PROGRAMMABLE EXECUTION OF MULTI-LAYERED NETWORKS FOR AUTOMATIC SPEECH RECOGNITION

A set of Multi-Layered Networks allows the integration of information extracted with variable resolution in the time and frequency domains and to keep the number of links between nodes of the networks small for significant generalization during learning with a reasonable training set size.

YOSHUA BENGIO, RÊGIS CARDIN, RENATO DE MORI and ETTORE MERLO

Important efforts have been devoted in recent years to the coding of portions of the speech signal into *representations*. Characterizing Speech Units (SU) in terms of speech properties or speech parameters requires a form of *learning* with a relevant *generalization* capability. Structural and stochastic methods have been proposed for this purpose in [6, 9].

Recently, a large number of scientists have investigated and applied learning systems based on Multi-Layered Networks (MLN). Definitions of MLNs, motivations and algorithms for their use can be found in [3, 8, 11, 12, 13, 14]. Theoretical results have shown that MLNs can perform a variety of complex functions [12]. Furthermore, they allow competitive learning with an algorithm based on well established mathematical properties.

Our interest in the use of MLNs is justified by previously published work. We have introduced a data-driven paradigm for extracting acoustic properties from continuous speech [7], and have investigated methods based on fuzzy or stochastic performance models for relating acoustic properties with SUs. MLNs appear to be good operators for automatically learning how to extract acoustic properties and relate them with phonetic features and words automating most of the activity that formerly required a large amount of effort from a human expert. The human expert used knowledge acquired by generalizing observations of time-frequency-energy patterns. In this article we will investigate how such learning can be performed by a set of MLNs whose execution is decided by a data-driven strategy.

By applying an input pattern to an MLN and clamping the output to the values corresponding to the code of the desired output, weights of connections between MLN nodes can be learned using error-back propagation [11]. When a new input is applied to an MLN, its outputs may assume values between zero and one. If we interpret each output as representing a *phonetic property*, then the output value can be seen as a degree of evidence with which that property has been observed in the data [4].

If phonemes are coded using a known set of phonetic features, the MLNs will learn how to detect evidence of each feature without being told all the details of the acoustic properties relevant for that feature.

Statistical distributions of feature evidences can be collected in performance models of SUs conceived as Hidden Markov Models (HMM). These models can be used to represent the time evolution of feature evidences for each SU or word. It is also possible to compute distances between time evolutions of real and desired degrees of evidences and to use such distances to rank word hypotheses, each word being characterized by a desired time evolution of degrees of evidences.

ORGANIZATION OF MULTI-LAYERED NETWORKS

Figure 1 shows the general scheme of an MLN. The input layer is fed by a *Property Extractor* (PE), that acts

This work was supported by the Natural Science and Engineering Council of Canada (NSERC) and Fond Concerté d'Aide à la Recherche (FCAR) of the province of Québec. The Centre de Recherche en Informatique de Montréal (CRIM) kindly provided computing time on its facilities.

