

Nils J. Nilsson

RADC-TDR-64-33

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



AN EXPERIMENTAL COMPARISON OF THREE
LEARNING MACHINE TRAINING RULES

TECHNICAL DOCUMENTARY REPORT NO. RADC-TDR-64-33

February 1964

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

Project No. 5581, Task No. 558104

(Prepared under Contract AF30(602)-2943 by Nils J. Nilsson
Applied Physics Laboratory and Richard C. Singleton, Mathe-
matical Sciences Dept, Stanford Research Institute, Menlo Park,
Calif.)

DDC AVAILABILITY NOTICE

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

DDC release to OTS is authorized.

LEGAL NOTICE

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

DISPOSITION NOTICE

Do not return this copy. Retain or destroy.

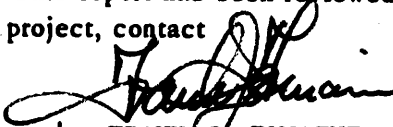
Key Words: Artificial Intelligence, Learning Machines, Adaptive Mechanisms, Pattern Recognition, Perceptron

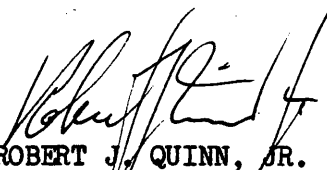
ABSTRACT

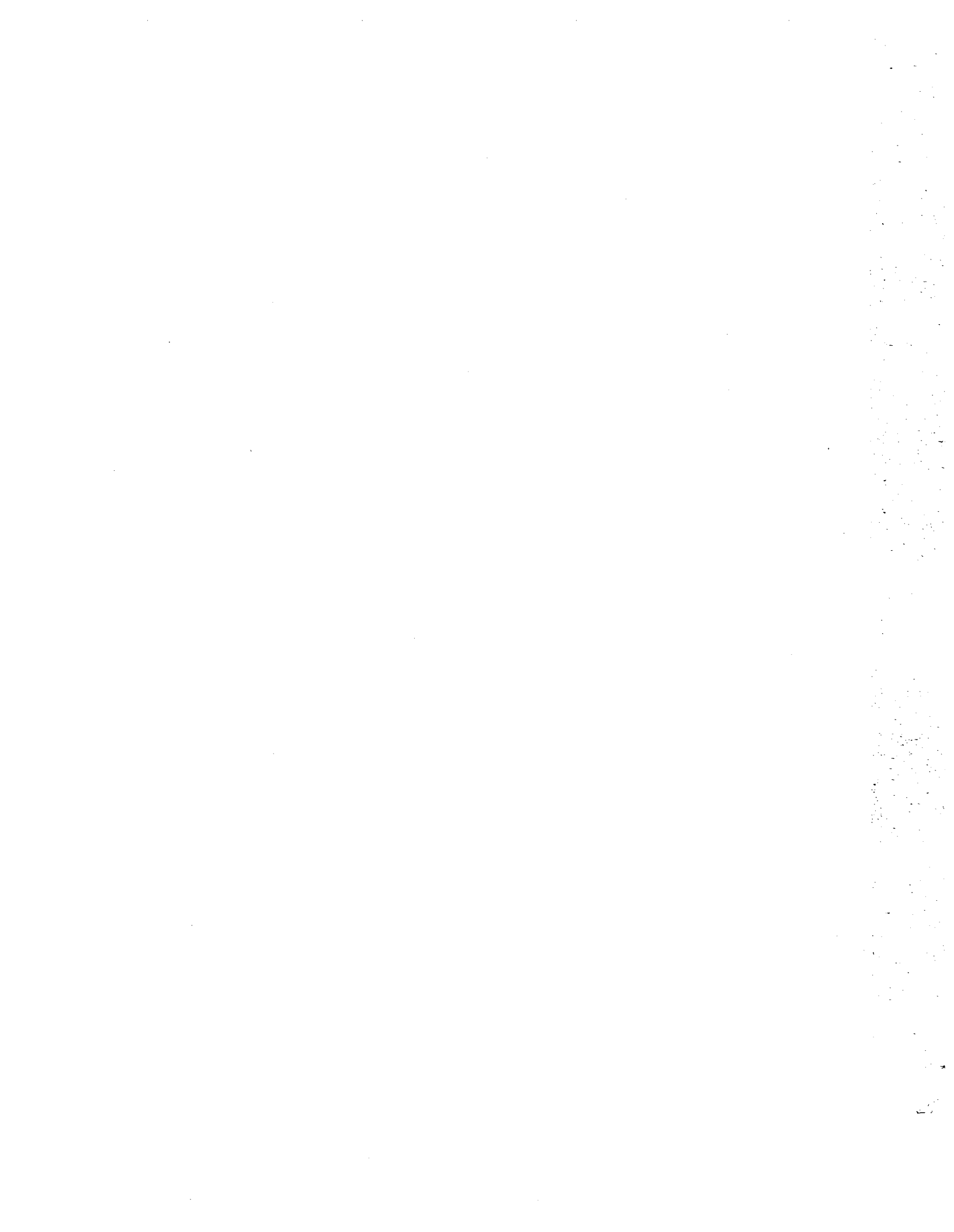
This report includes the results of a series of experiments to compare the efficiency of training methods using (1,0) and (+1,-1) representatives for patterns. It also presents a theoretical explanation which deals with a single TLV rather than with a network of TLV's as used in the experiments. The effects noted, however, can be expected to pertain to the network also.

