Learning the hidden structure of speech

Jeffrey L. Elman
Department of Linguistics, University of California, San Diego, La Jolla, California 92093

David Zipser
Institute for Cognitive Science, University of California, San Diego, La Jolla, California 92093

(Received 11 September 1987; accepted for publication 30 December 1987)

In the work described here, the backpropagation neural network learning procedure is applied to the analysis and recognition of speech. This procedure takes a set of input/output pattern pairs and attempts to learn their functional relationship; it develops the necessary representational features during the course of learning. A series of computer simulation studies was carried out to assess the ability of these networks to accurately label sounds, to learn to recognize sounds without labels, and to learn feature representations of continuous speech. These studies demonstrated that the networks can learn to label presegmented test tokens with accuracies of up to 95%. Networks trained on segmented sounds using a strategy that requires no external labels were able to recognize and delineate sounds in continuous speech. These networks developed rich internal representations that included units which corresponded to such traditional distinctions as vowels and consonants, as well as units that were sensitive to novel and nonstandard features. Networks trained on a large corpus of unsegmented, continuous speech without labels also developed interesting feature representations, which may be useful in both segmentation and label learning. The results of these studies, while preliminary, demonstrate that backpropagation learning can be used with complex, natural data to identify a feature structure that can serve as the basis for both analysis and nontrivial pattern recognition.

PACS numbers: 43.72.Ne, 43.72.Ar

INTRODUCTION

The recognition of speech is one of the many things that is carried out by humans with apparent ease, but that has been done by computers only at great cost, with high error, and in highly constrained situations. Whether or not one is interested in machine-based speech recognition *per se*, the difficulties encountered by such systems may be diagnostic of flaws in the theoretical frameworks that motivate them. We believe that part of the difficulty in such systems lies in the use of inappropriate features as units for recognizing and representing speech. But what are the appropriate units and how are they to be found? In this article, we describe studies designed to determine whether these units can be learned. We use a newly developed learning procedure for artificial neural networks.

The question "What are the units of speech perception?" has been long standing and controversial. The problem is that, when one looks at the acoustic waveform, there are rarely obvious clues as to the boundaries between segments, let alone an indication of what those segments are. The speech sound wave varies continuously and smoothly over time. There is nothing mysterious or unexpected in this; it is the acoustic consequence of the fact that the production of speech involves a high degree of coarticulation (i.e., the mutual influence of neighboring sounds) with smooth transitions from one sound to another.

While one has the impression that speech is made up of concatenated "sounds," just what those sounds are is debatable. At least eight different levels of representation have been proposed to intervene between the speech wave and the representation "word": spectral templates (Lowerre, 1976), features (Cole *et al.*, 1986), diphones (Dixon and Silverman, 1976; Klatt, 1980), context-sensitive allophones (Wickelgren, 1969), phonemes (Pisoni, 1981), demisyllables (Fujimura and Lovins, 1978), syllables (Mehler, 1981), and morphemes (Aronoff, 1976; Klatt, 1980). While these are all reasonable candidates for representing speech, the problem is that they have not been derived, in a canonical way, from the speech data itself. Learning provides a systematic way to find recognition features in data.

Whether or not the representations used in the perception of speech are innate or learned remains open, and we do not wish to take a strong position on this issue. However, much of the motivation for supposing that internal representations in perception are innate has come from the apparent poverty of data and the weakness of learning algorithms. Recent developments in parallel distributed processing (PDP) learning algorithms have demonstrated that a surprisingly small amount of data may contain sufficient cues to its intrinsic structure, so that this structure can be inferred using only rather simple learning rules. In the current article, we attempt to demonstrate the consequences of requiring that representations be learnable. To do this, we have taught trainable networks a series of speech recognition tasks and then examined the internal representations that are generated. The networks are adept at solving the recognition tasks. They spontaneously develop their own representations, which sometimes, but not always, correspond to our previous categories of representational units.

0001-4966/88/041615-12$00.80

I. BACKPROPAGATION OF ERROR

In these studies, we use a form of the "generalized delta rule" known as "backpropagation of error" (Rumelhart *et al.*, 1986; Le Cun, 1985; Parker, 1985). This algorithm provides a technique for training a multilayer network to associate many pairs of patterns. One member of the pair is the input and the other is the output. Generally, the input and output are related in some interesting manner (although the exact nature of the relationship may be unknown). The task of the network is to learn this relationship. More precisely, the complete set of pattern pairs to be learned can be thought of as the extensional definition of a vector-valued function whose domain is the set of input patterns and whose range is the set of outputs.

The set of functions that can be learned in such networks depends on several things, including their architecture and the learning algorithms employed. For example, the perceptron convergence procedure (Rosenblatt, 1962) can program networks to compute linear Boolean functions such as AND and OR, but not nonlinear ones such as XOR. The early generalizations of the perceptron rule that extended the learning set to patterns with continuous rather than Boolean values are also limited to learning linear functions. Backpropagation, on the other hand, can be used to teach multilayer networks to compute all the Boolean functions. Backpropagation is applicable to patterns with continuous component values, and thus can also deal with a much wider range of functions.

Much of the significance of backpropagation learning stems from the fact that it is defined on a neurallike network. An example of such a network is shown in Fig. 1. In the networks used in the present study, there was a set of units designed as *input units*, which receive external input; a set of units called *output units*, which transmit output; and internal units called *hidden units*, which receive from and transmit to other units in the network. These three types of units were organized into three distinct layers (*input, hidden, output*), with full feed-forward connections between adjacent layers. The learning algorithm works as follows.

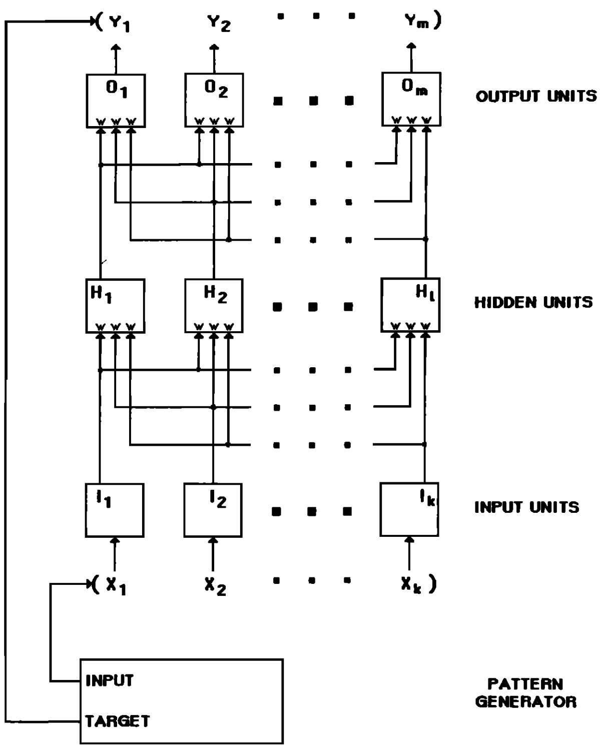

FIG. 1. Strictly layered backpropagation networks of the kind depicted in this figure were used for all the work described in this article. See text for details of the learning algorithm used to train the networks.