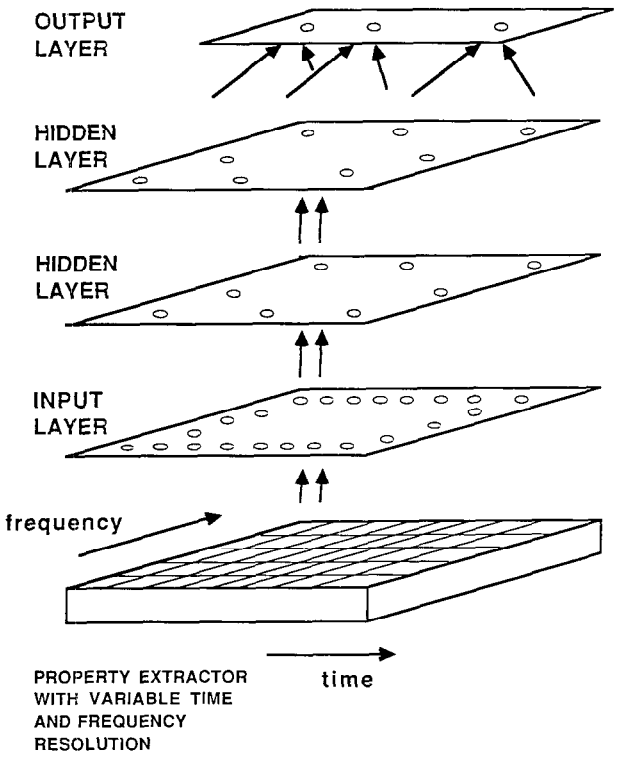

FIGURE 1. Multi-Layered Network with Variable Resolution Property Extractor

as a window analyzing the data with variable time and frequency resolution. PEs may also extract data from the speech waveform.

The MLN in Figure 1 has two hidden layers and one output layer. Different MLNs may be used concurrently.

The following considerations motivate the use of different PE extractors and of different MLNs:

- In the speech signal there are events characterized by abrupt transients. A plosive sound or the beginning of an utterance may produce events of that nature. In such situations, indicated as S_1, it is important to analyze possible bursts requiring a PE with high time resolution and not very high frequency resolution in high frequency bands.
- It is also important to detect voicing with a PE spanning low frequencies for a relatively long time interval and to analyze possible formant transitions with PEs examining frequency bands between 0.3 and 3.5 kHz.
- Recognition performance is improved by taking into account acoustic properties related to the morphology of time evolution of certain speech parameters following the approach proposed in [6].

The network in Figure 2 shows five PEs. Most of them are positioned on a speech spectrogram at the onset time after a silence, a buzz-bar or a frication noise interval.

The PEs are mostly rectangular windows subdivided into cells as shown in Figure 1. A vector of time resolutions (VT) and a vector of frequency resolutions (VF) describe the size of the cells in each PE (time values are in msecs, frequency values are in kHz). A symbol `t*` is inserted into VT to indicate the time reference for the position of the window. The PEs introduced in Figure 2 have the following VTs and VFs:

```
PE11:  VT = (30,30,t*,10,10,10,10)
       VF = (0.1,0.25,0.3,0.5)
```

meaning that two time intervals of 30 msecs each are analyzed before `t*` and four time intervals of 10 msecs each are analyzed after `t*`. The analysis is based on filters whose bands are delimited by two successive values of VF. There are 20 nodes on the first layer above PE11 and 10 nodes of the second layer.

PE12 has 39 filters each spanning three successive time intervals of 40 msecs. The filter bandwidth is 200 Hz, the position in frequency is decided based on spectral lines as defined in [10], the first filter contains the spectral line that corresponds to the first formant in the last time interval. This allows speaker normalization by aligning filters with spectral lines. Default conditions are established in order to ensure that filters are positioned in prescribed frequency ranges even if spectral lines are not detected.

Each filter receives at its input a normalized value of the energy in its time-frequency window. For each filter there is a corresponding input that receives points of spectral lines detected inside the window corresponding to the time and frequency resolutions of the filter. There are 20 nodes in the first hidden layer above PE12 and 10 nodes on the second hidden layer.