PUBLICATION REVIEW

This report has been reviewed and is approved. For further technical information on this project, contact


Approved: FRANK J. TOMAINI
Chief, Info Processing Branch


Approved: ROBERT J. QUINN, JR.
Col, USAF
Chief, Intel and Info Processing Div



AN EXPERIMENTAL COMPARISON
OF THREE LEARNING MACHINE TRAINING RULES

by

Nils J. Nilsson

and

Richard C. Singleton

A. BACKGROUND

In the design of learning machine systems such as MINOS II,* it is necessary to specify whether the binary inputs to the learning machine shall be represented by plus ones and minus ones (+1, -1) or by ones and zeros (1,0). The standard error-correction training rule affects the input-output behavior of the learning machine in a significantly different way, depending on whether the rule is applied to a (+1, -1) input machine or to a (1,0) input machine. This difference is due to the fact that for a (1,0) machine, only active weights (weights connected to a one input) are adapted, whereas for a (+1, -1) machine, all of the weights are adapted. A limited amount of past experience has indicated that the (+1, -1) machines converge faster than do (1,0) machines. It was decided to conduct a series of computer-simulation experiments to determine whether or not the training rate of (+1, -1) machines was appreciably faster. In this report we shall describe these experiments and present a partial theoretical explanation of the results.

* MINOS II is the name of a large-scale, self-contained learning machine system built at the Stanford Research Institute under sponsorship of the U.S. Army Electronics Research and Development Laboratory (Contract DA 36-039 SC-78343). The experimental results contained in this report were made available to USAERDL in Quarterly Progress Report No. 11 because of their obvious importance to the design of MINOS II.

B. DESCRIPTION OF THE EXPERIMENTS

To test the performance of the two methods, a one-bit output majority-rule learning machine was simulated on the IBM 7090 computer. This simulated learning machine was a scaled-down version of one of the six parallel units of MINOS II. A schematic diagram of the simulated machine is shown in Fig. 1. Twenty binary inputs are operated on by five threshold logic units (TLUs), which produce a one-bit output according to majority-rule logic. The five threshold logic units are connected to

