# Connected-digit speaker-dependent speech recognition using a neural network with time-delayed connections

**3 authors**, including:

**K.P. Unnikrishnan**
NorthShore University HealthSystem
**46** PUBLICATIONS **1,032** CITATIONS

**John J Hopfield**
Princeton University
**223** PUBLICATIONS **47,129** CITATIONS

# Connected-Digit Speaker-Dependent Speech Recognition Using a Neural Network with Time-Delayed Connections

K. P. Unnikrishnan, John J. Hopfield, and David W. Tank

*Abstract*—An analog neural network that can be taught to recognize stimulus sequences has been used to recognize the digits in connected speech. The circuit computes in the analog domain, using linear circuits for signal filtering and nonlinear circuits for simple decisions, feature extraction, and noise suppression. An analog perceptron learning rule is used to organize the subset of connections used in the circuit that are specific to the chosen vocabulary. Computer simulations of the learning algorithm and circuit demonstrate recognition scores >99% for a single speaker connected-digit data base. There is no clock; the circuit is data driven, and there is no necessity for endpoint detection or segmentation of the speech signal during recognition. Training in the presence of noise provides noise immunity up to the trained level. For the speech problem studied here, the circuit connections need only be accurate to about 3-b digitization depth for optimum performance. The algorithm used maps efficiently onto analog neural network hardware: single chip microelectronic circuits based upon this algorithm can probably be built with current technology.

## I. INTRODUCTION

A UTOMATIC recognition of words in connected speech can be considered a statistical pattern recognition problem. Both dynamic time warping (DTW) and hidden Markov models (HMM) have been frequently used as algorithms for solving this pattern matching problem. In DTW, a dynamic programming local minimization routine is used to best match a given set of data observations in the input speech waveform against a set of stored templates [1], [2]. The templates represent words in the vocabulary and are produced from a training set of speech utterances. The input observations are temporally distorted to find the best match with each of the templates. That template with the best overall match is identified as the spoken word. In a common form of HMM, each word in the vocabulary has an associated Markov model [3]-[5]. The probabilities of data observations and the transition probabilities between hidden states are determined from a training set by a reestimation algorithm. In use, each word model is checked for the probability that it could have produced a given speech utterance to be recognized. The model with the highest probability is assigned as the spoken word.

K. P. Unnikrishnan was with the Molecular Biophysics Research Department, AT&T Bell Laboratories, Murray Hill, NJ. He is now with GM Research Laboratories, Warren, MI 48090.
J. J. Hopfield was with the Molecular Biophysics Research Department, AT&T Bell Laboratories, Murray Hill, NJ. He is now with the Divisions of Chemistry and Biology, California Institute of Technology, Pasadena, CA 91125.
D. W. Tank is with the Molecular Biophysics Research Department, AT&T Bell Laboratories, Murray Hill, NJ 07974.
IEEE Log Number 9041590.

We have applied a different statistical pattern recognition algorithm that is efficiently implemented in neural network architectures to a small vocabulary connected speech recognition problem. The work we present combines our earlier analog network with time-delayed connections for sequence recognition [6], [7] with an analog perceptron learning rule [8]. The algorithm and its method of implementation is distinct from DTW and HMM approaches but it shares several attributes: 1) the speech signal is coded by a series of features which describe the short-time spectral characteristics, 2) the parameters of the recognition system are determined from a training set by a learning algorithm (analogous to a reestimation formula), and 3) spoken words are determined from parameters related to probabilities computed by the algorithm in a maximum likelihood framework. A variety of other algorithms for neural network architectures have been investigated for speech recognition (for example, see [9]-[12]), and these will be discussed later in connection with our results.

In general, the numerical precision necessary for the implementation of both DTW and HMM algorithms is high. Real-world speech recognition systems based upon these algorithms are typically implemented using digital signal processing hardware because of the necessity to perform high-accuracy calculations. Connectionist architectures emphasize the efficient use of low precision parallel hardware. The algorithm presented here is demonstrated to work with this form of low precision hardware.

Although we have only studied one of the simpler speech recognition problems, the method and form of statistical acoustic signal processing used in the network described here may be applicable to a broader question in human speech perception. A complete theory of human speech perception must describe how its algorithms can be implemented in the hardware of neurobiology. The algorithm presented here can be implemented in a "connectionist" network of analog processing elements fashioned after neurons and synapses [13]. It is difficult to envision how HMM and DTW, in their common forms, could have such a neurobiological implementation, although preliminary attempts to map these algorithms onto high-precision connectionist architectures have been made [14].

## II. AN ANALOG NETWORK FOR A SIMPLE TONE SEQUENCE RECOGNITION PROBLEM

The organization of the speech recognition neural network can be understood by examining a simpler circuit with few components that illustrates most of the computational features present in the speech circuit. Consider the problem of distinguishing simple sequences of 4 tones each having duration $\tau$. Imagine that we have 6 possible tones labelled $T_1$ through $T_6$ and for
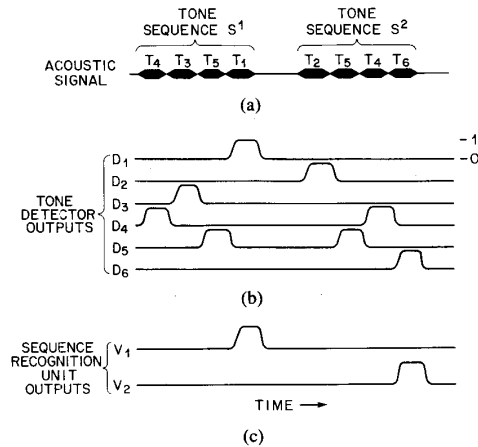
Fig. 1. (a) Two tone sequences to be recognized by a simple analog neural network. (b) The outputs versus time of tone detectors for an acoustic signal illustrated in (a). Each detector $D_i(t)$ is tuned to tone $T_i$. (c) The desired outputs of the two sequence recognition units that are to become active when their corresponding tone sequences are present in the acoustic data stream.
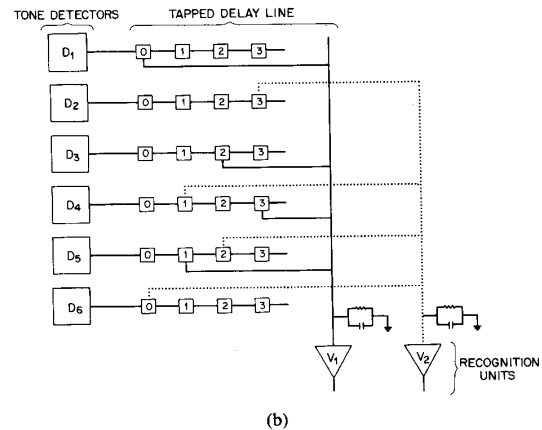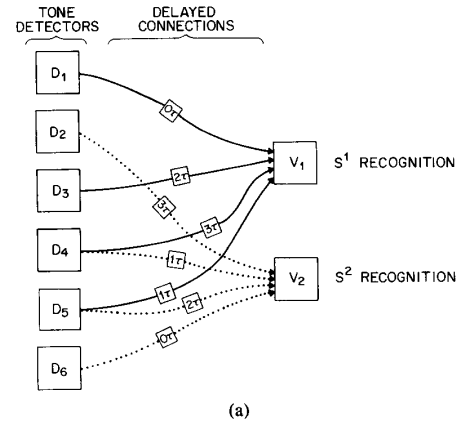


Fig. 2. (a) A symbolic diagram of the organization of a network to solve the tone sequence recognition problem. Detector outputs are connected to recognition units with a pattern of time-delayed connections appropriate to the tone sequence to be recognized. (b) How the tone sequence recognition circuit could be constructed as a simple network of electrical circuit elements. The recognition units are analog artificial neurons connected to appropriate individual taps of tapped delay lines that occur on the outputs of each tone detector.

each of these tones there is a corresponding detection device that has an output $D_i(t) = 1$ if tone $T_i$ is present at time $t$ and an output of 0 otherwise. A tone sequence is the serial presentation of individual tones in the input data stream. For example, the sequence $S^1$ might be $T_4 T_3 T_5 T_1$, meaning tone 4 is followed by tone 3 which is followed by tone 5, etc., while a second sequence $S^2$ might be $T_2 T_5 T_4 T_6$ (Fig. 1(a)). When each of these two sequences occurs in the input data stream to the circuit, the outputs of the tone detectors will have the patterns illustrated in Fig. 1(b).

Now consider the analog network shown schematically in Fig. 2(a) and as an electronic circuit in Fig. 2(b). Each of the two sequence recognition units corresponds to one of the two tone sequences we want to recognize and is supposed to produce appreciable output only when its corresponding sequence is recognized. Thus the two outputs of the sequence recognition units are sequence "spotters." The sequence detector units are nonlinear threshold elements having an output $V_i$ that is related to their input potential $u_i$ by a nonlinear gain function $V_i(t) = g(u_i(t))$ with a minimum of $V_i = 0$ and a maximum of $V_i = 1$. (More generally, in a noisy or ambiguous situation we might want the analog value of the output to represent the probability that the corresponding sequence occurred in the data stream. See Appendix A.) The desired outputs of the recognition units as a function of time during presentation of tone sequences $S_1$ and $S_2$ are drawn in Fig. 1(c).

The output of each of the 6 tone detectors is conected to a set of four boxes, (Fig. 2(a) or taps in a delay line shown in Fig. 2(b)) which delay the signal by amounts $3\tau$, $2\tau$, $1\tau$, and $0\tau$, respectively. Getting the sequence recognition outputs to behave in the desired fashion is simply a matter of choosing the right pattern of resistive connections from the outputs of the 6 × 4 time delay boxes to the inputs of the recognition units. For sequence detector 1, we use a resistor with resistance $R$ from the box whose output is the signal from tone detector $T_4$ delayed by $3\tau$. Similarly, tone detector $T_3$ is used with delay $2\tau$, a connection with delay $1\tau$ from $T_5$ and a connection with no delay from $T_1$. Similarly constructed time-delayed connections, appropriate for sequence $S^2$ connect to the input of sequence de-

tector 2. These connections result in a big input potential to the correct sequence detection unit when one of the sequences in our "vocabulary" is presented to the network. For a random sequence, there will typically be only one fourth as much input potential to the detection units and they will stay below threshold. (Choosing the connections is analogous to the electrical engineering problem of a matched filter [15].) The ability of the circuit to spot particular sequences is the result of the specific pattern of time-delayed connections.

## III. GENERALIZING THESE IDEAS TO SPEECH-LIKE SOUNDS

The same tone may be present at two different time points in the same sequence. If $S^1$ were the tone sequence $T_2 T_5 T_4 T_2$, then we would construct connections both with $3\tau$ delay and zero delay between the output of $D_2(t)$ and the input to the sequence detection unit "1." The temporal effect (or impulse response) of any individual detector on the input to a sequence recognition unit may be quite complex. But a large input to the sequence detection unit only occurs when all the signals add coherently.

Nothing in the discussion requires having only one tone present at a given time. If multiple tones are present at the same temporal position (a "chord") within the sequence to be recognized, then the outputs of each of the tone detectors present in that chord will have connections with the same time delays to the input of the corresponding sequence detection unit.