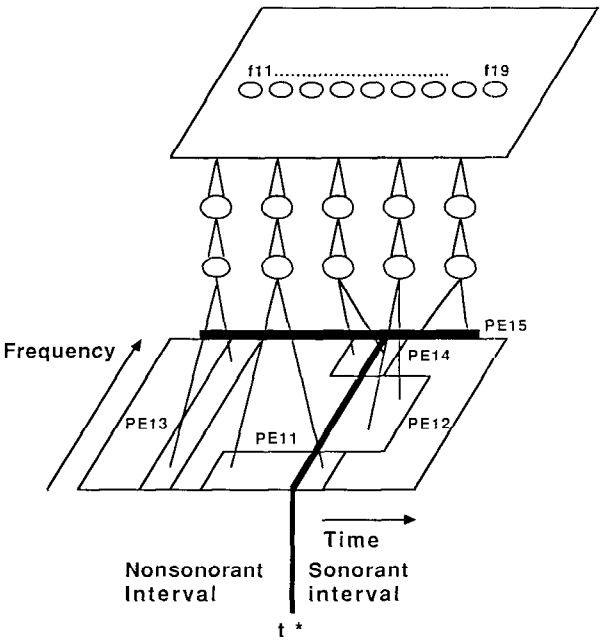

FIGURE 2. Property Extractors of MLN1

PE13 is supposed to capture properties of frication noise and is characterized by the following vectors:

```
PE13:  VT = {20,t*,20}
  VF = {1,2,3,4,5,6,7,8,9},
```

PE13 is executed every 20 msecs in the frication interval. It has 16 nodes in the first layer and 10 nodes in the second layer.

PE14 captures properties of the burst, when there is one, and is characterized as follows:

```
PE14:  VT = {5,5,t*, 5,5}
       VF = {2,3,4,5,6,7}.
```

PE15 receives at its input normalized properties of the time evolution of energies in various bands as well as properties extracted from the speech waveform. This is a subset of the properties defined in [6] and contains those properties not included in what is extracted by the other PEs. There are 20 nodes in the first layer above PE14 and PE15 and 10 nodes in the second layer.

Let MLN1 be the network shown in Figure 2. It is executed when a situation characterized by the following rule is detected:

```
SITUATION S1
  ((deep_dip)(t*)(peak)) or
  ((ns)(t*)(peak)) or
  (deep_dip)(sonorant-head)(t*)(peak))
    → execute(MLN1 at t*)
```
(1)

(deep_dip), (peak), (ns) are symbols of the PAC alphabet representing respectively a deep dip, a peak in the time evolution of the signal energy and a segment with broad-band noise; t* is the time at which the first description ends, sonorant-head is a property defined in [7]. Similar preconditions and networks are established for nonsonorant segments at the end of an utterance.

The output in Figure 2 corresponds to the features defined in Table I.

For example, phoneme /b/ will be described by the following values: (f11 = 1, f12 = 0, f13 = 0, f14 = 1, f15 = 0, f16 = 1, f17 = 0, f18 = 0, f19 = 0). The code in Table I is redundant and can be modified. We have chosen it because MLNs give degrees of evidence for each output (feature) and it will be possible to compare the performance of MLN 1, in which property extraction is based on automatically derived algorithms (i.e., learned with the weights) with past work done on property extraction performed by algorithms designed by a human expert [4].

The features defined in Table I have been used for recognizing letters of the E1 set defined as follows:

```
E1: {b,c,d,e,g,k,p,v,3}
```
(2)

In the learning phase the outputs were clamped according to the coding scheme of Table II.

By adding six other features to those in Table I, all the phonemes can be represented with phonetic features.

It was suggested that in the theory the number of examples required for obtaining a good generalization

TABLE I. Features Corresponding to Output Code of MLN1

Output	Feature
f11	voiced
f12	unvoiced
f13	sonorant
f14	plosive
f15	fricative
f16	labial
f17	alveolar
f18	palatal
f19	dental

TABLE II. Word Coding for the E1-Set

Word	f11	f12	f13	f14	f15	f16	f17	f18	f19
b	1	0	0	1	0	1	0	0	0
c	0	1	0	0	1	0	1	0	0
d	1	0	0	1	0	0	1	0	0
e	0	0	1	0	0	0	0	0	0
g	1	0	0	0	1	0	0	1	0
k	0	1	0	1	0	0	0	1	0
p	0	1	0	1	0	1	0	0	0
t	0	1	0	1	0	0	1	0	0
v	1	0	0	0	1	1	0	0	0
3	0	1	0	0	1	0	0	0	1

in MLNs grows with the number of weights (links) [11]. For this reason, a set of networks wherein each operates with a limited number of input nodes on a limited frequency interval can be better trained with a learning set of reasonable size. Furthermore, in spite of the theoretical complexity, a good generalization can be obtained with a reasonable number of experiments if the properties extracted from the inputs are chosen in such a way that a limited variation can be expected in their values if many speakers pronounce the same sound. (Experimental results on the recognition of the E1 set will be described in the section entitled "Use of MLNs in a Recognition System.")

