PRACTICAL APPLICATIONS FOR ADAPTIVE DATA-PROCESSING SYSTEMS

by

B. Widrow,* G. F. Groner,* M.J.C. Hu,* F. W. Smith,*
D. F. Specht,* L. R. Talbert*

A. Introduction

Adaptive data-processing systems which can be trained to classify complex digital and analog patterns have been under development over the past several years. Methods of adaptation, convergence rates, statistical memory capacities, and generalization capabilities have been studied for single threshold elements (Adaline) and for certain networks of adaptive threshold elements (Madaline). In addition, simple Adaline networks have been used successfully in recognition of speech, in weather forecasting, in an automatic control system, and in diagnosis of electrocardiographic waveforms. The inherent ability of adaptive neural nets to generalize, i.e., to extrapolate a means of behavior from a very limited amount of training, has been absolutely essential in making these applications possible.

B. Adaline and Madaline

A basic building block of the systems to be considered is an adaptive threshold element, sometimes called an adaptive "neuron." For the past several years, we at Stanford University have called this element Adaline (adaptive linear neuron). A functional diagram of this element is shown in Figure 1. It includes an adjustable threshold element and the adaptation machinery which automatically adjusts the variable weights. It has been demonstrated experimentally and theoretically that this element can be trained to react specifically to a wide variety of binary input signals and that it can be trained to generalize in certain ways, i.e., to react as desired with high reliability to inputs that it has not been specifically trained on.

In Figure 1, the binary input signals on the input lines have values of +1 or -1, rather than the usual values of 1 or 0. Within the neuron shown, a linear combination of the input signals is formed. The weights are the gains $w_1, w_2, \ldots$, which may have either positive or negative values. The output signal is +1 if this weighted sum is greater than a certain threshold, and -1 otherwise. The threshold level is determined by the setting of w_0, whose input is permanently connected to a +1 source. Varying w_0 varies a constant added to the linear combination of input signals.

For fixed gain settings, each of the 2^n possible input combinations would cause either a +1 or a -1 output. Thus, all possible inputs are classified into categories. The input-output relationship is determined by choice of the gains $w_0, \ldots, w_n$. In the adaptive neuron, these gains are set during the training procedure.

In general, there are 2^{2^n} different input-output relationships or truth functions by which the n input variables can be mapped into the single output variable. Only a subset of these, the linearly separable logic functions, can be realized by all possible choices of the gains. Although this subset is not all inclusive, it is a useful subset, and it is "searchable," i.e., the "best" function in many practical cases can be found iteratively without trying all functions within the subset. An iterative search procedure has been devised and is described below. This procedure is quite simple to implement, and can be analyzed by statistical methods that were originally developed for the analysis of adaptive sampled-data systems.[1]

An adaptive pattern classification machine has been constructed for the purpose of illustrating adaptive behavior and artificial learning. A photograph of this machine, which is an adjustable threshold element (called "KNOBBY ADALINE"), is shown in Figure 2.

During a training phase, simple geometric patterns are fed to the machine by setting the toggle switches in the 4 x 4 input switch array. All gains, including the threshold level, are to be changed by the same absolute magnitude such that the analog error (the difference between the desired meter reading and the actual meter reading) is brought to zero. This is accomplished by changing each gain in the direction which will diminish the error by 1/17. The 17 gains may be changed in any sequence, and after all changes are made, the error for the present input pattern is zero. The weights associated with switches up (+1 input signals) are incremented by rotation in the same direction as the desired meter needle rotation, the weights connected to switches in the down position are incremented opposite to the desired direction of rotation of the meter needle. The next pattern and its desired output is then presented, and the error is read. The same adjustment routine is followed and the error is brought to zero. If the first pattern were reapplied at this point, the error would be small but not necessarily zero. More patterns are inserted in like manner. Convergence is indicated by small errors (before adaption), with small fluctuations about stable weights. A least mean square adaption procedure (LMS) requires that adaption be made even if the quantized neuron output is correct. If, for example, the desired response is +1, the neuron is adapted to bring the analog response closer to the desired response, even if the analog response is more positive than +1.

The iterative training routine is purely mechanical. Electronic automation of this procedure will be discussed below.

The results of a typical adaption on six noiseless patterns is given in Figure 3. During adaption, the patterns were selected in a random sequence, and were classified into 3 categories. Each T was to be mapped to +30 on the meter dial, each G to 0, and each F to -30. As a measure of performance, after each

* Department of Electrical Engineering, Stanford University, Stanford, California.

adaption, all six patterns were read in (without adaption) and six analog errors were read. The sum of their squares denoted by Σe^2 was computed and plotted. Figure 3 shows the learning curve for the case in which all gains were initially zero.

It is shown in References 2 and 3 that making full correction with each adaption using the LMS procedure is a stable "performance feedback" process having an adaptive time constant equal to the number of weights. In the experiment of Figure 3, the time constant is 17 adaptions. It is also shown that changing each weight by the same magnitude in the appropriate directions is equivalent to utilization of the method of steepest descent on a mean square error surface. A number of other steepest descent adaption procedures have been devised by W. C. Ridgway III, [4] and C. H. Mays, These procedures have been analyzed by Mays with regard to proofs of convergence and bounds on the number of adaptions required for convergence. The decision to adapt may be based on one of the following rules: adapt only if the response is incorrect; or adapt only if the response is incorrect or within a "dead zone." If the decision to adapt is made, then the increment size might be fixed or might be proportional to the analog error, the difference between the analog sum and the desired output. These procedures are described in detail in Mays' Ph.D. thesis, [6] along with bounds on the number of adaptions needed for convergence.

The effects of adaptive feedback in Adaline networks on their ability to self-heal by adapting around internal defects are analogous to the effects of feedback in amplifiers and control systems in making system performances insensitive to gain changes and nonlinearities. P. R. Low [7] has studied, by simulation and by analysis, what he calls "defective" Adalines. One such Adaline has a set of weights whose integration speeds vary over a 5 to 1 ratio. These speeds are randomly selected from a uniform distribution of speeds. It was found that sometimes the non-uniformity in the adapt rates hinder and sometimes help, but on the average, this wide variation among the speeds increases the total number of adaptions required to achieve convergence, but by only 5 percent. The resultant weight values are essentially unaffected by this, as are the functions realizable and the statistical memory capacity.

An important question is, how many patterns or stimuli can the single adaptive neuron be trained to react to correctly at a time? This is a statistical question. Each pattern and desired output combination represents an inequality constraint on the weights. It is possible to have inconsistencies in sets of simultaneous inequalities just as with simultaneous equalities. When the patterns (i.e., the equations) are picked at random, the number which can be picked before an inconsistency is created is a random variable. As few as 4 patterns can form a nonlinearly separable set, regardless of the pattern size.

A series of experiments was devised by J. S. Koford and R. J. Brown where patterns containing unbiased random bits and random desired responses were applied to Adalines with varying numbers of inputs. It was found that the average of random patterns that can be absorbed by an Adaline is equal to twice the number of weights. This is one basic measure of memory capacity. It was recently proven by Brown that this experimental result is rigorously correct. Analytical curves showing the probability of being able to train in N patterns as a function of N/(n+1) are presented in Figure 4. The total number of adaptive weights is (n+1), including the threshold weight. Notice the sharpening of the break point of these curves at exactly the average capacity as the number of inputs to the Adaline increases.

Derivation of the capacity formula and of the curves in Figure 4 will be presented in a Ph.D. thesis by Brown.

Storage capacity in excess of that of a single Adaline can be readily achieved by use of parallel multi-Adaline networks. Several Adalines can be used to assist each other in solving problems by automatic load-sharing.

The configuration in Figure 5 shows a Madaline (multiple Adalines) of 5 Adalines with parallel-connected inputs in the first layer. In the second layer of fixed logic the Adaline outputs are connected to a majority-rule element whose output is the system output. The "job assigner," a purely mechanical device, automatically decides which Adalines, if any, need adaption. There are a variety of fixed logic schemes that could be used on the second layer. M. E. Hoff, Jr., in his doctoral thesis [8] described convergent adaption procedures that can be used with all possible fixed-logic second layers.