By increasing the complexity of the tone sequence, it can be made more analogous to speech. The natural variability of speech produces a degree of unreliability in the pattern of "tone detectors" being activated by a given sequence class (i.e., word). It is then appropriate to think of the statistical distribution of tone detector activation during a sequence to be recognized. For example, if two different tones are equally likely during the first time period of one of our tone sequence classes, then equal-valued time-delayed connections should be made to the sequence detector input. Since repeated tones are allowed, this general scheme applies also to the case of intervals having unequal lengths.

The sequences do not need to be "orthogonal." For example, two sequence detectors may share a time-delayed detector output, having the same tone at identical points in their two sequences. One might ask if this time-delayed connection is really necessary, since both of the corresponding sequence detection units will have the same connection. The answer to this depends upon the nature of the rest of the vocabulary the circuit will be exposed to. (The rest of the vocabulary should be considered the "noise model" in the matched filter analogy.) If we are expecting to see only those two sequences in the data stream (and are thus merely to make a binary choice between two alternatives), then the output from that detector for this time delay to the sequence detection units will superfluous; its signal provides no information about the choice to be made. Thus the connection strength may be set to zero with no loss of performance. If, however, other sequences do not contain this tone at this temporal location (a multiple discrimination is to be made), then these two connections still convey information, namely that these two sequences are different from the other sequences. Negative connections can also be used to advantage in sharpening the selectivity of the sequence "matched filters." They can be made to signals which are absent in a particular sequence, and will decrease the input to a recognition unit for incorrect sequences. Negative connections can be made to taps for which an output would provide information against the presence of the associated sequence in the data stream. In general, the analog value of the connections should be related to the quantity of information present at each particular time-delayed output for the speech sequence discrimination. These points, illustrated in the example above, are central to the more difficult problem that is the focus of this paper.

## IV. PREPROCESSING AND FEATURE EXTRACTION: MOVING FROM TONES TO FORMANTS

The circuit we have studied for speech recognition is similar to the tone sequence recognition network discussed above. Its general organization is the same as that shown in Fig. 2(b) except that more complex detectors, time delays, and recognition units are used. A block diagram of this circuit is shown in Fig. 3.

In the set of experiments reported in this paper, we have attempted to recognize spoken digits in connected strings at their respective temporal end points. Fig. 4(a) shows the waveform of an exemplar digit string where the spoken digits are
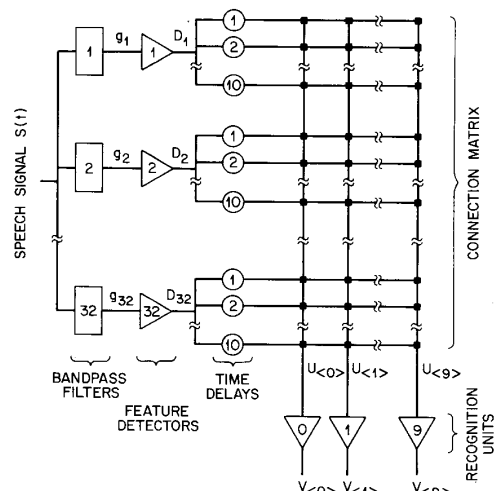


Fig. 3. A block diagram of the speech recognition circuit. The circuitry for bandpass filters, feature detectors, time delays, and the recognition units are fixed. The strength of connections between the output of time delays and the input to recognition units are assigned through a learning algorithm.
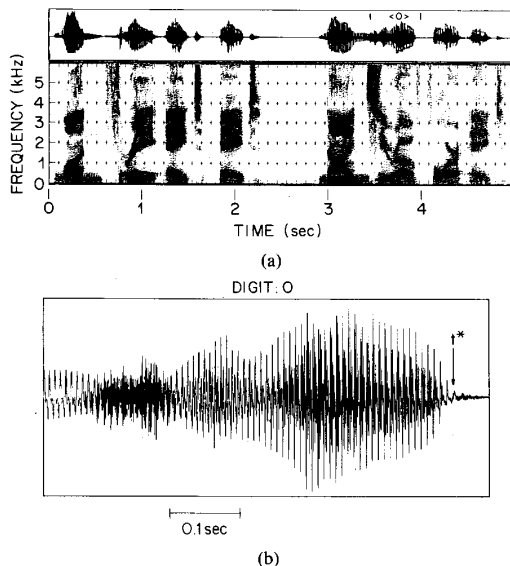


Fig. 4. (a) Speech waveform and spectrogram of a digit string used in the experiment. The digits spoken are ⟨ 1 3 8 8 9 0 4 8 ⟩. The speech data was collected at a rate of 12 000 samples/s. (b) Waveform of the digit ⟨ 0 ⟩ segmented out from the string in (a). Its temporal endpoint to be used for training is marked as $t^*$.

⟨ 1 3 8 8 9 0 4 8 ⟩. We have used detection of features in spectral energy for the recognition. The spectrogram below the waveform shows some of these features. Fig. 4(b) shows the waveform of digit ⟨ 0 ⟩ segmented out of the string and its visually identified temporal end point $t^*$.

In our network the speech signal is transformed into a temporal sequence of activation of acoustic feature detectors, analogous to activation of tone detectors in a tone sequence. A patterned set of time-delayed connections is constructed (as

discussed in the next section) which temporally organizes these features and determine the states of a network of sequence(word) recognition units connected together as an $n$-flop. The feature set used in the present work was determined by the locations of energy maxima in the short-time frequency spectrum of the acoustic signal. But the methods used are more general and can be applied to any set of computable acoustic or acoustic-phonetic features. (See the discussion.)

The selection of energy maxima and the corresponding activation of detector outputs was computed as follows. The normalized speech waveform was filtered by a set of thirty-two 2-pole bandpass filters. The center frequencies $f_n$ of the filters were chosen to be between $f_{min}$ = 200 Hz and $f_{max}$ = 4 kHz distributed according to the relation log $[1 + (f(\text{Hz}))/1000]$ (Mel scale).

The filtered outputs $h_n(t)$ were full-wave rectified and integrated with a time-constant $\tau_{filt}$ = 10 ms. Thus the rectified, filtered, and integrated outputs $g_n$; $n$ = 1, 32 were calculated from integration of the equation

$$\frac{dg_n}{dt} = -\frac{g_n}{\tau_{filt}} + |h_n(t)|. \qquad (1)$$

The set $g_n(t)$ for the word $\langle 0 \rangle$ is shown in Fig. 5(a). The time $t^*$ at which the word ends is also indicated in the figure.

The outputs $g_n$ of the filterbank-integrator network are analog signals corresponding to the rms amplitude in different frequency bands. Nonlinear signal processing and feature extraction was done by a second network for which the output signals $D_n$ represent the occurrence or absence of feature $n$. Feature $n$ was chosen to be a peak in the frequency distribution of the acoustic signal power occurring at frequency $f_n$:

$D_n(t)$ = 1 if

$$g_n(t) \geq \theta_1 \qquad (2a)$$

and

$$\frac{2g_n(t) - [g_{n+1}(t) + g_{n-1}(t)]}{g_n(t)} \geq \theta_2 \qquad (2b)$$

i.e.,

$$(2 - \theta_2)g_n(t) - [g_{n+1}(t) + g_{n-1}(t)] \geq 0 \qquad (2c)$$

$D_n(t)$ = 0    otherwise.

Thus detector $n$ is active at time $t$ with output $D_n(t)$ = 1 if two criteria are met: 1) the rectified and integrated output $g_n$ of the $n$th bandpass filter is greater than a threshold $\theta_1$ and 2) a discrete approximation of the second log derivative of $g_n$ with respect to $n$ is greater than a second threshold $\theta_2$. In our simulations, we have used $\theta_1$ = 150 and $\theta_2$ = 0.05. The maximum value of filter output was about 5500 and $\theta_1$ was chosen so that spurious speech signals do not turn on the detectors. $\theta_2$ was chosen to have a small nonzero value.

The detector outputs $D_n(t)$ computed for the presentation of the acoustic signal in the word $\langle 0 \rangle$ are shown in Fig. 5(b). Relatively invariant features not easily discernable across the bandpass filter outputs are easily seen in the $D_n(t)$ outputs. The nonlinear processing that occurs in this second network tends to suppress "unformant-like" noise and accentuate "formant-like" features. This representation is very good for discriminating vowel-like sounds, but is not particularly effective for discriminating consonants. (A low-order LPC represenation has similar characteristics.)

The operation of these feature detectors is similar to those of the "center-surround" cells found in the mammalian auditory system or visual system [16]–[18]. Fig. 5(c) shows a schematic diagram of a center-surround cell with threshold. The circuitry for this computation can be constructed from standard designs for artificial neural circuits. The net effect of the center-surround inhibitory network applied to the filter output is to increase the signal to noise of the features we have chosen to select (formant locations) and convert the inputs to the time delay network into a simple 0, 1 (binary) form.

## V. CONSTRUCTING THE DELAY CONNECTIONS

The outputs of detector $D_n(t)$ are followed by a set of time delays which delay the signal from each channel and also convolve it with a broadening function. If a connection is formed, the signal $x_n(m, t)$ that is available to a recognition unit at any particular delay tap $[n, m]$ corresponding to detector frequency $f_n$ and mean time delay $(m)\tau_{delay}$ is

$$x_n(m, t) = \int_0^\infty K(m, t')D_n(t - t') \, dt' \qquad (3)$$

where the delay kernel $K(m, t')$ is a maximum for $t'$ = $(m)\tau_{delay}$. The tap interval $\tau_{delay}$ was usually taken as 0.06 s. Most of the work described here used a kernel of the form

$$K(m, t') = \frac{1}{\sqrt{2\pi}\sigma} \exp \frac{-(t' - (m)\tau_{delay})^2}{2\sigma^2} \qquad (4)$$

with $\sigma^2 = \sigma_0^2(n - 4)^2$ and $\sigma_0$ = .015 s. (Rigid delays correspond to $\sigma_0$ = 0.) The impulse response functions of these time delays are shown in Fig. 6.