Before training begins, all the weights and biases[1] are set to random values between − 1.0 and 1.0. Training proceeds in cycles. At the beginning of each learning cycle, i.e., at time $t = 0$ for that cycle, an input pattern $(x_1 x_2 \cdots x_k)$ and a target pattern $(y_1 y_2 \cdots y_k)$ are selected at random from the set of input/output pattern pairs to be learned. At time $t = 1$, the activations of the input units are set equal to the input pattern; that is, $I_1 = x_1, I_2 = x_2, \ldots, I_k = x_k$. The output activations of the hidden units are set at time $t = 2$ to

$$H_i(t+1) = \text{squash}\left(\text{bias}_{H_i} + \sum_{j=1}^{j=k} W_{H_i j} I_j(t)\right),$$

where

$$\text{squash}(x) = (1 + e^{-x})^{-1}.$$

Similarly, the activations of the output units are set at $t = 3$ to

$$O_i(t+1) = \text{squash}\left(\text{bias}_{O_i} + \sum_{j=1}^{j=1} W_{O_i j} H_j(t)\right).$$

The resulting pattern of activation over the output units constitutes the network's output for this input pattern. This output pattern vector is then subtracted from the correct output pattern [i.e., the target pattern $(y_1 y_2 \cdots y_k)$] to produce an error pattern. This error, in turn, is used at $t = 4$ to adjust the weights of connections feeding into the output layer, using the rule

$$\Delta W_{O_i j} = \eta \delta_{O_i} H_j, \quad \Delta\text{bias}_{O_i} = \eta \delta_{O_i},$$

$$\delta_{O_i} = (y_i - O_i) O_i (1 - O_i),$$

$$\eta = \text{learning rate constant.}$$

Error is then propagated back to hidden units at $t = 5$ according to the rule

$$\Delta W_{H_i j} = \eta \delta_{H_i} I_j, \quad \Delta\text{bias}_{H_i} = \eta \delta_{H_i},$$

$$\delta_{H_i} = \eta H_i (1 - H_i) \sum_{k=1}^{k=m} \delta_{O_k} \omega_{O_k i}.$$

These five sequential events together make up one learning cycle.

When computations are programmed inductively in this manner, values from the range of the function (corresponding to the desired outputs of the network) must be available. Frequently, these values are supplied by an external source or "teacher," which assumes that such a teaching input is available; however, in certain situations, this assumption may be an unrealistic one. This raises the question of how to configure a system so it can learn, either without an external teacher or with the kind of information more realistically available.

Several solutions have been proposed. One of the simplest and most elegant of these is to use teaching patterns that are the same as the input or some fixed transformation of the input. While this would seem to limit us to learning the identity function (or some fixed transformation of it), it has been shown that with this procedure the hidden units learn to represent the input patterns in terms of salient features. When the number of hidden units is less than the number of input units, the information in the input is represented at a lower dimensionality. In many perceptual problems, this lower dimensional feature representation is just what is needed as a basis for further processing. We use this approach in some of the speech recognition studies reported here.

II. PRELIMINARY CONSIDERATIONS

Before backpropagation could be used to recognize speech, it was necessary to find a way to present the sound to the network. The speech signal itself changes with time, but the networks require fixed input and target samples. The approach we used here was to present the learning network with fixed input patterns, each of which consists of a set of sequential samples. In some cases, the individual samples in the input pattern were Fourier amplitude spectra; in other cases, the actual digitized sound samples were used. Preliminary studies were carried out to find the workable ranges for the number of frequencies and the amount of time that had to be represented in the input samples.

Another consideration was the way to normalize this data. The dynamic range of the Fourier spectra is large, and the vast majority of the points have very low values. We found that certain versions of the learning procedures had difficulty with this type of input, consisting of a vast sea of near-zero values with a few high peaks. The situation was further complicated by large amplitude differences between examples of the same sound.

We used two different strategies in preprocessing the input data. One strategy involved finding *ad hoc* methods to deal with the problems raised by the amplitude and dynamic range. The other strategy was to train a network to do the preprocessing itself. Details of these strategies will be given later.

III. DIRECT LEARNING OF PHONETIC LABELS

In our first series of studies, we asked how a backpropagation network might solve the problem of learning to label a set of highly confusable syllables. The basic idea was to use a spectrogram of a sound as the input pattern and a target bit pattern with one bit position for each of the types of sound. The task of the network was to learn to set the output unit corresponding to the sound type of the input to 1.0, while setting all the other output units to 0.0. We chose the syllables [ba], [bi], [bu], [da], [di], [du], [ga], [gi], and [gu] because this set of three voiced stops, paired with each of three vowels, is known to exhibit a high degree of variability due to coarticulation. Although listeners readily report all versions of (for example) the [d] as sounding the same, the acoustic patterns corresponding to the consonant differ greatly across the three vowel contexts. Indeed, this represents the paradigm case of perceptual invariance coupled with acoustic variability.

The stimuli for the experiment were prepared as follows. A single male speaker recorded a set of 505 tokens of the set of nine syllables (about 56 tokens of each syllable). Tokens were recorded in a moderately quiet environment, but with no particular effort at eliminating background noise; nor was an attempt made to ensure a constant rate of speech or uniformity of pronunciation. Recording was carried out through analog-to-digital conversion at a 10-kHz sampling rate and low-pass filtered at 3.5 kHz.

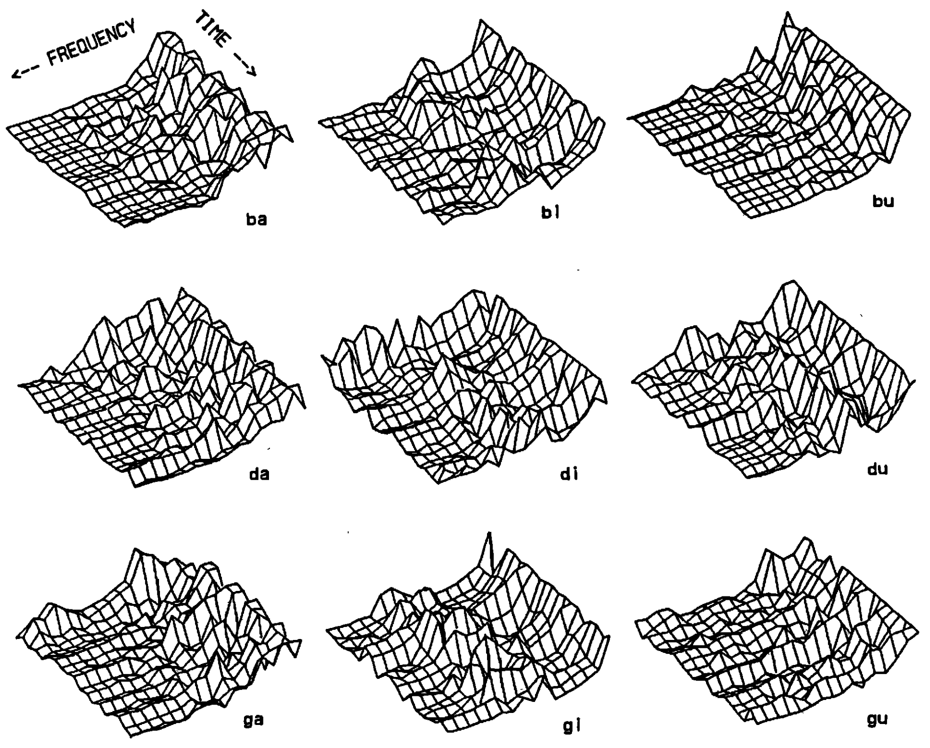

FIG. 2. Graphs of examples of the nine sounds [ba], [bi], [bu], [da], [di], [du], [ga], [gi], and [gu]. To generate these data, the release of each stop was located under computer control, and syllables were edited to include 5 ms before the release and sufficient time following release to allow 64 ms of FFT. A 64-point FFT was carried out over Hamming-windowed frames, advancing 32 points per frame. Each token consisted of an array of twenty 3.2-ms frames; within each frame, the FFT output was reformatted to give spectral magnitudes over 16 evenly spaced frequency ranges, then normalized to the average for the token. These values were then transformed using the logistic function $y = 1/(1 + e^{-\text{mag}})$. This procedure maps all values to the range 0 to 1, and the average to about 0.46. Since the median for the amplitude spectra is generally just a bit greater than the average, this procedure guarantees that about half of all values are mapped to each side of 0.5, which is beneficial for backpropagation learning.

