RECURRENT NEURAL NETWORKS FOR SYLLABIFICATION

Andrew Hunt

Speech Technology Group
Department of Electrical Engineering
The University of Sydney

ABSTRACT - An important procedure in many prosodic analysis systems is to locate syllables. The location of syllables is used for the identification of stress and pitch accents, and forms the basis for the analysis of rhythm. This paper presents a novel syllabification method using recurrent neural networks which is more accurate than previous techniques. This method achieves high accuracy on continuous speech, by finding 94% of syllables and by placing most syllable boundaries within 20msec of the desired location. The paper also investigates means of optimising the performance of recurrent neural networks.

INTRODUCTION

An important procedure in many prosodic analysis systems is to locate syllables. The location of syllables is used in the identification of stress and pitch accents, for the measurement of segmental durations, and for the analysis of rhythm.

To date, most syllabification methods used for prosodic analysis in speech recognition have employed crude feature detection mechanisms and basic classification algorithms. The poor performance of these methods has led to errors in later stages in prosodic processing. Many of these methods were developed for isolated word systems and will not transfer well to continuous speech applications. A brief review of such methods and their results will be presented.

This paper describes a novel neural network topology using a series of recurrent neural networks (RNN) for performing syllabification. Several optimisations for the RNN are introduced. A variety of input parameters derived from the acoustic signal are also described and compared. Best performance is achieved with bark-frequency cepstral-coefficients.

The system has been developed and tested on the TIMIT database, an American, speaker-independent, continuous-speech database. The system finds 94% of syllables with a 5% insertion rate and 5% deletion rate. The system also places the start and end points with high accuracy - usually within 20msec of the desired point. The system does have difficulty dealing with vowels occurring in clusters, with the labelling of rhoticised vowels in the TIMIT database, and with certain diphthongs.

SYLLABIFICATION

The aim of the syllabification system is to correctly locate all syllables in an utterance by identifying start and end points. However, most systems developed to date have located different points from the conventional syllable boundary. This system indentifies the start and end of the vowel around which the syllable is constructed. This definition is useful because the start of the vowel is reasonably close the perceived beat of the syllable (Marcus 81), and because it much easier than splitting the leading and trailing consonant clusters of a syllable.

This approach to syllabification can be viewed as a process of locating all vowels in an utterance. A two stage process is used. The first stage classifies each frame of utterance as vowel or non-vowel - this stage is referred to as VC classification. The second stage must then determine if a section labelled as all vowel is a single vowel or a 'vowel cluster' - this stage is referred to as vowel splitting. For example, in */hiIzsmat/* ('He is smart'), there are two vowel sections, */iI/* and */a/*. The first section is a vowel cluster containing two vowels, and the second section contains only one vowel. The first stage of syllabification should locate the two sections, and the second stage of syllabification should determine that the first section needs to be split but that the second cluster does not.

There are many examples of syllabification systems. Mermelstein (75) and Lea (80) describe algorithms based on detection of peaks and dips in the energy of the acoustic signal. Aull and Zue (85) and Waibel (88) describe techniques based on an intermediate classification of speech frames, followed by detailed analysis of sonorant segments. Most techniques report finding around 90% of syllables but none report the detail of accuracy for syllable boundary placement. Not all work on casual continuous speech.

SPEECH DATA AND SIGNAL PROCESSING

TIMIT Database

The TIMIT database is an American, speaker-independent, continuous-speech database of read sentences. The database covers eight dialect regions of the USA. Dialect region 1 (New England) was used for all experiments. The TIMIT database divides its speakers into training and testing speakers. For Dialect Region 1 there are 38 training speakers (14 female - 24 male) and 11 testing speakers (4 female - 7 male).

For each speaker in the database there are 10 sentences. There are 2 dialect sentences designed to expose the dialect variations of the speakers - the same two are used for all speakers. These sentences were not used for training or testing as recommended in the TIMIT manual. There are 5 phonetically-compact sentences for each speaker which are designed to provide a good coverage of pairs of phones. Most speakers in dialect region 1 speak different instances of the phonetically-compact sentences. There are 3 phonetically-diverse sentences for each speaker which are designed to provide examples of different contexts for phones. There are no repetitions of the phonetically-diverse sentences.

Analysis of the complete TIMIT sentence list showed that 92.7% of vowels occurred alone - i.e. not clustered with another vowel. 7.0% of the vowels occurred in two-vowel clusters, and less than 0.3% of vowels occurred in 3 vowel clusters. These figures indicate that the separation of vowel clusters is important for proper syllabification.

Parameter Extraction

A wide variety of parameters can be extracted from the acoustic signal of speech. The best choice of parameters depends on the nature of the recognition task, the recognition method being used (e.g. HMMs, feed-forward MLPs, RNNs), and on the type of speech data being classified (e.g. continuous speech, isolated words).

