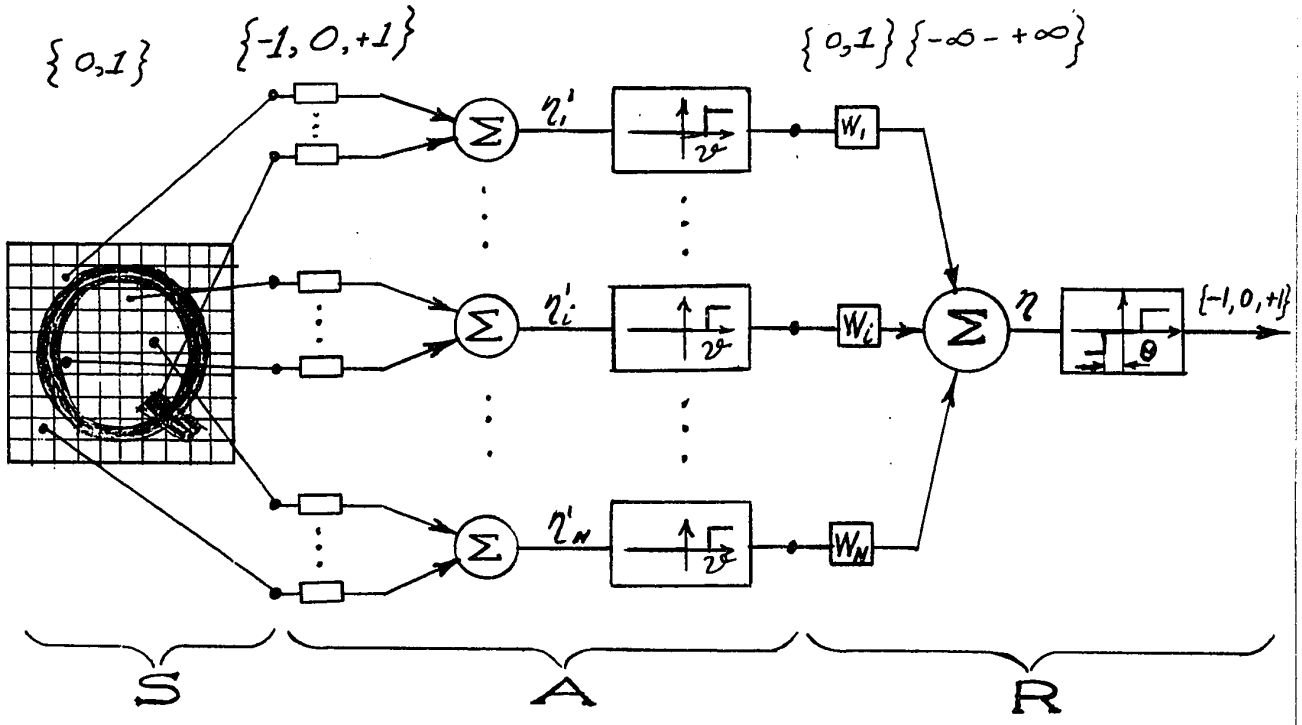

{0,1}
{-1, 0, +1}
{0,1}
{-∞ - +∞}
Σ
η'1
ϑ
W1
η'i
Wi
η
θ
{-1,0,+1}
η'N
WN
S
A
R

FIG. 3

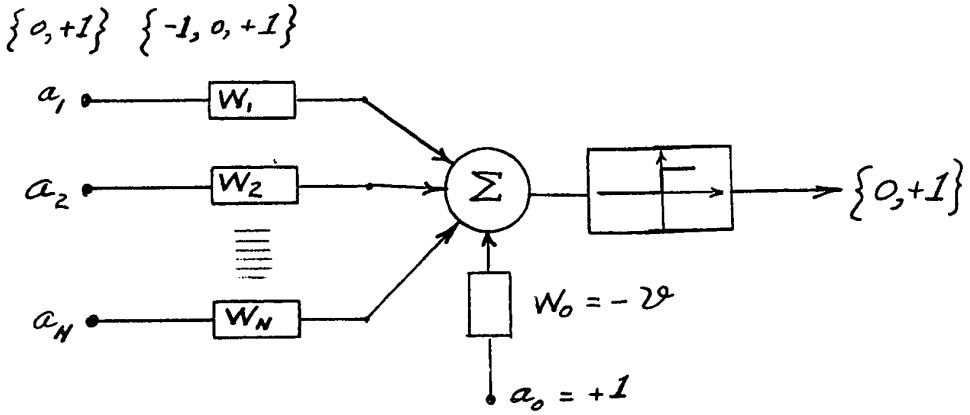
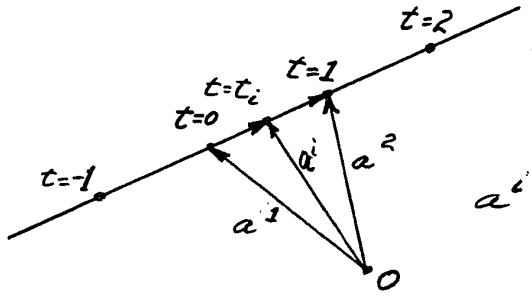
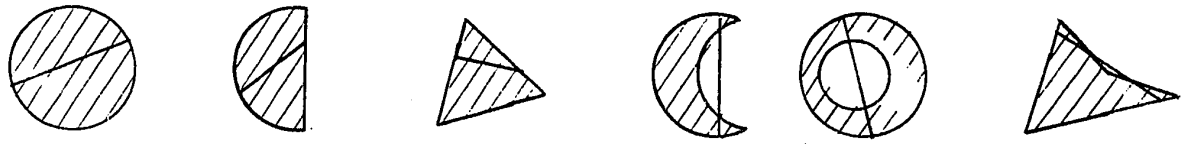