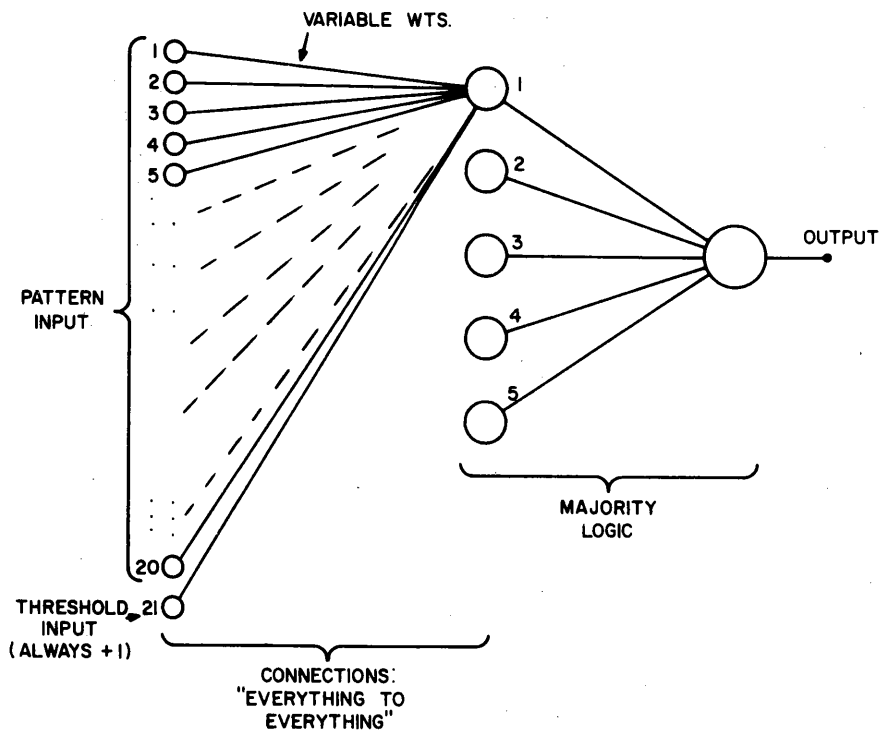


FIG. 1 SYSTEM ORGANIZATION FOR COMPARING (1, 0) AND (+1, -1) LEARNING MACHINES

the 20 inputs by an everything-to-everything scheme employing 100 adjustable weights. Each TLU also has an adjustable threshold simulated by adjustable weights connected to a 21st input, which always has the value (+ 1). The total number of adjustable weights is therefore equal

to 105. The initial values of all weights and thresholds, before training, were in all cases equal to zero. [These initial conditions represent equivalent starting positions for both (1, 0) input machines and (+ 1, - 1) input machines.]

Learning curves were obtained for random sets of randomly categorized patterns, first represented by the (1, 0) scheme and then by the (+ 1, - 1) scheme. Six different random pattern sets of 90 patterns each were used. These sets were divided into three groups. In Group I, (Pattern Sets 1 and 2), each pattern had exactly five ones; in Group II (Pattern Sets 3 and 4), each pattern had exactly ten ones; in Group III (Pattern Sets 5 and 6), each pattern had exactly fifteen ones. Each group had two sets of different random patterns, and two learning curves were obtained for each set. One learning curve is the result of training a machine whose input patterns were presented as ones and zeros; the other curve is the result of training a machine on the same patterns presented as ones and minus ones. In addition, a modified training rule was used for the patterns in Group III. This modified rule attempted to train a (1, 0) input machine in such a way that its learning performance approximated more closely that of a (+ 1, - 1) input machine operating under the ordinary training rule. Therefore, for Pattern Sets 5 and 6, a total of three learning curves were obtained.

A total of six pattern sets were used for the following reasons:

- (1) Three groups were chosen to see if the difference in training rates between (1, 0) and (+ 1, - 1) machines depended at all on the number of ones in the pattern.
- (2) Two different sets were included in each group to give an indication of the differences in learning curves for different pattern sets with the same number of ones.

A total of 90 patterns were used in accordance with a (local) rule-of-thumb that the number of random patterns that a machine can learn in a number of iterations appropriate for a practical application is roughly equal to the number of adjustable weights per output bit. The simulated machine had 105 adjustable weights.

C. TRAINING RULES

The three training rules tested all had the following characteristics:

When the learning machine output is in error, a determination is made of how many TLUs must have their responses reversed so that the majority will vote correctly. Let the minimum number of such reversals necessary be equal to k . Of those TLUs voting incorrectly, one selects the k whose analog sums are closest to threshold and prepares to reverse their responses.

Suppose the response of the i th TLU is to be reversed: Then, its weights must be adapted. The three training rules differ in the way in which this reversal is accomplished:

- (1) (1, 0) Training Rule [for (1, 0) input machines]--An increment is added to each active weight (a weight connected to a one input). The size and direction of each increment are the same for each active weight and are determined by the total change needed in the analog sum to effect a reversal of the TLU binary output.
- (2) (+ 1, - 1) Training Rule [for (+ 1, - 1) input machines]--An increment is added to all weights. Those weights connected to plus one inputs are altered in a direction opposite to that of weights connected to minus one inputs. The size of the increments is the same for all weights and the size and direction is determined by the total change needed in the analog sum to effect a reversal of the TLU binary output.
- (3) Modified (1, 0) Training Rule [for (1, 0) input machines]--An increment is added to all weights. Those weights connected to plus one inputs are altered in a direction opposite to that of weights connected to zero inputs. The size of the increment is the same for all weights and the size and direction is determined by the total change needed in the analog sum to effect the reversal of the TLU binary output.

The (1, 0) and (+ 1, - 1) training rules were applied to all six pattern sets, whereas the modified (1, 0) training rule was applied only to the Pattern Sets 5 and 6.

