Phoneme Recognition from the TIMIT database using Recurrent Error Propagation Networks

CUED/F-INFENG/TR.42

Tony Robinson and Frank Fallside
Cambridge University Engineering Department,
Trumpington Street, Cambridge, England.
Enquiries to: ajr@eng.cam.ac.uk

March 1990

Abstract

This report describes a speaker independent phoneme recognition system based on the recurrent error propagation network recogniser described in [1, 2].

This recogniser employs a preprocessor which generates a range of types of output including bark scaled spectrum, energy and estimates of formant positions. The preprocessor feeds a fully recurrent error propagation network whose outputs are estimates of the probability that the given frame is part of a particular phonetic segment. The network is trained with a new variation on the stochastic gradient descent procedure which updates the weights by an adaptive step size in the direction given by the sign of the gradient. Once trained, a dynamic programming match is made to find the most probable symbol string of phonetic segments. The recognition rate is improved considerably when duration and bigram probabilities are used to constrain the symbol string.

A set of recognition results is presented for the trade off between insertion and deletion errors. When these two errors balance, the recognition rate for all 61 TIMIT symbols is $68.6\% \pm 0.3\%$ correct ($62.5\% \pm 0.4\%$ including insertion errors) and on a reduced 39 symbol set the recognition rate is $75.1\% \pm 0.2\%$ correct ($68.9\% \pm 0.4\%$). This compares favourably with the results of other methods on the same database [3, 4, 5, 6, 7].

1 Introduction

The most promising approach to the problem of large vocabulary automatic speech recognition is to build a recogniser which works at the phoneme level and then map the resulting string of phonemes onto a string of words. Phonemes are the smallest linguistic unit that can be used to distinguish meaning [8, p 23]. By their symbolic nature they provide a natural boundary for speech recognition systems between the lower level distributed representations such as the acoustic waveform and its transformations, and the higher level symbolic representations such as words and the representation of syntactic and semantic knowledge. The phoneme recognition approach is practical because the number of phonemes is small (about 45) compared with the number of words in a large vocabulary task (over 1000). Thus speaker independent phoneme models may be trained with a much smaller speech corpus than would be required to train speaker independent word models.

The DARPA TIMIT Acoustic Phonetic Continuous Speech Database [9] (hereafter referred to as the TIMIT database) has been designed to be used for training phoneme recognisers. It is becoming the most widely available database of its size and type. This report uses the

December 1988 Prototype CD-ROM which contains 420 talkers uttering 4200 sentences sampled at 16kHz and constitutes the training set of the complete TIMIT database. Accurate comparison of different phoneme recognition systems using different databases is difficult, it is therefore important to evaluate recognisers on a standard database.

Currently the best established technique for large scale automatic speech recognition uses Hidden Markov Models (HMMs) [10, 11, 12]. Recently, connectionist models [13, 14] and more particularly, error propagation networks [15] have been used with some success in this field [16, 17, 18]. The main differences between the HMM and connectionist approach using error propagation networks are:

- Error propagation networks provide a discriminant decision, i.e. the training minimises the distance to the target class and maximises the distance to the other classes. Standard HMMs lack this ability although work is now being done to develop discriminant HMMs [19, 20].
- Recurrent nets have an inherent mechanism for adapting to speaker variability in that information relating to the type of speaker (e.g. female/male) can be propagated in time through the state vector. There is no such mechanism in word HMMs which consist of concatenated independent phoneme models, although this can be done through an external mechanism such as the remapping of codebooks.
- Error propagation networks are trained by a gradient descent procedure which is considerably slower than HMM Baum-Welch parameter reestimation.
- Explicit target values are needed at each frame to train error propagation networks (e.g. a time aligned phonetic transcription). The HMM approach needs only the correct sequence of models.
- The sequential nature of the speech signal is more naturally expressed by the state transitions in a HMM than by the development of the state vector in recurrent nets. As a result, the state sequence of phoneme HMMs can be concatenated to yield the state sequence for word models but no equivalent operation has been applied to recurrent nets.

The first two points may yield a higher recognition accuracy for recurrent nets and the last two points may be overcome with sufficient computational resources, an adequately labelled database and suitable postprocessing. This suggests that recurrent error propagation networks are worth investigating as an alternative to HMMs. Previous connectionist work has been limited to a subset of the English phonemes or has been speaker dependent.