The beginning of each consonant was located and an FFT analysis was carried out over 6.4-ms frames, advancing 3.2 ms per frame for 20 frames. The output of the FFT was reformatted to give spectral magnitudes over 16 frequency ranges. As a result, each token was represented as a 16×20 array of positive values. Finally, these FFT magnitudes were normalized to the token average and "squashed" using the logistic function $\text{squash}(x) = (1 + e^{-\text{mag}})^{-1}$. This confines the magnitude values to the range 0.0 to 1.0. Examples of the resulting values for several tokens are graphed in Fig. 2.

The network used consisted of 320 input units, between two and six hidden units (in different conditions), and as many output units as there were types to be labeled. The network used here and throughout this work was strictly layered; i.e., every unit of the input and hidden layer was connected to every unit on the layer above.

The data set of sounds was divided into two equal parts. One was used for training and the other was kept for testing performance on untrained examples. On each cycle of the teaching phase, the following occured: (a) One of the syllables from the training set was selected at random and applied to the input layer; (b) activation was propagated up from the input layer to the hidden layer, and from the hidden layer to the output layer; (c) the error for each output unit was calculated and backpropagated through the network, and the weights were adjusted according to the learning rule.

The trained network was tested by scanning through all members of the training and test data sets. The result for a token was scored as correct only if the value of the output unit that should be 1 was greater than 0.5, and the values of all the other output units were less than 0.5.

Three versions of simple phonetic labeling were run.[2] In one, nine labels were used, corresponding to the nine sound types. In the other two, three labels were used, corresponding either to the three vowels (ignoring the consonant in the same syllable) or else to the three consonants (ignoring the vowels). In all cases, more than 100 000 training cycles were run. This extensive training was used because we were interested in ultimate performance. Recently, we have found that higher learning rates give the same performance levels in about 10 000 training cycles.

In all three versions, the networks were always able to learn to label the training set perfectly; that is, there were no errors in the classification of either sound type (the syllable, the vowel, or the consonant). When presented with the data from the test set, the network trained to label whole syllables made an average of 16% errors. The networks trained to label vowels and consonants made an average of 1.5% and 7.9% errors, respectively. The results for the vowels and consonants were about the same whether two or three hidden units were used.

We found that the performance could be improved by introducing certain kinds of noise into the samples. Simply adding random noise to the inputs degraded performance. However, if the samples were distorted by adding noise proportional to each value, performance was significantly improved. This random distortion was accomplished by replacing each input value x by $(x + xr)$, where r is randomly chosen from the range -0.5–0.5. When tested without noise, the training sets still learned perfectly. On test data, the performance for syllables, vowels, and consonants was 10%, 0.3%, and 5.0% errors, respectively. This means that a recognizer based on vowels and consonants would have an accuracy of about 95%.

There are several reasons why training in noise should increase the model's ability to generalize. First, the noise effectively expands the data set; each syllable is represented by a larger number of exemplars. Second, and probably more important, the introduction of noise helps to blur stimulus idiosyncrasies that might be learned in place of the phonetically valid features. This results in greater error during the teaching phase, but better generalization.

Now let us turn to the hidden units. They restructure the input patterns in such a way as to provide input for the final (output) layer. In the process of carrying out this mapping, they encode the input patterns as feature types. One can ask what sorts of features become represented in the hidden units as a result of the teaching phase. These internal representations may provide a clue as to how the phonetic categorization is accomplished.

In order to visualize the relationship between hidden unit activity and input sound type, we used a technique that displays the average activity of each hidden unit at a different spatial position for each sound type. Examples of the hidden unit activity patterns obtained in this way are shown in Fig. 3. Every hidden unit has become absolutely associated with a subset of sound types. Hidden units have outputs of 1 for some sound types in this set and 0 for others. In addition, some hidden units produce a wide range of output values for tokens that they are not associated with. The association subsets can be vowellike or consonantlike, in that a unit is completely on or completely off for some consonant or vowel. In the example illustrated for the nine label case, for example, two of the hidden units are vowellike and two are consonantlike. Not infrequently a unit cleanly represents a single vowel or consonant in its on activity. It is interesting that each time the learning procedure is rerun, using different random initial weights, a different pattern of hidden unit correlations is observed. However, while several unit patterns occur often, some never appear at all. For example, no hidden unit has ever been found that represents the [u] sound alone by an on unit. This contrasts with [a] and [i], which can be so represented.

Another version of label learning was carried out in which a larger number of phonetic labels were employed, reflecting a finer-grain phonetic analysis. Each of the nine syllables was divided into a *consonantal* portion and a *vocalic portion*. The *consonantal* stimulus corresponded to the first 32 ms of the syllable (starting 5 ms before release of closure), and the *vocalic* stimulus corresponded to the 32 ms of the syllable that occurs 150 ms after the release of closure. This yielded 18 new stimuli. Each of the 18 stimuli types was given its own digital label, with labels randomly assigned to 9-bit codes. A network consisting of 320 input units, six hidden units, and nine output units was trained on 1 000 000 learning cycles of these 1010 (505×2) stimuli.

The correlations between hidden unit activity and

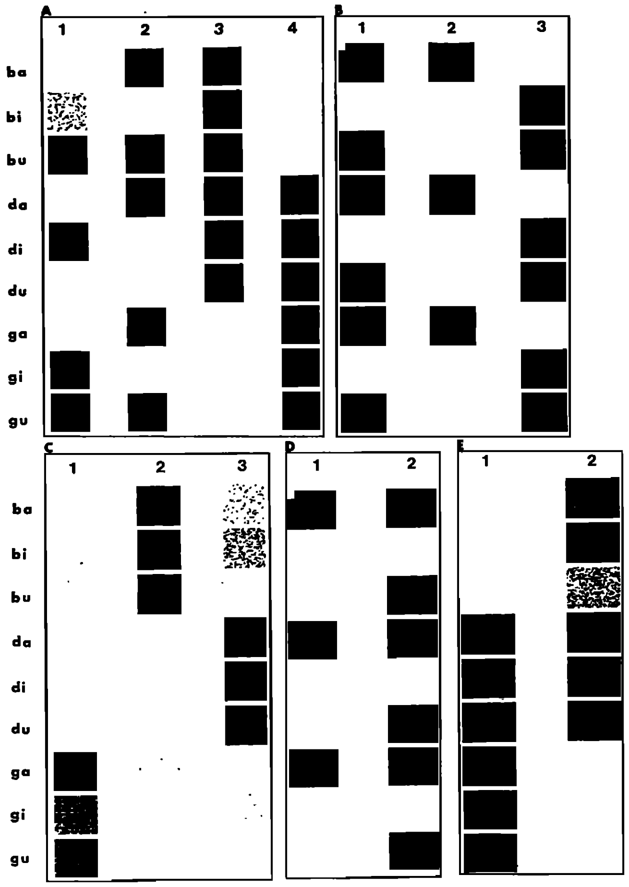

FIG. 3. Panels A–E display hidden unit response patterns for different versions of the backpropagation phonetic labeling networks. Each column shows the behavior of a single hidden unit for all nine sounds. The activity of the units is coded in the degree of darkening of the rectangle associated with each sound. A completely black rectangle indicates a unit with average activity of about 1.0 for that sound. Likewise, a white rectangle (not delineated against the background) indicates an average activity near 0.0. The shaded rectangles indicate intermediate average activities. Panel A is from a four hidden unit network trained with nine labels signifying syllables. Panels B and D are from networks trained with three vowel labels, while C and E were trained to recognize three consonants.