D. RESULTS OF EXPERIMENTS

The learning curves for each of the six sets of patterns are illustrated in Figs. 2 through 7. Each learning curve depicts the number of errors made (out of 90 patterns) during a test procedure conducted after each iteration through the pattern set. The following conclusions seem warranted as a result of comparing the (1, 0) rule curves with the (+ 1, - 1) rule curves:

- (1) In all cases, the (+ 1, - 1) rule converges to zero errors faster and more directly than does the (1, 0) training rule.
- (2) The disparity between convergence times for the (1, 0) and (+ 1, - 1) training rules increases with the percentage of ones in the patterns, being least noticeable for the case of 25% ones and increasing to a large factor in the case of 75% ones.
- (3) The convergence time for the (+ 1, - 1) training rule is little affected by the number of ones in the patterns.

It can be shown theoretically that the modified (1, 0) rule would exhibit a learning curve almost identical with that of the (+ 1, - 1) rule when the percentage of ones in each pattern is equal to 50%. For this reason the modified (1, 0) rule was not tried on Pattern Sets 4 and 5.

Examination of Figs. 6 and 7 indicate that the modified (1, 0) training rule results in a learning curve whose convergence time is intermediate between those of the (1, 0) and (+ 1, - 1) rules. For this reason, the modified (1, 0) rule was not tested on Pattern Sets 1 and 2, where the (1, 0) and (+ 1, - 1) rules produced very similar curves.

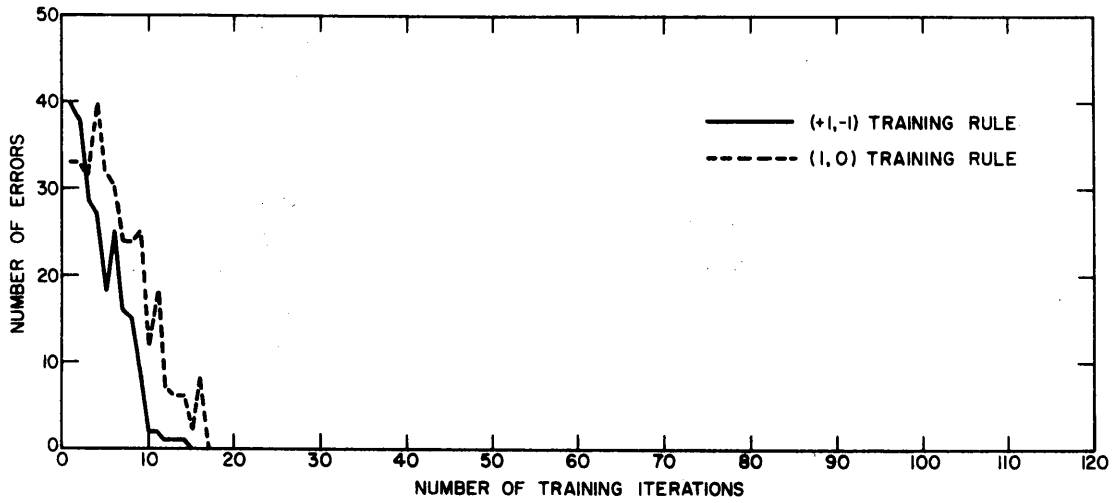


FIG. 2 LEARNING CURVES FOR PATTERN SET 1
(Each pattern containing exactly five ones)