The strategy adopted here is to pass frames of windowed speech through a preprocessor which are then fed to a recurrent net. This net is trained to model the frame-by-frame classification of the TIMIT database. A postprocessor is then used to convert this distributed representation into a string of phoneme symbols representing the sentence.

The results reported here are the results of improvements made upon two previous versions. The first version [1, 21] was limited to a set of 28 symbols which covered the 7 speaker database of four utterances of 31 sentences. This was multiple speaker, not speaker independent work and no results from other established methods were available for comparison. The second version of this system is described briefly in [2] and provided true speaker independent phoneme recognition from an established database (TIMIT). The aim of this report is to provide an overview of the system, to present the details that were missing from the second report and to describe the latest improvements.

2 Preprocessor

The preprocessor was designed as a synthesis of several established techniques. More specifically, for each 16ms frame there are:

- 1 channel for the number of zero crossings;
- 4 channels for the cube root of the energy in quarter frames;
- 20 channels for cube roots of the energies in bark scale bins of the short time Fourier transform;
- 1 channel for the pitch as determined by a peak in the cepstrum;
- 1 channel for the height of the pitch peak to measure the degree of voicing;
- 4 channels for the position of the first four peaks in the homomorphically smoothed power spectra.

The bark scaling and cube roots were derived by simplifying the auditory model presented by Bladon and Lindblom [22]. The preprocessor used in the first version had only the energies and bark scale power spectra and a noticeable improvement was found with the inclusion of the extra features.

The second version had the above features and an additional 12 channels for lpc derived log area ratios. By examination of the resulting weight matrix, the weight connected to these channels were found to be near zero, and subsequent removal without degradation in performance confirmed that these channels were redundant. In the latest version the training data was preprocessed with four different offsets to better cover the variability in the windowed speech. This improved the frame-by-frame recognition rate by about 5%, as can be seen in table 1, although it should be noted that part of the increase is as a result of increasing the time constant for smoothing the weight changes used in training (the "momentum" term [15]).

no. of offsets	frame-by-frame recognition rate
1	61.1%
2	64.2%
4	66.0%

Table 1: Effect of multiple offsets on recognition rate

The preprocessor also truncated initial and final silences longer than 160ms. This was done to reduce size of the training data and provide a more even distribution of symbols amongst frames.

3 Recurrent Net

A recurrent net can be considered as a sequence of error propagation networks [15] where the input and output vectors are divided into external and internal portions. The external input vector, $u_{0\ldots L-1}$, consists of the 31 channels from the preprocessor; and the external output vector, $y_{0\ldots M-1}$ has 61 dimensions and is fed to the postprocessor. The internal output forms a state vector, $x_{0\ldots N-1}$, of 128 dimensions and is fed to the same network in the next time period as shown in figure 1. This network operates by concatenating the input and output vectors:

$$o_i^{(t)} = \begin{cases} 1 & \text{for } i = 0 \\ u_{i-1}^{(t)} & \text{for } 1 \le i \le L \\ x_{i-L-1}^{(t)} & \text{for } L+1 \le i \le N+L \end{cases} \quad (1)$$

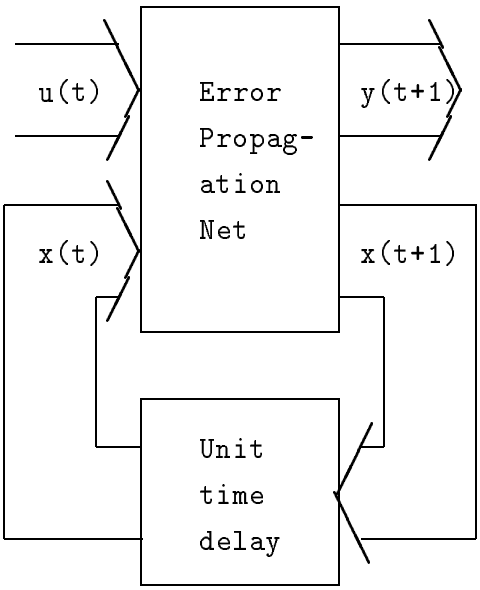

Figure 1: Recurrent network

which is then passed forwards through the network by performing a matrix multiplication followed by the application of a non-linear squashing function:

$$x_i^{(t+1)} = \frac{1}{1+\exp\left(\sum_{j=0}^{L+N} w_{ij} o_j^{(t)}\right)} \quad \text{for } 0 \le i \le N-1 \tag{2}$$

$$y_{i-N}^{(t+1)} = \frac{1}{1+\exp\left(\sum_{j=0}^{L+N} w_{ij} o_j^{(t)}\right)} \quad \text{for } N \le i \le N+M-1 \tag{3}$$

