ARTIFICIAL NEURAL NETWORKS

Venue
IEE, Savoy Place, London
17 – 18 October 1989

TWO-LEVEL RECOGNITION OF ISOLATED WORD USING NEURAL NETS

I.S. Howard and M.A. Huckvale

Department of Phonetics and Linguistics, University College London, Gower Street, London WC1.

INTRODUCTION

This paper describes a neural-net based isolated word recogniser that has a better performance on a standard multi-speaker database than our reference Hidden Markov Model recogniser. The complete neural net recogniser is formed from two parts: a front-end which transforms the complex acoustic specification of the speech into a simplified phonetic feature specification, and a whole-word discriminator net. Each level was trained separately, thus considerably reducing the time necessary to train the overall system.

Isolated Word Recognition

Treating isolated word recognition (IWR) as a pattern classification problem, performance is related to the discriminability of the pattern vectors (words) in pattern space. There are three problems associated with the recognition of isolated words that weaken discriminability:

1) Occasion-to-occasion variations in the productions of words create large non-linear distributions of pattern vectors.

2) Increasing vocabulary size reduces the average distance between pattern centres.

3) Changes in speaker can cause short-term acoustic specifications of words to overlap.

Statistical models have proved adept at tackling 1), and Hidden Markov Models have been shown to accommodate occasion-to-occasion variability. HMMs have more problems with 2) and 3). With 2) since each model is trained on the examples of a particular word only, there is no constraint that models of different words should not overlap. With 3) the statistical model of a word should be conditioned on speaker characteristics, but unfortunately there is no mechanism by which this might be done.

Automatic Speech Recognition Using The Multi-Layer Perceptron

We know that neural-net pattern recognition techniques, such as the multi-layer perceptron are capable of performing complex non-linear pattern recognition tasks, Rumelhart et al (1). Previous experiments have shown them to have similar performance to HMM recognisers on a standard digit database Peeling and Moore (2). Thus neural nets are also capable of addressing problem 1). However, it is possible to build nets which discriminate between patterns, not only identify them, and so address 2) since a network can be constructed that determines which aspects of a pattern vector are important for discrimination. Another advantage offered by nets is their ability to make use of contextual information. Thus neural nets may also be able to address 3) since their classification procedure is sensitive to global aspects of the pattern vector, aspects which may be characteristic of the speaker.

Neural networks have other useful properties for the word-recognition problem: Their uniform structure makes them well suited for real-time hardware implementations (e.g. Howard and Walliker (3)), their intrinsic redundancy allows them to degrade gracefully in noise, and they appear well suited for solving a wide range of different problems, at different abstract levels, in speech processing. Consequently they provide a formalism which can integrate processing at the different levels into one system. However there are severe problems in training networks large enough to be useful in large vocabulary speech recognition.

Incorporation Of Speech Knowledge

The Multi-Layer Perceptron (MLP) classifier in its fully interconnected form constitutes a pattern processing tool that is very general in its transformation capabilities. Clearly the more degrees of freedom a network has, the harder it will be to train. That is, it will require correspondingly more computation and training data as its degrees of freedom increase. One may expect advantages in lower training times and need less training data in the case of a more constrained network, providing the constraint is appropriate for the task.

We believe that it may be helpful to incorporate a priori knowledge in the network to ease the optimization task. There has been recent interest in specifying a priori some of the structure in a MLP, Waibel (4). By specifying network topology one limits the degrees of freedom. By specifying the weights one may be able to make the training simpler, provided one can ensure that the optimization starts ``nearer'' the solution in weight space. Ideally, if sections of the network can be completely specified, no more further training would be required. Modular construction of this type is used widely in the solution of engineering problems. It is important to note, however, that the specification of weights in an MLP constitutes `soft' constraints which can be improved or even undone by further training.

In order to incorporate speech knowledge into the MLP a priori, it is necessary to have a suitable model of speech. A hierarchical analysis for ASR is appropriate because of the structured nature of the speech generation process. The approach we adopt here is to break the problem down into two stages, to train these levels independently and then to join them together. However, it is necessary to be able to define the intermediate representation that is used. One requires a more abstract description of the quantity of interest, without losing information useful in the discrimination process. Such a representation should then be simpler to process than the original input data in order to achieve word recognition.

