Pattern Recognition With An Adaptive Network

Lawrence G. Roberts
Lincoln Laboratory
Massachusetts Institute of Technology

Summary

This paper presents the results of several experiments with a class of adaptive networks, in which recognition of hand-printed characters selected from the english alphabet reached the 94% success level in 40 trials per character. The object of the experiments and analysis was to develop efficient reinforcement procedures for such adaptive systems. A major result was the development of a new, symmetric reward function.

Introduction

An adaptive system is one which can improve its performance with successive trials. The movement is produced by reinforcement which modifies the system on the basis of the consequences of each output. The adaptive systems considered here consist of logical networks which transform a large input pattern into a few outputs using the values of internal weights to determine the transformation. Self-organizing systems which classify patterns in this way have been investigated (1) (2), and the perception experiments described by Rosenblatt (3) are similar to the initial experiments described below.

In order to study these adaptive networks, a series of programs was written to recognize hand- printed characters. The network structure and reinforcement criteria were analyzed and varied so as to obtain the best recognition. with the simplest system. Three successive experiments are described which demonstrate the techniques used and the results obtained.

Initial Experiments

The first experiments on character recognition were performed on the TX-O computer at M. I. T . Input of free-hand characters was obtained by drawing them on the computer-controlled display scope with a light-sensing pen. A character could thus be represented on a 64 x 64 bit matrix. After several unsuccessful attempts to recognize these free-hand characters, a program was written which normalized the center of density and average radius of the characters.

Two Character Program

The first successful program classified the input characters into two pattern types. The network consisted of 2048 cells divided into two equal groups. Each cell was connected randomly to eight input bits. For each cell on aj was formed,

$$\alpha_j = \left[\sum_{i \in S_j} b_i \right]^2$$

where the set Sj included the eight input bits (bi) for the cell j. Each cell had a weight, Wj, and two correlations were performed, one for each group.

$$z_1 = \sum_{j \in S_1} (\alpha_j w_j) \qquad z_2 = \sum_{j \in S_2} (\alpha_j w_j)$$

Comparing the correlation coefficients, Z1 and Z2, a decision was made on the basis of Which was larger, this being the machine's choice of pattern type.

Reinforcement was applied to the weights of the cells in the correct set after each trial. The reward function used was of the standard type,

$$r_j = \alpha_j - \beta W_j$$

Where the r_j was added to the W_j to form the next weight and b was the decay constant. The best results seemed to be obtained with $\beta = .01$.

Results

If the inputs were chosen to be "x" and "0" the system recognized the two figures correctly 95 percent of the time, within twenty trials. Characters which had been rotated and distorted could be classified just as well after 100 trials. For the inputs "0" and "Q", the system could obtain 80 percent performance after 100 trials. These results were for the normalized characters; without normalization the program could not recognize more than 60 percent of the characters.

Six Character Program

The successor of the two character experiment was a program called Adapt I. The cellular construction of this network was the same as in the preceding program except that the cells were divided into six groups of 170 cells each instead of two groups of 1024 cells. 1his provided six outputs, the greatest of which determined the pattern choice. The reinforcement procedure was changed, however, to modify the random connections to the cells as well as the weights. The only purpose in allowing connection changes was to save cells, since inactive cells are only wasted. Thus, the following conditions constituted the reward function for Adapt I.

If $W_j \leq w$ and $a_j < 4$
then a new random group must be found such that $a_j \geq 4$.

$$r_j = \alpha_j - \beta W_j$$

where w is the average weight for the group.

Results

Since all of the weights were equal at the start , all the cells were forced to find good connections upon the first instance of any character. This meant that Adapt I could recognize any six characters on the second trial of each if they were fairly close to the originals, and within forty trials if they were sloppy, all with an accuracy above 90 percent. However, the program was very slow, taking about seven seconds per character, and its extension to more characters would make it proportionally slower.

Adapt II

In order to study the recognition of the entire hand-printed alphabet with a reasonable processing time, a new program, Adapt II, written for the TX-2 computer (4) at Lincoln Laboratory is being tested. This computer is considerably faster and more powerful than the TX-O, thus speeding up the recognition task. One major difference between Adapt II and the previous systems is that the connections are not random. The 36 x 36 input matrix is subdivided into 36 squares with eight cells connected to each square. (See Figure 1) Each cell has an input of eight bits but since these bits are all from the same area they constitute a local test on the character rather than an overall test.

In the only tests conducted with Adapt II to date, the outputs of the first layer of 288 cells have been the linear sums of the eight input bits.

$$a_j = \sum_{i \in s_j} (b_i)$$

This creates 288 aj's which are now the representation of the original 1296 bits .The second layer of cells perform the correlation of aj and an set of weights, wjk, for each of 44 outputs.

$$z_k = \sum_j a_j w_{jk}$$

Thus an output is obtained for each of 44 characters, the largest indicating which character is chosen.