The resulting output is compared with the desired output vector, $d_{0\ldots M-1}$, according to a cost function. Following Hinton [23], Baum and Wilczek [24] and Stolla, Levin and Fleisher [25] the cross-entropy cost function is used:

$$\log(p^{(t)}) = \sum_{i=0}^{M-1} d_i^{(t)} \log(y_i^{(t)}) + (1-d_i^{(t)}) \log(1-y_i^{(t)}) \tag{4}$$

The first version of this system used the standard least mean squares cost function. The maximum likelihood metric has two advantages: firstly that faster convergence was observed and secondly that it results in a statistically more rigorous interface to probabilistic grammatical constraints which is important for higher level processing.

Training is performed on a 64-processor array of T800 transputers with the training data distributed evenly over the processors. Each processor has a copy of all the weights which are used to make a forward and backward pass for 32 consecutive frames on each processor. The resulting partial derivatives are summed to give an estimate of the gradient based on 2048 frames. There is a trade off in the number of frames processed before updating the weights; a large number gives a more accurate gradient signal, and a small number allows for more frequent weight updates. Typical training time was two days (about 10^{13} floating point operations).

This version of the model used a new algorithm for updating the weights. A step size for each weight is used, and the weight is changed by the magnitude of the step size multiplied by the sign of the local gradient. Initially all steps were equal, and the step is adapted by multiplying (or dividing) by a scaling factor if the local gradient agrees (or disagrees) in sign with the smoothed gradient. The scaling factor used was 1.1 and the momentum term started at 0.5 and

increased over the first few passes through the training set until it was sufficient to smooth the local gradient over the whole of the training set. The step sizes were hard limited to be not greater that a factor of 16 above or below the mean step size. This method has the disadvantage that changes in the magnitude of the step size can occur more rapidly than the changes in the smoothed gradient. Thus it is possible to have a large smoothed gradient which consistently disagrees with the sign of the local gradient which results in a rapid reduction of the step size to the lower threshold, so inhibiting further motion of that weight. In spite of this disadvantage, this method was found to converge faster for this problem than the technique of Chan and Fallside [26] used in the first version, and the similar technique developed by Jacobs [27].