{0,+1}
{-1, 0, +1}
a1
W1
a2
W2
Σ
{0,+1}
aN
WN
W0 = - ϑ
a0 = +1

FIG. 4



$$a^i = t_i (a^2 - a^1) + a^1 = t_i a^2 + (1 - t_i) a^1$$

FIG. 5



a) CONVEX SETS

b) NONCONVEX SETS

FIG. 6

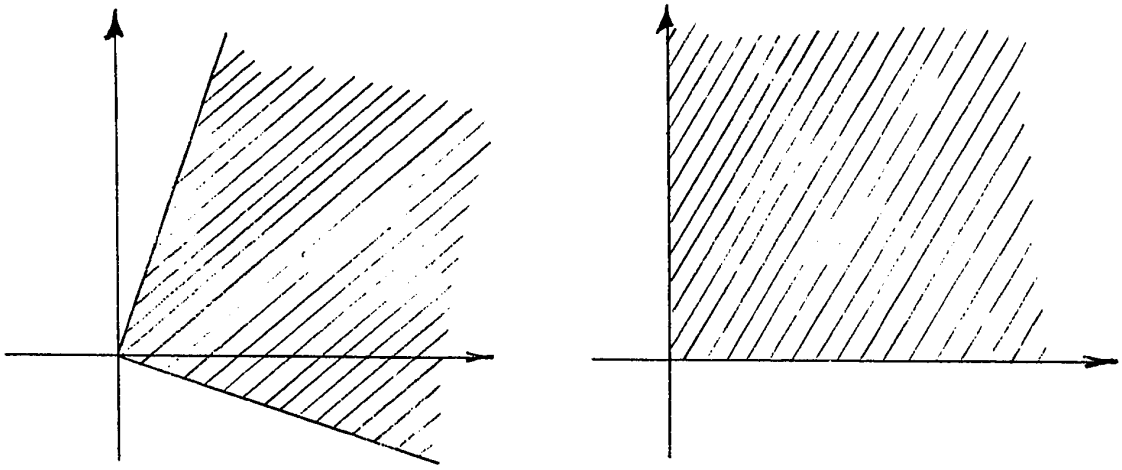


FIG. 7

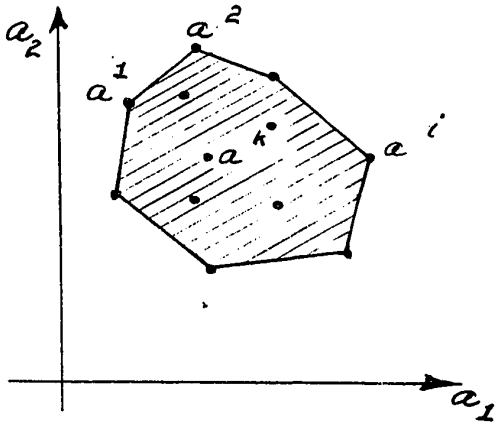


FIG. 8

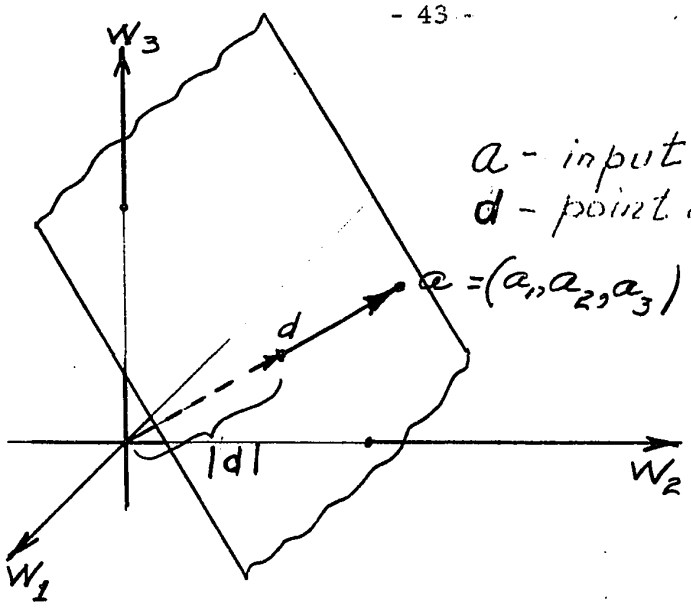
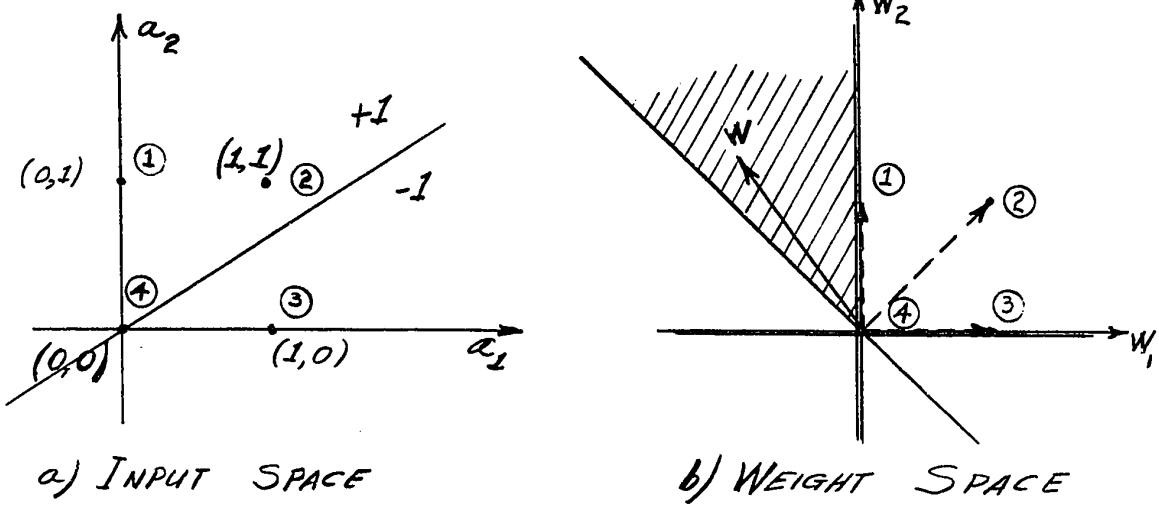