One procedure for training these networks is to use the "minimum change" rule. Under this rule:

i) No adaption is performed if the system output is correct.

ii) If the system output is in error, a minimum number of the incorrect Adalines are adapted. The Adalines chosen for adaption are those whose analog responses require the least amount of change to give the proper response.

When adaption is performed according to the minimum change rule, various Adalines tend to take "responsibility" for certain parts of the training problem. Thus, this rule produces load sharing among the Adalines by assigning responsibility to the Adaline or Adalines that can most easily assume it.

The adaptive system of Figure 5 was suggested by common sense, was tested by simulation, and was found to work very well. It was subsequently proven by Ridgway in his doctoral thesis [4] that this system will converge on a solution, if a set of weights exists that will solve the training problem. The essence of the proof lies in showing that the probability of a given Adaline taking responsibility for adaption to a given pattern, desired-response pair is greatest if that Adaline had taken such responsibility during the previous adapt cycle in which the pattern was presented. The division of responsibility stabilizes at the same time that the responses of the individual Adalines to their share of the load stabilizes. In the case that the training problem is not perfectly separable by this system, it can be shown that the adaptation process tends to minimize error probability.

The memory capacities of Madaline structures utilizing both the majority element and the OR element have been measured by Koford. Although the logic functions that can be realized with these output elements are different, both types of elements yield structures with the same statistical storage capacity. The average number of patterns that a Madaline can be adapted to equals the capacity per Adaline multiplied by the number of Adalines. The memory capacity is therefore equal to twice the number of weights. This result remains to be proven analytically.

With suitable pattern-response examples and the proper training procedures, generalizations can be trained into Adalines. The kinds of generalizations that have been produced are concerned with the training of Adalines to be statistically insensitive to noise, and to be sensitive or insensitive to translation, rotation, and size. Adalines can be forced to react consistently on a training set of patterns for all possible positions, for example, and then they will react consistently in all positions with high reliability on new patterns never seen before, which are quite unrelated to the training set. Single Adalines and simple Madalines have been trained to be insensitive to left-right translation, size, and rotation simultaneously, and have performed with very high reliability, much greater than chance. Reference 9 gives examples of these phenomena and the weights that resulted in the adaptive structures.

C. A Real-Time Adaptive Speech Recognition System

A list of the design requirements for a machine which recognizes speech would surely include as desirable features: (1) the use of several parameters of the speech waveform which contain the information required to characterize many sounds; (2) a decision network which is sophisticated enough to handle many speech recognition tasks yet is easily designed; and (3) the ability to recognize voices other than those on which its design is based.

A real-time adaptive speech recognition system[10] has been constructed which is composed of a preprocessor with a microphone input which feeds a bank of eight band-pass filters spaced throughout the audio spectrum. Spectral power is quantized and sampled to form a 240-bit pattern for each spoken word. When these patterns are applied to Adaline networks for classification, the reliability is high. With an 18-word vocabulary, after training on only 8 samples per word, the system was able to make correct identification of new samples of these words with better than 95 percent reliability. When interrogated by new voices of the same sex, the average reliability was better than 85 percent.

(1) System description

The input transducer of the system utilizes sound spectra of words because the manner in which the energy at various frequencies changes during the course of an utterance can serve to identify the word spoken. Figure 6 is a pictorial representation of the sound spectrum for the word "zero."

The three variables to consider are frequency, energy, and time, each of which must be quantized for presentation to the Adaline network. Figure 7 is a block diagram of the adaptive speech system. The speech sound is passed from a microphone through a processor, consisting of an audio amplifier and an automatic level control, to eight band-pass filters which have center frequencies spaced from 300 cps to 4.5 kc and bandwidths on the order of 200 cps. The outputs of the filters are rectified and integrated to produce waveforms similar to the slices which make up Figure 6.

Figure 8 shows in greater detail the procedure for time sampling and energy quantization. For simplicity the waveform for the 2.8 kc filter of Figure 6 has been redrawn. Three quantum levels have been drawn horizontally across the waveform, dividing the amplitude scale into four quantum zones. Each of these quantum zones has associated with it a three-bit binary code, as shown in Figure 8. Time sampling is initiated and terminated by a voice-controlled trigger, which is a switch operated by the speech intensity. During the utterance, the waveform is sampled ten times; each time a sample is taken, the corresponding three-bit pattern is stored. When this process is complete, the waveform has been translated into a 30-bit binary pattern which can be used with an Adaline network. With the actual system, eight such waveforms are translated into a 240-bit binary pattern for each spoken word.

The patterns formed for a given word will change if the rate at which the word is spoken is changed. The effect of this rate variation is reduced by a linear time normalization process. Real-time sampling takes place at a rate of about 250 samples per second. Most samples are discarded. Normalized patterns are formed by taking 10 samples at intervals such that each interval is one eleventh of the total word length. The combined effect of amplitude and time normalization is to reduce the error rate for a wide variety of test situations by a factor of 5 to 10.

The sampler, digital storage, Adaline network, and decoder are simulated on an IBM 1620 digital computer - an approach which enables the experimenter to make changes in the system with only a change in the computer program. Several programs written to simulate a variety of Adaline networks and speech pattern forming techniques have nearly the same operating features. The set of words to be recognized is entered into the computer at the beginning of the training phase. The training speaker(s), directed by the computer, then speaks examples of these words for training. After all the training examples and their associated output codes have been accepted by the computer, their corresponding patterns are sequentially presented to the Adaline network which is trained until all of these patterns are classified correctly. When the training is completed, any person may speak a word, and the system forms the pattern, presents it to the Adaline network, decodes the output, and types the associated word on the computer typewriter.

(2) System Performance

A series of experiments was performed to determine

how well the adaptive speech system can recognize speech patterns not included in the training set. Sets of binary patterns made by various speakers were recorded on punched cards and were later used for training and testing the system. This allowed the same patterns to be used for comparing various system organizations, and prevented speakers from purposely changing their voices during an experiment in an attempt to lower error rates. There was some background noise in the laboratory during the recording sessions.

Two sets of words were considered, (1) the ten decimal digits ("zero" through "nine"), and (2) ten phonetically balanced monosyllable words.[11] In each case the system was trained by a single speaker (10 examples per word), then tested by several persons, each speaking 10 examples per word. When the "training" speaker spoke new examples of the words for testing, the error rates were 2 percent, and 0 percent respectively for the two word sets. When other speakers tested the system, the average error rates were 15 percent, and 5 percent for these word sets. The system not only learned the distinguishing characteristics of the words, but also the characteristics of the training speaker. It still, however, demonstrated an ability to do well on the voices with which it had no previous experience.

The effect of the size of the word set was studied by adding randomly selected phonetically balanced words to a list, then training and testing with a single speaker each time the length of the list was increased. The error rate was less than 2 percent until there were 18 words in the list. The error rate for 18 words was 4 percent; that for a 20 word set was 14 percent.

Four more experiments were performed to study the versatility of the system. In the first, the system was trained to identify, in English, the digits "1" through "4", spoken in four languages. The languages used were Chinese (Mandarin), English, French, and Portuguese, spoken by native speakers. When tested by the speakers after training, the system identified all words spoken with an average error rate of 14 percent. In the second experiment, the system was trained to identify three speakers saying the word "you." After training, the system identified the speakers with an average error rate of 23 percent. In the third experiment, words were spoken to the system through a conventional telephone line by a person 4 miles away. After training the system on the digits "1" through "4", this person spoke 10 new examples of each of these words. The system recognized this set of 40 words with 100 percent accuracy. The computer program was modified in the fourth experiment so that several words could be spoken in a series during the recognition phase. When spoken with a slight pause between words, whole sentences were recognized with the same error rates as when the words were spoken one at a time.