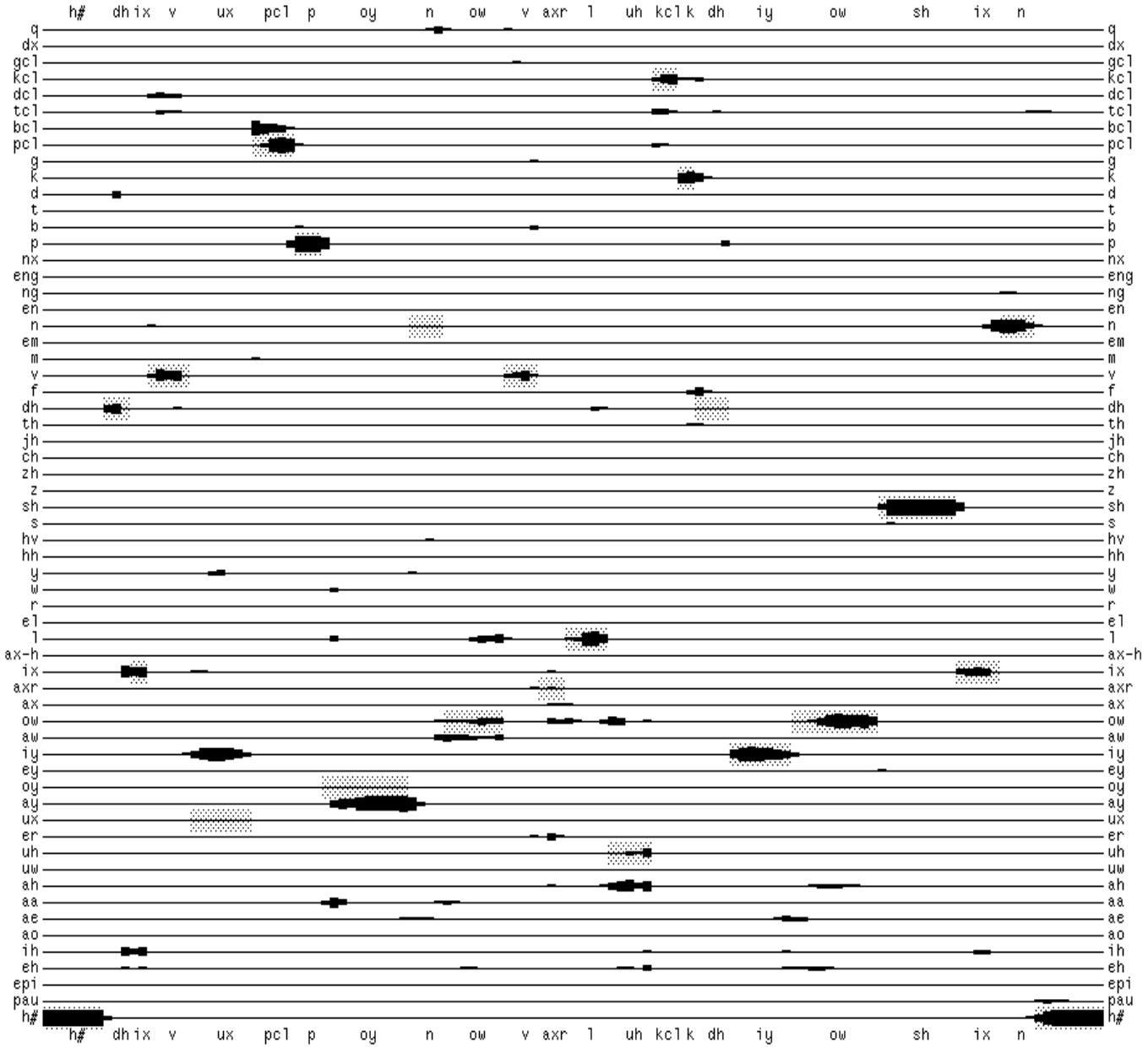

Figure 2: Example output from the recurrent net.

Example output of the model from the test set is given in figure 2 plotted against time as a variable width line. The hand labels are indicated on the horizontal axis of the diagram and the recogniser labels on the vertical axis. The shaded rectangles represent the target outputs. The sentence is "The viewpoint overlooked the ocean" (TIMIT file: train/dr7/flas0/sx228/sx228.adc) which is of 2 seconds duration.

4 Postprocessor

The distributed output is converted to a symbolic form by finding the most likely sequence of phonetic segments which match the observed output. This may be efficiently achieved using the technique of dynamic programming [28, p 311]. In addition to the distance measure used for training (equation 4), two additional probabilities, duration and bigram, were used. The duration probability for a symbol was calculated from a histogram of the duration in frames of all the occurrences of that symbol in the training set. Similarly the bigram probability was calculated from a matrix of the number of co-occurrences of the two symbols in the training set. To avoid zero probabilities, a small constant (0.5) was added to each frequency count before normalisation.

TIMIT	REDUCED	IPA	TIMIT	REDUCED	IPA
p	p	p	b	b	b
t	t	t	d	d	d
k	k	k	g	g	g
pcl	sil	p^{o}	bcl	sil	b^{o}
tcl	sil	t^{o}	dcl	sil	d^{o}
kcl	sil	k^{o}	gcl	sil	g^{o}
dx	dx	ɾ	q		ʔ
m	m	m	em	m	m̩
n	n	n	en	n	n̩
ng	ng	ŋ	eng	ng	ŋ̩
nx	n	ɾ̃			
s	s	s	sh	sh	š
z	z	z	zh	sh	ž
ch	ch	č	jh	jh	ǰ
th	th	θ	dh	dh	ð
f	f	f	v	v	v
l	l	l	el	l	l̩
r	r	r	w	w	w
y	y	y	h#	sil	□
pau	sil	□	epi	sil	▯
hh	hh	h	hv	hh	ɦ
eh	eh	ɛ	ih	ih	ɪ
ao	aa	ɔ	ae	ae	æ
aa	aa	ɑ	ah	ah	ʌ
uw	uw	u	uh	uh	ʊ
er	er	ɝ	ux	uw	ü
ay	ay	$ɑ^{y}$	oy	oy	$ɔ^{y}$
ey	ey	e^{y}	iy	iy	i^{y}
aw	aw	$ɑ^{w}$	ow	ow	o^{w}
ax	ah	ə	axr	er	ɚ
ix	ih	ɨ	ax-h	ah	ə̥