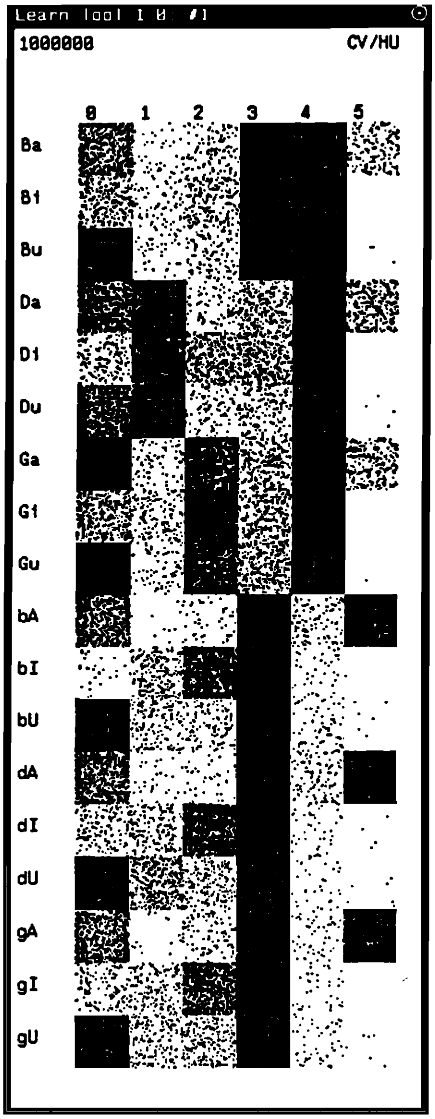

FIG. 4. A graph of the hidden unit activity associated with each of the 18 speech inputs. Each column shows the behavior of a single hidden unit for all 18 speech sounds. Upper-case letters indicate which portion of a CV syllable was presented (consonant or vowel); lower-case letters indicate the context.

sound type are displayed in Fig. 4. There are six columns, corresponding to the six hidden units, and 18 rows, corresponding to the 18 stimulus types. The phonetic segment is indicated by an upper-case letter, and its context by a lower-case letter. Thus "Ba" refers to tokens of a voiced bilabial stop, extracted from the syllable [ba], whereas "dI" refers to tokens of a high front vowel, extracted from the syllable [di].

Figure 4 allows us to look at the internal representation that has been developed in order to encode the 1010 tokens as 18 phonetic types. The representation is interesting in several respects. First, we see that one hidden unit (unit 4) is always on for the first nine types, and off for the last nine types. It thus serves as a consonant/vowel detector. Note that the learning task has not explicitly required that this distinction be drawn. The network has simply been asked to learn 18 phonetic labels. It happens that the consonant/vowel is a useful dimension along which to classify types, and this dimension is implicit in the stimuli. In other cases, we see that it is not as easy to interpret single hidden units by themselves. When both units 2 and 4 are on, a velar stop (Ga, Gi, Gu) is signaled; otherwise, the vowel [i] is indicated.

One very striking result is the response pattern for unit 1. This unit is always on (and only on) for the alveolar stops (Da, Di, Du). What makes this so surprising is that the alveolar stops exhibit a great deal of acoustic variability across different vowel contexts. The task simply required that the network learn labels for the three different alveolar allophones; it was not required to group these together in any way (indeed, there was no feedback that informed the network that any relationship existed between these three types). Nonetheless, the network has spontaneously determined a representation in which these allophones are grouped together.

The weights connecting the input to the hidden units are a kind of filter through which the sound stimuli pass to determine hidden unit activity. The shape of this filter is indicative of the sound features recognized by the hidden units.

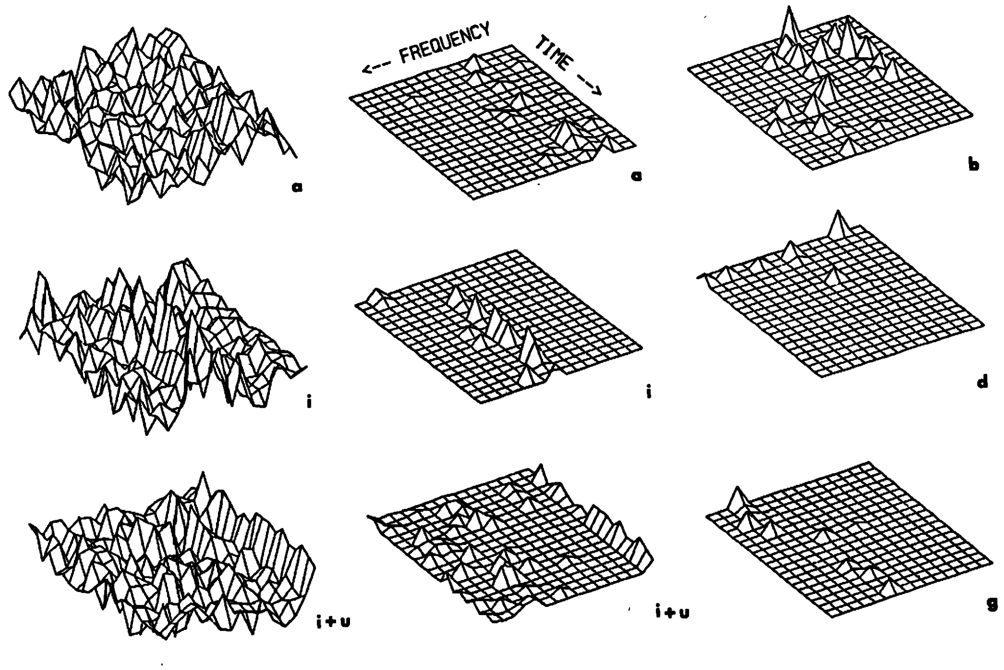

FIG. 5. These are graphs of the weights connecting the input sound array to the hidden units. Each graph represents a single hidden unit. The weights are plotted at positions that correspond to the frequency and time of their attached input. The format of time and frequency in these graphs is the same as in Fig. 2. The hidden units are all from networks trained to recognize either vowels or consonants. The hidden units are on only for the sounds indicated in the lower right-hand corner of each graph. The left-hand column has complete graphs. The other two columns have flood graphs, representing only that portion of the graph above a high tide of 0.7, and the whole range of weight values. All values below the high-tide value are set to the high-tide value. The center column is based on the same data as the one to the left. The complete graphs for the right-hand column are not shown.

Examining these weight profiles can give us some understanding of these features. Figure 5 shows graphs of the input weights for hidden units with outputs at 1 for only a single sound or a pair of sounds. In the column on the left, the weights for several vowel-recognizing units are depicted. The patterns are very complex and little can be gleaned from them. In the next column, a more interpretable "flood" plot of the same data is shown; the flood plot shows only those peaks above some "high-tide" level. The important differences between the weight arrays become apparent, showing some of the basis for distinguishing between the various sounds. The last column on the right is a flood plot of some consonant-recognizing units. The flood plots of the vowel-recognizing units and the consonant-recognizing units reveal only part of the story. The negative peaks are also important in the recognition process. And the importance of the finer scale structure of the weight matrix is not yet known.

Results from these studies of phonetic label learning indicate that this approach has considerable power and can be successful even given a highly confusable set of stimuli. Furthermore, the backpropagation technique results in internal representations whose interpretations are, in some cases, obvious; in other cases, the representations are novel and may suggest alternative ways of categorizing speech.

One important difficulty with this approach is the origin of the phonetic labels. The direct teaching technique requires that, for each speech stimulus, the correct label be known. It would seem desirable not to have to make this assumption. For example, from the viewpoint of child language acquisition, this requires the circular assumption that children know the labels of the sounds they are learning (that is, know which labels go with which sounds), before they learn them. This consideration led us to investigate identity mapping as a way to learn phonetic features.