Connections are made from the time delay outputs to the input of a recognition unit, each providing a weighted contribution from its associated time delay. The summed signal from all delays is filtered with a time constant $\tau_{rec}$ at the input of each recognition unit. The input potential $u_i(t)$ of recognition unit $i$ is given by

$$\frac{du_i}{dt} = \frac{-u_i}{\tau_{rec}} + \left[ \sum_{n,m} T_{i,n,m}x_n(m, t) + q_i \right] \qquad (5)$$

where $T_{i,n,m}$ is the magnitude of the connection from the output of the $m$th delay for the $n$th feature detector to the input of recognition unit $i$. $q_n$ represents the constant input bias current that determines the threshold for each recognition unit. For purposes of describing the learning algorithm used to determine the values of $T_{i,n,m}$ and $q_i$, it is convenient to describe this input potential in a different form. Since convolution and filtering are linear operations, the input potential to recognition unit $i$ at time $t$ can be written as a sum of terms

$$u_i(t) = \sum_{n,m} T_{i,n,m}X(m, t) + Q_i \qquad (6a)$$

where

$$X_n(m, t) = \int_0^\infty x_n(m, t - t') \exp \left( \frac{-t'}{\tau_{rec}} \right) dt' \qquad (6b)$$

and

$$Q_i = \int_0^\infty q_i \exp \frac{-t'}{\tau_{rec}} dt'. \qquad (6c)$$

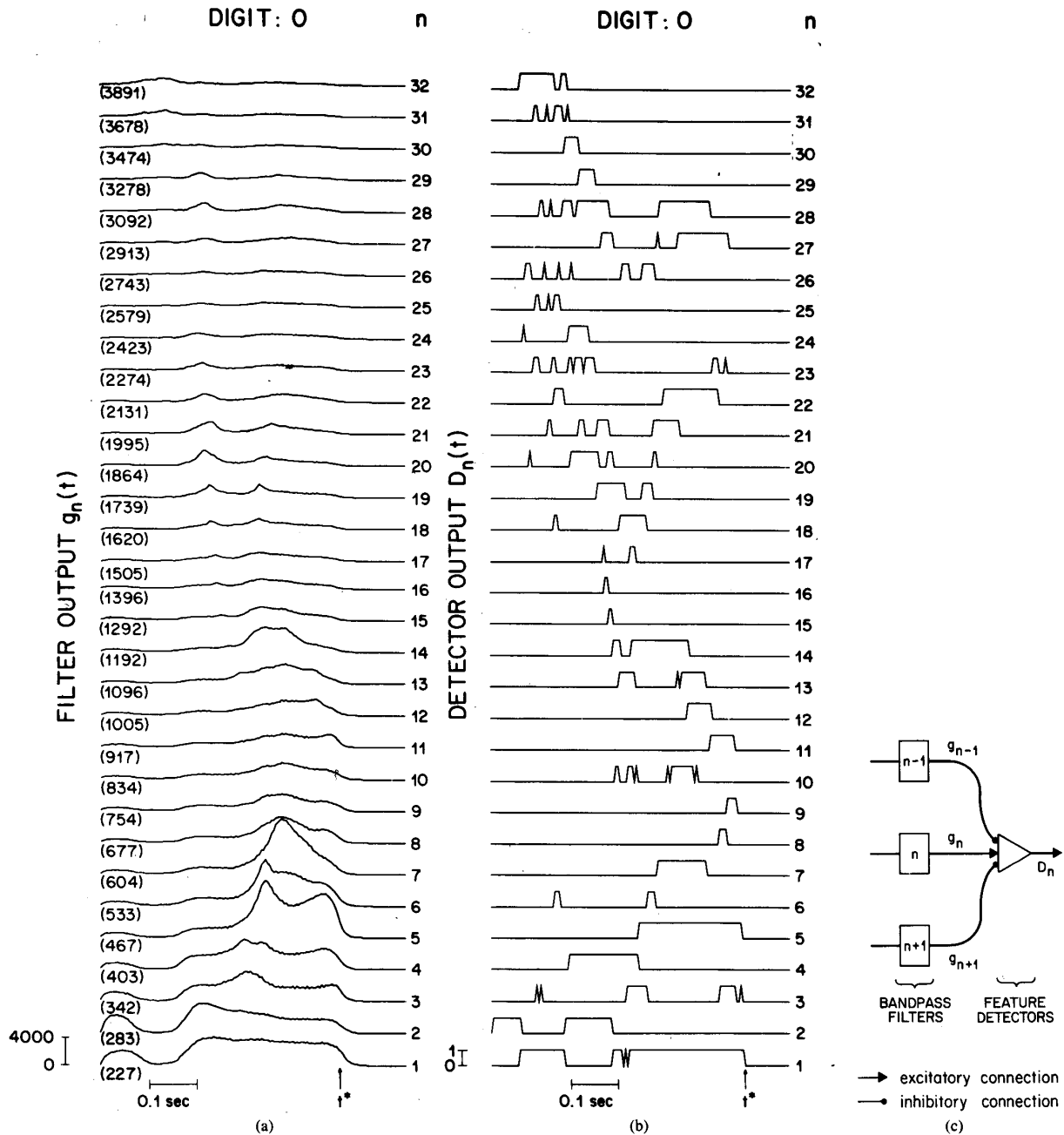We have used $\tau_{rec}$ = 0.04 s in most of the experiments.

Fig. 5. (a) Rectified and integrated outputs from the 32 bandpass filters for the exemplar of ⟨ 0 ⟩ shown in Fig. 3(b). The center frequencies of filters are given in parentheses. (b) Outputs of the 32 center-surround detectors for the same exemplar of ⟨ 0 ⟩. The two thresholds used in the detectors were $\theta_1 = 150$ and $\theta_2 = 0.05$. The maximum vale of filter output for this utterance was about 5500. (c) Schematic diagram of the "center-surround" mechanism used for feature detection.

Given the input potential $u_i(t)$ determined as above, the output $V_i$ of the recognition unit for word $i$ is defined as

$$V_i(t) = \tfrac{1}{2}\left[1 + \tanh\left(u_i(t)\right)\right]. \tag{7}$$

Given a set of connection strengths between the time delayed outputs and the inputs to the recognition units (which we will refer to as the time delayed connections), the above circuit equations define the output of the word recognition units for any input signal. As in the simple tone sequence circuit described earlier, we desire the output of a word recognition unit to be near 1 when that word has just occurred in the input signal and be near 0 otherwise. More specifically, the duration of an utterance is divided into nine intervals, defined by ten equally
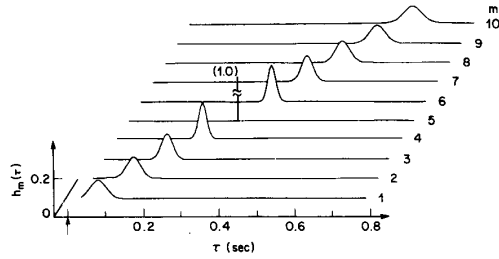
Fig. 6. Impulse response functions of time delays used in the speech recognition circuit. The delays are spaced with a fixed time of 0.06 s. They have Gaussian profiles with widths proportional to the distance from the middle delay. Time delay 1 is "no-delay" and time delay 10 is 0.54 s. The arrow on the x axis shows the impulse time at $t = 0$ s.
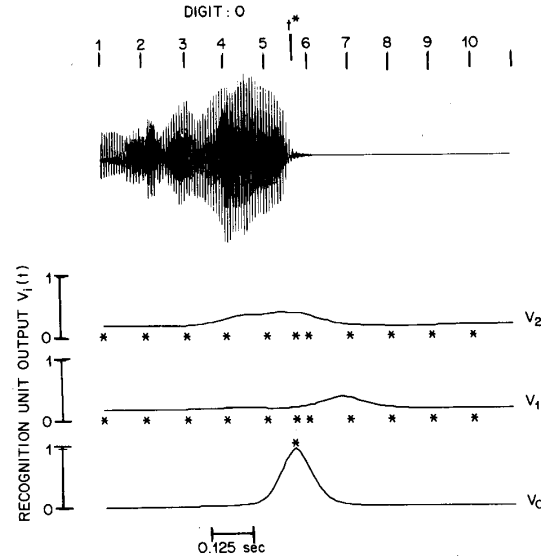


Fig. 7. Waveform of the exemplar of $\langle 0 \rangle$ from Fig. 4(b) and the outputs of recognizers $\langle 0 \rangle$, $\langle 1 \rangle$, and $\langle 2 \rangle$ after 10 learning cycles. Learning takes place at 10 time points 125 ms apart and at the temporal end point of the word, marked as $t^*$. The targets for learning at each of these time points are marked with asterisks.

spaced time points. The endpoint $t_i^*$ of each exemplar utterance lies within this interval. Target goals for the word recognition unit outputs are defined by the following rules, illustrated by a particular example for the recognition unit $\langle 0 \rangle$.

1) For word recognition unit "zero", for each "nonzero" utterance $k$ at its ten equally spaced time points and its endpoint $t_k^*$ the appropriate target output is 0.

2) For word recognition unit "zero", for each utterance $i$ which is an example of "zero", at the time $t_i^*$ the appropriate target output is 1.

This is illustrated in Fig. 7. It shows the waveform of the exemplar of $\langle 0 \rangle$ from Fig. 3 along with its endpoint. The 10 equally spaced time points are 125 ms apart. (This time window was chosen to be able to train on all exemplars in the vocabulary.) The figure also shows the targets at each of these time points for word recognition units $\langle 0 \rangle$, $\langle 1 \rangle$, and $\langle 2 \rangle$ for this exemplar of $\langle 0 \rangle$ along with their actual outputs after 10 learning cycles.

Given these target outputs and a set of training utterances, an analog perceptron learning algorithm was used for determining the optimal connections $T_{i,n,m}$ for yielding the desired responses. Beginning with all $T_{i,n,m} = 0$, the connections were iteratively updated by the following algorithm. It is based on minimizing an entropic measure of error and is described in more detail in Appendix A.

$$\Delta T_{i,n,m} = \epsilon_1 \cdot \sum_{examples} \sum_{time\ points} (\text{Target}^{ex} - V_i^{ex}) \cdot X_n^{ex}(m, t)$$

(8a)

where $\Delta T_{i,n,m}$ is the change for connection $T_{i,n,m}$, $\epsilon_1$ is a learning coefficient. For an example ex, $X_n^{ex}(m, t)$ refers to the convolved output of time delays and $V_i^{ex}$ refers to the actual output of the $i$th recognition unit at the ten learning points. The targets at these learning points in time is either 1 or 0, set according to the rules mentioned above. The adjustable threshold $q_i$ for each word recognition unit is also updated as follows:

$$\Delta q_i = \epsilon_2 \cdot \sum_{examples} \sum_{time\ points} (\text{Target}^{ex} - V_i^{ex}) \qquad (8b)$$

where $\Delta q_i$ is the change for theshold $q_i$. If the learning coefficients $\epsilon_1$ and $\epsilon_2$ are small, this algorithm is a gradient descent optimization procedure. (See Appendix A for details.)

The function $X_n(m, t)$ is obtained by convolving and filtering the output of quantizer channel $n$ from times prior to $t$, and does not depend on the strengths of the connections. As a result, the problem of finding appropriate connections at a particular time has been transformed into the form of a simple perceptron learn-

ing problem. For a recogntion unit driven by connections from a set of inputs (no hidden units), the learning problem is convex (see Appendix A) and there is a unique set of best connection weights defined by any sufficiently large set of training data.

When the circuit is used for recognition after training, an utterance which is not spoken clearly may not be completely reliably identifiable as a particular spoken word. Also, the output $D_n(t)$ of the detector contains far less information than was present in the initial data stream, and this output may be ambiguous even if the original sound was not. In order to give meaning to the output $V_i(t)$ of a recognition unit for a particular word under these circumstances, a procedure was used which trains each recognition unit to evaluate the probability that its particular word was the actual sound which just occurred (see Appendix A). When the optimum set of connections is found, the value of the output of a word-recognition unit during an utterance represents the network estimate of the probability that this particular word has just been completed. For scoring the results we convert these various probabilities to "yes or no" answers to compare with what was actually said (see results section). Thus the decision of what word was spoken is computed by the network in a maximum likelihood framework (see also [6]).