Table 2: The TIMIT symbol set with the CMU/MIT reduction and IPA symbols

In order to compare with other techniques, the 61 TIMIT symbols were grouped into 39 symbols according to a CMU/MIT mapping taken from [6]. The TIMIT symbols, the reduced set of symbols and the IPA symbols are given in table 2 which is an adaptation of a similar table by Seneff and Zue [29]. A full explanation of the IPA symbols can be found in [30]. All

occurrences of the the glottal stop, q, were discounted in the reduced set.

5 Results

The TIMIT "sa" sentences were considered unsuitable for training or testing as they consist of only two phrases and would introduce an unnatural bias in the distribution of phonemes and their contexts. Of the other two types of sentence, "si" and "sx", three out of every four speakers were used for training, and the remainder retained for testing.

Table 3 gives the number and percentage of insertion, substitution and deletion errors in the 32638 symbol test set after a filter which merges duplicate instances of repeating symbols was applied. This table also shows the recognition accuracy which is defined to be 100% minus the percentage of insertion, substitution and deletion errors. The table shows that either the duration or the bigram probabilities are needed to achieve reasonable recognition rate. When both duration and bigrams are used, it is unnecessary to merge adjacent identical symbols and so this filter is not used in subsequent processing.

context	correct	insertion	substitution	deletion	accuracy
none	76.0%	39.2%	21.3%	2.7%	36.8%
duration	68.6%	7.1%	24.3%	7.1%	61.5%
bigram	66.7%	3.4%	24.2%	9.1%	63.3%
combined	65.8%	2.6%	23.7%	10.5%	63.2%

Table 3: Recognition rates for variations in context

Longer symbol sequences that span the same sentence involve a larger number of duration and bigram probabilities. As these probabilities are less than one, extra symbols are penalised and there is a tendency towards short sequences and a greater number of deletion errors than insertion errors. This may be compensated for by adding a bias to every transition; the effect of this bias is shown in table 4. It is assumed that insertion and deletion errors are equally detrimental to the performance of the recogniser, so from the table it is found that a bias of 3.0 is appropriate and this value is used in all future analysis.

bias	correct	insertion	substitution	deletion	accuracy
0.0	65.8%	2.7%	23.8%	10.4%	63.1%
1.0	66.7%	3.5%	24.3%	9.0%	63.3%
2.0	67.7%	4.6%	24.7%	7.7%	63.0%
3.0	68.6%	6.1%	25.1%	6.3%	62.5%
4.0	69.3%	8.3%	25.5%	5.2%	61.0%
5.0	70.2%	11.4%	25.8%	4.1%	58.8%
6.0	71.1%	16.6%	25.9%	3.0%	54.5%
7.0	72.1%	27.0%	25.7%	2.2%	45.2%

Table 4: Recognition rates for variations in transition bias

A confusion matrix for these results is given in figure 3. The hand labels are on the vertical axis and the recogniser labels are on the horizontal axis. The null symbol, -, is added so that insertion and deletion errors may be shown. The area of the square at the intersection of two symbols is proportional to the number of such points in the test set. The resolution is three phonemes per pixel rounded up (about 0.01% of the total). The strong diagonal represents the