IV. IDENTITY MAPPING OF SPECTROGRAMS

In the previous study, the input and output patterns were different; the input was a known speech stimulus, and the output was its abstract phonetic label. This interpretation of input and output is neither available nor necessary to the operation of the learning algorithm. It is simply learning a function that relates two patterns. One can apply the learning algorithm in a different mode in which the input and output pattern are the same. This mode, which we call *identity mapping* [it is also known as *auto-association* (Ackley *et al.*, 1985)], does not require an external teacher. Identity mapping a large pattern via a layer of a few hidden units has been shown to yield useful internal representations that give explicit information about the structure of the input patterns (Cottrell *et al.*, in press; Zipser, in press). We have used identity mapping with the same sounds described above to see if useful internal representations can be learned in the absence of *a priori* knowledge about the "meaning" or "names" of patterns. We find that the hidden units in identity mapping come to represent both previously identified speech features and new, not easily described, features.

The stimuli for this experiment were identical to those used previously. They consisted of 505 tokens of the nine consonant–vowel (CV) syllables, represented by normalized and squashed power spectra. The network had 320 input units, between 2 and 10 hidden units, and 320 output units. The training phase was similar to the labeling studies, except that the target output pattern was always identical to the input pattern.

In Fig. 6(a), we see an example of the hidden unit activations that developed after about 150 000 learning cycles. Unit 3 is a vowel unit since it is strongly on for all [a] and off for the other vowels. Units 2 and 4 are consonantlike units since they are on quite strongly for two consonants and off

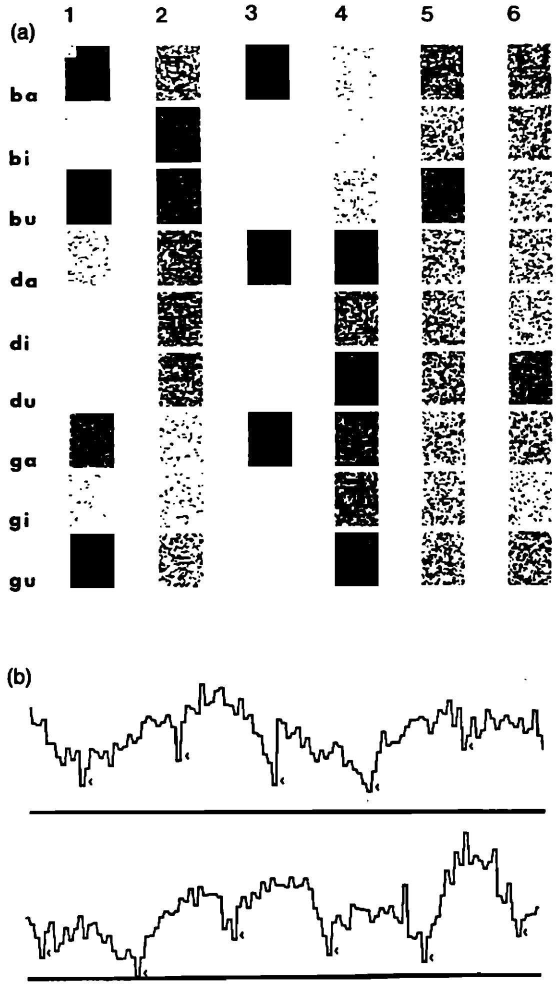

FIG. 6. (a) Hidden unit response patterns from a network that had been trained to identity map the nine syllables listed on the left-hand side. The average activity values are encoded in the same way as in Fig. 3. (b) Strip chartlike plots of the total error from an identity mapping network as sound tokens are continuously shifted through the input space as described in the text. The network is the same one whose hidden unit activities are shown in (a) above. The "<" marks indicate times when syllables are in register in the input space. The lower panel is a continuation of the right end of the upper one.

for a third. Unit 3 is vowellike but encodes some consonant information also. Units 5 and 6 cannot be characterized in terms of vowels and consonants; they represent some feature that is not easily described.

Different hidden unit activation patterns are obtained on each independent run of the same learning problem, but the same general kinds of hidden units are found. Sometimes hidden units represent a single vowel or consonant. More often they represent a strongly correlated encoding of mixed sound types, as is the case with unit 1. Units like 5 and 6, which recognize some enigmatic feature, are also quite common. In general, the fewer the number of hidden units, the more strongly correlated with sound types they become.

While the identity mapping network we have described was trained on speech that was not phonetically labeled, the speech tokens had been laboriously presegmented into syllables. It occurred to us that the identity mapping network might be able to segment continuous speech. The reason this might be possible is that error would be expected to be at a minimum when the sounds used for training were in register on the input units. These error minima would then signal the boundaries between syllables.[3]

To test this possibility, we synthesized a pseudocontinuous speech by stringing together examples of the nine sound types in random order and shifting this sequence through the input one time step per cycle. This resulted in a stimulus that had a complete sound token correctly in register with the inputs only once every 20 cycles. On all the rest of the cycles, the input consisted of part of the end of one token and part of the beginning of another. Networks that had been fully trained to identity map presegmented sounds were used for this study, but their learning mechansims was turned off. On each shift cycle the total er.or was computed. (This error is just the sum-squared difference between the input and output patterns.) The results are shown in Fig. 6(b).

The error signal has a clear periodic component, decreasing to an identifiable minimum each time a single token is in register with the input. The reason for this is that, when a CV syllable is in sync, the network "recognizes" one of the input patterns on which it has been previously trained. This results in low error. On the next testing cycle, the shifted input pattern still resembles one of the learned patterns so error is relatively low, but, as the shifting stimulus gets increasingly out of registration, the error increases. The results of this simplified study indicate that identity mapping networks might be used for segmenting continuous speech. One can also envisage using multiple networks to process speech in parallel. A network trained on identity mapping could be used to locate syllable boundaries; an error minimum could then be used to activate analysis by a second network that had been trained to do phonetic labeling.

We have seen that identity mapping can be used to learn salient features without labeling and may also be useful in segmenting speech. But we are limited by the need to presegment the input for training purposes. This is an undesirable limitation because it requires a teaching environment that may be richer than that available to the human learner. In the next section, we try to remove the requirement for presegmentation of the sound stimulus.

V. IDENTITY MAPPING OF CONTINUOUS SPECTROGRAMS

The goal in this study was to see what sort of representation might result if a network were trained on continuous speech. As in the previous study, the task is to identity map the input. In this case, we drop the restriction that the input must correspond to a CV syllable. Instead, the speech input consisted of a corpus of 15 min of running speech. A text was created that contained: (a) the digits from 0 to 9; (b) the 500 most frequent words from the Kucera–Francis corpus (Kucera and Francis, 1967); (c) a phonetically balanced word list of 100 items; and (d) a prose passage. The corpus was read in a conversational manner by a male speaker at a moderate rate under relatively quiet conditions, filtered at 3.5 kHz, and digitized at a 10-kHz sampling rate. The speech was Hamming windowed and analyzed by a 128-point FFT

