WORD RECOGNITION FROM THE DARPA RESOURCE MANAGEMENT DATABASE WITH THE CAMBRIDGE RECURRENT ERROR PROPAGATION NETWORK SPEECH RECOGNITION SYSTEM

Tony Robinson and Frank Fallside
Cambridge University Engineering Department

ABSTRACT - Recent work with Recurrent Error Propagation Networks has shown that they can perform at least as well as the current best Hidden Markov Models for speaker independent phoneme recognition on the TIMIT task. Accurate phoneme recognition is a prerequisite for very large vocabulary word recognition and this paper extends the previous work to word recognition from the DARPA 1000 word Resource Management task. This preliminary work achieves 52.1% word recognition rate (43.3% accuracy) with no grammar when trained on the TIMIT database using single pronunciation word models from the SPHINX system. The paper concludes with a list of topics that should be addressed in order to improve the recognition rate.

1 INTRODUCTION

Progress in large vocabulary speech recognition is dependent on good phoneme recognition techniques. The phoneme level is necessary both to reduce the storage and computational load during recognition, and also to allow for the recognition of words where an insufficient number of examples are available during training.

The Recurrent Error Propagation Network (REPN) phoneme recogniser has been shown to perform as well as, if not slightly better than, the current best Hidden Markov Models (HMM) for speaker independent phoneme recognition on the TIMIT task [Robinson and Fallside, 1990, Robinson *et al.*, 1990, Lee and Hon, 1989]. This paper starts with an overview of the phomene recogniser, then discusses the generation of the dictionary and the use of dynamic programming to extend the recogniser to the word level. The storage of the dictionary in a tree structure and the pruning of the search of this tree are discussed as they both yield a significant reduction in the overall computation required. These savings, combined with the use of context independent phoneme models, yield an efficient scheme for large-vocabulary speaker-independent word recognition from continuous speech.

2 THE REPN PHONEME RECOGNISER

This section gives an overview of the REPN phoneme recogniser which is fully described in earlier work [Robinson and Fallside, 1990]. The 16kHz digitised speech from the TIMIT database is Hamming windowed with a duration of 32ms and a frame separation of 16ms. The windowed speech is Fourier transformed to yield a power spectrum which is then downsampled by grouping in to 20 bins evenly spaced on a bark scale. This power spectrum

is then normalised and cube rooted to reduce the dynamic range. In addition to the spectral information, the cube root of the power in the frame is added, yielding 21 input values for the network. This design is the result of a comparison of many different types of preprocessor [Robinson *et al.*, 1990]. The same study covered a range of network configurations and the resulting best recogniser is used in this paper.

The recurrent network falls into the framework described by Rumelhart, Hinton and Williams [Rumelhart *et al.*, 1985]. It may be viewed as a single layer error propagation (back propagation) network, part of whose output is fed back to the input after a single frame time delay. This is shown in figure 1 where the external input, $u(t)$, and the state input, $x(t)$, together form the input vector, the output vector being composed of the external output, $y(t+1)$, and the state output, $x(t+1)$. In practice, the external output is not trained to classify the current input vector, $u(t)$, but that of four frames previously, $u(t-4)$. This is to allow some forward context in the classification, backward context being already available through the state vector. The output from the net is interpreted as a vector of probabilities that the frame was labelled with a particular phoneme. An example is given in figure 2 for the sentence "`show me all alerts`".

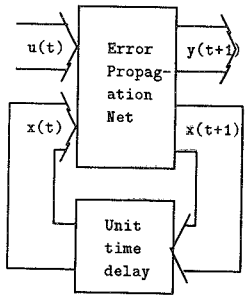

Figure 1: The recurrent network

3 THE 992 WORD DICTIONARY

The network has been tested on the DARPA 1000-word Resource Managment database [Price *et al.*, 1988]. All six speakers on the first CD-ROM of the speaker-dependent training data (September 1989 release) were used for testing, with 610 sentences per speaker.

The dictionary was based on that used in the SPHINX system [Lee, 1989, Appendix II]. This SPHINX phoneme set was expanded and the output of the recogniser (TIMIT symbols) was reduced according to table 1. In addition, multiple closures, such as `/kcl tcl/` were reduced to the first symbol. No attempt was made to deal with glottal stops, epenthetic silences or pauses.