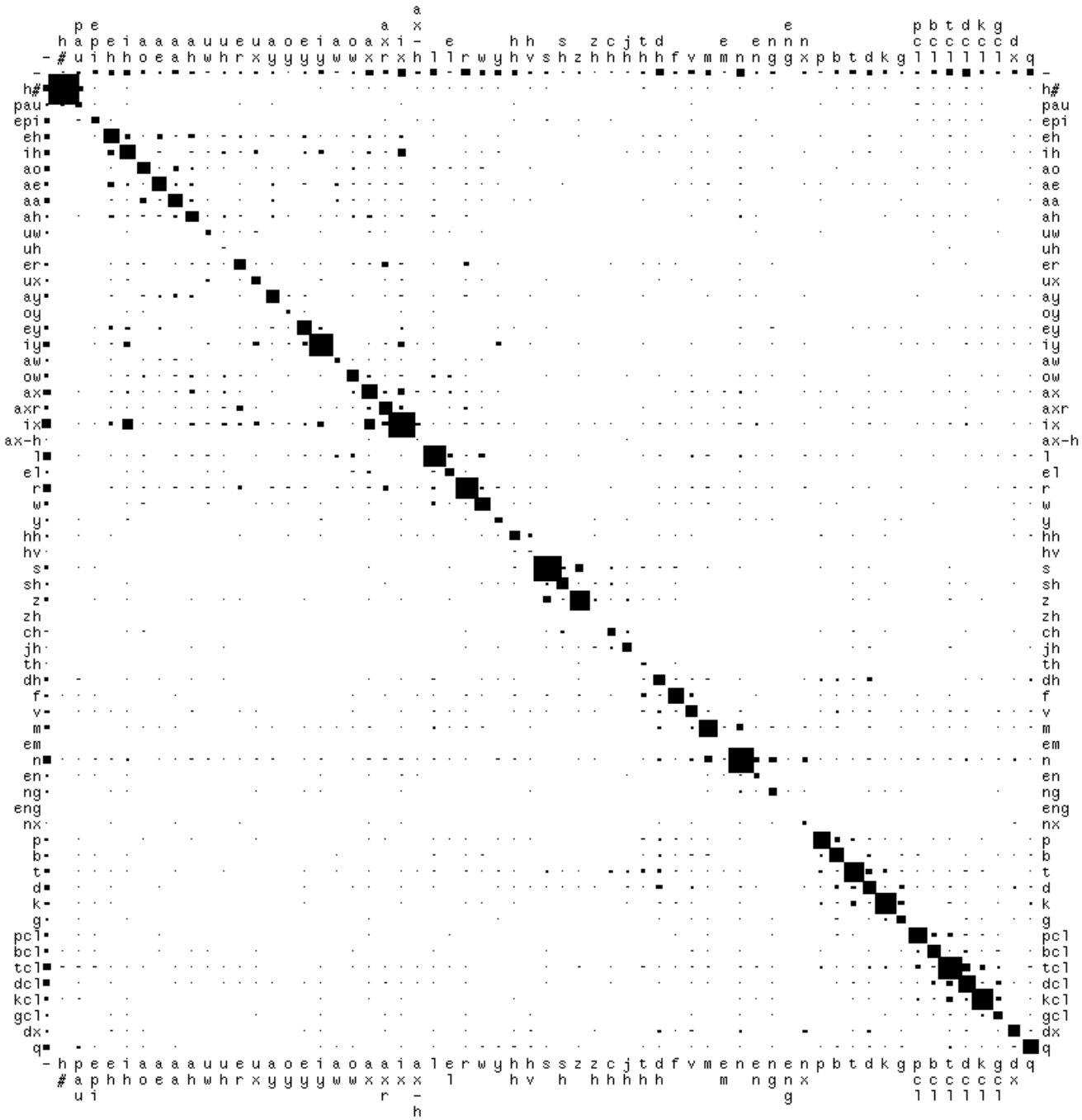

Figure 3: Confusion matrix

62.5% of symbols recognised correctly, and the remainder are errors. The top ten errors from this matrix are given in table 5. As expected, the most common errors are between symbols belonging to the same broad class.

hand label	recogniser label	percentage of all errors
ix	ax	1.67%
ix	ih	1.61%
ix	-	1.27%
r	-	0.94%
-	dcl	0.91%
n	-	0.91%
s	z	0.91%
tcl	dcl	0.91%
l	-	0.89%
ih	ix	0.88%

Table 5: The ten most common errors

Two methods for reducing the recogniser symbol set were tried. The first involved a simple rewrite of the symbolic outputs of the full symbol set to the reduced set. This yielded 75.1% correct, 6.2% insertion, 18.6% substitution and 6.3% deletion errors (68.9% accuracy). The second method was to merge the distributed representations of the symbols by summing the probabilities of the relevant classes and reestimating the duration and bigram probabilities. This yielded 75.2% correct, 7.1% insertion, 18.4% substitution and 6.4% deletion errors (68.1% accuracy). The main difference is the higher percentage of insertion errors with the second method which may be attributed to poorer duration and bigram modelling resulting from the merging of different types of symbol (e.g. the closures and silence).