Sonorant segments can be extracted from continuous speech using a procedure described in [5]. Sonorant segments are characterized by narrow-band resonances from which spectral lines as introduced in [10] are extracted. For sonorant segments an MLN called MLN2 is used.

MLN2 is executed in situation S_2 characterized by peaks and high energy valleys of the signal energy in which frication noise has not been detected. MLN2 is applied every 20 msecs in the sonorant region.

The approach just introduced can be generalized in a formal way. In general, MLNs may have PEs that act as running windows advancing on a spectrogram by fixed time intervals. For each time reference, the output of the invoked MLNs is represented by a vector M of degrees of evidence for each of the feature outputs:

$$M = \{\mu_1, \mu_2, \ldots \mu_j, \ldots \mu_J\} \quad (3)$$

where μ_j is the degree of evidence of feature f_j. As time reference varies from the beginning to the end of the speech signal, if T is the sampling interval and nT is the nth set of feature evidences, these evidences will be represented by a vector M(n):

M(n) represents a code for a speech interval. Time evolutions of feature evidences can be used for building word models conceived as Markov sources.

USE OF MLNs IN A RECOGNITION SYSTEM

The speech signal is initially segmented into two types of regions. These regions and transitions between them define situations. Each region is labelled with one of the following symbols: SON (attached to a segment with narrow band resonances—typically vowels, nasals and sonorant consonants), NS (attached to segments with a spread of energy in higher frequencies—typically fricative sounds, or to segments with a very low total energy). Each type of segment may contain every phoneme, but different sets of MLNs and MLN inputs are used for different types of segments.

The system that performs a data-driven execution of MLNs is described by a Procedural Network (PN). A general purpose environment for developing various types of PNs has been developed [7]. A simple PN has been conceived to implement the programmed execution of MLNs. It can be described as a supervised Augmented Transition Network (ATN) with measures associated to its arcs. A supervisor implements a dynamic programming strategy which combines the measures of different hypotheses in order to find the one which is optimal according to some criterion. The transitions may have associated conditions (COND), actions (EXE), hierarchical nesting (PUSH), default conditions (DEFAULT) and termination (POP). The semantic of these conditions and actions is detailed in [7].

The speech signal is considered as a sequence of segments of type SON, NS, and of transitions between segments.

In Figure 3 the step corresponding to the analysis of a transition followed by an NS or SON segment is shown. This step is repeated as many times as the length of the signal requires.

Figure 4 shows the PN representation of a two-steps variable-depth strategy associated to a segment. Depending on the preconditions, a particular set of MLNs is activated. The variable-depth paradigm is described in the following. If two or more candidates have scores which are close enough to trigger the variable-depth analysis, then an MLN specialized to solve the specific conflict is executed. If the candidates are well discriminated, the execution of specialized MLNs is not required and the default transition is taken. The number of conflict sets is finite and small. Several stages of

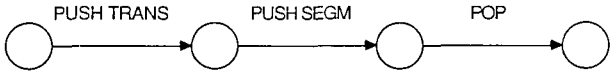

FIGURE 3. Analysis of a Transition Followed by a Segment

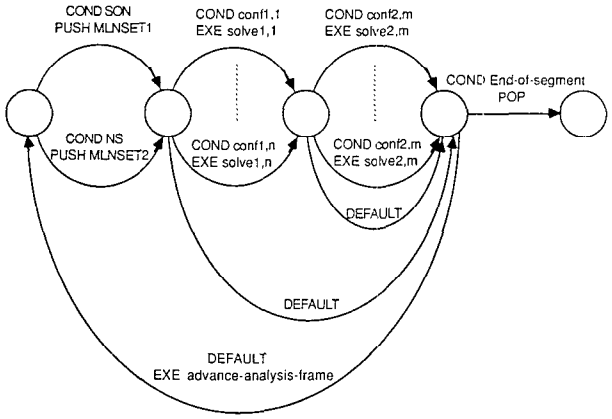

FIGURE 4. Two-Steps Variable-Depth Strategy

variable-depth analysis can be considered, although, in practice, there are no more than two of them. Variable-depth analysis is particularly useful, for example, to discriminate between /m/ and /n/ when these sounds have close degrees of evidence or to discriminate among pairs of plosive sounds.