When there are $N_{ex}$ examples of each digit, and $N_{tp}$ equally spaced time points chosen for training, each word recognition unit would then have $(N_{tp} + 1) \cdot (10 - 1) \cdot (N_{ex})$ negative training examples, and $N_{ex}$ positive training examples. The utterances used in training can be either isolated words, or words clipped from continuous speech, which may also deliberately include leading or trailing parts of other words. The amount of computation necessary to establish a set of connections using 288 training examples is about 10 min on an Alliant FX-8 computer with vectorized Fortran.

The learning algorithm establishes delayed connections whose numerical value is related to the amount of information that the given detector channel provides at that delay for the given word
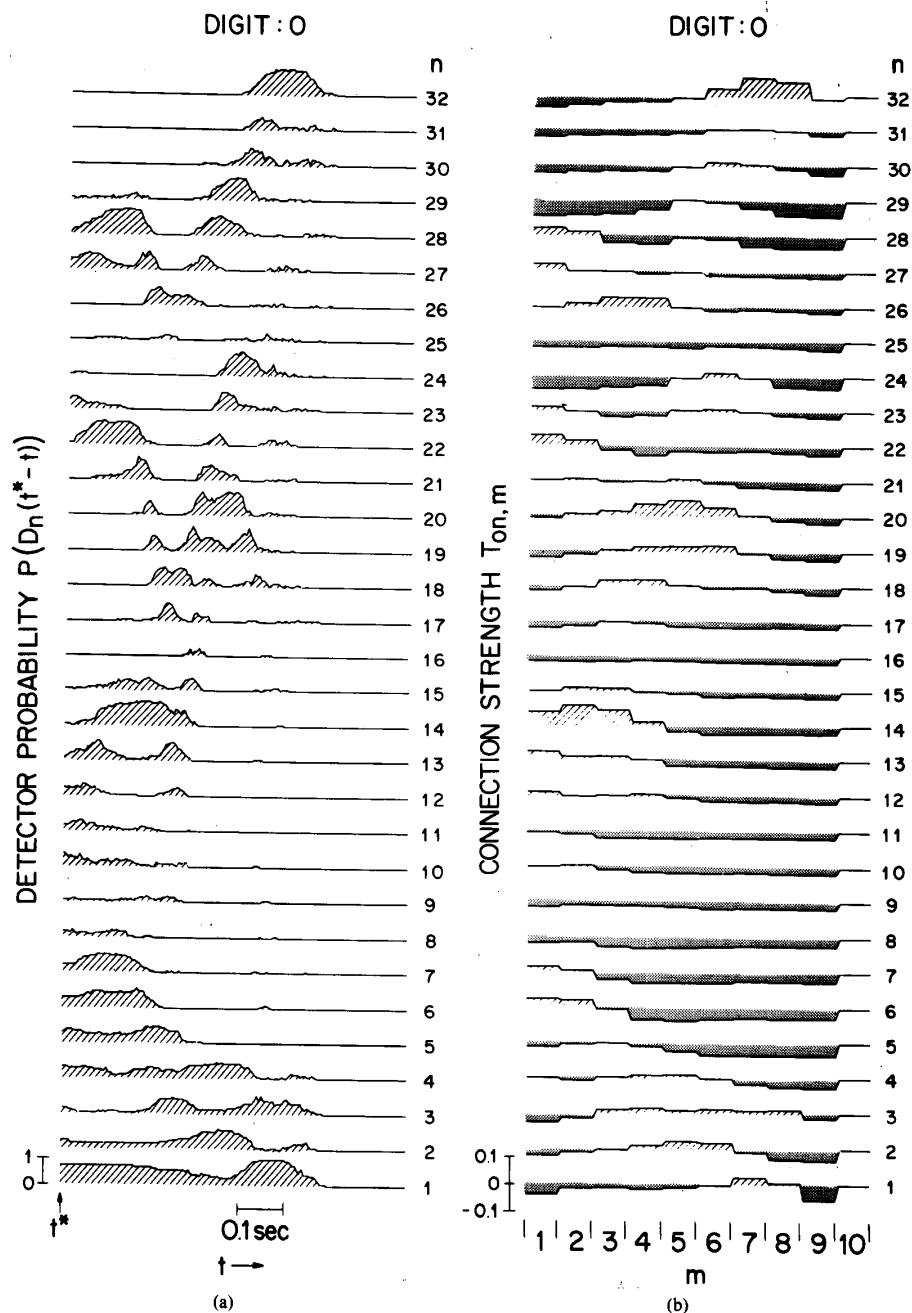
Fig. 8. (a) Detector output probabilities for the 19 exemplars of $\langle 0 \rangle$ in the training data. The probabilities are plotted as a function of the time before the word endpoint $t^*$. (b) The set of connections for the recognition of $\langle 0 \rangle$, learned after 256 iterations. The value of the connection at time delay $m$ for detector $n$ is indicated by the distance of the line above or below the horizontal axis at position $m$ on line $n$. Positive connections are shaded by lines and negative connections are shaded by dots. (In Fig. 8(a), the leftmost point corresponds to end of the word. In Fig. 8(b), the leftmost segment corresponds to zero time delay ($m = 0$) and longer delays lie along the positive $x$ axis.)

to be recognized. This is illustrated by considering the connections learned for word $\langle 0 \rangle$. The probability for each detector to turn on at different time points with respect to the endpoint $t^*$ for all exemplars of $\langle 0 \rangle$ in the training set is shown in Fig. 8(a). Regions with high probability ($P(D_n(t^* - t) \approx 1$) show

the features of $\langle 0 \rangle$ which provide reliable information about that word. A set of connections learned by the network for the recognition of $\langle 0 \rangle$ is shown in Fig. 8(b). Positive connections will be learned at those locations where exclusive information about the digit $\langle 0 \rangle$ against all other digits in the vocabulary is
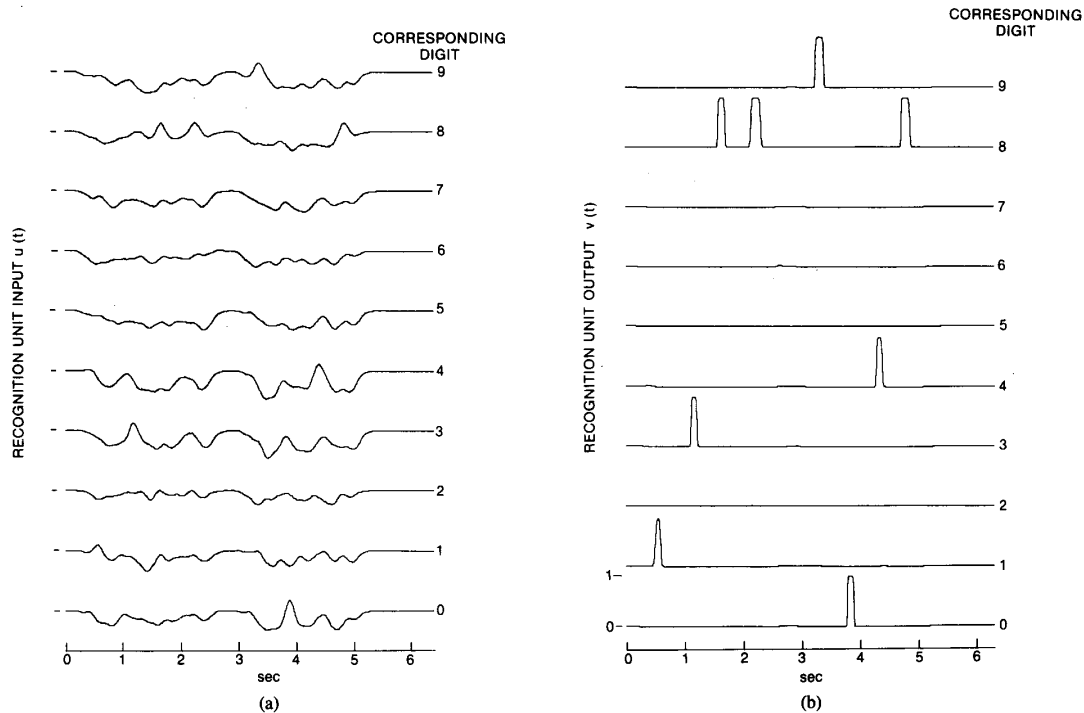
Fig. 9. Recognition unit outputs for presentation of the test digit sequence ⟨ 1 3 8 8 9 0 4 8 ⟩. (a) The figure shows the input potential ($u_i$) to the recognition units as a function of time. (b) The figure shows the output of nonlinear recognition units. No mutual inhibition between recognition units was used in these simulations.

present. This could be anticipated from the meaning of the mutual discrimination error measure being minimized (see Appendix A). This can be seen in channel 32 where the detector turns on with high reliability during the initial portions of the word. Positive connections are learned for the time delays 6, 7, and 8 corresponding to the region in time when $D_{32}(t) = 1$ with high probability. An opposite case can be seen in channel 1 where $D_1(t)$ has a high probability of being activated during the end of the word, but no positive connections are learned for the corresponding time delays (1 through 4). This suggests the presence of overlapping information from other digits during these time points. Hence there is no mutually exclusive information for the recognition of ⟨ 0 ⟩ from that region of channel 1.

The input and output potentials of recognition units in a network using a set of learned connections is shown in Fig. 9. The digit sequence presented to the network was ⟨ 1 3 8 8 9 0 4 8 ⟩. This sequence was not part of the training set. Fig. 9(a) shows the input potential $u_i(t)$. The markings on the left of each curve show the input potential level when no signal is presented to the network. The appropriate recognition units receive positive input potentials when their corresponding digits are presented. But incorrect recognition units also receive some positive input potential. For example, during the presentation of ⟨ 4 ⟩ recognition units for 0, 3, 5, 7, and 9 receive positive input potential. But these are not enough to drive the output of these incorrect recognition units, as can be seen from the outputs $V_i(t)$ plotted in Fig. 9(b). For training, targets are set for the outputs of recognition units. The threshold for each recognition unit is also learned. Hence discrimination of digits based on thresholds becomes an easier task at the output of recognition units.

## VI. EXPERIMENTAL RESULTS

Digit strings from a single male speaker were low-pass filtered, sampled at 12 kHz, and digitized with a 14-b A/D converter. The recordings were made using an AKG D330BT microphone, over several days, in an office environment with a signal-to-noise ratio of approximately 40 dB. Each string containes eight digits, spoken at a rate and segmentation typical of a telephone number string. In about half of the digit pairs, there is no break between the digits.

A set of 36 digit strings were used for training and another set of 18 were used for testing. For training, the individual digits were segmented from the strings by visual inspection of wave forms and spectrograms (as shown in Fig. 3). This segmentation yielded a set of 288 digits that were used for training. The temporal endpoint and the total length of the word were determined for each clipped out digit by visual inspection of spectrograms. Table I shows the number of utterances of each digit used for training, and the longest and shortest utterance in each case.