In another set of experiments, the system was trained by one, three, and five speakers to recognize the ten digits. When tested by four other speakers, the average error rates were 18.7 percent, 18.5 percent, and 27.7 percent respectively. This was a surprising result because it had been presumed that the system would perform better if it had a greater variety of experience during training. Work on this aspect of the problem is continuing.

This speech recognition system is an extremely simple one and has performed quite well. Research is under way to improve it. This system has been an excellent source of patterns for testing Adaline-Madaline types of recognition systems in a real and practical context.

D. Adaptive Weather Forecasting

When meteorologists forecast the weather based on their past experience, they are actually recognizing relationships between current and future meteorological events. Weather forecasting can thus be thought of as a type of pattern-recognition problem. Adaline, used as a pattern recognition device, might serve as a weather forecaster since it has the ability to make decisions, based on past training experience, to previously unseen problems.

(1) Adaptive Methods

An adaptive imitator[12] that is capable of imitating weather conditions is also capable of giving weather forecasts. Figure 9 represents an adaptive weather imitator.

The complex meteorological system which the adaptive imitator tries to imitate is the "World". The input and output of the "World" are "yesterday's" and "today's" weather respectively. The "adaptive process" is an automatic operator whose function is to correct the adaptive imitator whenever it disagrees with the system that it is attempting to model. The adaptive imitator could be as simple as a single Adaline.

An adaptive weather forecasting system can be operated in the following manner. "Yesterday's" weather conditions are presented to the system, and it is trained to read "today's" weather. Then if "today's" weather is presented as an input, it will give tomorrow's weather. The training procedure should actually go on for days, weeks and years, until the system has seen a large variety of weather situations and learns to forecast with very few errors on the average. In practice, the system could be trained on past weather information from previous years.

(2) An Experiment

Weather data from January to April 1961 were obtained from Mr. H. E. Root of the U.S. Weather Bureau at the San Francisco International Airport. Several experiments have been performed. In one experiment, a single Adaline was trained to "read" surface-pressure maps, and to forecast fair or rain for the San Francisco Bay Area.

Figure 10 shows that the inputs to the Adaline are the quantized grid-point pressures. In this experiment, pressure information over an area of approximately 5,000,000 square miles (bounded by latitudes 25^{o}N to 55^{o}N and by longitudes 110^{o}W to 150^{o}W) was used.

To train the Adaline to recognize locations of high and low pressure centers, the area shown in Figure 10 was divided into forty-eight 5^o by 5^o squares. Instead of using the average pressure at each grid point, the average pressure over each 5^o by 5^o square was used as an input to the Adaline. The range of pressures was linearly divided up into ten levels, i.e., each 5^o by 5^o square was quantized into ten levels of pressure.

An adaptive weather forecasting system consisting of three independent multilevel-input and binary-output Adalines was simulated on an IBM 1620 computer. A steepest descent adaption procedure designed for analog inputs was used. This system was trained to forecast fair or rainy weather in the San Francisco Bay Area for the succeeding period of 36 hours, at 12-hour intervals. The first Adaline was trained to be an expert at forecasting for the first 12 hours, the second Adaline for the second 12 hours, and the third Adaline for the last 12 hours.

The experiment consisted of using three types of weather information. The three-Adaline system was trained to recognize 33 patterns of each of the following types.

1. Today's 0400 PST (Pacific Standard Time) surface-pressure map.

2. Today's 0400 PST and yesterday's 0400 PST surface-pressure maps.

3. Today's 0400 PST surface-pressure map and the difference ($\Delta P/\Delta t$) between today's and yesterday's quantized pressures.

The system that was trained on patterns from 2 and 3 required twice as many inputs, 96 instead of 48 inputs, since the amount of input information had been doubled.

After training, the three-Adaline system was then tested on 18 patterns from each of the types mentioned above. The forecaster's percent probability of rain were interpreted to be fair if the percent probability of rain was less than 50 percent, and to be rain if the percent probability was 50 percent or more. The performance of the three-Adaline system was then compared with the official forecast for those 18 days. The results are tabulated below.

WEATHER FORECAST FOR	OFFICIAL FORECAST	ADALINE FORECAST Using Patterns		
		1	2	3
	Score (%)	Score (%)	Score (%)	Score (%)
TODAY 8:00 a.m. - 8:00 p.m.	78	72	78	78
TONIGHT 8:00 p.m. - 8:00 a.m.	89	67	78	89
TOMORROW 8:00 a.m. - 8:00 p.m.	67	67	78	83

The results of this test and of subsequent tests indicate that Adalines using today's surface pressure, and $\Delta P/\Delta t$ weather information can do as well as human forecasters on the initial 24 hours, and can do better on the last 12 hour forecast. The results of this test are interesting since the 33 training days and 18 testing days were chosen during the rainy season and at transitions. These were days that forecasters had trouble in forecasting.

Other interesting results can be obtained by studying the final weight settings of the Adaline after it has been trained. Areas of significance will show up as weights with large values, which can furnish additional information to forecasters when they make their weather forecasts. Areas with large weight values have appeared which correspond to propagation paths of storm movements.

(3) Other Applications

There are numerable problems in weather forecasting where Adalines or Madalines might find use. For example, a system of Adalines or Madalines could be trained to give weather forecasts for Seattle, San Francisco, Los Angeles and San Diego all at once using the same weather data; or they could be trained to forecast the amount of precipitation for different locations; or they may be trained to reduce information from pictures of cloud patterns taken by weather satellites.

The results of these experiments and of a more recent experiment using 200 training patterns and 100 testing patterns during the rainy seasons over the past seven years have demonstrated that an Adaline, using only limited amounts of data, can be used as an objective weather forecaster. Based on Adaline's performance, it is conceivable that an adaptive weather forecasting system can become an effective meteorological operational and research tool.

E. Adaptive EKG Diagnosis

Another promising area for the application of adaptive systems is that of diagnosis in clinical medicine. In this area, correlation between symptoms, test results, and the malady-to-be-discovered is certainly present, but it is often difficult to describe.

With the cooperation of Drs. Toole and Von der Groeben of the Department of Cardiology of the Stanford University Medical School, a simulated Madaline structure was given the task of discovering the correlations between vectorcardiograms and heart disease with the intention that it would thereby become a reliable diagnostic aid for heart disease.

The heart muscle, in its normal function of contracting and expanding, is a current source which generates an electric field in the human body. More precisely, it can be represented by numerous small current sources which are simulated into conduction sequentially. An electrocardiogram is a recording of the changing electric potential between various points on the surface of the body resulting from redistribution of charge within the heart tissue.

A clinical electrocardiogram (EKG) consists of 12 or more tracings recorded sequentially. On the other hand, vectorcardiograms (which are not in common use clinically) reportedly contain as much or more information in only 3 tracings recorded simultaneously. The name derives from the fact that the three voltages approximate three orthogonal components (x = right-left, y = head-foot, z = anterior-posterior) of a voltage vector which is the resultant effect of the electrical activity going on within the heart. It was chosen for this study since it clearly has less redundancy than the clinical EKG and it retains the phase information which is lost by the sequantial recording of the clinical EKG. An example of a normal vectorcardiogram is shown in the upper part of Figure 11.

The vectorcardiogram consists of three time-varying analog voltages. Although adaptive threshold elements can be made to accept analog inputs as well as binary, time-varying inputs cause the output to be a function of time also. Since it is desired that the output be a function of the total waveform, the solution is to sample the waveform in time and to apply each of the samples to a separate input of each of the adaptive threshold elements.

Experimental work has been limited so far to analysis of the QRS portion of the signal taking samples every 5 msec up to 75 msec measured from the onset of QRS. The QRS portions of the waveform in the upper part of Figure 11 is shown,after sampling, in the lower part of that figure.

The first (and perhaps most important) task is the separation of normals from abnormals. The separation of abnormals into the different diseases is a second phase. For either phase, the QRS samples are applied to a Madaline of the structure shown in Figure 12. The 45 analog samples of Figure 11 are seen here being applied simultaneously to a number of 45-input Adalines.