Reinforcement

A reward function usually operates upon the weights of a system in such a way as to improve the correlation between the correct set of weights and the aj's. This operation is usually independent of the transformation from the input to the aj's .Thus, the studies made on reinforcement procedures in Adapt II also apply to similar sys- tems qn which the aj's need not be linear sums of the input.

The first reward function tested was the standard function used in previous models, How- ever, it was desirable to keep the mean value of the weights stable at zero, so a zero average term, was used instead of aj. Also, it was necessary for computational efficiency to fix an upper and lower bound on the weights such that. Both a reward, rj and an inhibition, rj, were used, where the rj were added to the wj of the correct character and themissing textwere added to the wj of an incorrectly chosen character :

The term k is a constant to restrict the weights, While a and b are variables determining the strength of the reward and inhibition.

Results of Standard Reward

Adapt II was tested on a series of unnormalized hand printed characters chosen from examples of printing supplied by many individuals. The 256 samples used for the tests consisted of about thirty-eight characters fairly evenly distributed as to class. Recognition scores achieved by humans for individual characters without context was about 95 percent.

The program was tested upon these characters, rewarding itself on the basis of the human labeling after having made its own decision. A count was kept of the correctness of the program I s decision and graphs made of the probability of correct response, Pc .The parameters a and b were varied to obtain a maximally correct response. The graphs for three sets of parameter values appear in Figure 2. For the set of parameters a = 2, b = 2, the maximum recognition was Pc =.57. This was the best recognition that the standard reward was found to produce, and there was no indication that the performance would improve with time. These poor results prompted the design of a new reward function.

Revised Reinforcement

If the condition that rj = 0 if |wj| = M is required of a reward function, then the following reward functions, labeled sj and sj , are perhaps the simplest .

These functions differ from the rj functions in that the maximum reward or inhibition occurs When wj = 0 instead of when wj = M. Since negative values of wj imply as much as positive values about the input aj, the symmetry of the "s" functions is pleasing.

Results of s-Function Tests

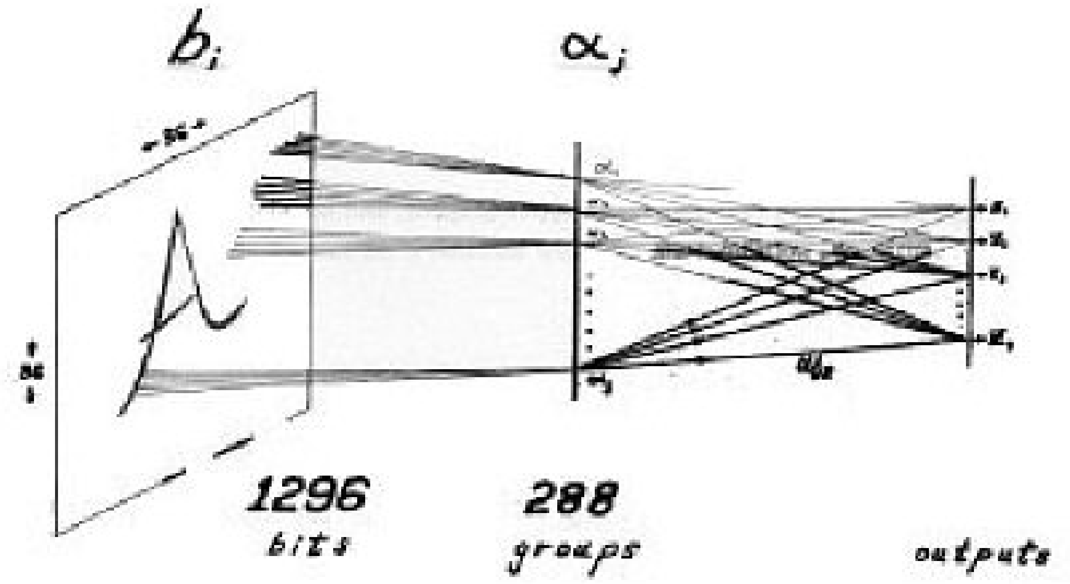
The tests on the "s" rewards were conducted in the same manner as for the standard functions and showed a decided improvement. The results appear in Figures 2 and 3 wi th the bottom graphs in Figure 2 being those for the rj functions . The parameters found to result in the fastest improvement for Pc < .6 were c = d = 2, and the combination c = 1, d = 3 was found to produce the greatest final value of Pc. The combination of these two sets of parameters with a change-over at Pc = .6 resulted in the dashed curve in Figure 2. The graph represents the maximum performance obtained from Adapt II to date, showing adaptation to the complete alphabet with a probabilityof .94, in about forty trials per character.