The percentage time warp, defined as

$$\frac{(\text{longest utterance} - \text{shortest utterance}) \times 100}{\text{mean length}}$$

ranged between 28–71% for the training set.

The complete time-delay neural network contains both fixed circuitry and modifiable connections. All of the circuitry that is used to produce the detector outputs is fixed. The filter bank characteristics and distributions were determined from psychophysical evidence and practical experience with filter bank front ends in other recognition systems [19]. The integration time

TABLE I
WORD LENGTH DISTRIBUTION IN TRAINING DATA

| Digit | Number of Utterances | Longest Utterance (sec) | Shortest Utterance (sec) | Mean Length (sec) | Time Warp (%) |
|---|---|---|---|---|---|
| 0 | 19 | 0.611 | 0.457 | 0.535 | 29 |
| 1 | 31 | 0.566 | 0.304 | 0.435 | 60 |
| 2 | 44 | 0.504 | 0.282 | 0.366 | 61 |
| 3 | 27 | 0.573 | 0.322 | 0.455 | 55 |
| 4 | 36 | 0.540 | 0.319 | 0.409 | 54 |
| 5 | 31 | 0.613 | 0.331 | 0.466 | 60 |
| 6 | 32 | 0.681 | 0.365 | 0.527 | 60 |
| 7 | 28 | 0.621 | 0.465 | 0.547 | 28 |
| 8 | 23 | 0.523 | 0.243 | 0.392 | 71 |
| 9 | 17 | 0.653 | 0.417 | 0.508 | 46 |

TABLE II
RECOGNITION ACCURACY[1] AS A FUNCTION OF LEARNING
ITERATIONS

| Number of Learning Iterations | Recognition Accuracy (%) | |
|---|---|---|
| | Training Data (segmented) | Testing Data (connected) |
| 8 | 0 | 0 |
| 16 | 43.4 | 47.2 |
| 24 | 82.3 | 73.6 |
| 32 | 91 | 84 |
| 40 | 94.8 | 89.5 |
| 48 | 95.1 | 90.9 |
| 56 | 96.9 | 94.4 |
| 64 | 97.6 | 96.5 |
| 128 | 100 | 99.3 (100) |
| 192 | 100 | 99.3 (100) |
| 256 | 100 | 99.3 (100) |

constants of the filter bank outputs were also fixed at 10 ms. The connections for center-surrounded inhibition were also predetermined as described above.

Several fixed time constants of the time-delayed connections and recognition units must also be chosen: i) the integration time constant of the recognition unit, ii) the mean delay between different taps on the delay lines, and iii) the width of delay functions (their standard deviations). A set of standard values for these parameters were used for routine training and testing of the network. We will first describe the performance of the network with these parameter values and then demonstrate the degree to which the recognition accuracy depends upon parameter value choice. The parameters used in routine training were determined by optimization of individual parameters. A systematic multidimensional search of the parameter space was not attempted. The standard values used in the present work were $\tau_{rec} = 0.04$ s, $\tau_{delay} = 0.06$ s, and $\sigma = 0.015$ s.

The number of time delays used for the recognition of each digit was determined according to the average length of that digit in the training data. For example, 9 time delays (32 × 9 connections) were used for ⟨0⟩ recognition and 6 time delays for ⟨2⟩ recognition (i.e., connections for the 4 longest time delays were set to zero for the ⟨2⟩ recognition unit). From the total of 3200 modifiable connections in the network only 2496 were changed during training; the rest were set to zero.

The performance of the network in correctly recognizing both the segmented digits in the training set and the 144 connected digits in the test set was determined as follows: the network dynamics was scored as correct for a given digit utterance only if the output of the correct recognition unit was greater than 0.5 and all incorrect units had outputs less than 0.5. In addition, if wrong units had outputs greater than 0.5, but at the same time the correct unit had a larger output, the network dynamics was also evaluated as correct. According to the above criteria, the network dynamics will be scored incorrect if all the recognition units had outputs less than 0.5. (A simple neural network can do this n-flop or winner-take-all decision task.) For comparison with scoring procedures that have been used in hidden Markov models (where the model with the maximum output probability is taken as the correct one even if its probability is small) in a few cases we also evaluated the network dynamics using an area criterion: the areas under all output recognition unit responses were calculated during word presentation and the unit with the largest area chosen to determine the network's decision of spoken word. In the following, recognition accuracies calculated using the area criterion will be given within brackets next to the other results. The recognition score with the threshold criteria can be considered as the word-spotting accuracy for this vocab-

ulary and the recognition score with the area criteria as word recognition accuracy.

It should be noted that training is done at 10 equally spaced time points 0.125 s apart. But after the training phase, the network output is continuous in time. For circuit simulations on the computer, we sample these every 5 ms. Evaluation of the network is done using these time varying outputs.

The neural network using the learning algorithm solves the connected-digit recognition problem for a single speaker with over 99% accuracy. The ability of the network to correctly recognize the utterances in the segmented-digit training set and the connected-digit testing set monotonically increased with successive iterations of the learning algorithm, as shown in Table II. By 128 iterations, the network using standard parameter values reached its peak performance level, correctly recognizing 100% of the segmented digits from the training data and 99.3% of the digits in the connected digit test strings. By the area criterion, the recognition accuracy of the time-delay network for the connected digit test strings increased to 100%. In several other test cases that convergence was analyzed by determining recognition accuracy versus iteration number, similar convergence rates were measured. Based upon these results, other learning experiments were routinely stopped at 256 iterations of the learning algorithm.

The fact that the recognition accuracy for the connected digit test set is not 100% could be due either to the inability of the network to capture the true variability in the speech set or due to the use of a training data set that is too small to capture the range of speech variance. To test these alternatives, the test set and training set were combined to form an enlarged training set. The resulting network recognizes 100% correctly all the training segments and all the strings from which these segments were made, demonstrating that the variance in the larger set could be captured by the network. Even more speech data would be required to test fairly the network that resulted from the enlarged data set, in order to determine if its performance on untrained speech segments was improved. This was not done since we did not feel that how well the single speaker problem could be solved above the 99% level is a significant question.

Although the criterion used to determine the above recogni-

[1]In all tables, the recognition accuracy with the threshold criterion is shown outside the parenthesis and the recognition accuracy with the area criterion is shown within parenthesis. The recognition accuracy for the training data was determined using segments and that for testing data was determined using actual connected strings.
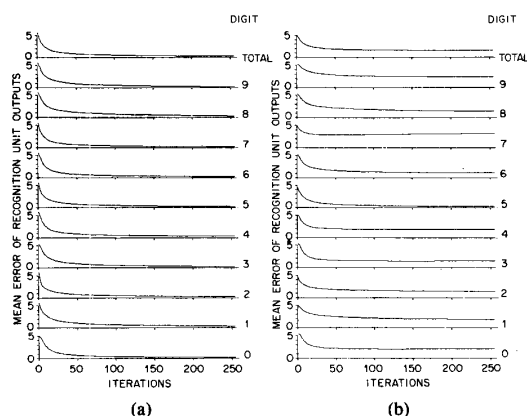
Fig. 10. Mutual information error measure for (a) the training data and (b) the testing data as a function of learning iterations. The error is plotted for each individual digit and also for the whole data set. See text for details of calculation of the error.

tion accuracies is that of practical interest in evaluating a recognition system, the network performance that is actually being maximized by the learning algorithm is determined by the mutual discrimination error measure in the gradient descent learning algorithm. (See Appendix A.) This total error measure is the sum of contributions provided by each digit category. In Fig. 10(a), these individual digit contributions and the total error are plotted versus learning algorithm iteration for the training set. The error expression can also be evaluated for the testing data, although the network is not trained on this set. It is shown in Fig. 10(b). While the error in the training set is still decreasing, the error is increasing for some of the sequences in the test set (digit $\langle 7 \rangle$, for example). This is again evidence that the training set is too small. In the later part of the learning procedure, peculiarities of the small training set are being learned, at the cost of appropriate generalization.

The integration time constant for the recognition neurons, the width of time delays, and the mean spacing between each of the delays were varied and the network performance determined. In all cases, the standard parameter values were used as a starting point, and the parameter under test was varied. The recognition accuracy is reduced when integration time constants of the recognition units is made substantially larger or smaller (Table III). Below $\tau_{rec} = 0.030$ s the recognition scores on both the training and test sets drop. Above $\tau_{rec} = 0.060$ s, the network is able to learn the training data very well, but its performance on testing data deteriorates. The performance on connected digits is worse at larger values of the integration time constant because of interference from signals of the preceeding digit.

Tables IV and V list the recognition accuracy as a function of the standard deviation ($\sigma$) of Gaussian time delays and the mean spacing ($\tau_{delay}$) between them. For the speaker-dependent connected digit data base used in these experiments, there is no sensitivity to these parameters over a wide range of values. We expect more sensitivity in the more difficult problem of speaker-independent recognition.

The tolerance to speech noise and imperfections in the hardware are important issues in the application of speech recognition algorithms and their implementations to real-world problems. The importance of the numerical precision of the connections on network performance was evaluated by quantization of the connections after learning. Connections were

### TABLE III
RECOGNITION ACCURACY AS A FUNCTION IF INTEGRATION TIME CONSTANT ($\tau_{rec}$)

| Integration Time Constant (sec) | Recognition Accuracy (%) | |
| --- | --- | --- |
| | Training Data (segmented) | Testing Data (connected) |
| .01 | 85.1 | 77.7 |
| .02 | 99.3 | 95.8 |
| .03 | 100 | 98.6 |
| .04 | 100 | 99.3 |
| .05 | 100 | 99.3 |
| .06 | 100 | 97.9 |
| .07 | 100 | 95.1 |
| .08 | 100 | 95.1 |

### TABLE IV
RECOGNITION ACCURACY AS A FUNCTION OF STANDARD DEVIATION OF GAUSSIAN TIME DELAYS ($\sigma$)

| Standard Deviation of Gaussian Time Delays (sec) | Recognition Accuracy (%) | |
| --- | --- | --- |
| | Training Data (segmented) | Testing Data (connected) |
| .00 | 100 | 99.3 |
| .01 | 100 | 99.3 |
| .02 | 100 | 99.3 |
| .03 | 100 | 99.3 |
| .04 | 100 | 99.3 |
| .05 | 100 | 99.3 |

### TABLE V
RECOGNITION ACCURACY AS A FUNCTION OF MEAN SPACING BETWEEN TIME DELAYS ($\tau_{delay}$)

| Mean Spacing Between Time Delays (sec) | Recognition Accuracy (%) | |
| --- | --- | --- |
| | Training Data (segmented) | Testing Data (connected) |
| .04 | 100 | 99.3 |
| .05 | 100 | 99.3 |
| .06 | 100 | 99.3 |
| .07 | 100 | 99.3 |
| .08 | 100 | 99.3 |

learned at full precision for 256 iterations of the learning algorithm. The learned connection set was subsequently quantized to a fixed number of quantization levels per connection. The quantization levels tested ranged between 21 (greater than 4 b of information per connection) to 3 (less than 2 b of information per connection). Table VI lists the recognition accuracy versus the number of quantized levels per connection. Performance of the network remains unchanged until the information per connection is reduced to less than 3 b (7 quantized levels of connections in the table). Even at 1.5 b of information per connection (3 quantized levels, $-1, 0$, and $+1$), the recognition score is still above 96% by our standard amplitude criterion and above 99% by the area criterion.