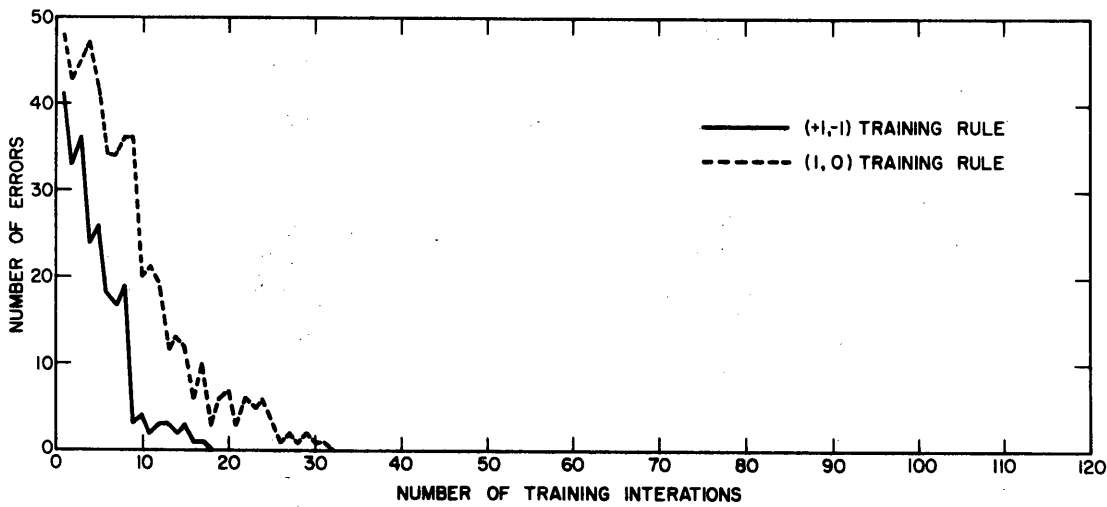


FIG. 3 LEARNING CURVES FOR PATTERN SET 2
(Each pattern containing exactly five ones)

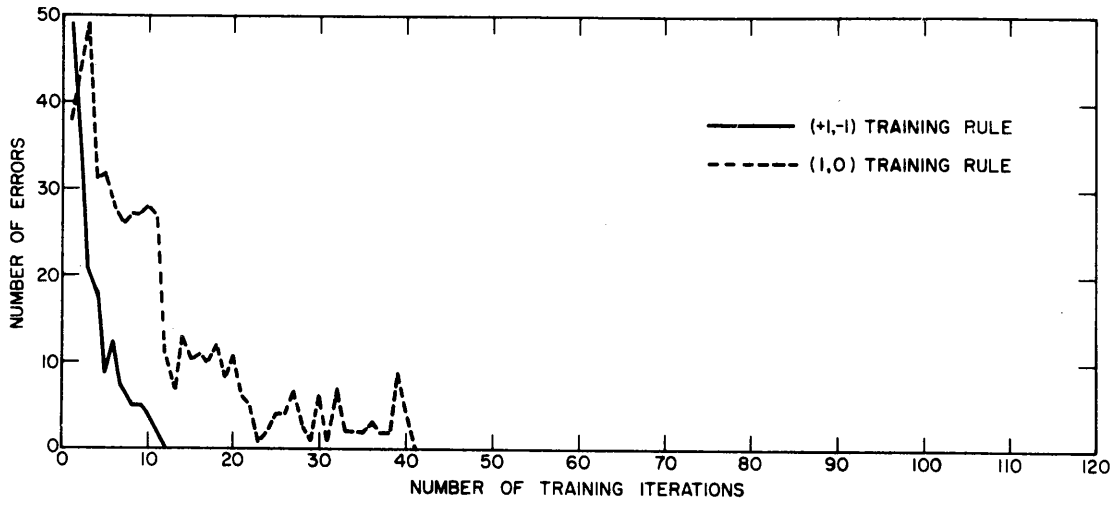


FIG. 4 LEARNING CURVES FOR PATTERN SET 3
(Each pattern containing exactly ten ones)

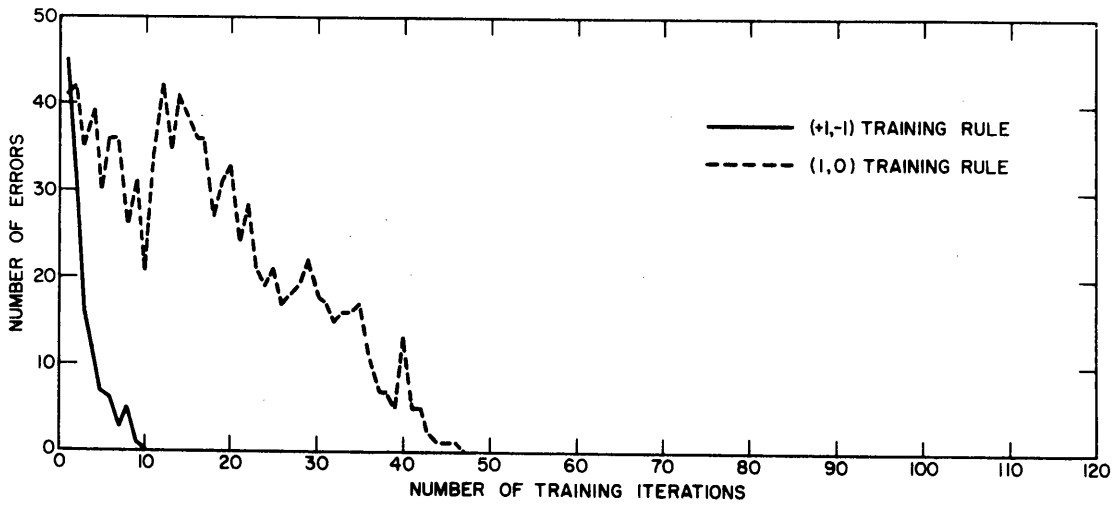


FIG. 5 LEARNING CURVES FOR PATTERN SET 4
(Each pattern containing exactly ten ones)