Acoustic-Phonetic Intermediate Representation

The representation that we have chosen for our experiments is based on a traditional phonetic and phonological analysis of speech. It consists of a time-aligned acoustic-phonetic feature matrix, representing properties of the signal such as voicing, frication, nasality and vowel quality. This representation is useful not only because it can directly related to a phonological analysis of words, but also because these features normalize cross-speaker differences that occur at the acoustic-level. The bank of feature detectors are trained using a time-aligned transcription of the signal and so enhance those aspects of the signal that are necessary for word discrimination.

Word Recogniser

The feature detector stage provides the input to a word classifier network. In this way the network can be trained in two steps. This two-level training results in a network capable of performing the required task with much less training time than would be necessary than if the network was trained together initially.

The use of the feature detector network has a similarity with the TDNN of Waibel (5). The difference is that we consider the replication unit as a module that is capable of generating the necessary transformation. Thus our approach uses `cloning' at the feature bank level, rather than at the node level.

This paper describes an isolated word recognition experiment, using a standard database, in which neural-net and standard approaches are compared. The next two sections describes the database and the experiment, while the following section describes the results.

DATABASE

The database used for training and testing comprised 8000 isolated digits, which were recordings of 400 digits from each of the `least consistent' twenty speakers of the RSRE 40-speaker digit database. All the digits had been detected, cut from a recording and processing by means of a 19-channel filterbank, Holmes (6). The resulting output energies were then quantised into 16 levels in 20ms frames. The first 15 speakers were exploited for training material, while the last 5 speakers were reserved for testing.

Acoustic-Phonetic Feature Detectors

To train the acoustic-phonetic front end, each digit was automatically annotated by a dynamic programming time-alignment procedure. A time-aligned feature matrix was then generated, for each vocoder frame. More information appears in Howard and Huckvale (7).

The input to the front-end MLP consisted of five frames of vocoder data, spanning 100ms in time. Thus each input had a context of two frames either side of the labelled frame. The MLP had 34 hidden units and 17 outputs, one for each feature. Full connectivity was used between layers. The training consisted of 25 passes over 100 digits from each of the 15 training speakers. The recognition performance of the feature detectors on the test set was as follows: (percentages are for the `equal-error' point, i.e. where the miss rate equals the false-alarm rate):

FEATURE		PERFORMANCE
SIL	(silence)	91.9%
FRIC	(frication)	90.7%
VOC	(voicing)	93.0%
NAS	(nasality)	84.8%
VFRIC	(voiced frication)	76.9%
S	(/s/ fricative)	93.4%
FTH	(/f/,/T/ fricative)	75.1%
?	(glottal stop)	88.8%
K-REL	(/k/-release)	86.2%
T-ASP	(/t/-aspiration)	60.8%
EE-IH	(/i/,/I/ vowel)	90.0%
EH	(/e/ vowel)	94.0%
UH	(/V/ vowel)	94.4%
ER	(schwa vowel)	77.3%
AW	(/O/ vowel	98.9%
UE	(/u/,/U/ vowel)	60.6%
R	(/r/ glide)	95.3%

EXPERIMENT

Dynamic Time Warping Recogniser

To obtain good speaker-independent performance using DTW, it is necessary to use a representative set of templates for each training speaker, which leads to a large number of templates and a large amount of processing. The dynamic programming templates consisted of one example of each digit from 5 of the training speakers (50 entries in total). The DTW recogniser was run with slope constrains p=1, Sakoe and Chiba (8).

Hidden Markov Model Recognizer

The HMM recogniser used was a continuous density distribution type with training and recognition performed on the input vocoder data by means of the forward-backward algorithm, (e.g. Russell et al (9)). Eight states were used, with no skip transitions and with self-transition probabilities initialised to 0.8.

Initial models for the Hidden Markov Models were generated using the time aligned phonetic transcriptions and average vocoder data for each phonetic label. The training data consisted of 10 repetitions of the digits for each of 5 training speakers. Re-estimation consisted of running 20 iterations over the training set for each digit. It was found that no increase in recognition resulted from the use of more iterations.

Multi-layer perceptron recognizer.