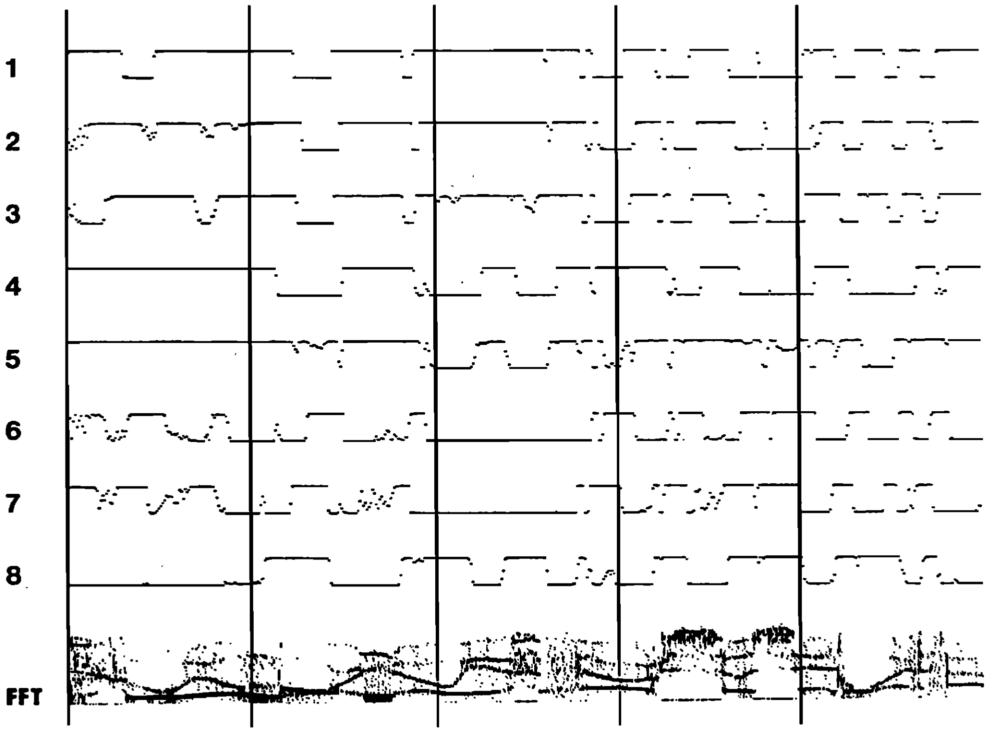

FIG. 7. Hidden unit activity from a network that has been trained on a large, unsegmented speech corpus. The output activity of the eight hidden units is plotted on strip chartlike graphs as a sample of the speech corpus is shifted through the input space of the network. The bottom segment of the figure shows an FFT of the sound to which the hidden units are responding. The vertical timing lines are 600 ms apart. The numbers 1–8 on the left-hand side indicate the individual hidden units.

using overlapping windows that advanced 64 points per frame. The power spectra were reduced to 32 frequency bins, normalized and squashed in a manner similar to that described above.

The network was made up of three layers; the input layer contained 640 units, the hidden layer had eight units, and the output layer contained 640 units.

Given the magnitude of the corpus, the computational requirements of learning such a database to an acceptable level of error are considerable. Pilot studies ran for approximately 2 weeks on a VAX11/750 digital computer (with FPA). For this reason, we carried out further studies on the Cray XMP-4 computer of the San Diego Super Computer Center.

The network was trained on the corpus for 1 000 000 learning cycles. We experimented with two modes of presentation. In one mode, the speech was passed through the input layer of the network in a continuous fashion; after each learning cycle, the input layer of the network was advanced by one time interval so that there was considerable overlap from one cycle to the next. In the second mode, a section of the corpus was selected at random for identity mapping; eventually, all possible (overlapping) portions of the utterance were seen by the network. In pilot work, we found no differences between the two modes; the random mode is the one we adopted for our studies.

At the completion of the learning phase, the network had been trained on an extensive body of speech. The speech corpus contains approximately 140 000 different input patterns (each pattern consisting of 640 numbers). Our hope was that this sample was both repesentative of the variety of speech patterns for the speaker, while, at the same time, containing enough regularity that the network would be able to successfully encode the patterns. One graphic view of the representation that is built up is shown in Fig. 7. At the bottom of the figure we see a spectrogram of a section of the training corpus. Shown above are eight lines that graph the activations of the hidden units when the speech shown at the bottom is passed through the network. These plots can be thought of as a kind of feature representation of the speech. It is clear that features have steady states that last for roughly syllable-sized periods of time. It has proved difficult, however, to give an interpretation of the content of these features. An important question is how much information has been preserved by the encoding contained in the eight hidden units. One can test this by seeing whether it is possible to teach a network the phonetic labels for speech sounds when we use the hidden unit representation of the identity mapped speech rather than the speech itself. This involves two steps. First, we take the nine CV syllables used before, pass them through the network previously trained on the Cray (using the 15-min speech corpus), and then save the hidden unit activations that result. In the second step, we use the hidden unit activations to train a second network to label the activation patterns as [ba], [bi], [bu], etc. This step is analogous to the previous labeling studies, with the important difference that the input now is not speech but the representation of speech derived from the Cray-trained network. The labeling network had 560 input units (to accommodate 70 time slices, each time slice lasting 6.4 ms and being represented by eight hidden unit values), 8 hidden units, and 9 output units. The nine output units were used to encode the nine different syllable types.

After approximately 100 000 learning cycles, the hidden unit activity of the labeling network had a reasonably distinct pattern that distinguished the nine different syllables. A more rigorous test is to see how many categorization errors are made by this network. There is an overall error rate of 13.5% (false reject = 10.4%, false accept = 3.1%), but most of it is due to a very high error rate in labeling [ga]. If this syllable is removed, the overall error rate drops to 5.8%.

This result is quite encouraging. It indicates that a degree of dimension reduction has been achieved by the corpus-trained network using only eight hidden units, without an enormous loss of information. The encoded representation is rich enough that the identity of the original speech can still be extracted. Furthermore, we also see that we were able to create a feature representation without segmenting the data or knowing its phonetic identity. Indeed, features obtained in this way may eventually be useful for the task of segmentation.

One assumption we made in these studies was that the power spectra of the speech provides a good initial representation of the speech. This is not an unreasonable assumption, and there are, in fact, many additional assumptions one might have made (such as the use of critical bands, preemphasis of higher frequencies, etc.). Nonetheless, an important goal of this investigation has been to see how much of the structure of speech could be discovered with minimal *a priori* assumptions about what were meaningful transformations or representations of the data.

In this spirit, we wondered what might happen if we abandoned the use of the FFT to train the network. Suppose we simply presented the network with the unanalyzed digitalized waveforms? This is what we did in the final study.

VI. IDENTITY MAPPING OF CONTINUOUS RAW SPEECH

In this study, we attempted to see whether a useful input representation could be built up from continuous speech using the digital waveform itself.

The first speech corpus we used consisted of the simple sentence, *This is the voice of the neural network* (with a duration of approximately 4 s). The speech was kept as a series of 16-bit samples, each value representing an A/D converter voltage, with samples occurring at 100-μs intervals. The network was made of 50 input units, 20 hidden units, an 50 output units. As in the previous experiment, random sections of the corpus were selected and presented as input and target for identity mapping. Since each such section contained 50 samples, the network's input window covered 5 ms of speech.

The use of pulse code modulated (PCM) speech in identity mapping makes it very easy to test the network's performance simply by collecting the output and converting it back to analog form (since the network output is then simply a set of digital values that can be passed through a D/A conveter). This requires that the output layer contain linear units; so, while the hidden layer remained nonlinear, the output layer was linear.

After one million learning cycles, we fixed the weight values (so that no further learning could occur) and fed the whole training data set through the network as a sequential stream of nonoverlapping 50 sample inputs. The output was converted to analog form and played over speakers. The authors judged the speech to be of high quality; a 0.96 correlation between input and output was obtained. Spectrograms of both the input and output are shown in Fig. 8.

We were interested in seeing how well the network weights would generalize to novel speech. To test this, we retrained the network using 4 min of the full speech data base that was used in the Cray training study (but in PCM form rather than as power spectra). We reasoned that this larger and more varied training set would be needed in order, to learn features that would have general applicability. Learning proceeded for one million cycles, using the same presentation method as with the simple sentence. It is worth noting that, because this data set contained 2.5 million different 50-sample patterns, less than half the data was seen, and any pattern that was presented was typically seen only once.

The resultant network was then used as a filter for the original neural network sentence. A spectrogram of the output is shown in Fig. 9; the output is somewhat degraded compare with the filter that is trained on the sentence itself, but is still quite understandable.