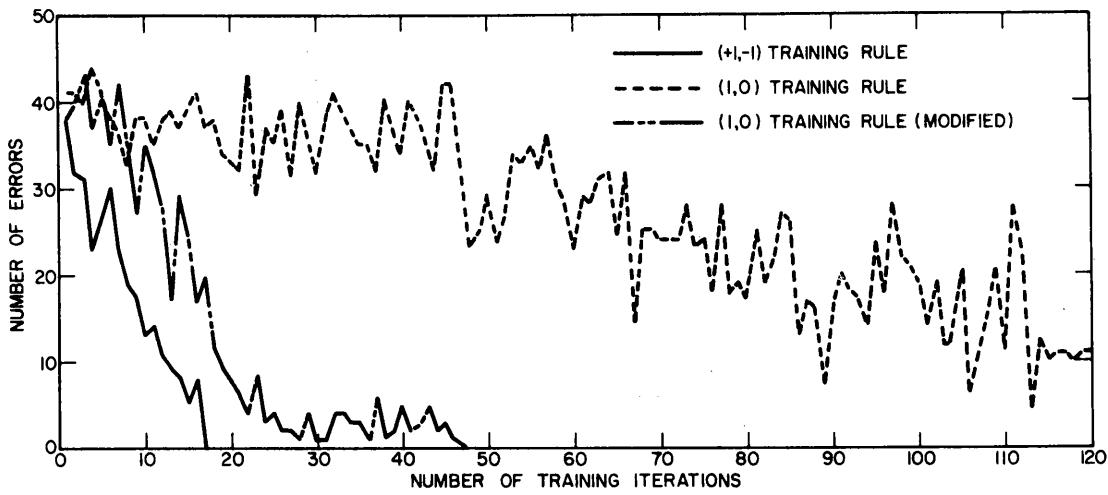


FIG. 6 LEARNING CURVES FOR PATTERN SET 5
(Each pattern containing exactly fifteen ones)

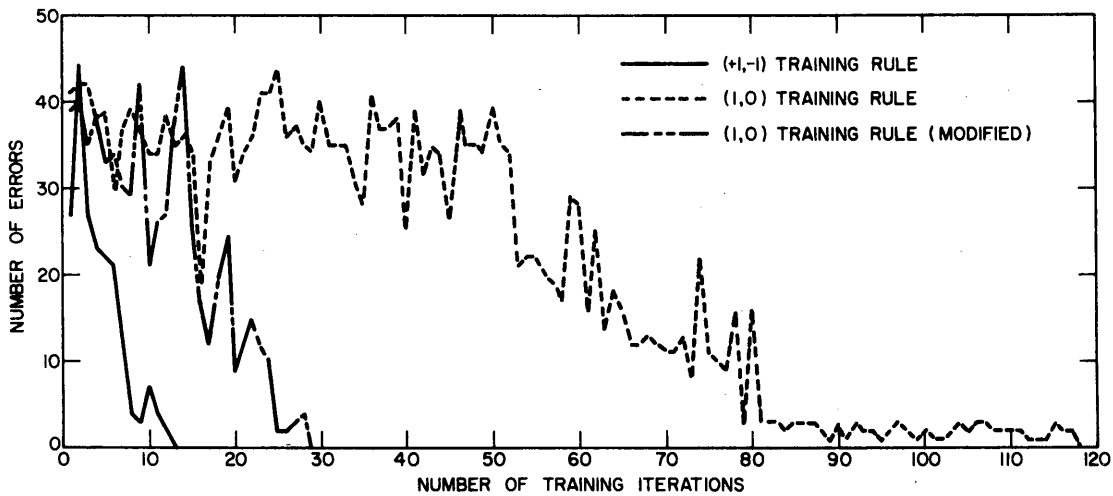


FIG. 7 LEARNING CURVES FOR PATTERN SET 6
(Each pattern containing exactly fifteen ones)

E. A PARTIAL THEORETICAL EXPLANATION*

In attempting a theoretical explanation of the apparently faster convergence rate for patterns in the (+1, -1) representation, as opposed to the (1,0) representation, we shall consider only the training of single TLU's. When observed in the past, this difference in convergence rate was attributed

- (1) to the fact that in the (+1, -1) representation the pattern vectors, and thus the correction vectors, were of equal length, and
- (2) to the tendency for pattern vectors to be more nearly orthogonal to each other when in the (+1, -1) representation.