For the MLP experiments, the digits were processed into fixed length vectors by centering each one into a 50-frame (1 second) window, padded with silence.

Three MLP word recognition networks were constructed:

MLP1) Linear MLP classifier operating directly on the vocoder energies. The network had 19x54 inputs and 10 outputs.

MLP2) Linear MLP classifier operating on the output of the featurebank as applied to the digits. The network had 17x50 inputs and 10 outputs.

MLP3) Combined featurebank and linear MLP classifier operating on the vocoder energies. The network had 50 copies of the feature detectors feeding a linear classifier, namely (19x5 inputs, 34 hidden units, 17 outputs) cloned 50 times with 10 outputs, see Fig 1. This network was initialised with the weights already obtained from the featurebank and network 2.

Training of the networks was performed on 100 digits from each of the 15 training speakers using back-propagation and fixed learning parameters. MLP1 was trained for 900 passes, MLP2 for 450 passes and MLP3 for 50 passes. The slope of the error curve was used to gauge when training was complete.

RESULTS

The overall digit recognition results obtained from the various recognition experiments are shown below.

a) DTW (vocoder data)	98.8%
b) HMM (vocoder data)	95.6%
c) MLP1 (vocoder data)	50.3%
d) MLP2 (features)	96.2%
e) MLP3 (combined)	96.2%

Confusion matrices for these results are given in Fig 2. The best performance came from the DTW algorithm, which also had the largest computational load at recognition time. Results for the MLP2 network are better than published multiple-speaker results using simple MLPs (with a similar number of weights) on the same database by Peeling and Moore (2). The combined network did not increase its performance on the test set despite further training, although performance on the training set did improve from 98.1% to 99.6%, suggesting insufficiency of training material.

The confusion matrices for the feature-based recognition shows particular difficulties between digits `zero' and `two', this is probably due to the failure of the `UE' vowel quality detector. Indeed the pattern of digit errors for the network operating on the features is repeated when the HMM is run on the feature data (Fig 2f).

CONCLUSIONS

We have demonstrated that is possible to train a MLP network in two stages for the purpose of IWR. The results show that the use of acoustic-phonetic features provide a relevant intermediate representation for speech in the digit experiment. In almost all cases, recognition performance on the AP features was as good as the raw acoustic data. The main problem was due to the inadequacy of the features to discriminate the digits ``2'' and ``0''. For the other digits, performance was better using the AP features.

The MLP word classifier, with AP intermediate representation, performed as well as the HMM. The major benefit of the MLP approach was that the AP feature net and the word classifier net could then be combine into another net. To train the latter from start-up would have required a larger amount of processing than was required to train the net in two sections.

The retraining of the combine MLP network was very slow and resulted in no better recognition performance. This is because the data set used for training was too small for the task.

Comparison with the results obtained by Peeling and Moore (2) in multiple-speaker IWR show that for a comparable sized network with similar numbers of weights, our network performs considerably better on the harder task of speaker-independent IWR. However, our linear MLP operating on raw vocoder data performed worse than the result obtained by RSRE for their linear network. It is always possible that our MLP would have improved with more training. The main point is that it took longer to train than our 2-layer approach that gave better performance.

FUTURE WORK

We believe it is necessary to allow for the variability that occurs in speech in the structure of the recogniser. That is to say, we believe that provision should be made at different levels of abstraction in the decoding hierarchy to deal with this variability. This could be achieved by the use of contextual input at different levels in the decoding hierarchy.

It may be beneficial to include decimation in time as the input signal is processed by higher and higher levels in the decoding hierarchy. This provides a means of data reduction that would be necessary for the operation of systems with large vocabularies.