Conclusions

The networks described demonstrate that it is possible to recognize characters with probabilities as high as .94 with a training period of forty trials per character if a suitable reward function is used. The new "s" reward developed for Adapt II provided a considerable improvement over the standard reward, pointing out the sensitivity of adaptive systems to the kind of reinforcement. It should be possible to improve on the reported performance with further investigation into reward procedures. Research in this area has been greatly facilitated by the logical simplicity and speed of Adapt II Which contains only 200 TX-2 instructions and processes characters at a rate of four per second.

Acknowledgment

The author is indebted to Dr. H. L. Sherman for the use of his data, to H. P. Peterson for his earlier work in the reduction of these hand- printed characters to digital form, and to the staffs of the TX-O and TX-2 for their assistance and cooperation in the use of the computers.



$$\alpha_j = \sum_{i \in S_j} b_i \qquad Z_k = \sum_j \alpha_j w_{jk}$$

Fig. 1 Organization of Adapt II.

Lawrence G. Roberts
Lincoln Laboratory
Massachusetts Institute of Technology

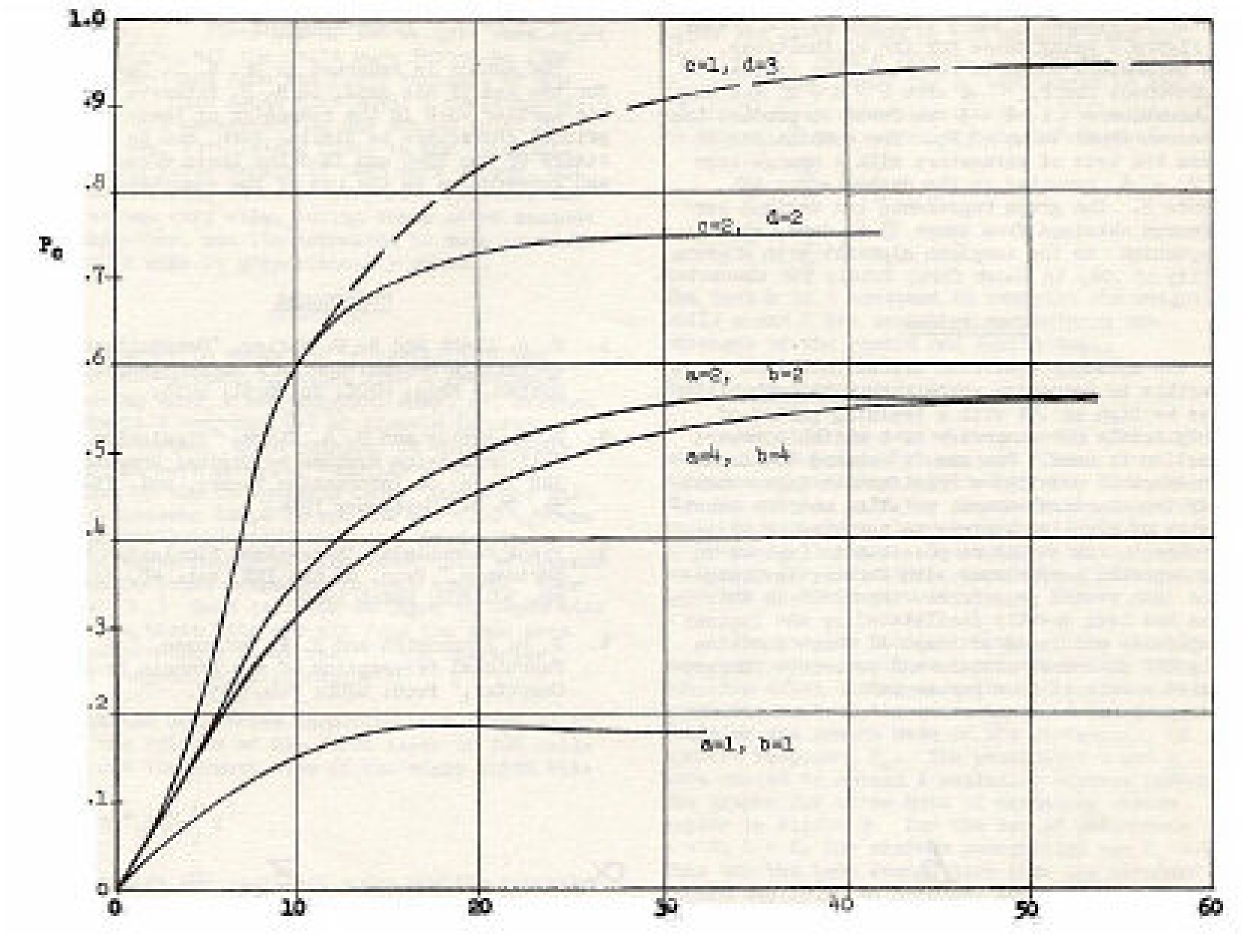


Fig. 2. Learning curves for s reward function (upper graphs), and for standard reward (lower three graphs), with Adapt II.