The parameters used in various combinations in this work were:-

- log of energy in a 32msec sliding window,
- log of peak-to-peak amplitude in a sliding 32msec window,
- log of energy in 19 mel-scale bands from 0-4kHz,
- log of energy in 17 bark-scale bands from 0-4kHz,
- first 12 mel-frequency cepstral coefficients (MFCC),
- first 12 bark-frequency cepstral coefficients (BFCC).

Mel-frequency cepstral coefficients (MFCC) are derived by the mechanism described by Davis & Mermelstein (80). Equivalent parameters were derived for the bark-scale parameters by substituting the bark-scale band edges for mel-scale bands edges. These new values are referred to in this paper as bark-frequency cepstral coefficients (BFCC). Both the mel-scale and bark-scale are perceptually motivated frequency scales with similar shapes.

Other parameters such as formant frequencies, formant bandwidths, zero-crossing rate and pitch were found to be less effective in a pilot experiment of VC classification using feed-forward networks (unpublished).

The RNN described in this paper works best with input parameters scaled to range from -1 to 1. Self-normalisation using the mean and standard deviation of each parameter was used to scale the values.

RECURRENT NEURAL NETWORK TOPOLOGY

A RNN is similar in most respects to a conventional feed-forward network. The architecture and training mechanism used in this paper was described by Werbos (90). The mechanism feeds forward in time the output of a hidden layer to the input of the hidden layer in the next time frame. Figure 1 illustrates the network connections. Werbos (90) described a method of back-propagating error through time which updates connection weights by a globally optimal value.

RNNs are particularly powerful at performing discrimination on time-series events. They are able to learn internal states and to classify complex temporal patterns. Thus, they are an appropriate topology for many speech recognition applications. As an example, the current best phoneme recognition performance uses a similar RNN topology (Robinson & Fallside 91).

Werbos (90) described a general form of RNN with all nodes being fully connected, and with recurrent

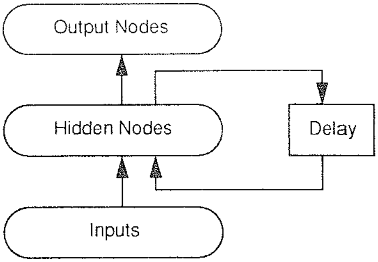

Figure 1. Recurrent Neural Network Topology.

connections through 2 time periods. In this work the connections are simplified. The hidden and output layers are not fully connected and the recurrent connection is through only one time period. This does not appear to reduce performance on the syllabification problem, but does reduce the number of weights and improves the training time.

RNN Optimisation

RNNs do show tremendous discrimination power but have large computing requirements for training. The delta-bar-delta weight update algorithm (Jacobs 88) was introduced the improve training time. The delta-bar-delta algorithm provides dynamic adaption of the learning rate of each connection weight in the network. It provided at least one order of magnitude improvement in training time.

The 'tanh' function was used as the transfer function of the nodes in preference to the conventional sigmoid function. Two reasons are offered. First, the tanh function is symmetric which is suggested to improve learning rate slightly (Haffner et al 88). Second, the distance metric used with the tanh function is a statistical likelihood function which is more appropriate to a classification function than the euclidean distance measure. However, no significant change in performance was observed from using tanh.

For most of this work speech frames were analysed with 10msec spacing. The phoneme labels of the TIMIT database cannot be relied on for 5 msec accuracy which is the accuracy required by performing classification on 10msec frames. A novel method was implemented in the RNN training program. The training program could ignore outputs and errors within a given distance of the vowel-consonant boundaries. This is achieved by setting the error at those output nodes to zero before back-propagation. This feature improved training time at the cost of a slight loss in accuracy.

Output Delay

The RNN operation can be enhanced to provide better modelling of the context required to classify speech. Coarticulation in continuous speech leads to the pronunciation of one phoneme being affected by the phonemes prior and following it. Thus, it is desirable for the network to have access to context information to be able to model, or compensate for, coarticulation. Figure 2 shows a representation of context relevant to speech processing. An ideal network might look at 100msec before and after the current point to include 1 or 2 adjoining phones. This can be achieved for a RNN by delaying the output with respect to the input. Without output delay the RNN can only use the left context, but the output delay provides both left and right context for discrimination and therefore should improve performance. The diagram shows RNN context as decaying over time. RNNs can, however, learn to use prior input in much more complex ways and in ways more appropriate to the problem. This area deserves further research.