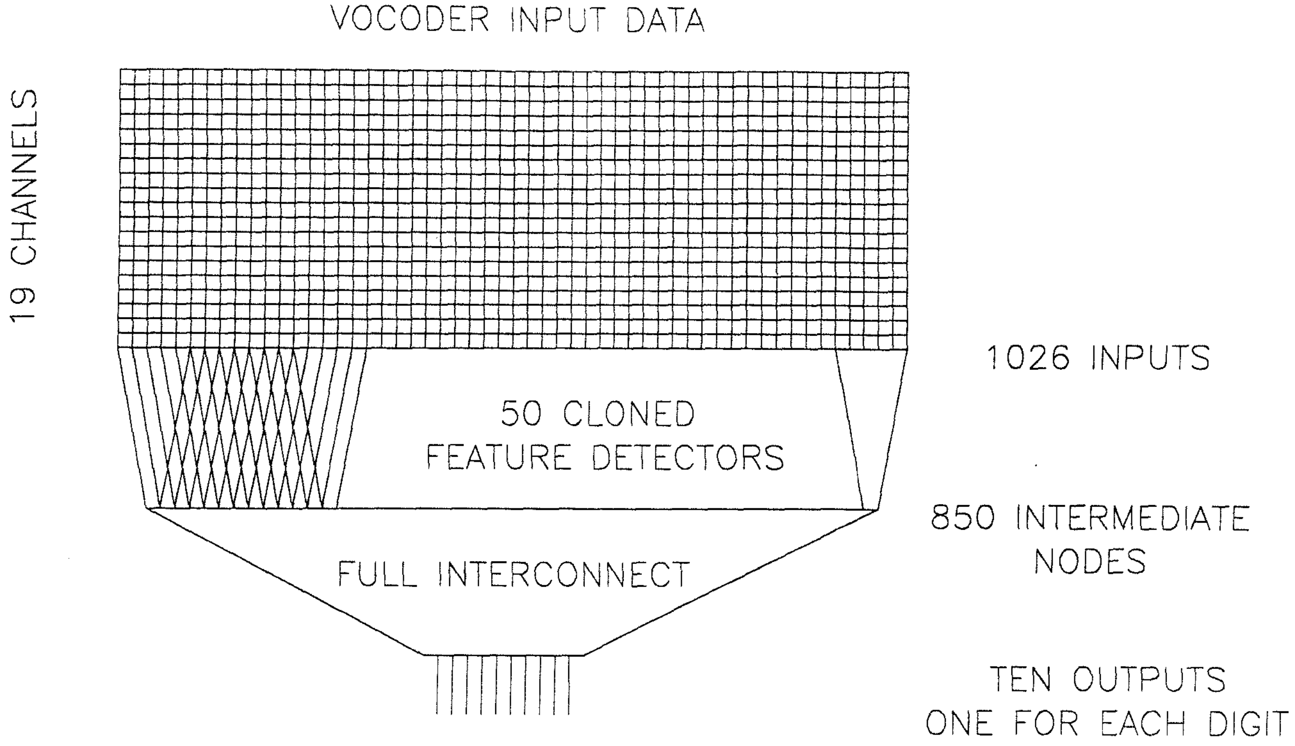

FIGURE 1. SCHEMATIC OF THE 2-LEVEL MLP FOR ISOLATED WORD RECOGNITION.

	0	1	2	3	4	5	6	7	8	9
0	50	0	0	0	0	0	0	0	0	0
1	0	50	0	0	0	0	0	0	0	0
2	0	0	49	0	0	0	0	1	0	0
3	0	0	1	49	0	0	0	0	0	0
4	0	0	0	0	50	0	0	0	0	0
5	0	3	0	0	0	47	0	0	0	0
6	0	0	0	0	0	0	50	0	0	0
7	0	0	0	0	0	0	0	50	0	0
8	0	0	0	0	0	0	0	0	50	0
9	0	1	0	0	0	0	0	0	0	49

Number of matches = 500
Recognition rate = 98.8%

2a) Result for DTW (vocoder).

	0	1	2	3	4	5	6	7	8	9
0	50	0	0	0	0	0	0	0	0	0
1	0	49	0	0	0	1	0	0	0	0
2	6	0	42	0	0	0	0	2	0	0
3	0	0	0	48	0	0	0	0	2	0
4	0	0	0	0	50	0	0	0	0	0
5	0	1	0	0	0	49	0	0	0	0
6	0	0	0	0	0	0	50	0	0	0
7	0	0	0	0	0	0	0	50	0	0
8	0	0	0	0	0	0	6	0	44	0
9	0	2	0	0	0	2	0	0	0	46

Number of matches = 500
Recognition rate = 95.6%

2b) Result for HMM (vocoder).