In order to establish the reliability of these recognition rates, the test database was divided into sixteen parts and the recognition rates computed for each part. As the total test database contained 840 sentences each part contained 52 sentences. Table 6 gives the mean and the variance in the mean for each type of error.

no of symbols	correct	insertion	substitution	deletion	accuracy
61	$68.6\% \pm 0.3\%$	$6.1\% \pm 0.2\%$	$25.1\% \pm 0.3\%$	$6.3\% \pm 0.1\%$	$62.5\% \pm 0.4\%$
39	$75.1\% \pm 0.2\%$	$6.2\% \pm 0.2\%$	$18.6\% \pm 0.3\%$	$6.3\% \pm 0.2\%$	$68.9\% \pm 0.4\%$

Table 6: Recognition rates with variances

6 Discussion

The results presented in the previous section compare favourably with other TIMIT results. For the classification task of labelling segmented speech, Zue, Glass, Phillips and Seneff [3] report a 70% classification rate on the full TIMIT symbol set and Digalakis, Ostendorf and Rohlicek [4] report 73%. There are no previous connectionist techniques for the full range of phonemes though Hataoka and Waibel [5] reported 60.5% on the 16 English vowels. For the segmentation and labelling task, Lee and Hon [6] reported 73.80% with 7.72% insertions for a covering set of 39 phonemes and Levinson, Liberman, Ljolje and Miller [7] report 52% with 12% insertions on a 51 symbol set.

The most serious limitation of the current approach is the time taken to train the network. Whilst the algorithm described in section 3 was faster than previous gradient descent based techniques for this task, it is still slower than training an equivalent HMM recogniser. As the performance is only slightly better than the HMM approach it suggests that future work should be aimed towards improving the efficiency of the training algorithm and/or increasing the performance of the network.

The network had 30240 free parameters (weights) arranged as a simple rectangular matrix. Whilst this structure is simple, restricted connectivity, fixed or replicated weights may give a better trade-off between the number of state units and the number of weights, whilst retaining sufficient information processing capacity and so result in faster learning and recognition.

The network performance was poor when no additional information was used in the production of the symbol string. The use of the log likelihood distance metric allows other probabilistic constraints to be applied, such as the duration and bigram probabilities described in this report. It is expected that further work extending this approach to large vocabulary speech recognition would discard the bigram models in favour of transitional constraints imposed by word models built from a pronunciation dictionary.

Use of the dynamic programming match allows a known string of phonemes to be time aligned with the output of the net. This has possibilities for resegmenting the training set, and for creating segment boundaries in speech where only the transcription is known.

Future work will also investigate semi-automatic techniques for selecting which channels in the preprocessor carry information useful to the recognition process, and will also compare the existing preprocessor to a full auditory model [31].

In conclusion, the method presented in this report has made a small but significant improvement in recognition accuracy over the best existing HMM techniques on the TIMIT database. This demonstrates that recurrent error propagation networks are a suitable method for performing low level automatic speech recognition and suggests their use in the task of large vocabulary automatic speech recognition.

7 Acknowledgements

The work described in this report 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 TIMIT database and the ParSiFal project IKBS/146 which developed the transputer array. They also wish to thank all members of the Speech, Vision and Robotics group of Cambridge University Engineering Department for their advice, but in particular Mike Chong, Patrick Gosling, Tim Marsland, Richard Prager and Georges Wong.

References

[1] A. J. Robinson and F. Fallside. A dynamic connectionist model for phoneme recognition. In *Neural Networks from Models to Applications: Proceedings of nEuro'88*, pages 541–550. I.D.S.E.T., Paris, 1989.

[2] F. Fallside, H. Lucke, T. P. Marsland, P. J. O'Shea, M. St. J. Owen, R. W. Prager, A. J. Robinson, and N. H. Russell. Continuous speech recognition for the TIMIT database using neural networks. In *Proc. ICASSP*, 1990.

[3] Victor Zue, James Glass, Michael Phillips, and Stephanie Seneff. Acoustic segmentation and phonetic classification in the SUMMIT system. In *Proc. ICASSP*, pages 389–392, 1989.

[4] Vassilios Digalakis, Mari Ostendorf, and J. Robin Rohlicek. Improvements in the stochastic segment model for phoneme recognition. In *Proceedings of the DARPA Workshop*, October 1989.

[5] Nobuo Hataoka and Alex H. Waibel. Speaker-independent phoneme recognition on TIMIT database using integrated time-delay neural networks (TDNNs). Technical Report CMU-CS-89-190, Carnegie-Mellon University, November 1989.