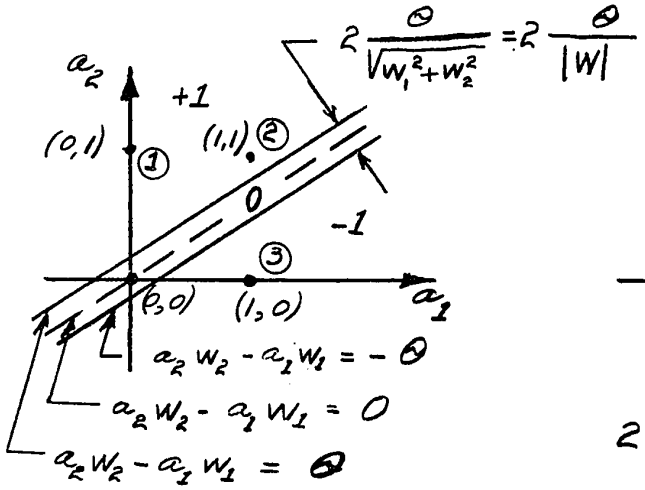
FIG. 9



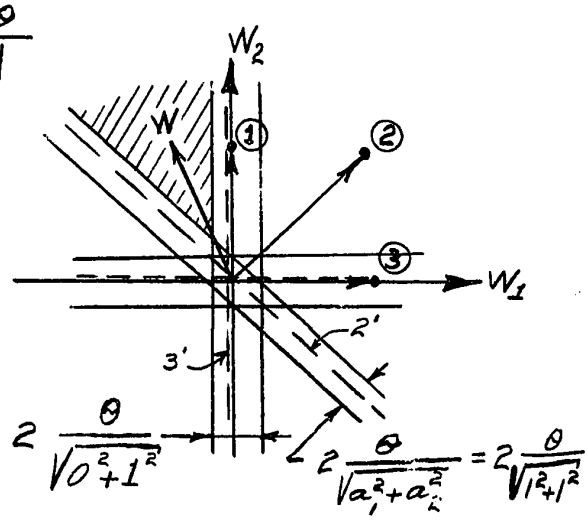
a) INPUT SPACE

b) WEIGHT SPACE

FIG. 10

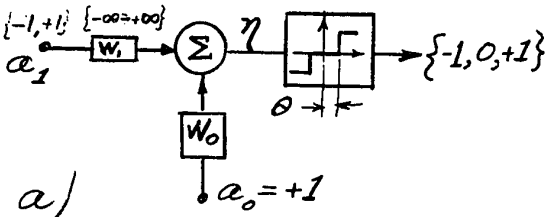


a)



b)

FIG. 11



a)

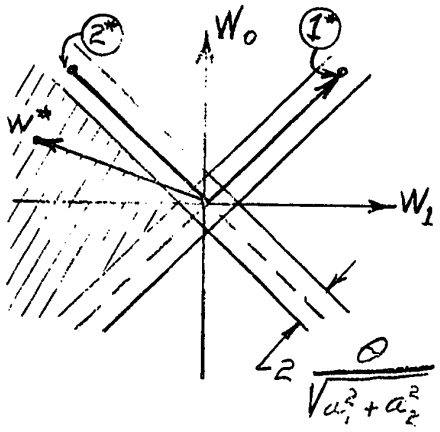


Fig 12

b)