	0	1	2	3	4	5	6	7	8	9
0	0	0	4	6	0	0	3	37	0	0
1	0	49	0	0	1	0	0	0	0	0
2	0	2	2	12	2	0	2	30	0	0
3	0	0	0	50	0	0	0	0	0	0
4	0	0	0	0	50	0	0	0	0	0
5	0	8	12	17	4	0	0	9	0	0
6	0	0	0	0	0	0	50	0	0	0
7	0	0	0	0	0	0	0	50	0	0
8	0	2	1	14	4	0	22	7	0	0
9	0	19	5	16	2	0	0	8	0	0

Number of matches = 500
Recognition rate = 50.2%

2c) Result for MLP1 (vocoder).

	0	1	2	3	4	5	6	7	8	9
0	50	0	0	0	0	0	0	0	0	0
1	1	48	0	0	0	0	0	0	0	1
2	9	0	39	1	1	0	0	0	0	0
3	0	0	0	50	0	0	0	0	0	0
4	0	0	0	0	50	0	0	0	0	0
5	0	0	0	0	0	50	0	0	0	0
6	0	0	0	0	0	0	50	0	0	0
7	0	0	0	0	0	0	0	50	0	0
8	0	0	2	0	0	0	1	0	47	0
9	0	1	0	0	0	2	0	0	0	47

Number of matches = 500
Recognition rate = 96.2%

2d) Result for MLP2 (feature).

	0	1	2	3	4	5	6	7	8	9
0	50	0	0	0	0	0	0	0	0	0
1	0	49	0	0	0	0	0	0	0	1
2	8	0	37	1	2	0	0	2	0	0
3	0	0	0	50	0	0	0	0	0	0
4	0	0	0	0	50	0	0	0	0	0
5	0	0	0	0	0	50	0	0	0	0
6	0	0	0	0	0	0	50	0	0	0
7	0	0	0	0	0	0	0	50	0	0
8	0	0	0	0	0	0	2	0	48	0
9	0	0	0	0	1	1	0	1	0	47

Number of matches = 500
Recognition rate = 96.2%

2e) Result for MLP3 (combine).

	0	1	2	3	4	5	6	7	8	9
0	50	0	0	0	0	0	0	0	0	0
1	0	48	0	0	0	0	0	2	0	0
2	6	0	38	2	1	0	0	3	0	0
3	0	0	0	49	0	0	0	0	0	1
4	0	0	0	0	50	0	0	0	0	0
5	0	0	0	0	0	49	0	0	0	1
6	0	0	0	0	0	0	47	0	3	0
7	0	0	0	0	0	0	0	50	0	0
8	0	0	0	0	0	0	0	0	50	0
9	0	1	0	0	0	0	0	0	0	49

Number of matches = 500
Recognition rate = 96.0%

2f) Result for HMM (feature).

FIGURE 2. CONFUSION MATRICES FROM ISOLATED WORD RECOGNITION EXPERIMENTS.

ACKNOWLEDGEMENTS

The HMM program was based upon a Coral program that was written by Martin Russell, RSRE. The digits database was used with kind permission of RSRE. This work was supported by RSRE research agreement 2047/137 with SRU and by a SERC fellowship.

REFERENCES

1. Rumelhart, D.E., Hinton, G.E., Williams, R.J., (1985), in Rumelhart & McClelland, ``Parallel distributed processing'', MIT press.

2. Peeling, S.M., Moore, R.K., (1987), RSRE Memorandum 4073.

3. Howard, I.S., Walliker, J.R., (1989), EUROSPEECH, Paris.

4. Waibel, A., (1989), Neural Computation, 1, No. 1, p39-46.

5. Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., Lang, K., (1988), IEEE Trans. ASSP, ASSP-37, p328.

6. Holmes, J., (1980), IEE Proc., 127, Part F, No. 1.

7. Howard, I.S. & Huckvale, M.A., (1988), 7th FASE Symposium, SPEECH-88, Edinburgh.

8. Sakoe, H,. & Chiba, S., (1978), IEEE Trans. on ASSP, ASSP-23, 67-72.

9. Russell, M.J., & Cook, A.E., (1986), Proc. IOA, 8, Part 7, p291-297.