In the experiments in this report, the pattern vectors in each trial were of equal length, since the number of ones was held constant. Thus we consider the second factor, the expected angle between pattern vectors.

If two n-dimensional pattern vectors, each having w ones, are selected at random without replacement, the probability of agreement in k ones is

$$\Pr\{k\} = \frac{\binom{w}{k} \binom{n-w}{w-k}}{\binom{n}{w} - 1} \quad \text{for } k = 0, 1, \dots, w-1.$$

In the experimental trials, n = 20 and w = 5, 10, and 15. Since $\binom{n}{w} \gg 1$, we will simplify our calculations by substituting the corresponding probability for sampling with replacement:

$$\Pr\{k\} = \frac{\binom{w}{k} \binom{n-w}{w-k}}{\binom{n}{w}} \quad \text{for } k = 0, 1, \dots, w.$$

* A similar theoretical explanation has been advanced independently by C. Kesler and G. Nagy in the following document: Collected Technical Papers, Vol. 2, F. Rosenblatt (Ed), Cognitive Systems Research Program, Cornell University, Ithaca, N. Y., p. 11, 30 July 1963.

The j th factorial moments for this distribution are:

$$E\left[\frac{k!}{(k-j)!}\right] = \sum_{k=j}^w \frac{k!}{(k-j)!} \Pr\{k\}$$

$$= \left[\frac{w!}{(w-j)!}\right]^2 \frac{(n-j)!}{n!} \quad \text{for } j = 1, 2, \dots, w.$$

Thus, the mean is

$$\mu = E(k) = \frac{w^2}{n}$$

and the variance is

$$\sigma^2 = E[k - \mu]^2 = E[k(k-1)] + \mu(1-\mu)$$

$$= \frac{w^2(n-w)^2}{n^2(n-1)}.$$

The standard derivation of \underline{k} is thus

$$sd_{\underline{k}} = w(n-w)/n \sqrt{(n-1)}.$$

We assume that the n -dimensional pattern vectors are augmented by an additional one as an $(n+1)$ st component to allow for a floating threshold level. In investigating the expected value of the cosine of the angle between two randomly selected pattern vectors, we treat the following two cases of interest:

(1, 0) Representation: Two pattern vectors agreeing in \underline{k} ones will have an angle

$$\cos \theta = (k+1)/w.$$

Using the assumed probability distribution, the expected value will be

$$E = [\cos \theta] = (w^2 + n)/n(w+1).$$

When viewed as a continuous function of w/n , $E[\cos \theta]$ has a single minimum at $w/n = [\sqrt{(n+1)} - 1]/n$.

(+1, -1) Representation: The angle between two pattern vectors agreeing in \underline{k} ones is

$$\cos \theta = [1 - 4(w-k)/(n+1)].$$

The expected value of this angle is

$$E[\cos \theta] = \left[1 - \frac{4w}{n+1} \left(1 - \frac{w}{n}\right)\right].$$

The function $E[\cos \theta]$ is symmetric about $w/n = 1/2$, and has a minimum value of $1/(n+1)$ at $w/n = 1/2$.

Example: For $n = 20$, as in the experimental trials, the following table of the expected value of $\cos \theta$ versus w is obtained:

w	E[cos θ]	
	(0, 1)	(+1, -1)
2	0.400	0.657
5	0.375	0.286
8	0.467	0.086
10	0.545	0.048
12	0.631	0.086
15	0.765	0.286
18	0.905	0.657

$E[\cos \theta]$ has a minimum of 0.36 at $w = 4$ in the (1, 0) case, and a minimum of 0.048 at $w = 10$ in the (+1, -1) case.

In general, a correction made for one pattern in a set tends to be incorrect for some others in the set; corrections are independent of each other only if the pattern vectors in the set are mutually orthogonal. Except for very small ratios of w/n , the (+1, -1) representation leads to the expectation of more nearly orthogonal pattern vectors than the (1, 0) representation, and thus, we might expect, to more rapid convergence for a single TLU.