The network performance was also evaluated for speech data degraded by Gaussian noise. Noisy data was produced by adding speech data at high signal-to-noise ratio to Gaussian noise such that the signal power and noise power over the length of the connected digit string had the desired ratio. As shown in

TABLE VI
RECOGNITION ACCURACY OF A NETWORK TRAINED WITH
FULL PRECISION FOR CONNECTION STRENGTHS AND THEN
QUANTIZED TO DIFFERENT LEVELS OF ACCURACY PER
CONNECTION

| Number of Quantized Levels Per Connection | Recognition Accuracy (%) | |
| --- | --- | --- |
| | Training Data (segmented) | Testing Data (connected) |
| full precision | 100 | 99.3 (100) |
| 21 | 100 | 99.3 (100) |
| 15 (<4 bits) | 100 | 99.3 (100) |
| 11 | 100 | 99.3 (100) |
| 9 | 100 | 99.3 (100) |
| 7 (<3 bits) | 100 | 99.3 (100) |
| 5 | 98.6 (99.7) | 97.9 (100) |
| 3 (<2 bits) | 96.1 (99.7) | 96.5 (100) |

TABLE VII
RECOGNITION ACCURACY OF A NETWORK TRAINED WITH
STANDARD DATA AND TESTED WITH DIFFERENT NOISY DATA

| S/N of Gaussian Noise (dB) | Recognition Accuracy (%) | |
| --- | --- | --- |
| | Training Data (segmented) | Testing Data (connected) |
| >40 | | 99.3 |
| 31 | | 98.6 |
| 21 | | 77.0 |
| 11 | | 16.7 |

TABLE VIII
RECOGNITION ACCURACY OF A NETWORK TRAINED WITH
11-dB SNR DATA

| S/N of Gaussian Noise (db) | Recognition Accuracy (%) | |
| --- | --- | --- |
| | Training Data (segmented) | Testing Data (connected) |
| >40 | 85.1 (86.0) | |
| 31 | 85.0 (86.7) | |
| 21 | 85.1 (86.7) | |
| 16 | 86.0 (86.7) | |
| 11 | 85.3 (86.7) | |

TABLE IX
RECOGNITION ACCURACY OF A NETWORK TRAINED WITH
EQUAL AMOUNTS OF STANDARD AND 11-dB SNR DATA

| S/N of Gaussian Noise (db) | Recognition Accuracy (%) | |
| --- | --- | --- |
| | Training Data (segmented) | Testing Data (connected) |
| >40 | 100 | 99.3 (100) |
| 31 | 100 | 99.3 (100) |
| 21 | 100 | 99.3 (100) |
| 11 | 99.7 (100) | 99.3 (100) |
| 6 | 99.0 (100) | 97.9 (99.3) |

Table VII, performance of a network trained on noiseless data ( >40 dB SNR in the bandwidth 200 to 4000 Hz) deteriorates substantially when tested with speech having less than 30 dB signal-to-noise ratio. In general, degradation of performance was measured with test data having a lower signal-to-noise ratio than that of the training data. However, recognition accuracies of test data having a larger signal-to-noise ratio than the training were equal to that measured for the training data. For example, as shown in Table VIII, a network trained on a data set with 11-dB signal-to-noise ratio performs equally effectively on test sets of 21, 31, and >40 dB signal-to-noise ratios, although at a poorer overall accuracy than that observed for the noiseless data.

In the third experiment, 288 noiseless utterances in the standard training set was combined with the same utterances mixed with Gaussian noise at the 6 dB level. As listed in Table IX, when trained on this data set, the recognition accuracy for the test set of 144 utterances was 99.3% for all SNR above 6 dB, dropping only to 97.9% for the 6-dB test set.

The recognition of digits from noisy speech data and using low-precision connections is illustrated in Fig. 11. Fig. 11(a) shows the output of recognition units as a function of time during the presentation of a test digit sequence ⟨ 1 3 8 8 9 0 4 8 ⟩. (These are the same outputs shown in Fig. 9(b)) The speech waveform of the string and the word endpoints of each spoken digits are also shown in this figure. This network used a standard set of parameters as described earlier and the signal to noise ratio of speech data was about 40 dB. This figure shows that a fully trained network is able to recognize the different digits very close to their end points, even though the data was not segmented. Fig. 11B shows the performance of the same net-

work when the speech data contains 6 dB SNR of noise. The output potential of recognition units are almost identical to those shown in Fig. 11(a) even though there is substantial noise in the speech signal. The set of connections used in the above experiments were obtained by training the network with a mixture of standard speech data and similar speech data with 6 dB of added noise. To generate the responses shown in Fig. 11(c), (d), this set of connections were quantized to have only 3 values. Again the responses for standard and noisy data are almost identical to those in Fig. 11(a) and all the digits are recognized very close to their endpoints.

Fig. 12 shows an example of word spotting by the network. The waveform of the spoken sentence, "Our phone number is 206 519 3847," is plotted along with the output of recognition units as a function of time. Noise rejection properties of the network can be seen during the initial portions of the sentence. The segment, "our phone number is," does not turn on any of the digit recognition units above the 0.5 level except for "phone" turning on the ⟨ 1 ⟩ unit. Word-spotting properties of the network can be improved by adding words like "phone" to the vocabulary and learning not to recognize those words. Again all the digits are correctly recognized.

## VII. DISCUSSION

The network we have described functions by directly computing how much information the detection of a feature at time $t$ provides to the discrimination of a word in the vocabulary at a future time $t^*$. The input current that is produced at the recognition units at time $t^*$ is directly related to this information. Since this input current is proportional to the strength of the time-delayed synapses from the outputs of the detectors to the inputs of the recognition units, information about the significance of a feature detection to the recognition of a word in the vocabulary lies in the pattern of time-delayed connections. In the circuit studied here, each word in the vocabulary is associ-
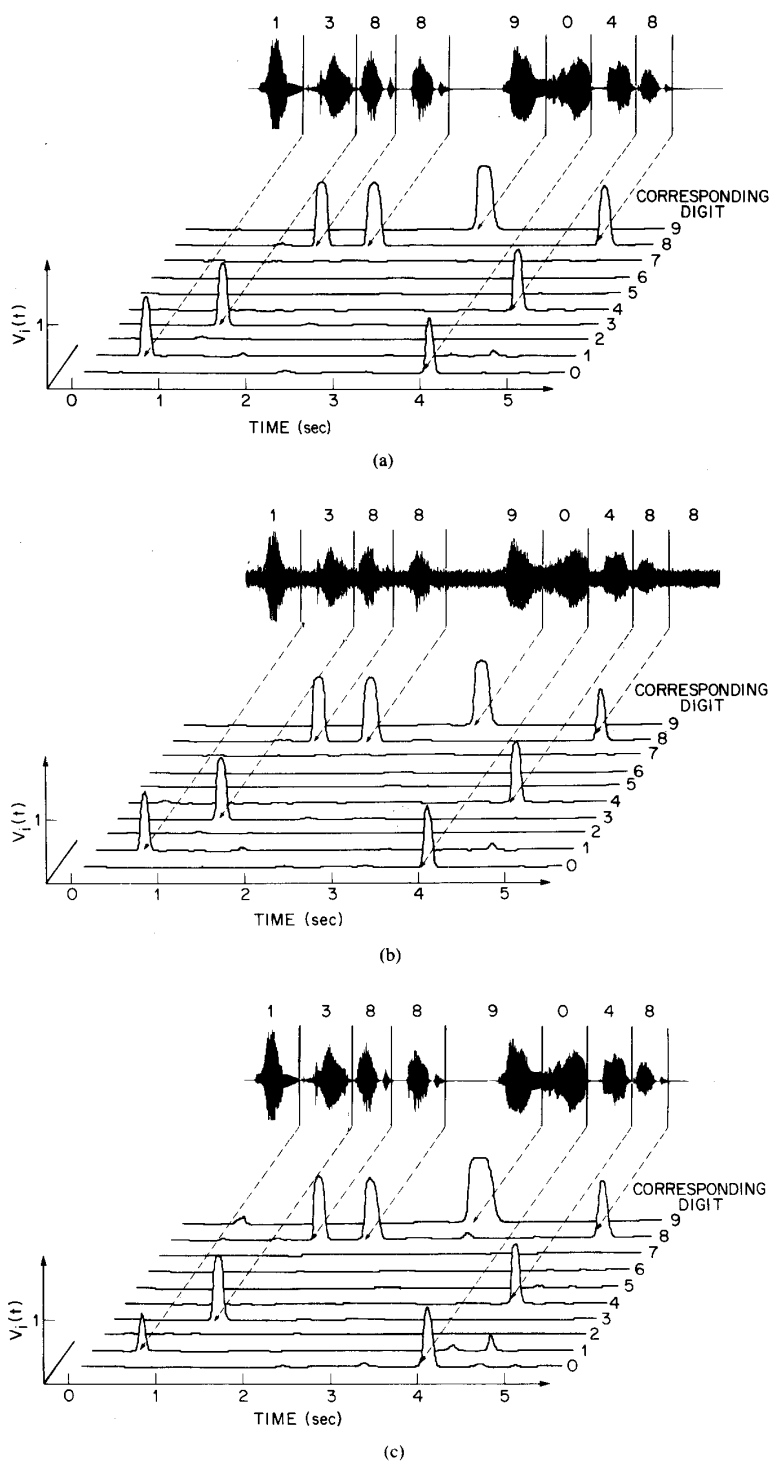
Fig. 11. (a) Response of recognition units plotted alongside the speech waveform for the test digit sequence ⟨ 1 3 8 8 9 0 4 8 ⟩. Approximate time points where the digits ended are marked on the waveform. Though the network had no knowledge of these time points, we can see that recognition of digits occurred approximately at their endpoints. (b) Recognition of a noisy sample of the same digit string. Gaussian white noise of 6-dB SNR was added to the original sample. Note that the response waveforms in cases (a) and (b) are almost identical in spite of the large amount of noise in case (b). (c) Recognition of the normal test string after the set of connections are quantized to have only 1.5 b of information per connection. Under this condition, only 18.5% of the available 3200 connections are nonzero. (*Continued on next page.*)
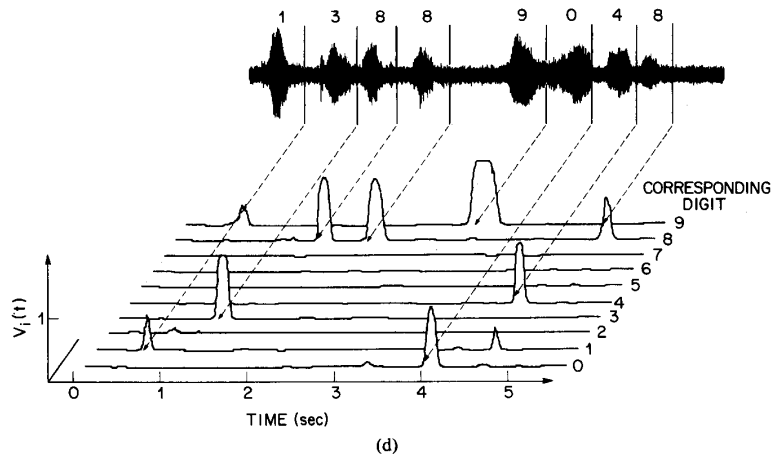
Fig. 11. (con't.) (d) Recognition unit outputs for the same quantized connections for the 6-dB SNR test data. The network is robust with respect to noise in the speech signal and accuracy of connection strengths.
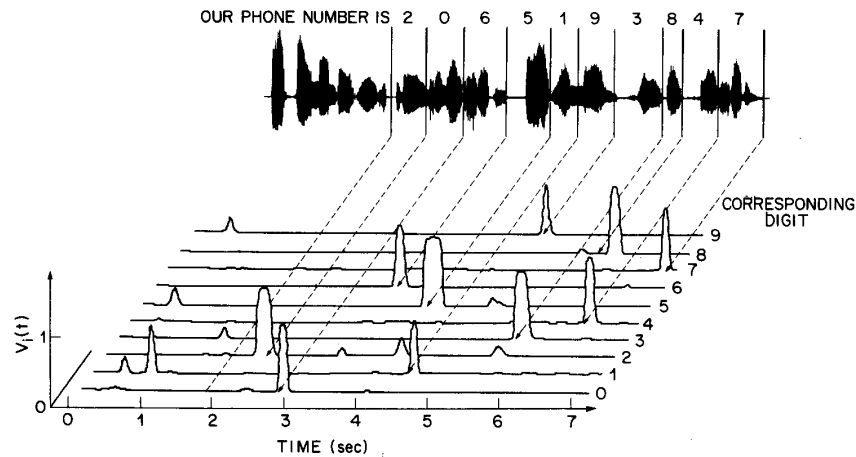