The adaptive threshold element outputs are combined by an OR element, and are also connected to a maximum detector. The output of the OR element is entirely adequate for classification alone, but the analog outputs of the elements were retained to serve as a "level of confidence" indicator to the physician.

To assure reliability of the classification of the training samples used in the experiments, the clinical EKG eiagnosis was supplemented by a complete physical examination, a study of the patient's medical history, and, in many cases, hemodynamic studies. Finally an independent set of samples (also with known diagnoses) was used to test the trained Madaline. The Madaline responses were then compared with the known diagnoses.

In one experiment, 107 vectorcardiograms were obtained for training. Five Adalines were used to separate the normals from the abnormals in the training set. To check the generalizing ability, the trained Madaline was then tested on 57 new cases with results as shown below:

RECOGNITION RATE ON THE TESTING SET

	True Normals (27 cases)	True Abnormals (30 cases)
Clinical EKG	> 95%	54%
Generalized Adaptive Approach	89%	73%
Improvement	-6%	+19%

There is ample evidence that the comparison will be even more favorable when the Madaline is trained on a much larger, and a more truly representative sample of patients.

Work is presently in progress (using the present small sample) in identification of abnormalities. A preliminary experiment in separation of Right Ventricular Hypertrophy from other abnormalities indicates generalization to 85 percent of cases not in the training set.

F. Pattern-Recognizing Adaptive Control Systems

The conditions of a dynamic system can be completely described at any instant by the values of its state variables (such as the error, the error derivative, etc.). Control decisions depend only on the present values of the state variables. These values can be encoded as a sequence of binary digits, the collection of which forms a pattern. Proper control of a dynamic system by an Adaline or Madaline becomes a matter of classification of the patterns which represent the different states of a dynamic system. Just as an Adaline can be taught to classify patterns into two groups, it can be taught to control a dynamic system in a "bang-bang" or +1, -1 manner.

When the state variables are encoded using what has been called a "linearly separable code", the task of learning control strategies is quite natural for an Adaline.

i) The large sets of patterns representing the control strategy for all possible regions of state space are often either linearly separable or separable with simple Madaline structures. The number of patterns which the Adaline is able to correctly classify is generally an order of magnitude or more greater than its statistical capacity.

ii) The Adaline generalizes in a known and predictable way. Namely, the Adaline can correctly classify all the patterns of a control strategy after learning to correctly classify only the patterns bordering on the switching surface.

iii) Because of this strong generalizing property and because of the special interrelationships among the

many patterns, the Adaline is much easier to train than it would be for a similar number of random or near random patterns.

(1) Trainable Controllers

Figure 13 shows in block-diagram form the general situation in which a Madaline would be used as a trainable controller for a dynamic system. The state variables $y_1, \ldots, y_m$ are assumed to be the system error.

The teaching controller supplies the desired output to the Adaline during the training process. This controller could be automatic or possibly human. The Adaline controller and the teaching controller need not have the same inputs, provided both receive the same or related information. For instance, the Adaline controller could be receiving the state variables as electronic signals while a human teacher could be receiving information about the system by actually watching its motions.

For the purposes of discussion the teacher will be assumed to be represented by a function $f(y_1, \ldots y_m)$. The switching surface $f(y_1, \ldots, y_m) = 0$ indicates where the teacher changes his reaction from "force plus" to "force minus." The Adaline with its encoder is basically a trainable function generator which forms the function $\hat{f}(y_1, \ldots, y_m)$. During the training, the Adaline analog output is adjusted so that its switching surface $f(y_1, \ldots, y_m) = 0$ is made to approximate the switching surface of the teacher.

The Adaline controller consists of an encoder and an Adaline. For simplicity, a single Adaline is shown here in the controller; more typically a Madaline might be used. The encoder produces patterns by quantizing or dividing the range over which each of the state variables varies into a finite number of zones. Each zone of a state variable y_i is represented by binary number or partial pattern. The m partial patterns make up the total pattern which represents a particular hypercube of state space. The pattern inputs to the Adaline change continually as the state variables change.

Figure 14 illustrates the quantization of a two-dimensional state space. Each square in the figure is represented by a particular pattern for the Adaline. The continuous curve $f(y_1, y_2) = 0$ represents a typical desired switching surface (a curved line in this case). The jagged curve $\hat{f}(y_1, y_2) = 0$ is the switching curve that an Adaline controller might use to approximate the teaching controller.

The system has two modes of operation:

i) During the training mode, the teaching controller controls the dynamic system. The adapt logic in the Adaline continuously compares the binary output of the Adaline with that of the teacher. Whenever they differ, the Adaline is adapted in the direction which would make them agree. Because the patterns change rapidly, there may not be time for a full correction. However, the pattern is bound to reoccur, at which time adaption can be continued. During the training mode the Adaline controller "watches" the teacher zero the error after various large disturbances or initial conditions.

ii) During the Adaline control mode, the teaching controller is not used and may be completely removed from the system.

(2) Coding

The choice of codes used to represent the zones of the quantized state variables as partial patterns largely determines how well the Madaline controller will be able to imitate its teacher. Certain linearly separable codes make the classification very easy for the Adaline. Two of these codes are illustrated in Figure 15. A linearly separable code is any code which has a non-singular partial pattern matrix. This matrix has the partial patterns as rows plus an extra column of ones (if necessary). The partial pattern matrix for codes (a) and (b) of Figure 15 are respectively:

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Both matrices are obviously invertible. When linearly separable coding is used, the Adaline will be able to exactly imitate (except for quantization effects) any teacher whose function does not contain cross-product terms, i.e., terms of the form $y_i y_j$, $i \neq j$, regardless of the number of patterns.

The proof that an Adaline using linearly separable coding has such classifying power can be given by demonstrating how the Adaline matches its function to that of the teacher.[13] The analysis of an Adaline is much simpler for classification problems in which a decision is based on the encoded values of state variables than for most problems. The simplification occurs because the Adaline can be fully described in terms of the m state variables instead of the n binary inputs, ($n >> m$ generally). The weights for the inputs used to encode a particular state variable are considered not as separate entities but as a single function of the state variable. The abilities and limitations of a single Adaline in pattern classification and generalization become apparent. Also, the weights can often be calculated in many seemingly complicated problems.

This new interpretation of the Adaline is for analytical purposes only. The Adaline is trained in the usual way. The function matching to be described goes on automatically "inside" the Adaline.

Because each state variable is encoded independently of the others the switching function realized by the Adaline can have no cross terms. Therefore, it and the teacher are limited to functions of the form:

$$f = \sum_{i=1}^{m} f_i(y_i) = 0 \ .$$

Furthermore, because the y_i, and thus the $f_i(y_i)$, can vary independently of each other the individual partial sums, the $\hat{f}_i(y_i)$, of the Adaline must match the corresponding sums, the $f_i(y_i)$, of the teacher. Therefore, a study of the coding needs only to consider how the functions of one variable are matched.

The matching of $\hat{f}_i(y_i)$ and $f_i(y_i)$ places constraints on the weights associated with y_i. If the functions are matched once per quantum zone, there is a constraint for every zone. These constraints can be expressed as a set of linear equations:

$$[A] \quad \vec{w}_i = \vec{f}_i \qquad (4)$$

[A] is the partial pattern matrix described above. If the Adaline is to have a threshold weight, the first column contains the +1's of the threshold inputs. The vector $\vec{w}_i$ contains the weights associated with y_i. [Row of A] $\cdot \vec{w}_i = f_i(y_i)$ is the constraint for one zone. The vector $\vec{f}_i$ contains the values of $f_i(y_i)$ where $f_i(y_i)$ is exactly matched to $\hat{f}_i(y_i)$, i.e., $f_i(y_i) = \hat{f}_i(y_i)$.

When [A] has an inverse, the weights necessary for matching always exist since then

$$\vec{w}_i = [A]^{-1} \vec{f}_i , \qquad (5)$$

regardless of the form of $f_i(y_i)$ (and $\vec{f}_i$). Thus, an invertible partial pattern matrix [A] guarantees that the functions can be matched. There are many possible [A]'s, one for each linearly separable code.