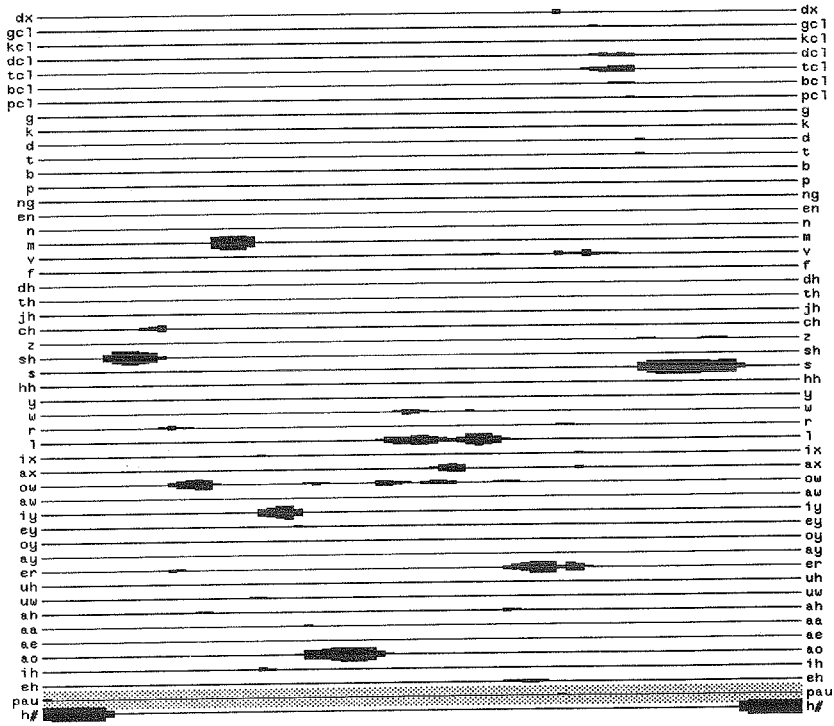

Figure 2: Example output from the recurrent net – Show me all alerts

TIMIT	dictionary
ux	uw
axr	er
ax-h	ax
hv	hh
zh	z
el	l
eng	ng
nx	n

SPHINX	dictionary	SPHINX	dictionary
B	bcl b	BD	bcl
D	dcl d	DD	dcl
G	gcl g	GD	gcl
P	pcl p	PD	pcl
T	tcl t	TD	tcl
K	kcl k	KD	kcl
TS	tcl t s	SIL	h#

Table 1: TIMIT and SPHINX to intermediate symbols

It is interesting to evaluate the relative performance of the phoneme recogniser on the TIMIT and the Resource Management databases. Table 2 gives the phoneme recognition rates on the TIMIT database for the full 61 and two reduced symbol sets of 50 and 39 elements. The last entry in this table is for the 50 symbol set on the Resource Management database using the transcriptions obtained from the dictionary. The table shows that there is a 45% increase in the number of errors when porting between these databases. The increased error rate has two main sources: firstly, there may be variations in the recording conditions of the two databases; and secondly, there are a range of pronunciations which are acceptable for a given word, so limiting the transcription to a single pronunciation will introduce errors.

Database	nsymbol	correct	insert	subst.	delet.	accur.
TIMIT	39	76.1%	5.7%	17.4%	6.5%	70.4%
TIMIT	50	71.1%	5.6%	22.4%	6.6%	65.5%
TIMIT	61	69.5%	5.5%	24.0%	6.4%	64.0%
R.M.	50	62.6%	12.1%	29.3%	8.2%	50.5%

Table 2: TIMIT and Resource Management phoneme recognition rates

4 SEARCHING AND PRUNING

The output from the network is interpreted as the vector of probabilities that the input should be labeled with a particular phoneme. A phoneme is realised as a sequence of frames labelled accordingly and a word is taken to be a string of phoneme symbols. Dynamic programming may be used to parse this probability stream for the most likely sequence of words [Ney, 1984]. This method is used with Hidden Markov Models (HMM) where it is known as a Viterbi search and is also used in most hybrid HMM/connectionist systems (for example [Morgan and Bourlard, 1990]).

The computation may be decreased if the dictionary is ordered as a tree structure as phonemes that occur at the beginning of many words need only be searched for once. Ordering the dictionary in this way gave a search space of 3058 nodes as compared with 6366 nodes with no shared phonemes.

Not all the nodes in the tree need be searched. If the probability is lower than the most likely node by some threshold then that node can be pruned. Table 3 shows the effect of various log likelihood thresholds on the fraction of nodes that need to be searched, the execution time, and the resulting recognition rates. The SPHINX HMM could prune 80–90% of the nodes [Lee, 1989, p60] – from the table it appears that a slightly higher pruning factor is available with this approach (about 90% for an increase of 2% in the number of errors). This is to be expected as the output of the network is highly discriminant, as can be seen in figure 2.