EXPERIMENTAL RESULTS

In order to test the ideas proposed in this paper, the E1-set as defined in (2) as well as other words were used.

The 10 words of the E1 set were pronounced twice by 80 speakers (40 males and 40 females). Data acquisition was performed with a Hewlett-Packard 9000-236 especially equipped for speech processing. Sampling rate was performed with a 12-bit A/D converter at 16 kHz. Learning and recognition were performed on a Vax 8650. The data from 70 speakers were used as a training set while the data from the remaining 10 speakers (6 males and 4 females) were used for the test. The MLN was positioned at t* detected by an algorithm described in [5].

An overall error rate of 9.5 percent was obtained with a maximum error of 20 percent for the letter /d/. This result is much better than ones we obtained before which were published recently [6]. It compares with results recently obtained by [1] working only with male speakers on nine letters using competitive learning based on cross entropy. Most of the errors represent cases that appear to be difficult even in human perception. Such cases are confusions b→e and d→e representing a low evidence of burst and formant transitions in voiced plosives (this might also be due to our poor resolution in data acquisition), confusions b→v, v→b, d→b, p→t, t→p, t→k indicating wrong estimation of the place of articulation, and confusions d→t, p→b, e→b indicating errors in the characterization of voicing.

Variable-depth analysis was introduced to reduce errors among plosive sounds. An error rate of 4 percent was obtained on a test set for the highly confusable set {B,D,E,V} in the same experimental conditions as in the E-set.

Another experiment was performed using the ear

model followed by a three layer MLN. The task was that of recognizing 10 English vowels pronounced five times by seven new speakers. An error rate of 4.3 percent on the test set was obtained.

A comparable error rate was obtained by using three networks for the recognition of the place and the manner of articulation plus the "tenseness" of a vowel. These networks performed equally well on vowels and diphthongs not used for learning but described by integration of features present in different combination in the training data.

The research is continuing toward the design of a speech coder capable of generating feature hypotheses for every speech sound.

The learning and recognition algorithms based on error-back propagation (EBP) have been executed on a SUN 4/280 and on a Vax 8650. For an MLN of about 10,000 links, the time was 115 CPU msecs for the recognition of a spoken letter and 317 msecs for the learning of a spoken letter on the SUN 4/280. A 20 percent reduction was obtained on the Vax 8650.

The theoretical complexity of the recognition algorithm is linear with the number of links.

MAJOR CONTRIBUTIONS

New evidence is provided in this article that speech sounds can be coded by features related to the place and manner of articulation. The presence of these features can be represented by degrees of evidence based on which stochastic performance models can be derived.

Second, it is shown how data-driven property extractors based on knowledge about acoustic correlates of phonetic features can provide useful information for training a multi-layered network with a reasonable number of data. Under these conditions significant performance has been achieved.

Third, a remarkable gain in recognition accuracy (in accordance with [1]) can be achieved if learning is competitive. Furthermore, our use of MLNs seems to perform speaker normalization equally well across male and female speakers.

Under the conditions described in this article, MLNs can be trained effectively with a reasonable number of data to generate a single model for several sounds that can be trained incrementally.