An added advantage of including output delay is that the discrimination capability of the network is enhanced. Lippman (88) describes the form of decision region for multi-layer feed-forward networks. Increasing the number of hidden layers increases the complexity of the decision regions that can be formed. If we unfold the RNN in time, we find that increasing the output delay increases the number of hidden nodes between input and output, and hence increases the discriminating ability of the network. This will apply for up to 3 delays in the output.

There is one significant disadvantage of increasing the output delay. As the output delay is increased, the

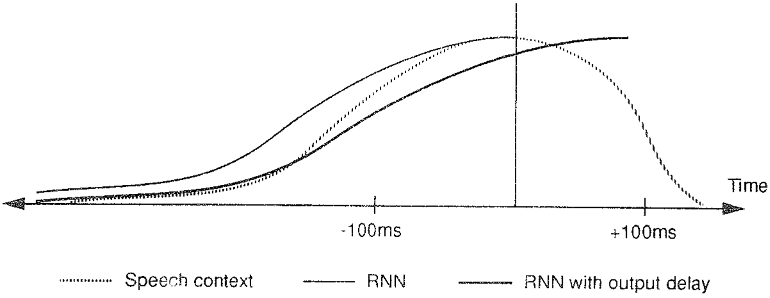

Figure 2. Context for RNNs and Speech

number of layers between input frames and the corresponding outputs increases. Thus the back-propagation of errors must go through more layers. At each layer the error is 'diluted' and the greater the dilution, the slower the network will learn. The problem is greatest at the start of training with a random weight set.

A novel mechanism was found for minimising this dilution problem. The output delay was set to zero for an initial training pass. Because the dilution of error is low the network trains quickly to an optimum performance. The output delay is then increased by one, without any change in the connection weights. Because the data used in the problem exhibits high correlation between successive frames the accuracy is still high, and the network again quickly learns the slightly modified task. Again, the output delay can be increased, and re-training performed. This procedure can significantly reduce training time. This process can be performed through tens of layers, with the requirement that the network have enough hidden nodes to store the required history, though for VC classification a delay of 4 frames was optimal.

Syllabification Architecture

As described earlier, syllabification can be performed in two stages. The first stage finds vowel sections, and the second stage separates vowel clusters. Stage 1 is the VC classifier using a single RNN. Stage 2 is an architecture which indicates which section need to be separated and where to place the separation boundary. A novel mechanism was employed for stage 2. A RNN was trained using the input vectors for each vowel section. The output (in time) remained Low for the entire length of the first vowel, and went High at the start of a second vowel in the vowel clusters. Thus, if there was only vowel in the section the output would not go High. Note that each section is treated as a separate forward pass in time.

One difficulty appeared with this mechanism, the output would often go high at the end of single vowels or in the transition of diphthongs. The behaviour is attributable to rapid changes in the input parameters which are an indication of the start of a new phoneme. The problem was mostly eliminated by training a second, almost identical, network. The second network with identical architecture was trained on the same data, but backwards in time. The results indicate that the chance of both networks making the same error would be low. Work in ongoing for determining when to split vowel clusters.

RESULTS

Vowel - Consonant Recognition

Table 1 presents the results of training the VC network with various input parameters. The accuracy given is the frame-by-frame accuracy of classification on independent test data - it does not represent the number of correctly located syllables. The results given are the best performance of three networks trained from different random starting points. All network had an output delay of four frames (40msec).

Unpublished pilot experiments on VC classification using feed-forward networks (mentioned previously) achieved accuracies of only 85.0%. The RNNs achieve markedly better results on this task.

The networks all trained to similar accuracy independent of the initial random weights. The networks

Input Parameters	# Inputs	# Hidden Nodes	Accuracy
log-amplitude	1	20	78.3%
log-amplitude + 15 linear scaled band energies	16	20	84.6%
19 mel-frequency bands log-energy	19	20	92.6%
17 bark-frequency bands log-energy	17	20	92.4%
12 mel-frequency cepstral coefficients (MFCC)	12	20	93.2%
12 bark-frequency cepstral coefficients (BFCC)	12	20	93.5%

Table 1: CV Classification using RNN.

appeared to generalise well as performance on test data was usually within 1% of accuracy for the training data.

The accuracy on bark-scale-based parameters was always marginally better than for mel-scale-based parameters. Further, the training was always faster for bark-scale-based parameters.

The performance on the cepstral transformed parameters was better than the untransformed parameters. Since the transformation is purely a linear transform, the improvement must be due to either improved representation of useful features in the cepstral parameters, or reduced noise in the parameters.

A particularly promising feature of the result is that for both the MFCC and BFCC training, 51% of errors occurred at vowel-consonant boundaries, and 69% of errors occurred within 15msec of the boundaries.