One natural question to ask is what kind of encoding the

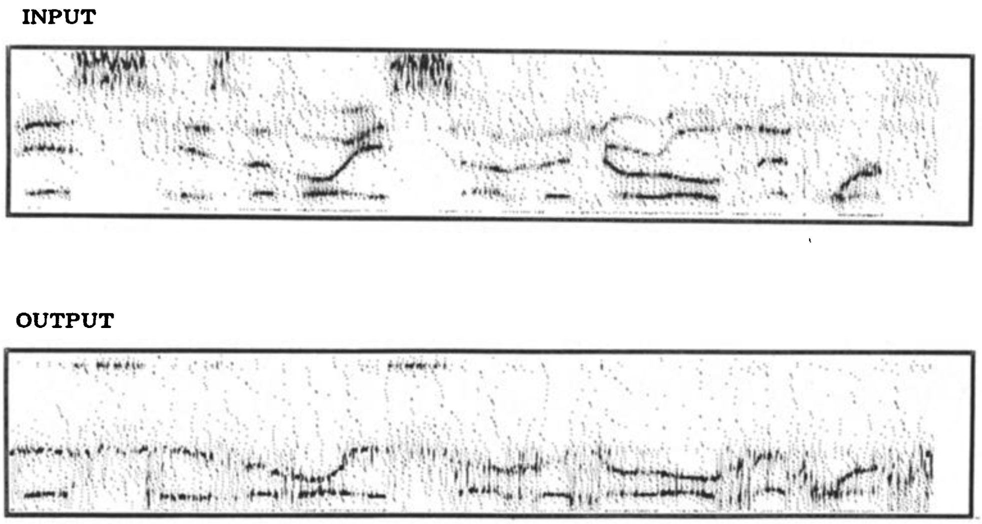

FIG. 8. Spectrograms of both the input and output of a network trained to identity map PCM speech..

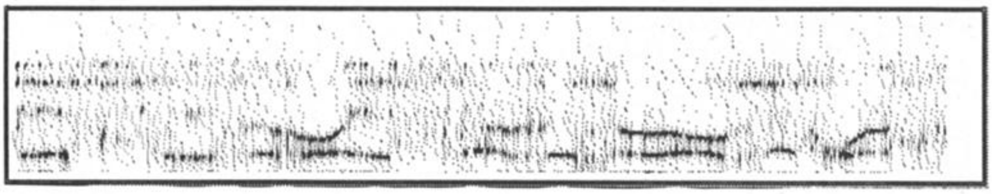

FIG. 9. A spectrogram of the output of the network of Fig. 8 given as input the "neural network" sentence that it had never seen.

hidden units have discovered. In Fig. 10, we see spectrograms of the outputs of the individual units in response to the neural network sentence. It is clear that the responses do not resemble single sines or cosines, showing that the units have not learned a Fourier decomposition. One thing that is striking is the extent to which the hidden units' spectral responses are similar. If one compares the spectrogram of the input sentence itself (Fig. 8) with those of the hidden units, one sees the way in which this is so. Most of the hidden unit response frequencies tend to center around the regions of the spectrum that are relevant for speech (this is not the case for all units; there are some that are distinctly different). The units have thus concentrated mainly on those areas of the spectrum that are relevant for encoding the speech data. The results obtained here with sound are analogous to those found for visual images by Cottrell *et al.* (in press). Since these authors were able to demonstrate considerable bandwidth compression using the hidden unit representations, we would expect that bandwidth compression has also occurred for sound.

The time scale of the hidden unit features is quite short (the features encode events on the order of 5 ms). We are

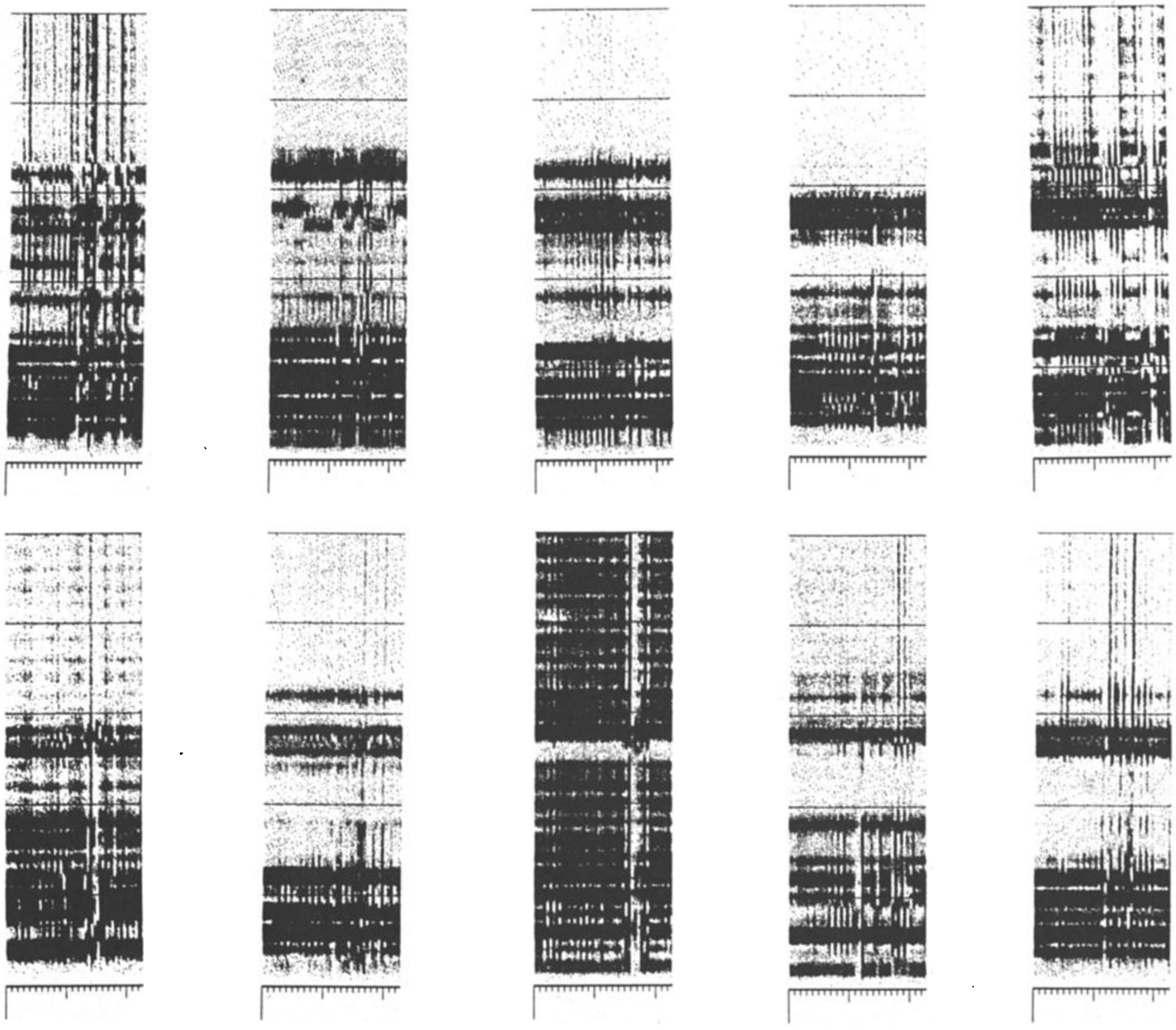

FIG. 10. Spectrograms of the outputs of the 10 (of the 20) individual hidden units, from the network of Fig. 9, in response to the "neural network" sentence. Time on the abscissa is marked in 10-ms intervals and frequency on the ordinate in 1000-Hz intervals.

primarily interested in features, but also in representations that encode larger events. Therefore, we constructed a second-level network that took as its input the hidden unit representation derived from the first network and identity mapped it to an output layer. This second network had 400 input units, 10 hidden units, and 400 output units. It was able to look at representations corresponding to 100 ms (i.e., it saw 20 groups of 20 hidden unit inputs at a time, and each group of 20 hidden units came from 5 ms of speech in the first network).