REFERENCES

1. Bahl, L.R., Brown, P.F., De Souza, P.U., and Mercer, R.L. Speech recognition with continuous-parameter hidden Markov models. In *Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP-88)* (New York, Apr. 1988), pp. 40–43.
2. Bengio, Y., and De Mori, R. Speaker normalization and automatic speech recognition using spectral lines and neural networks. In *Proceedings of Canadian Conference on Artificial Intelligence (CSCSI-88)* (Edmonton, May 1988), pp. 213–220.
3. Bourlard, H., and Wellekens, C.J. Multilayer perception and automatic speech recognition, In *Proceedings of IEEE International Conference on Neural Networks* (ICNN-87) (San Diego, June 1987), pp. IV407-IV406.
4. De Mori, R. *Computer Models of Speech Using Fuzzy Algorithms.* Plenum Press. New York, 1983.
5. De Mori, R., Laface, P., and Mong, Y. Parallel algorithms for syllable recognition in continuous speech. *IEEE Trans. Patt. Anal. and Mach. Intell. PAMI-7*, 1 (Jan. 1985), 56–69.
6. De Mori, R., Lam, L., and Gilloux, M. Learning and plan refinement in a knowledge-based system for automatic speech recognition, *IEEE Trans. Patt. Anal. and Mach. Intell. PAMI-9*, 2 (Mar. 1987), 289–305.
7. De Mori, R., Merlo, E., Palakal, M., and Rouat, J. Use of procedural knowledge for automatic speech recognition. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCA-87)*, (Aug. 1987), pp. 840–844.
8. Hinton, G.E., and Sejnowski, T.J. Learning and relearning in Boltzmann machines. In *Parallel Distributed Processing: Exploration in the Microstructure of Cognition, Volume 1*, MIT Press, Cambridge, Mass., 1986, 282–317.
9. Jelinek, F. The development of an experimental discrete dictation recognizer. *IEEE Proceedings*, Nov. 1984, 1616–1624.
10. Merlo, E., De Mori, R., Mercier, G., and Palakal, M. A continuous parameter and frequency domain based Markov model. In *Proceedings of International Conference on Acoustics, Speech Signal Processing (ICASSP-86)*, (Apr. 1986), pp. 1597–1600.
11. Plout D.C., and Hinton, G.E. Learning sets of filters using back propagation, *Computer Speech and Language*, 2, 2 (July 1987), 35–61.
12. Rumelhart, D.E., Hinton, G.E., and Williams, R.J. Learning internal representation by error propagation. In *Parallel Distributed Processing: Exploration in the Microstructure of Cognition, Volume 1*, MIT Press, Cambridge, Massachusetts, 1986, pp. 318–362.
13. Waibel, A., Hanazawa, T., and Shikano, K. Phoneme recognition: Neural networks vs. hidden Markov models. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP-88)*, (Apr. 1988), pp. 107–110.
14. Watrous, R.L., and Shastri, L. Learning phonetic features using connectionist networks. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI-87)*, (Aug. 1987), pp. 851–854.

CR Categories and Subject Descriptors: I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*speech recognition and understanding*
General Terms: Neural Networks, Automatic Speech Recognition
Additional Key Words and Phrases: Speaker normalization, speech preprocessing

ABOUT THE AUTHORS:

YOSHUA BENGIO is a Ph.D. candidate at the School of Computer Science at McGill University, where he received a masters of computer science and a bachelor's in electrical engineering. His research work is on the application of connectionist models to automatic speech recognition.

REGIS CARDIN has a master's degree in computer science from Concordia University in Montreal. He is completing his Ph.D. thesis at McGill; the topic is speech coding. His current interests include speech recognition, neural network algorithms, signal processing and computer arithmetic.

RENATO DE MORI received a doctorate degree in electronic engineering from Politecnico di Torino, Torino, Italy, in 1967. He was named director of the School of Computer Science at McGill University in January 1986. As of March, 1987 he has served as vice president and director of research at the Centre de Recherche en Informatique de Montreal (CRIM), a research center involving five universities and more than 10 companies.

ETTORE MERLO received a laureate degree in computer science from the University of Turin in 1983. He is currently finishing his Ph.D. thesis at McGill on artificial intelligence models for speech recognition. His research interests are in AI, speech recognition and software engineering. The authors' present address is McGill University, School of Computer Science, 805 Sherbrooke St. W., Montréal, Quebec H3A 2K6, Canada.