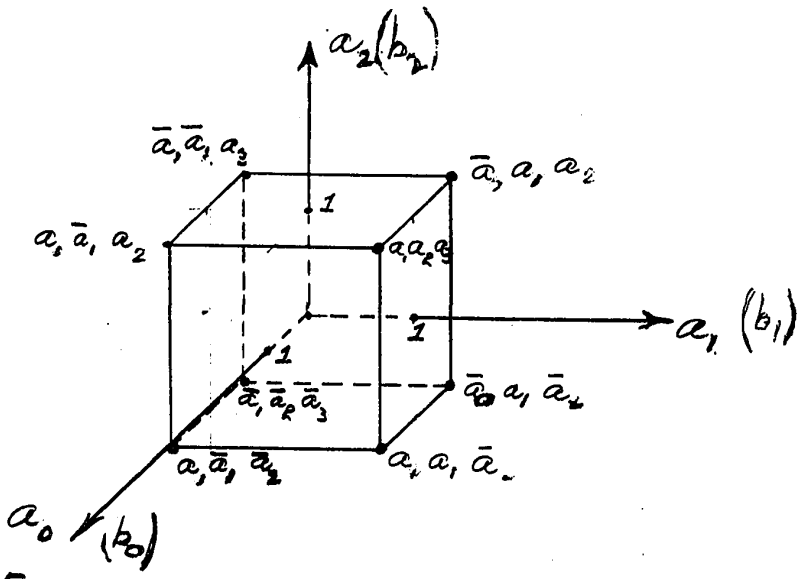


FIG. 15

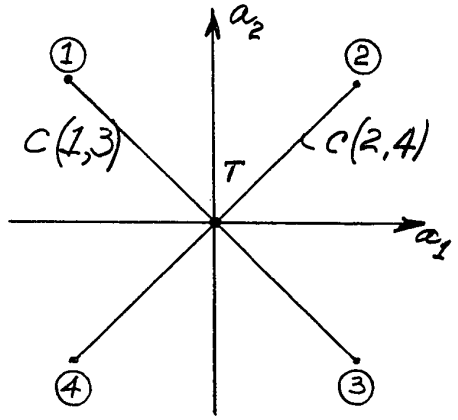
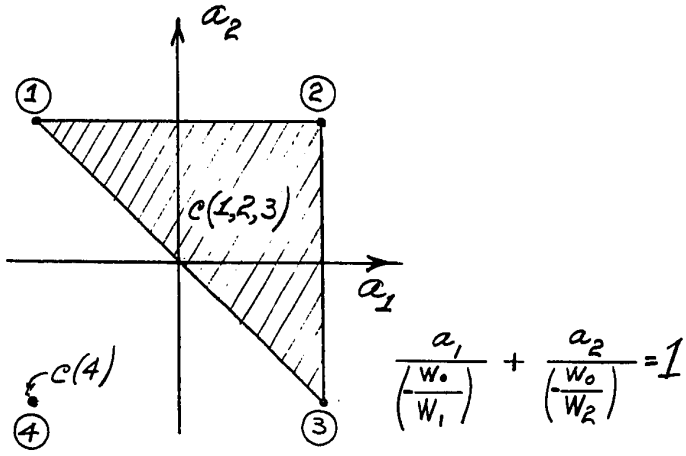


FIG. 13 a)



b)