The first network in this two-level system was the one trained on the extended speech corpus, and had its weights fixed. It was then given a repeated sequence of nonsense syllables, [ba] [ba] [ba] [ba] (in PCM form), as a continuous stream of nonoverlapping 5-ms inputs. Every 5 ms, the 20 hidden unit activations from this first network were passed to the input layer of the second network. After 400 such units had been collected by the second network, it did 1 cycle of learning in an identity mapping mode. On each succeeding learning cycle, the input pattern in the second network was shifted left by 20 units, and a new 20 units were received from the first network. This sequence of events continued until approximately 800 000 learning cycles had occurred in the second network (remember that the first network had already been trained and was not subject to learning; it was simply acting as a preprocessor that formated the speech in a manner analogous to the FFT used previously).

After the second network was trained on the hidden unit representations, we froze the weights in the second network. We then ran the input through both networks in a continuous stream. As we did this, we examined the pattern of ten hidden unit activations in the second network. These hidden unit patterns were apparently associated with syllable onsets.

VII. CONCLUSION

The series of experiments reported here is clearly preliminary in nature. There are a large number of questions that are raised by this work, and it is easy to think of many alternative ways of posing the problems we have presented in the networks. Nonetheless, we find this approach to analyzing and recognizing speech exciting for a number of reasons.

(1) *Power*. The domain of speech processing is an extremely difficult one. There are a large number of problems that remain unsolved of both a practical and theoretical nature. We believe that the experiments described here demonstrate that the PDP framework and the backpropagation method for learning are extremely powerful.

In the first labeling study, we saw that it was possible to build a system that could be taught to correctly categorize a number of highly confusable phonetic segments; and that, having learned this, the network was able to generalize the categorization to novel data. We are optimistic that the performance—which was good—can be improved with refinements in the technique. We are particularly impressed with the fact that an encoding was found in which one hidden unit became active whenever an alveolar stop was presented, regardless of vocalic context, and of another which did the same for velar stops. It is well known that both of these consonants exhibit a great deal of contextual variability, and the spontaneous discovery of an invariant feature is interesting.

(2) *Representations*. An important goal in this work was to study the representations that result from applying the backpropagation learning algorithm to speech. In some cases, the representations that are discovered are intuitively sensible and easy to interpret. In the labeling studies, the consonant/vowel distinction was encoded by a single unit. In other cases, we saw that the representation itself assumed a distributed form, with groups of hidden units participating in, for example, the encoding of place of articulation. It is interesting that the specific representations may vary when learning experiments are replicated. This suggests that multiple networks learning the same data may provide a richer representation than any single network. Finally, we saw in the PCM identity mapping studies that the algorithm finds solutions that are highly expedient. The spectral decomposition that was carried out was clearly tuned for speech; the majority of units had responses that focused on several regions of the spectrum, and these were precisely those regions that are highly relevant for speech.

(3) *Innate versus learned representations*. We do not believe that the work described here necessarily makes strong claims that the perceptual representations by humans are learned. On the other hand, we believe that the work does argue *against* making strong claims that such representations must be innate. The tendency, in linguistics perhaps more than psychology, has been to assume that much of the representation apparatus used in processing language must be innate. In large part, that is because it has seemed to many people that the representations are complex and often arbitrary and that the input data available to the language learner for those representations are impoverished.

We feel that this study encourages the belief that more information about the structure of speech is extractable from the input than has been supposed. The back-propagation method of learning may not, in fact, be what is used by humans. Still, it at least demonstrates that one relatively simple algorithm does exist that is capable of uncovering a great deal of structure in a small sample of speech. It is our hope, based on these preliminary studies, that it will be possible to construct a hierarchy of learning networks that will spontaneously learn to recognize speech using only extensive examples of input speech, loosely synchronized with transcribed text. We further hope that this task can be accomplished in such a way as to shed light on the actual mechanisms used by the brain.

ACKNOWLEDGMENTS

We wish to thank Dave Rumelhart for many useful discussions, and Jeffrey Teeters, Arthur Shapiro, and David Pare for programming assistance. This work was supported by Contract No. N00014-85-K-0076 from the Office of Naval Research, a grant from the System Development Foundation, and the Air Force Office of Scientific Research, Air Force Systems Command, USAF, Grant No. AFOSR-86-0062.

[1] Biases provide constant inputs to units. They may be thought of as the weighted input from a unit whose activation is always 1.0; the weight from this unit to every unit in the network is learned just like other weights in the network.

[2] The simulations reported in this article were run on several different machines: a Symbolics 3600 Lisp machine, a VAX 11/750 (with floating point accelerator), and a Sun 3/160 (with floating point accelerator). Simulations involving smaller data sets and running for a few thousand iterations took a few hours to complete. The longest simulation ran for 1 000 000 iterations and took 4 days.

[3] Kohonen *et al.* (1984) have used a similar scheme in their speech recognition system.

Ackley, D., Hinton, G. E., and Sejnowski, T. (**1985**). "A learning algorithm for Boltzmann machines," Cognitive Sci. **9**, 147–169.

Aronoff, M. (**1976**). "Word formation in generative grammar," in *Linguistic Inquiry Monograh 1* (MIT, Cambridge, MA).

Cole, R. A., Stern, R. M., and Lasry, M. J. (**1986**). "Performing fine phonetic distinctions: Templates vs. features," in *Invariance and Variability in Speech Processes*, edited by J. S. Perkell and D. H. Klatt (Erlbaum, Hillsdale, NJ).

Cottrell, G. W., Munro, P. W., and Zipser, D. (**1987**). "Image compression by back propagation: A demonstration of extensional programming," in *Advances in Cognitive Science*, edited by N. E. Sharkey (Ablex, Norwood, NJ), Vol. 2.

Dixon, N. R., and Silverman, H. F. (**1976**). "The 1976 modular acoustic processor (MAP)," IEEE Trans. Acoust. Speech Signal Process. **25**, 367–378.

Fujimura, O., and Lovins, J. B. (**1978**). "Syllables as concatenative phonetic units," in *Syllables and Segments* edited by A. Bell and J. B. Hooper (North-Holland, Amsterdam).

Klatt, D. H. (**1980**). "Scriber and LAFS: Two new approaches to speech analysis," in *Trends in Speech Recognition*, edited by W. A. Lea (Prentice-Hall, Englewood Cliffs, NJ).

Kohonen, T., Riittinen, H. Reuhkala, E., and Haltsonen, S. (**1984**). "Online recognition of spoken words from a large vocabulary," Inf. Sci. **33**, 3–30.

Kucera, H., and Francis, W. (**1967**). *Computational Analysis of Present-Day American English* (Brown U.P., Providence, RI).

Le Cun, Y. (**1985**). "A learning procedure for asymmetric threshold network," Proc. Cognitiva **85**, 599–604.

Lowerre, B. T. (**1976**). "The Harpy speech recognition system," Doctoral dissertation, Carnegie-Mellon University, Pittsburgh (unpublished).

Mehler, J. (**1981**). "The role of syllables in speech processing: Infant and adult data," Philos. Trans. R. Soc. London Ser. B **295**, 333–352.

Parker, D. B. (**1985**). Learning-Logic (TR-47) Tech. Rep. Center for Computational Research in Economics and Management Science, MIT, Cambridge, MA.

Pisoni, D. B. (**1981**). "In defense of segmental representations in speech perception," J. Acoust. Soc. Am. Suppl. 1 **69**, S32.

Rosenblatt, F. (**1962**). *Principles of Neurodynamics* (Spartan, New York).

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (**1986**). "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1. Foundations*, edited by D. E. Rumelhart and J. L. McClelland (MIT, Cambridge, MA), pp. 318–362.

Wickelgren, W. A. (**1969**). "Context-sensitive coding, associative memory, and serial order in (speech) behavior," Psychol. Rev. **76**, 1–15.

Zipser, D. (**in press**). "Programming neural nets to do spatial computations," in *Advances in Cognitive Science*, edited by N. E. Sharkey (Ablex, Norwood, NJ), Vol. 2.