Two possible ways of encoding the state variables are illustrated in Figure 15. The "single spot" code of Figure 15(a) is easy to analyze mathematically because most of the weights would have zero coefficients. If the threshold weight is not used, the weights are:

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \vec{f}_i \qquad (6)$$

The "multi-spot" code of Figure 15(b) is illustrated because it is usually quite easy to inplement, and also because this code usually allows the weights of the Adaline to be quite small. Other authors[5,6] have shown that, in general, the smaller the magnitudes of the weights (after proper normalization) the easier it will be to train the Adaline.

The development leading to linearly separable coding shows that it is sufficient to guarantee that the Adaline function generator be able to imitate a teaching function that has no cross-product terms. By a more involved argument, it can be shown that linearly separable coding is necessary if the Adaline controller is to do this using a minimum number of weights. A proof of necessity is not needed when the non-statistical capacity of an Adaline using linearly separable coding is considered. For instance, when each of the state variables is quantized into n' zones, there are $(n')^m$ possible patterns. The statistical capacity of this Adaline is approximately $2mn'$.

(3) Imitation of Functions with Cross-Product Terms

Previously, it was stated that a single Adaline controller can imitate only teaching functions which do not have cross-product terms. Functions containing cross-products can be realized in two ways. One way would be to encode the desired cross-product terms as additional variables. Another and more satisfactory approach would be to use several Adalines together in a Madaline. Encoding additional variables has the disadvantage in most problems that there will be an extremely large number of possible cross-product terms which may be significant. For generality they should all be encoded. A Madaline does not need a weight for every cross-product term because it can organize its total structure in such a way as to take into account the most significant cross-product terms while ignoring the rest.

The situation illustrated in Figure 16 demonstrates the ability of a Madaline structure to imitate a teaching function with cross-product terms. The teaching function is a rotated ellipse with equation:

$$f(y_1, y_2) = 5y_1^2 - 6y_1y_2 + 5y_2^2 - 2 = 0. \qquad (7)$$

This curve was chosen as a familiar nonlinear function. Two Adalines are used. Adaline I in Figure 16(b) has the U shaped switching line which approximates the switching line of half of the teaching function. Adaline II in Figure 16(c) has the inverted U shaped switching line which approximates the other half of the teaching function. The Adaline outputs are combined in an OR circuit. The logic or the OR circuit is: both Adaline outputs -1 then Madaline output -1 otherwise Madaline output +1. With the polarity of the Adaline outputs as shown on the figure the OR circuit causes the interior of the ellipse to be +1 as desired. The functions approximated by the individual Adalines can be shown to contain no cross terms.

(4) Examples of Adaline Controllers

The application of the ideas of this section to control systems can be readily demonstrated. The switching line of the teaching controller in Figure 14 is that of the well-known minimum time optimum controller for an oscillatory undamped second-order dynamic system.[14] A single Adaline using linearly separable coding is able to imitate the essential features of this highly nonlinear curve.

An Adaline employing adaptive components has been used in the controller of the "broom-balancing machine." The dynamic system being controlled is a cart carrying an inverted pendulum (or "broom"). This is an undamped and inherently unstable fourth-order dynamic system.

The Adaline controller contains one 16 input Adaline. The range of each of the state variables is divided into five approximately equal zones. The state variables are encoded into 4 bit partial patterns using a linearly separable code similar to the one illustrated in Figure 15(b). The controller is taught by having it observe the teacher return the system to the origin of state space after it has received various large disturbances. Training time is usually several minutes, after which, the Adaline is able to take over and balance the "broom."

G. Realization of Adaptive Circuits by Memistors

G. Realization of Adaptive Circuits by Memistors

In large networks of adaptive neurons, it is imperative that the adaptive processes be fully automated. The structure of the Adaline neuron and the adaption procedures used with it are sufficiently simple that it has been possible to develop electronic automatically-adapted neurons which are reliable, contain few parts, and are suitable for mass production. In such neurons it is necessary to be able to store weight values, analog quantities which can be positive or negative, in such a way that these values can be changed electronically.

A new electrochemical circuit element called the Memistor (a resistor with memory) has been devised by B. Widrow and M. E. Hoff for the realization of automatically-adapted Adalines. The Memistor provides a single variable gain element. Each neuron therefore employs a number of Memistors equal to the number of input lines, plus one for the threshold.

A Memistor consists of a conductive substrate with insulated connecting leads, and a metallic anode, all in an electrolytic plating bath. The conductance of the element is reversibly controlled by electroplating. Like the transistor, the Memistor is a 3-terminal element. The conductance between two of the terminals is controlled by the time integral of the current in the third terminal, rather than by its instantaneous value, as in the transistor. Reproducible elements have been made which are continuously variable, which vary in resistance from 50 ohms to 2 ohms, and cover the range in about 15 seconds with several tenths of a milliampere of plating current. Adaptation is accomplished by direct current, while sensing is accomplished nondestructively with alternating current.

Although the Memistor is still an experimental device, it is in limited commercial production. Figure 17 shows a partially fabricated sheet of Memistors, 21 at a time on a common substrate. Each cell has a volume of about 2 drops. The entire unit is encapsulated in epoxy.

The "broom-balancer" has been controlled by an adaptive machine called Madaline I, containing 102 memistors. This machine was constructed a year and a half ago hastily over a one and one half month period. The Memistors were not tested before installation in the machine, and some were defective when first made. A number of wiring errors existed; some weights were adapting to diverge rather than converge. There were a number of short circuits, open circuits, cold solder joints, etc. This machine worked very well when first turned on, and has functioned with very little attention over the past year and a half. After several weeks of experimentation, the individual weights were checked. Twenty-five percent of them were not adapting, yet the machine was able to adapt around these internal flaws and was able to be trained to make very complex pattern discriminations. Self-repairing systems are a very real and vital possibility.

H. Acknowledgments

This work was performed under Office of Naval Research Contract Nonr 225(24), NR 373 360, jointly supported by the U. S. Army Signal Corps, the U. S. Air Force and the U. S. Navy (Office of Naval Research), under Air Force Contract AF33(616)7726 supported by Aeronautical Systems Division, Air Force System Command, Wright-Patterson Air Force Base, and under Contract DA-04-200-AMC-57(Z) supported by the U. S. Army Zeus Project Office, Redstone Arsenal, Huntsville, Alabama.

I. References

1. B. Widrow, "Adaptive Sampled-Data Systems - A Statistical Theory of Adaption," 1959 WESCON Convention Record, Part 4.

2. B. Widrow and M. E. Hoff, "Adaptive Switching Circuits," 1960 WESCON Convention Record, Part IV, pp. 96-104; Aug. 23, 1960.

3. B. Widrow and M. E. Hoff, "Adaptive Switching Circuits," Tech. Rep. No. 1553-1, Stanford Elec. Lab., Stanford, Calif.; June 30, 1960.

4. W. C. Ridgway III, "An Adaptive Logic System with Generalizing Properties," Tech. Rep. No. 1556-1, Stanford Elec. Lab., Stanford, Calif.; April, 1962.

5. C. H. Mays, "Effects of Adaptation Parameters on Convergence Time and Tolerance for Adaptive Threshold Elements," submitted to IEEE Transactions on Circuit Theory.

6. C. H. Mays, "Adaptive Threshold Logic," Tech. Rep. No. 1557-1, Stanford Elec. Lab., Stanford, Calif.; April, 1963.

7. P. R. Low, "Influence of Component Imperfections on Trainable System Performance," Tech. Rep. No. 4654-1, Stanford Elec. Lab., Stanford, Calif.; July, 1963.

8. M. E. Hoff, Jr., "Learning Phenomena in Networks of Adaptive Switching Circuits," Tech. Rep. No. 1554-1, Stanford Elec. Lab., Stanford, Calif.; July, 1962.