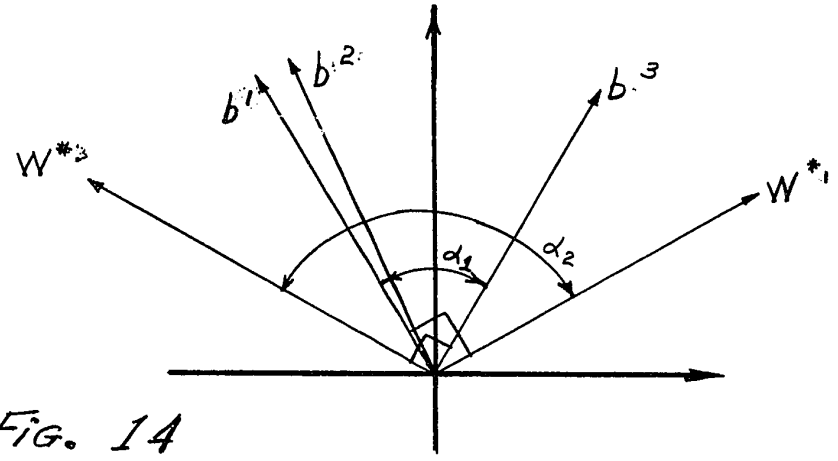


FIG. 14

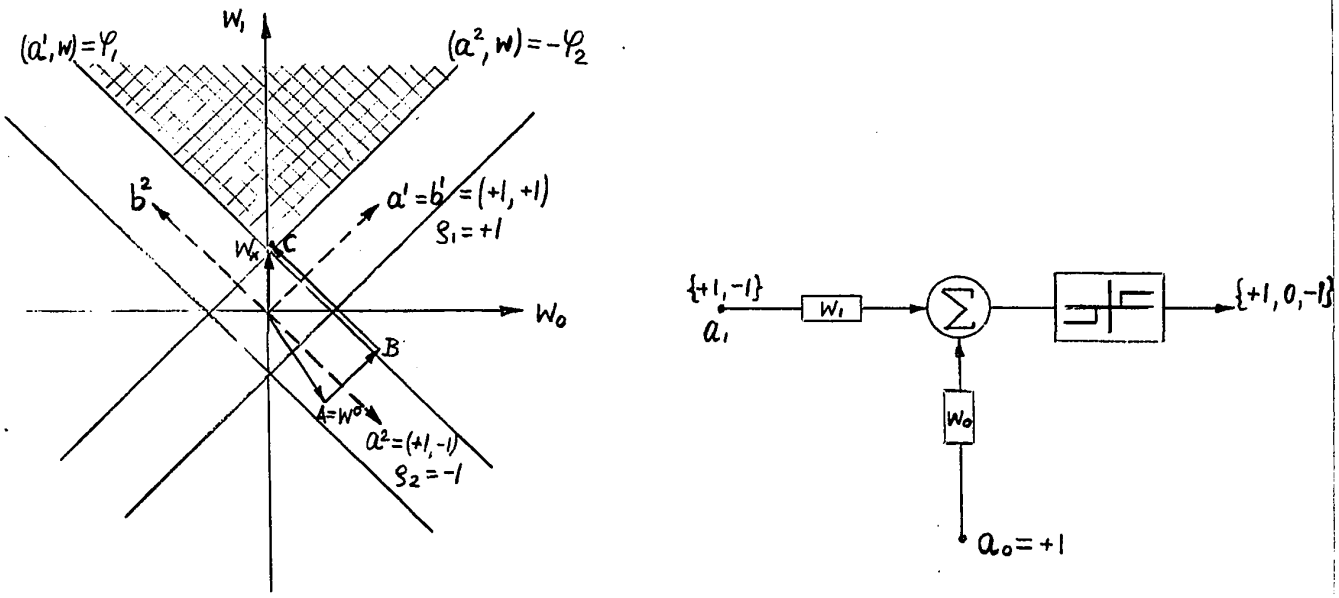


Fig 16

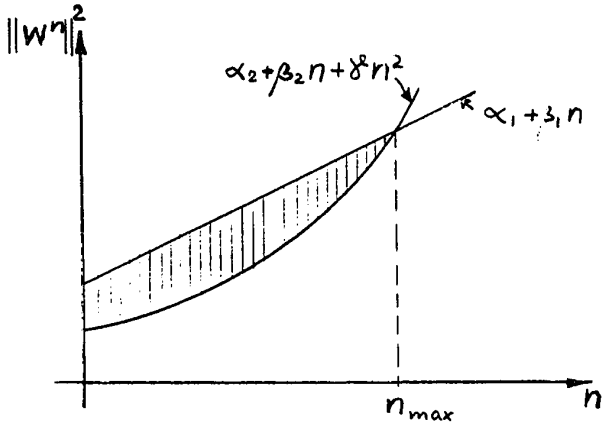


Fig 17

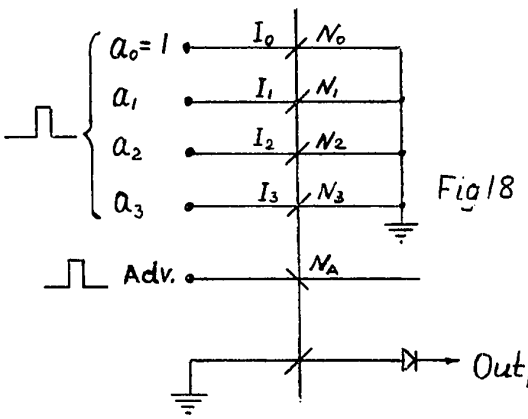


Fig 18

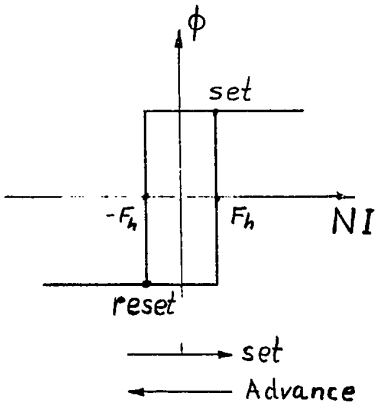


Fig 19