Another factor that should be taken into account is expected variation in angle as well as the mean angle. For the (1, 0) representation, the standard deviation of $\cos \theta$ is $4/(n+1)$ times sd_k , and for the (+1, -1) representation, $1/w$ times sd_k . Just how to interpret this is hard to say, but increased standard deviation most likely leads to slower convergence.

The observed variations in learning time as a function of w for the $(1, 0)$ and $(+1, -1)$ representations during the experimental trials using a majority logic configuration of five TLU's was very much as what one might anticipate for a single TLU, based on the expected angles between pattern vectors. While it cannot be claimed that our theoretical analysis explains the experimental results, it does appear that the expected angle between pattern vectors is related to convergence of multiple TLU systems as well as single TLU's.

Discussion of Training Rules

We consider a set of $(n + 1)$ -dimensional pattern vectors $\{a_i\}$ in $(1, 0)$ representation, where the $(n + 1)$ -st component is always one, and a corresponding set $\{\delta_i\}$ of categorization multipliers, where $\delta_i = 1$ if the i th pattern is in the first class and -1 if the i th pattern is in the second class. Suppose the $(1, 0)$ training rule is: if $(\delta_i a_i, w_j) \leq 0$, then $w_{j+1} = w_j + \delta_i a_i$, where w_j is the j th trial weight vector and (s, t) denotes the inner product of the vectors s and t . The pattern vectors can alternatively be expressed in $(+1, -1)$ form by the transformation $a'_i = 2a_i - 1$. If the same training procedure is followed, the rule is: if $(\delta_i a'_i, w'_j) \leq 0$, then $w'_{j+1} = w'_j + \delta_i a'_i$. In either representation, the rule leads to a solution weight vector when one exists, and the number of corrections is finite and bounded. But, as we have seen, the convergence rate may be widely different in the two representations.

The modified $(1, 0)$ training rule used in the experimental trials was essentially of the form: if $(\delta_i a_i, w''_j) \leq 0$, then $w''_{j+1} = w''_j + \delta_i a'_i$. This rule tests with the $(1, 0)$ pattern vectors, but corrects with $(+1, -1)$ pattern vectors. Although this rule worked well in the trials, it may not converge in all cases in which a solution weight vector exists. At least, no convergence proof is available for the rule as yet.

When using the $(+1, -1)$ training rule, it is possible to test using the $(1, 0)$ pattern vectors if the threshold weight is appropriately modified during training. What is needed is a sequence w''_j of weight vectors such that

$$(\delta_i a'_i, w'_j) = (\delta_i a_i, w''_j),$$

i.e., such that

$$w_j'' = 2w_j' - (1, w_j')\tau,$$

where τ is a vector with n zeros and with a one in the $(n + 1)$ st position.

The necessary correction rule is: if $(\delta_i a_i, w_j'') \leq 0$, then

$$w_{j+1}'' = w_j'' + \delta_i [2a_i' - (1, a_i')\tau].$$

For a pattern with k ones

$(1, a_i') = 2k - n + 1$. This training rule is equivalent to the $(+1, -1)$

training rule given earlier, except that the test is made using the $(1, 0)$ pattern vectors.

In general, if the training rule is based on the transformed pattern vectors $a_i'' = \alpha a_i' - \beta 1$, where $\alpha > \beta > 0$, then the $(1, 0)$ pattern can be used for testing if the training rule is: if $(\delta_i a_i'', w_j'') \leq 0$, then

$$w_{j+1}'' = w_j'' + \delta_i [\alpha a_i' - \beta(1, a_i')\tau].$$

By suitable choice of α and β for a given collection of pattern vectors, it may be possible to get more rapid convergence than with either the $(1, 0)$ or $(+1, -1)$ representations.

F. CONCLUSIONS

We have presented the results of a series of experiments to compare the efficiency of training methods using $(1, 0)$ and $(+1, -1)$ representations for patterns. For patterns with a large number of ones, the $(+1, -1)$ representation leads to substantially shorter training times.

A partial theoretical explanation was attempted. While this explanation deals with a single TLU rather than with a network of TLUs, as used in the experiments, the effect noted can reasonably be expected to pertain to the network also. As a consequence of the theoretical explanation, it is predicted that the $(1, 0)$ representation would lead to somewhat faster training when the patterns contain a small number of ones.

Future research might well be directed toward the following topics:

- (1) A theoretical comparison of the two representations applicable to networks of TLUs.
- (2) Investigation of the effects on training time of a dispersion in the number of ones in the patterns.
- (3) Derivation, for any given set of patterns, of the optimum values of α and β for a $(+\alpha, -\beta)$ representation.