9. B. Widrow, "Generalization and Information Storage in Networks of Adaline 'Neurons'," Self-Organizing Systems, pp. 435-461, Spartan Books, Washington, D. C., 1962.

10. L. R. Talbert, G. F. Groner, J. S. Koford, R. J. Brown, P. R. Low, and C. H. Mays, "A Real-Time Adaptive Speech Recognition System," Tech. Rep. No. 6760-1, Stanford Elec. Lab., Stanford, Calif.; May, 1963.

11. J. P. Eagen, "Articulation Testing Methods," The Laryngoscope, Vol. 58, pp. 955-991, 1948.

12. M. J. C. Hu, "A Trainable Weather-Forecasting System," Tech. Rep. No. 6759-1, Stanford Elec. Lab., Stanford, Calif.; June, 1963.

13. B. Widrow and F. W. Smith, "Pattern Recognizing Control Systems," Computer and Information

Sciences (COINS) Symposium Proceedings, Spartan Books, Washington, D. C., 1963.

14. L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko, The Mathematical Theory of Optimal Processes, Interscience Publication, John Wiley and Sons; New York, 1962.

J. Bibliography

N. Abramson and D. Braverman, "Learning to Recognize Patterns in a Random Environment," IRE Trans. on Info. Theory, Vol. IT-8, No. 5, pp. 58-63, Sept., 1962.

G. H. Ball, "An Invariant Input for a Pattern Recognition Machine," Tech. Rep. No. 2003-4, Stanford Elec. Lab., Stanford, Calif.; April, 1962.

D. Braverman, "Learning Filters for Optimum Pattern Recognition," IRE Trans. on Info. Theory, Vol. IT-8, No. 4, p. 279, July, 1962.

H. S. Crafts, "A Magnetic Variable-Gain Component for Adaptive Networks," Tech. Rep. No. 1851-2, Stanford Elec. Lab., Stanford, Calif.; Dec., 1962.

M. Fischler, R. L. Mattson, O. Firschein, and L. D. Healy, "An Approach to General Pattern Recognition," IRE Trans. on Info. Theory, Vol. IT-8, No. 5, pp. 64-73, Sept., 1962.

I. Flores and L. Grey, "Optimization of Reference Signals for Character Recognition Systems," IRE Trans. on Elect. Computers, Vol. EC-9, No. 1, p. 54, March, 1960.

W. H. Highleyman, "Linear Decision Functions with Applications to Pattern Recognition," Proc. IRE, pp. 1501-1514, June, 1962.

P. M. Lewis, "Characteristic Selection Problem in Recognition Systems," IRE Trans. on Info. Theory, Vol. IT-8, No. 2, p. 171, Feb., 1962.

T. Marill, and D. M. Green, "Statistical Recognition Functions and the Design of Pattern Recognizers," IRE Trans. on Elect. Computers, Vol. EC-9, No. 4, p. 472, Dec., 1960.

R. L. Mattson, "A Self-Organizing Logical System," Eastern Joint Computer Conference Convention Record, pp. 212-217, New York, 1959.

R. L. Mattson, "An Approach to Pattern Recognition Using Linear Threshold Devices," Lockheed Missiles and Space Co. Report No. LMSD-702680, Sept., 1960.

R. L. Mattson, and O. Firschein, "Feature Word Constructions for Use with Pattern Recognition Algorithms: An Experimental Study," accepted for publication in the Journal of the ACM.

R. L. Mattson, O. Firschein, and M. Fischler, "An Experimental Investigation of a Class of Pattern Recognition Synthesis Algorithms," IEEE Trans. on Elect. Computers, Vol. EC-14, No. 3, June, 1963.

R. L. Mattson, "The Analysis and Synthesis of Adaptive Systems which use Networks of Threshold Elements," Tech. Rep. No. 1553-3, Stanford Elec. Lab., Stanford, Calif.; Dec., 1962.

W. H. Pierce, "Improving Reliability of Digital Systems by Redundancy and Adaption," Tech. Rep. No. 1552-3, Stanford Elec. Lab., Stanford, Calif.; July 17, 1961.

F. Rosenblatt, Principles of Neurodynamics, Spartan Books, Washington, D. C., 1962.

G. S. Sebestyen, "Recognition of Membership in Classes," IRE Trans. on Info. Theory, Vol. IT-7, No. 1, p. 44, Jan., 1961.

G. S. Sebestyen, Decision Making Processes in Pattern Recognition, Macmillan Co., New York, 1963.

F. B. Smith, Jr., "A Logical Net Mechanization for Time-Optimal Regulation," NASA Technical Note D-1678; Dec., 1962.

M. E. Stevens, "Automatic Character Recognition. A State-of-the-Art Report," National Bureau of Standards Technical Note No. 112, May, 1961.

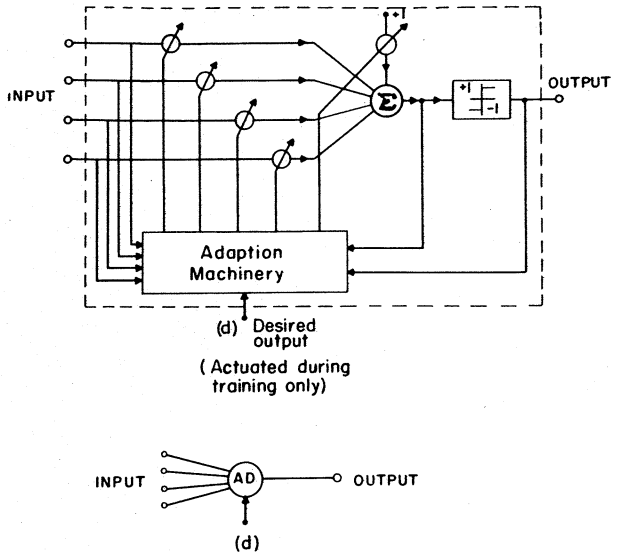
B. Widrow, W. H. Pierce, and J. B. Angell, "Birth, Life, and Death in Microelectronic Systems," IRE Trans. on Mil. Elect.; July, 1961.

B. Widrow," Rate of Adaption in Control Systems," ARS Journal, pp. 1378-1385, Sept., 1962.

B. Widrow, "Reliable, Trainable Networks for Computing and Control," Aerospace Engineering, Vol. 21, No. 9, pp. 78-123, Sept. 9, 1962.

B. Widrow, "Pattern Recognition and Adaptive Control," Joint Automatic Control Conference Proceedings, June, 1962.

M. C. Yovits, G. T. Jacobi, and G. D. Goldstein, Self-Organizing Systems 1962, Spartan Books, Washington, D. C., 1962.



ADALINE (Adaptive Linear " neuron")