Fig. 12. An example of word spotting by the network. The outputs of recognition units are plotted alongside the speech waveform for a test sentence, "Our phone number is 206 519 3847."

ated with 320 analog-valued time-delayed connections, 10 delays from each of 32 feature detectors. (The number of connection is often less due to setting connections with delays longer than the mean word length to zero, as described earlier.) The number of connections increases linearly with the number of words in the vocabulary. This is similar to the scaling of HMM and DTW algorithms that operate strictly on word level objects and do not have a hierarchical organization or level building from shorter speech sounds like phonemes, diphones or syllables. As in these other recognition systems, the scaling of the network connections with increasing vocabulary will be reduced by hierachical recognition strategies [20]. For example, complete networks of the type described here can be trained to recognize phonemes. The outputs of these phoneme detectors can be considered as the inputs to a word recognition network, replacing or being an addition to the formant detectors used in the network we have presented here. A hierarchical approach would also improve the performance on vocabularies with larger time warp. In the present circuit, time-warp insensitivity can be increased only by decreasing temporal resolution of feature order.

In most current HMM and DTW systems, the model parameters and templates for individual words contain the information that characterizes the acoustic signal for that word. This information depends very little on other words in the vocabulary. This is different from information in the connections of the time-delayed perceptron network. The value of individual connections is related to the amount of information provided by the detection of its associated feature to the discrimination of its associated word from other words. Since the relevance of an individual feature to the word discrimination problem depends intrinsically upon the other words in the vocabulary the numerical values will be strongly dependent upon the vocabulary and noise conditions. This was evident in the relationship between the connection strengths for the word ⟨ 0 ⟩ and the probabilities of detector outputs for the set of ⟨ 0 ⟩ utterances.

The feature set used in the network described here was based upon a simple "center-surround" determination of local peaks in the short-time frequency spectrum. These peaks roughly correspond to the formant positions, especially for vowels. The center-surround mechanism intrinsically relies upon the physical relationship between adjacent input channels. The learning

algorithm that organizes the time-delayed connections so as to minimize the mutual discrimination error measure (see Appendix A) does not depend upon any logical relationship between detector outputs. Any set of feature detectors can be used.

Some common descriptions of the acoustic signal (for example, the set of cepstral coefficients) cannot be directly used as a feature vector for the learning algorithm. The analog numbers of the cepstral vector components have no meaning as features. However, cepstral vectors could be used in conjunction with a vector quantizer, each codeword in the VQ codebook having an associated detector line with an output related to the likelihood that the observed cepstral vector was an instance of the codeword. Gold et al. have used this for recognition of isolated words [20]. The use of thresholded center-surround features greatly enhances the discriminability of this system (see Fig. 5). The recognition accuracy is reduced to 90% when the rectified outputs of filters are themselves used as input features.

In the work of others, increasing recognition accuracies have accompanied the use of more acoustic-phonetic attributes in feature vectors. For example, HMM recognition scores was appreciably increased by inclusion of delta-cepstral coefficients providing direct information on formant trajectories. The ability of the network to work with the arbitrary feature detectors we chose suggests its utility in recognition systems based more heavily upon acoustic-phonetic features. For example, it would be possible to directly use our recognition network with the more complex feature detector neural network under development by Mueller and coworkers [22]–[24]. In this work, simple center-surround mechanisms and delays were used to construct output units that respond to abrupt onsets and offsets of spectral energies and that can detect the local direction of formant motion in frequency. Our circuit can be considered a network implementation of a statistical pattern recognition algorithm that matches naturally to the kind of connectionist "front-end" implementation suggested by this work. Espy-Wilson has demonstrated how several complex acoustic-phonetic features can be computed from the speech signal along with their probability of being observed in a given data stream [25]. These computed outputs could be directly used as inputs to our recognition network, and it may be possible to implement these detectors also in connectionist architecture. Networks similar to those described here and based upon a spectral representation could be trained to have outputs that directly represent the probability that an acoustic phonetic feature has been observed in the data stream. The outputs of these trained complex feature detectors could be used as inputs to a larger, hierarchical network that is designed to recognize larger acoustic categories like syllables or words.

The delay network has no explicit determination of word endpoint. The speech signal is not explicitly segmented during the recognition process. (Indeed, the same snippet of sound may be taken to be both the end of one word and the beginning of next.)

It is intrinsically a word-spotting architecture. This was evident in the performance illustrated in Fig. 12. For optimum word-spotting performance, it may be important that the network be trained with a characteristic set of "noise" that the system will be exposed to in the recognition environment. This speech need not be characterized as words, but simply "not a word in the known vocabulary." The word spotting aspect of the delay network is in contrast to the inherent segmentation that has been a requirement of present HMM recognition systems and which renders them less effective in unknown word environments.

The experiments where connections were quantized demon-strate that the numerical precision necessary for implementation of the network is relatively low. We are not aware of similar experiments with HMM or DTW recognition algorithms, where published work makes explicit use of high-precision arithmetic.

Three aspects of the acoustic signal processing we have used have a basis in neurobiology. The Mel scale, describing the center frequencies of the band-pass filters, is based upon the representation of frequency along the mammalian cochlea. The spectral peak feature detectors circuitry employ a center-surround organization found in the mammalian auditory system. Our use of time delays to recognize a temporal pattern is analogous to the use of delay mechanisms in the biosonar system of the moustache bat [26] and sound localization by the barn owl [27].

Many other neural network architectures have been used for speech recognition. A majority of these networks employ multilayer perceptrons using the back-propagation error correction method [28] to learn the set of connections. Many of these networks reduce the temporal pattern recognition problem to one of spatial pattern recognition by using the spectral energy over a period of time as the input to the network. Ellman and Zipser used a network for the recognition of syllables, vowels, and consonants spoken by a single male speaker [10]. With a network trained to achieve 100% recognition on training data, they achieved recognition accuracies of 84%, 98.5%, and 92.1%, respectively, for the three tasks. The performance of the network improved when noise proportional to signal was added during training. Waibel et al. have used a similar network for phoneme recognition [9]. The hidden units in this network receive inputs from lower layers with different time delays. They achieved an average correct recognition of 98.5% for a single male speaker. Watrous and Shastri used a network to discriminate between the word pairs NO/GO by a single speaker [29]. Hidden units in this network have self recurrent links for accumulating information over time. The total error of the network for recognition wildly fluctuated during training. The best recognition score on a test set of 25 word pairs was 98%. Burr achieved 99.5% recognition accuracy for single speaker isolated digit recognition using a network with one layer of hidden units [30]. Plaut and Hinton used a similar network to filter out noise in formant-like patterns and recognize them as a riser or a nonriser [31]. The performance of this network improved 1) when they used different noisy exemplars each time during training and 2) by decaying the connection weights each time so that small weights tend to become zero. Prager et al. used a Boltzmann machine for detection of 11 steady state vowels [32]. They achieved a recognition score of 85% for the same speaker. Burr has used a single layer perceptron for recognition of E-set words and polysyllable words [33]. Gold has used a 120 neuron Hopfield net for recognition of vowels and consonents [12] and a neural network with time-delayed connections for recognition of isolated words [20], [21]. When half of the 120 b are given as input, the network converged perfectly in 79.6% of cases. Kohonen has used a self-organizing network to classify phonemes [11]. This network classifies phonemes from multiple male speakers with an accuracy of 92–97% and is capable of isolated word recognition on a 1000 word vocabulary with 96–98% accuracy.

Although based upon back-propagation of a quadratic error measure, the network used by Waibel et al. has a delay architecture very similar to the one discussed in this paper. We have used an information theoretic error measure that provides recognition unit outputs that have a probabilistic interpretation. In addition, we have not used a multilayer perceptron in order to

avoid problems associated with multiple minima on the error surface.

## APPENDIX A
## THE ANALOG PERCEPTRON LEARNING ALGORITHM USED

The analog perceptron learning rule discussed in detail elsewhere [8], was applied to the speech recognition network studied as follows. A given set of utterances (the training set) produces a sequence of detector outputs $D_n(t)$ versus time as described in the text. Let us consider each detector output at $m$ fixed time points located before the endpoint of the word, each time point corresponding to the mean delay of one of the $m$ delay lines used in our network. If the delays in our network were rigid with no dispersion, this set of detector outputs would be the combined information multiplied by the delay line connections, appearing the input current to the recognition units at the end of the utterance. By assuming rigid delays, we can think of our speech recognition network as a single layer perceptron where for each word $w$ in the set $W$ of training words, 320 (32 input channels × 10 delays) input values are to be mapped onto a set of 10 output values, corresponding to the outputs of the recognition units evaluated at the word endpoint $t^*$. For each word $w$ which belong to the set $W$, the input data consists of the 320 inputs $I_{n,m}^w(t) = D_n^w(t^* - m\tau_{\text{delay}})$, $m = 1, \cdots , 10$. At $t^*$ the output $V_i^w$ of the $i$th recognition unit for the input data provided by $w$ is

$$V_i^w = g\left( \sum_{n,m} T_{i,n,m} I_{n,m}^w \right) \qquad (9)$$

with

$$g(u) = \tfrac{1}{2}\left[1 + \tanh(\beta u)\right]. \qquad (10)$$

Because of the linearity of the convolution operator, a similar transformation works with nonrigid delays (finite $\sigma$) and nonzero $\tau_{\text{rec}}$. Associated with each word $w$ in the training set $W$, we have a set of probabilities $P_{+,i}^w$ that the word is an instance of category (i.e., digit) $i$, and a probability $P_{-,j}^w$ that it is not an instance of this category. For our data set there was no ambiguity, so these probabilities are 1's and 0's. However, the learning algorithm is more general than this and can be applied to situations where only a statistical measure is possible.