[6] 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.

[7] S. E. Levinson, M. Y. Liberman, A. Ljolje, and L. G. Miller. Speaker independent phonetic transcription of fluent speech for large vocabulary speech recognition. In *Proc. ICASSP*, pages 441–444, 1989.

[8] P. Ladefoged. *A Course in Phonetics.* Harcourt Brace Jovanovich, New York, second edition, 1982.

[9] John S. Garofolo. *Getting Started with the DARPA TIMIT CD-ROM: An Acoustic Phonetic Continuous Speech Database.* National Institute of Standards and Technology (NIST), Gaithersburgh, MD, 1988.

[10] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi. An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. *The Bell System Technical Journal*, 62(4):1035–1074, April 1983.

[11] L. R. Rabiner, S. E. Levinson, and M. M. Sondhi. On the application of vector quantization and hidden Markov models to speaker-independent, isolated word recognition. *The Bell System Technical Journal*, 62(4):1075–1105, April 1983.

[12] L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, pages 4–16, January 1986.

[13] D. E. Rumelhart and J. L. McClelland. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. I: Foundations.* MIT Press, Cambridge, MA, 1986.

[14] Teuvo Kohonen. *Self-Organization and Associative Memory.* Springer-Verlag, New York, second edition, 1988.

[15] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. I: Foundations.*, chapter 8. Bradford Books/MIT Press, Cambridge, MA, 1986.

[16] H. Bourlard and C. J. Wellekens. Multilayer perceptrons and automatic speech recognition. In *Proceedings of the IEEE First Annual International Conference on Neural Networks*, pages IV:407–416, San Diego, June 1987.

[17] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang. Phoneme recognition using time–delay neural networks. Technical report, ATR Interpreting Telephony Research Laboratories, October 1987.

[18] Michael A. Franzini, Michael J. Witbrock, and Kai-Fu Lee. A connectionist approach to continuous speech recognition. In *Proc. ICASSP*, pages 425–428, 1989.

[19] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer. Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *Proc. ICASSP*, pages 49–52, 1986.

[20] S. J. Young. Competitive training in hidden Markov models. In *Proc. ICASSP*, pages 681–684, 1990. Expanded in the technical report CUED/F-INFENG/TR.41, Cambridge University Engineering Department.

[21] A. J. Robinson. *Dynamic Error Propagation Networks*. PhD thesis, Cambridge University Engineering Department, February 1989.

[22] R. A. W. Bladon and Bjorn Lindblom. Modeling the judgement of vowel quality differences. *Journal of the Acoustical Society of America*, 69(5):1414–1422, May 1981.

[23] Geoffrey E. Hinton. Connectionist learning procedures. Technical Report CMU-CS-87-115, Computer Science Department, Carnegie-Mellon University, June 1987.

[24] Eric B. Baum and Frank Wilczek. Supervised learning of probability distributions by neural networks. In Dana Z. Anderson, editor, *Proceedings of Neural Information Processing Systems*, Denver, November 1987. American Institute of Physics.

[25] S. A. Solla, E. Levin, and M. Fleisher. Accelerated learning in layered neural networks. *Complex Systems*, 2, 1988.

[26] L. W. Chan and F. Fallside. An adaptive training algorithm for back propagation networks. *Computer Speech and Language*, 2(3/4):205–218, 1987.

[27] Robert A. Jacobs. Increased rates of convergence through learning rate adaptation. *Neural Networks*, 1:295–307, 1988.

[28] Alfred V. Aho, John E. Hopcroft, and Jeffery D. Ullman. *Data Structures and Algorithms*. Addison-Wesley, 1983.

[29] Stephanie Seneff and Victor W. Zue. Transcription and alignment of the TIMIT database. In John S. Garofolo, editor, *Getting Started with the DARPA TIMIT CD-ROM: An Acoustic Phonetic Continuous Speech Database*. National Institute of Standards and Technology (NIST), Gaithersburgh, MD, 1988.

[30] Geoffrey K. Pullum and William A. Ladusaw. *Phonetic Symbol Guide*. University of Chicago Press, Chicago, 1986.

[31] Roy D. Patterson and Tatsuya Hirahara. HMM speech recognition using DFT and auditory spectrograms. Technical Report TR-A-0063, ATR Auditory and Visual Perception Research Laboratories, 1989.