Figure 1 An Automatically-Adapted Threshold Element

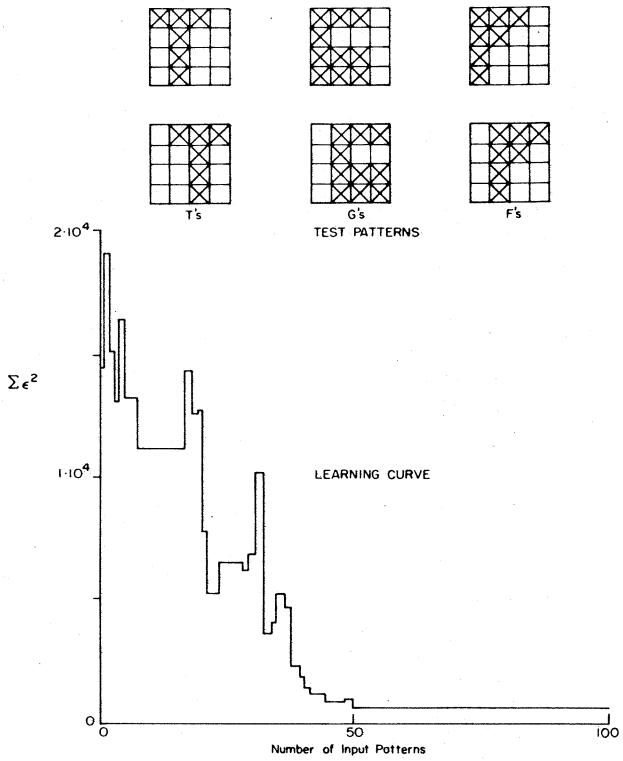


Figure 3 Measurement of Rate of Adaption

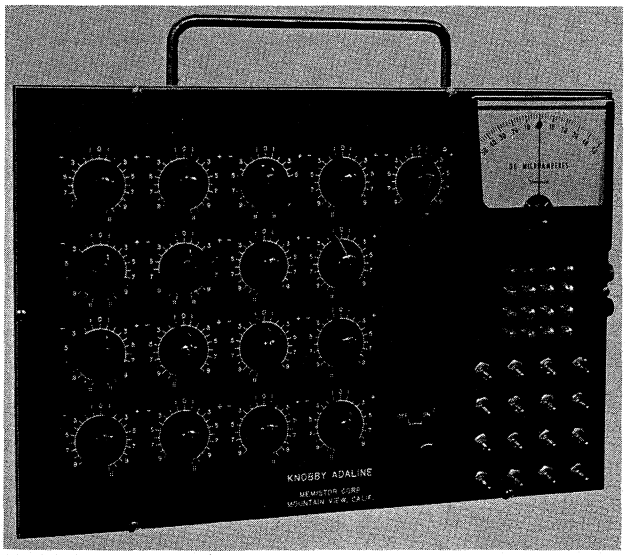


Figure 2 An Elementary Learning Machine

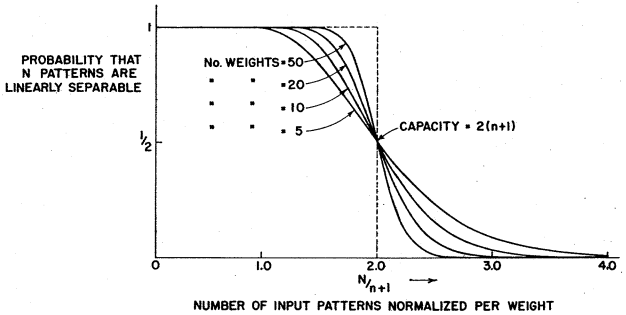
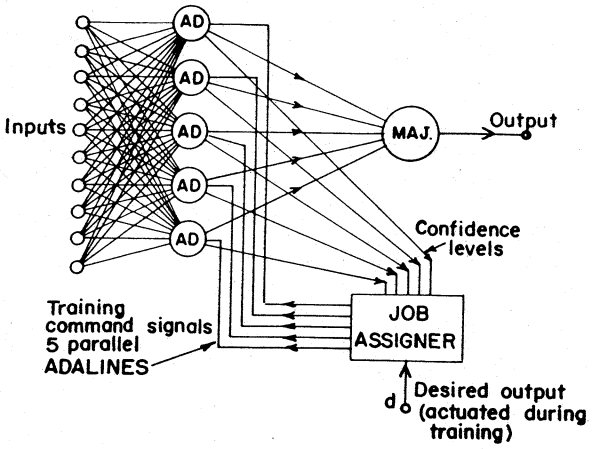


Figure 4 Adaline Memory Capacity Curves



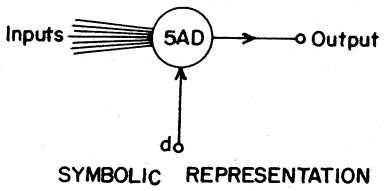


Figure 5 Configuration of Madaline

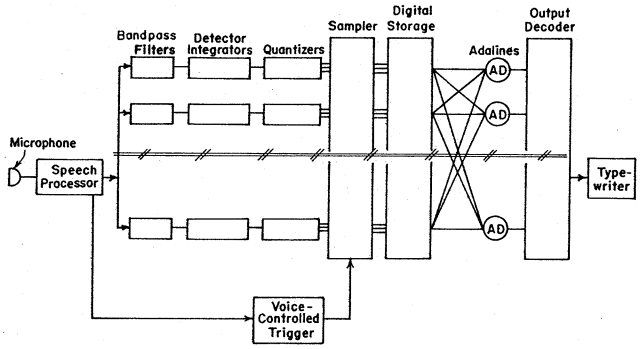


Figure 7 An Adaptive Speech System

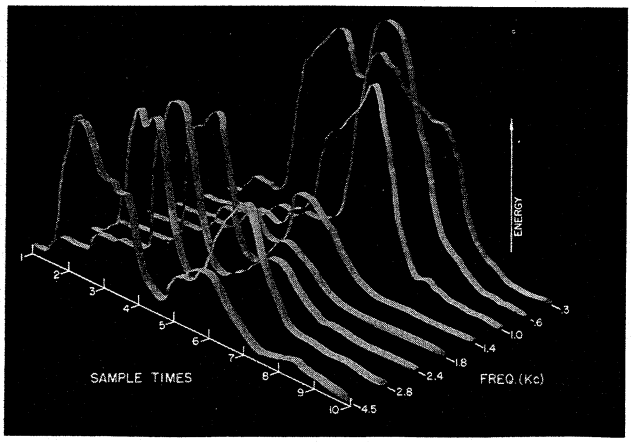


Figure 6 Spectral Structure of the Spoken Word "Zero"