The "speed" column in table 3 refers to the factor by which parsing exceeded real time when executed on a SparcServer (about 14 MIPS). This does not include the time taken to run the network. Further optimisation for speed is possible, although this stage is already

threshold	pruned	speed	correct	insert	subst.	delet.	accur.
0.0	100.0%	0.7	–	–	–	–	–
16.0	97.3%	1.1	44.1%	7.6%	45.1%	10.8%	36.5%
20.0	95.5%	1.3	49.2%	9.0%	42.1%	8.7%	40.2%
24.0	91.6%	1.7	51.1%	9.1%	40.6%	8.4%	42.0%
26.0	89.2%	1.9	51.5%	9.0%	40.2%	8.3%	42.5%
28.0	86.6%	2.2	51.8%	8.9%	39.9%	8.3%	42.9%
32.0	80.7%	2.9	51.9%	8.8%	39.8%	8.2%	43.1%
∞	0.0%	10.3	52.1%	8.8%	39.7%	8.2%	43.3%

Table 3: Degree of pruning

running in near real time. This speed is achieved because there are only a small number of context independent phoneme units, so the log likelihood of every reasonable position may be computed prior to the dynamic programming stage.

The network contains 56026 weights which are processed at every 16ms frame yielding 3.5 million multiply and accumulates per second. Thus a real time implementation should be possible on a processor which has support for the multiply and accumulate operation (e.g. i860, DSP32C, TMS32C30), with sufficient spare processing power to perform the parsing operation and the necessary data movement.

5 CONCLUSION

This paper has presented the first results of extending the REPN phoneme recogniser to the word level. A word recognition rate of 52.1% (43.3% accuracy) has been achieved which is reasonable considering the state of development of the recogniser. However, it is much worse than the most recent HMM results with the SPHINX system which provide a word recognition rate of 81.9% [Lee *et al.*, 1990].

The most obvious next step is to incorporate a word-pair grammar and forced alignment. This will provide a mechanism for redefinition of the dictionary and generation of multiple pronunciations, especially of short "function" words. It will also allow embedded training, allowing the network to train on the same recording conditions and in the reduced variation of phonetic context found in the Resource Management task. This is expected to yield large improvements in the recognition rate.

For real applications it is important that a recogniser can be implemented with reasonable computational cost. This paper has discussed the organisation of the dictionary and found that efficient pruning during the search is possible. This advantage, combined with the use of a small number of context independent phoneme models, results in an efficient segmentation of the frame probabilities into words. It is hoped that the speed advantage of this approach can be retained during the continued development of the recogniser.

6 ACKNOWLEDGEMENTS

The work described in this paper was carried out as part of an ESPRIT Basic Research Action project (3207). The authors would like to acknowledge NIST for the provision of the DARPA Resource Management and TIMIT databases and the ParSiFal project IKBS/146 which developed the transputer array.

REFERENCES

[Lee and Hon, 1989] Kai-Fu Lee and Hsiao-Wuen Hon. Speaker-independent phone recognition using hidden Markov models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(11):1641–1648, November 1989.

[Lee *et al.*, 1990] Kai-Fu Lee, Hsiao-Wuen Hon, Mei-Yuh Hwang, and Sanjoy Mahajan. Recent progress and future outlook of the SPHINX speech recognition system. *Computer Speech and Language*, 4:57–69, 1990.

[Lee, 1989] Kai-Fu Lee. *Automatic Speech Recognition: The Development of the SPHINX System.* Kluwer Academic Publishers, Boston, 1989.

[Morgan and Bourlard, 1990] N. Morgan and H. Bourlard. Continuous speech recognition using multilayer perceptrons with hidden Markov models. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1990.

[Ney, 1984] H. Ney. The use of a one-stage dynamic programming algorithm for connected word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):263–271, April 1984.

[Price *et al.*, 1988] Patti Price, William M. Fisher, Jared Bernstein, and David S. Pallett. The DARPA 1000-word Resource Management database for continuous speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 651–654, 1988.

[Robinson and Fallside, 1990] Tony Robinson and Frank Fallside. Phoneme recognition from the TIMIT database using recurrent error propagation networks. Technical Report CUED/F-INFENG/TR.42, Cambridge University Engineering Department, March 1990.

[Robinson *et al.*, 1990] Tony Robinson, John Holdsworth, Roy Patterson, and Frank Fallside. A comparison of preprocessors for the Cambridge recurrent error propagation network speech recognition system. In *International Conference on Spoken Language Processing*, Kobe, Japan, November 1990.

[Rumelhart *et al.*, 1985] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical Report ICS-8506, University of California, San Diego, September 1985.