The analog perceptron learning algorithm uses the entropy of $P$ with respect to $V$ given by

$$f = \sum_W \sum_j \sum_{+,-} P_{+,-,j}^w \ln(P_{+,-,j}^w / V_{+,-,j}^w) \qquad (11)$$

as the convex positive function to be minimized by the properly organized network. In our network, $V_i^w$ is the probability that word $w$ is an instance of category $i$. Thus in the above expression $V_+^w = V_i^w$ and $V_-^w = (1 - V_i^w)$. Direct differentiation of the above expression shows that gradient descent on $f$ in the space of connection weights is obtained with the iterative update rule

$$\Delta(T_{i,n,m}) = \epsilon \sum_w \left[(P_{+,i}^w - P_{-,i}^w) - \tanh(\beta u_i^w)\right] I_{i,m}^w. \qquad (12)$$

Where $\epsilon$ is the learning coefficient.

A second differentiation shows that the curveture of this function in connection space is always greater than or equal to zero, so that the gradient descent procedure will always find the global minimum of the function. (This expression reduces to the rules described in the section on constructing the delay connections.)

## APPENDIX B
## ISSUES PERTAINING TO ELECTRONIC IMPLEMENTATION

For small circuits with few feature detectors, it is easy to put individual delays of the appropriate amplitude and time course on the outputs of the feature detectors for direct connection to the recognition units, essentially as depicted in Fig. 2(a). For most of the large circuits of practical interest, it is more effective to build longer delays by combining simple short delays in succession and implementing a tapped delay line. Tapped delay lines can be built in strictly analog VLSI, as in the artificial cochlea built by Mead's group [29] or implemented as digital delay lines with analog taps, as in CCD tapped delay lines, or made completely in digital hardware using memory storage and CPU arithmetic to compute the effects of analog connections.

For any implementation, there are two fundamentally different ways of organizing the time-delayed connections. One is to put a tapped delay line on the output of each feature detector and to connect resistors representing the strength of a given time-delayed connection from the delay line tap to the input of the appropriate recognition unit. This is an efficient representation when there are many more recognition units than there are detectors. If there are $N_D$ detectors, $N_R$ recognition units, and $N_\tau$ delay line taps, the number of delay line segments will be $N_D * N_\tau$ and the number of connection resistors will be $N_D * N_\tau * N_R$. When the number of detectors exceeds the number of recognition units it is more efficient to put a tapped delay line on the input to a recognition unit and to have summing junctions at each tap. The number of connections needed is the same, but the number of delay line elements is now $N_R * N_\tau$.

## REFERENCES

[1] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 66–72, Feb. 1975.

[2] C. S. Myers and L. R. Rabiner, "A level building dynamic time warping algorithm for connected word recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 284–297, Apr. 1981.

[3] J. K. Baker, "Stochastic modeling for automatic speech understanding," in *Speech Recognition*, R. Reddy, Ed. New York: Academic, 1975.

[4] F. Jelinek, "Continuous speech recognition by statistical methods," *Proc. IEEE*, vol. 64, pp. 532–536, Apr. 1976.

[5] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *Bell Syst. Tech. J.*, vol. 62, pp. 1035–1074, Apr. 1983.

[6] D. W. Tank and J. J. Hopfield, "Neural computation by concentrating information in time," *Proc. Nat. Acad. Sci. U.S.*, vol. 84, pp. 1896–1900, Apr. 1987.

[7] D. W. Tank and J. J. Hopfield, "Concentrating information in time: Analog neural networks with applications to speech recognition problems," in *Proc. IEEE First Int. Conf. Neural Networks* (San Diego, CA), June 1987.

[8] J. J. Hopfield, "Learning algorithms and probability distributions in feedforward and feedback networks," *Proc. Nat. Acad. Sci. U.S.*, vol. 84, pp. 8429–8433, Dec. 1987.

[9] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang,

"Phoneme recognition using time-delay neural networks," ATR Interpreting Telephony Research Laboratories, Japan, Tech. Rep. TR-I-0006, 1987.

[10] J. L. Ellman and D. Zipser, "Learning the hidden structure of speech," *J. Acoust. Soc. Amer.*, vol. 83, pp. 1615-1626, Apr. 1988.

[11] T. Kohonen, "The neural phonetic typewriter," *Computer*, vol. 21, pp. 11-22, Mar. 1988.

[12] B. Gold, "Hopfield model applied to vowel and consonant discrimination," M.I.T. Lincoln Lab., Lexington, MA, Tech. Rep. 747, June 1986.

[13] J. J. Hopfield and D. W. Tank, "Computing with neural circuits: A model," *Science*, vol. 233, pp. 625-633, Aug. 1986.

[14] R. P. Lippman, "An introduction to computing with neural nets," *IEEE ASSP Mag.*, vol. 4, pp. 4-22, Apr. 1987.

[15] C. E. Cook and M. Bernfield, *Radar Signals*. New York: Academic, 1967.

[16] S. A. Shamma, "Speech processing in the auditory system, Part II: Lateral inhibition and the central processing of speech evoked activity in the auditory nerve," *J. Acoust. Soc. Amer.*, vol. 78, pp. 1622-1632, 1985.

[17] S. A. Kaltenbach and Y. A. Saunders, "Spectral and temporal response patterns of single units in Chinchilla dorsal cochlear nucleus," *Exp. Neurol.*, vol. 96, pp. 406-428, 1987.

[18] S. W. Kuffler, "Discharge patterns and functional organization of the mammalian retina," *J. Neurophysiol.*, vol. 16, pp. 37-68, 1953.

[19] B. A. Dautrich, L. R. Rabiner, and T. B. Martin, "On the effects of varying filter bank parameters on isolated word recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, p. 193, Aug. 1983.

[20] B. Gold, R. P. Lippmann, and M. L. Malpass, "Some neural net recognition results on isolated words," in *Proc. IEEE First Int. Conf. Neural Networks* (San Diego, CA), June 1987.

[21] "A neural network for isolated word recognition," in *Proc. Int. Conf. Acoust., Speech, Signal Processing* (New York, NY), 1988.

[22] P. Mueller and J. Lazzaro, "A machine for neural computation of acoustical patterns with application to real-time speech recognition," in *Neural Networks for Computing*. New York: American Institute of Physics, 1986.

[23] P. Mueller, T. Martin, and F. Putzrath, "General principles of operations in neuron nets with application to acoustical pattern recognition," in *Biological Prototypes and Synthetic Systems*. New York: Plenum, 1962.

[24] P. Mueller, "Principles of temporal pattern recognition in artificial neuron nets with application to speech recognition," in *Artificial Intelligence*, S-142. New York: The Institute of Electrical and Electronics Engineers, 1963.

[25] C. Y. Espy-Wilson, "An acoustic-phonetic approach to speech recognition: Applications to the semivowels," doctoral dissertation, Massachusetts Institute of Technology, May 1987.

[26] N. Suga, "The extent to which biosonar information is represented in the bat auditory cortex," in *Dynamic Aspects of Neocortical Function*. New York: Wiley, 1984, pp. 315-373.

[27] C. E. Carr and M. Konishi, "Axonal delay lines for time measurement in the owl's brain stem," *Proc. Nat. Acad. Sci. U.S.*, vol. 85, pp. 8311-8315, 1988.

[28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533-536, Oct. 1986.

[29] R. L. Watrous and L. Shastri, "Learning phonetic features using connectionist networks: An experiment in speech recognition," Univ. of Pennsylvania, Philadelphia, PA, Tech. Rep. MS-CIS-86-78, Oct. 1986.

[30] D. J. Burr, "A neural network digit recognizer," in *Proc. IEEE Int. Conf. Syst. Man, Cybern.* (Atlanta, GA), 1986.

[31] D. C. Plaut and G. E. Hinton, "Learning sets of filters using back-propagation," *Comput. Speech Language*, vol. 2, pp. 35-61, 1987.

[32] R. W. Prager, T. D. Harrison, and F. Fallside, "Boltzmann machines for speech recognition," *Comput. Speech Language*, vol. 1, pp. 3-27, 1986.

[33] D. J. Burr, "Speech recognition experiments with perceptrons," in *Proc. Neural Inform. Processing Syst.—Natural and Synthetic* (Denver, CO), 1987.

[34] R. F. Lyon and C. Mead, "An analog electronic cochlea," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-36, pp. 1119-1134, June 1988.

[35] M. A. Bush and G. E. Kopec, "Network-based connected digit recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 1401-1413, Oct. 1987.
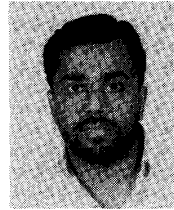
**K. P. Unnikrishnan** received the B.Sc. in physics from Calicut University, India, in 1979, the M.Sc. degree in physics from Cochin University, India, in 1981, and the Ph.D. degree in biophysics from Syracuse University, Syracuse, NY, in 1987.

From 1987 to 1989 he was a postdoctoral Member of the Technical Staff at AT&T Bell Laboratories, Murray Hill, NJ. He is currently a Senior Research Scientist at GM Research Laboratories, Warren, MI. His research interests concern neural computations in sensory systems.



**John J. Hopfield** received the B.A. degree from Swarthmore College in 1954 and the Ph.D. degree in physics from Cornell Universisty, Ithaca, NY, in 1958.

His research has included studies of neural networks in biological computation, work on electron transfer in photosynthesis, accuracy and proofreading in biomolecular synthesis, and optical properties and impurity levels of semiconductors. He is currently the Roscoe G. Dickinson Professor of Chemistry and Biology at the California Institute of Technology where he leads the program in computation and neural systems. He was formerly a member of the Biophysics Research Department at AT&T Bell Laboratories, Murray Hill, NJ.
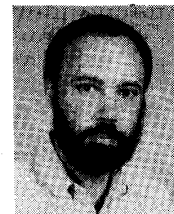


**David W. Tank** was born in Cleveland, OH, on June 3, 1953. He received his undergraduate education from Case Western Reserve University, Cleveland, OH, and Hobart College, Geneva, NY. He received the Ph.D. degree in physics from Cornell University, Ithaca, NY in 1983.

From 1983 to 1984 he was a postdoctoral fellow at AT&T Bell Laboratories in Murray Hill, NJ. He has remained at Bell Laboratories, joining the Biophysics Research Department in 1984. His research interests concern the biophysics of individual nerve cells, neural representation and coding, and the computational properties of neural circuits.