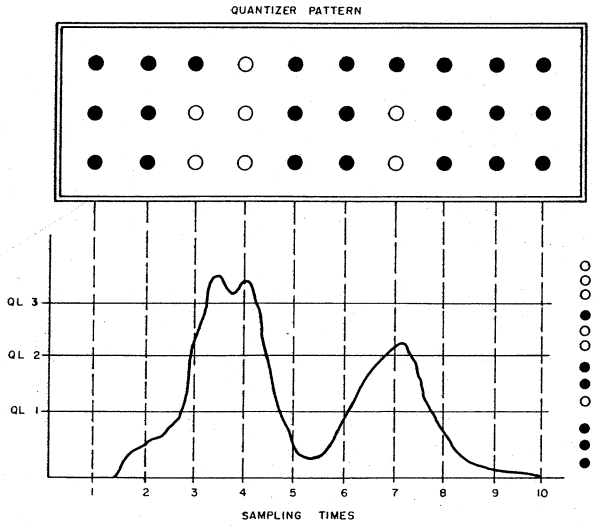


Figure 8 The Formation of a Digital Speech Pattern

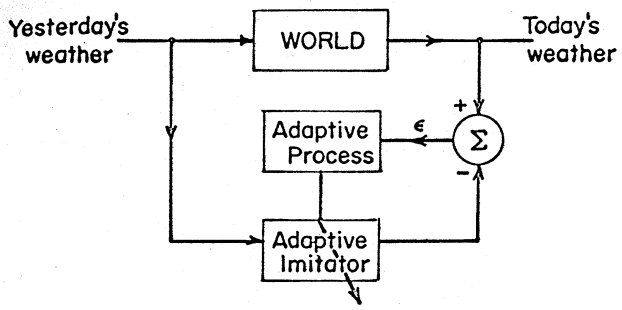


Figure 9 An Adaptive Weather Imitator

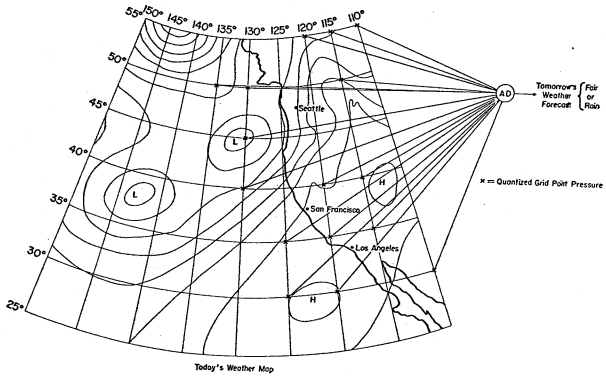


Figure 10 Training an Adaline to Read Weather Maps

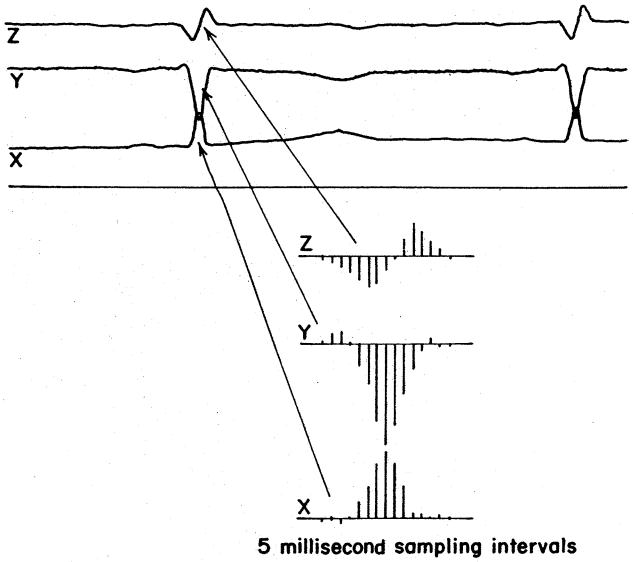


Figure 11 A Typical Normal Vectorcardiogram - (upper) complete; (lower) sampled QRS complex

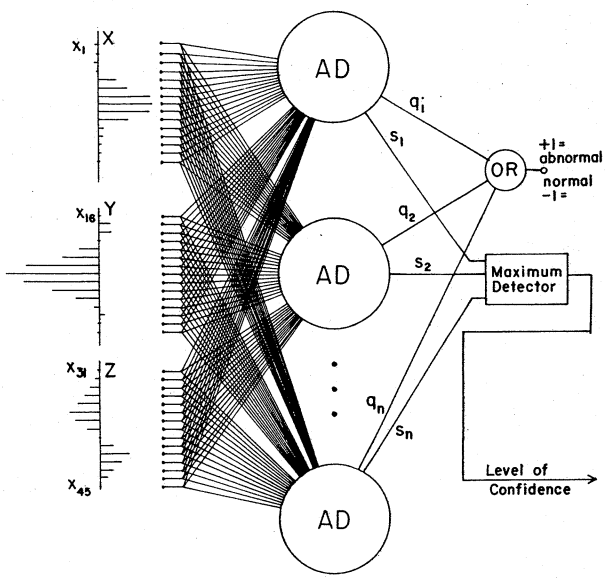


Figure 12 A Madaline Structure for EKG Diagnosis

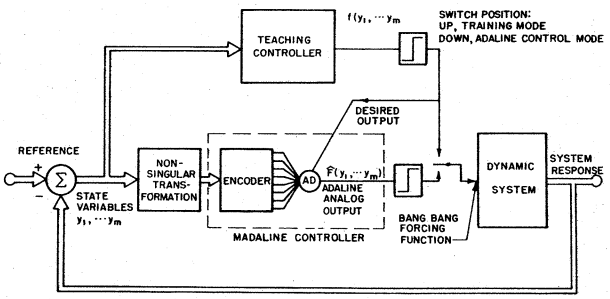


Figure 13 Block Diagram of a Dynamic System

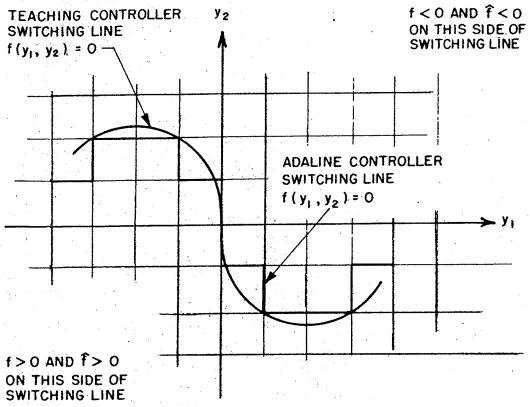


Figure 14 Quantized State Space

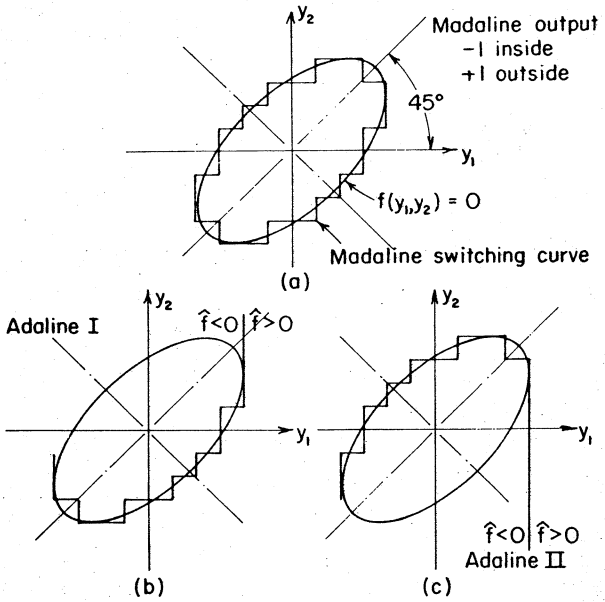


Figure 16 Rotated Ellipse with Madaline Realization

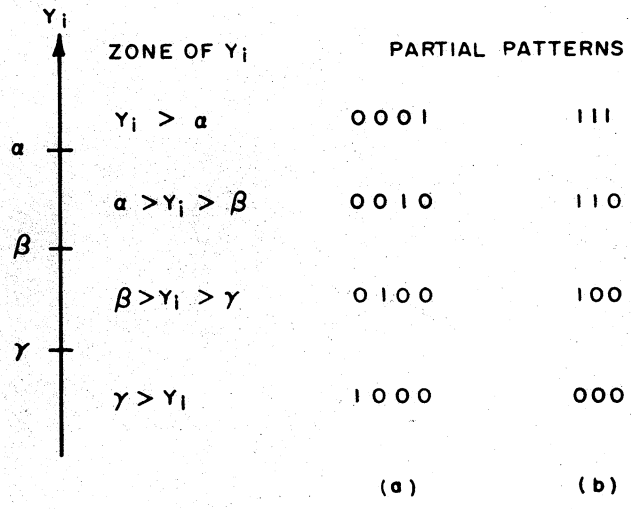


Figure 15 Possible "Linearly Separable Codes"

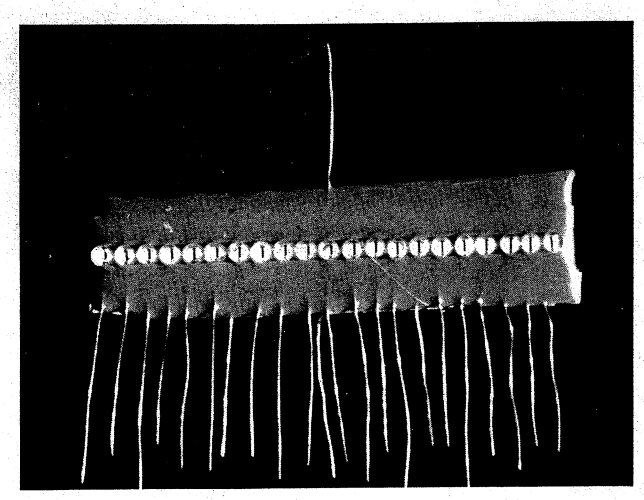


Figure 17 A Partially-Fabricated Sheet of Memistors

1963 WESTERN ELECTRONIC SHOW AND CONVENTION/SAN FRANCISCO, AUGUST 20-23

PRACTICAL APPLICATIONS FOR ADAPTIVE DATA-PROCESSING SYSTEMS

By B. Widrow,* G. F. Groner,* M. J. C. Hu,* F. W. Smith,*
D. F. Specht,* L. R. Talbert*

11.4