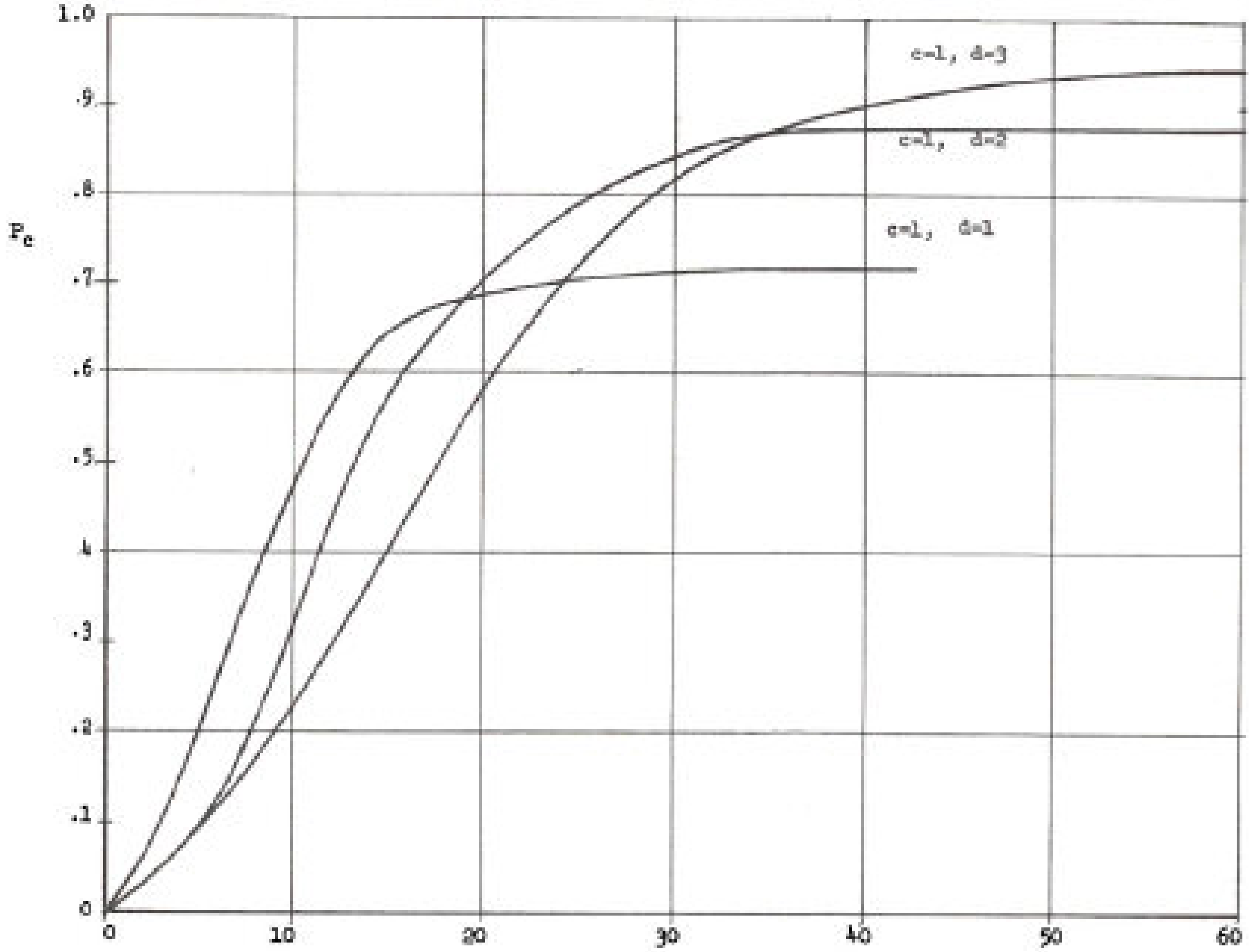


Fig. 3. Learning curves for a function with varying inhibition strengths.

References

W. A. Clark and B. G. Far1ey, "Generalization of Pattern Recognition in a Se1f-Organizing System," Proc. WJCC, pp 86-91; 1955.

B. G. Farley and W. A. Clark, "Simulation of Self-Organizing Systems by Digital Computer, " IRE Trans. on Information Theory, vol. IT-4, pp. 76-84; September 1954.

Frank Rosenblatt, "Perception Simulation Experiments," Proc. of the IRE, vo1. 48, No. 3, pp. 301-309; March 1960.

J. M. Frankovich and H. P. Peterson," A Functional Description of the Lincoln TX-2 Computer," Proc. WJCC; Feb. 1957.

*Operated with support from the U.S. Army, Navy, and Air Force.