Splitting Vowel Sections

The BFCC parameters were chosen as the input for stage 2 of syllabification; the splitting of vowel sections. The accuracy for the networks was identical in both forward and backward directions. On a frame-by-frame basis, the accuracy was 92.2%. Again, the majority of errors occurred at the boundary points - 31% occurred on the boundary, and 51% occurred within 15msec of the boundary.

Syllable Recognition

The VC classification RNN and vowel splitting RNNs join to make a complete syllabification process. The results for the syllabification are

Located syllables	94.0%
Insertion rate	5.4%
Deletion rate	5.6%

Of the located syllables, 87.0% correctly matched with the expected vowel position. A further 93.% were single vowel which were incorrectly split, and 3.7% were vowel clusters which were not split as required.

The accuracy of placement of boundaries is indicated by two measures: the error in the placement of the start of the vowel, and the ratio of expected to measured vowel duration. For correctly placed vowels, 87% have a starting time placed within 20msec of the expected point, and 80% have a vowel duration within 30% of the expected value.

DISCUSSION

The results appear to be better than previously reported results for syllabification of continuous speech. Unfortunately, most papers of syllabification do not report detail of the accuracy of the syllabification, and none include measures of the accuracy of the placement of boundaries.

Analysis of the results and the speech data showed two specific problems. First, the labelling of rhoticised vowels in the TIMIT database caused problems. In the database the same label was used for both the colouring of a vowel and the phoneme */r/* (i.e. the difference between the */r/* in *'here'* and *'hero'*). This was a particular problem for the VC classification. The two instances of the same label need to be handled separately; one as an extension of a single vowel, and the other as a consonant. This analysis of the labelling will need to be performed by hand and may form part of future work.

The second problem was the handling of diphthongs by the vowel splitter. Many of the errors of the vowel splitting process were due to diphthongs being split, or due to not splitting a vowel cluster which would be incorrectly treated as a diphthong. More training with a larger network and more data should reduce this problem.

Another important observation is that most of the deleted syllables were reduced syllables. Since the first application for the syllabification system will be for a stress and pitch accent detector the omission of reduced (and therefore unstressed) syllables is not a major problem.

In general, the performance of the system reached expectation. The processing requirements of the operational system are minimal and can be performed faster than real-time.

Research is taking place on development of a speaker-adaptive front-end to the architecture described in this paper (Hunt, sub). Also, work is being done to utilise more powerful computing resources to train the RNNs on the full TIMIT database, instead of only dialect region 1.

CONCLUSION

A novel architecture for performing syllabification of continuous speech using recurrent neural networks was described. The system is more accurate than previous reported systems. The major drawbacks encountered are due to the use of the 'r' label in the TIMIT database, and problems in determining when vowel sections should be split. Various optimisations of the RNN architecture were described, including detailed analysis of the use of delaying output for training. Work is continuing on improving a few aspects of the system.

ACKNOWLEDGEMENT

The author wishes to thank Laurens Leerink for his help with implementing recurrent neural network software and for help on training methods. The author is supported by a Telecom Research Laboratory Fellowship and an Australian Postgraduate Research Award.

REFERENCES

Aull, A. & Zue, V., (1985) *Lexical Stress Determination and its Application to Large Vocabulary Speech Recognition*, ICASSP '85, pp 4111-4114.

Davis, S.B. & Mermelstein, P., (1980) *Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences*, IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. 28, pp 357-366.

Haffner, P., Waibel, A. & Shikano, K., (1988) *Fast Back-Propagation Methods for Neural Networks in Speech*, Proc. from the Fall Meeting of the Acoustical Society of Japan, October 1988.

Hunt, A.J., (1992) *Speaker Adaption of a Recurrent Neural Network*, submitted manuscript.

Jacobs, R.A., (1988) *Increased Rates of Convergence Through Learning Rate Adaptation*, Neural Networks, Vol. 1, pp 295-307.

Lea, W., (1980) *Prosodic Aids to Speech Recognition*, W. Lea, Trends in Speech Recognition, (Prentice-Hall: Englewood Cliffs, NJ), pp 166-205.

Lippman, R.P., (1988) *Neural Nets for Computing*, ICASSP 88, Vol. 1, pp 1-6.

Lippman, R.P., (1989) *Review of Neural Networks for Speech Recognition*, Neural Computation, Vol. 1, pp 1-38.

Marcus, S., (1981) *Acoustic Determinants of Perceptual Centres*, Perception and Psychophysics, Vol. 30, pp 247-256.

Mermelstein, P., (1975) *Automatic segmentation of speech into syllabic units*, JASA, Vol. 58, pp 880-883.

Robinson, T. & Fallside, F., (1991) *A recurrent error propagation network speech recognition system*, Computer Speech and Language, Vol. 5, pp 259-274.

Waibel, A., (1988) *Prosody and Speech Recognition*, (Pitman Publishing: London).

Werbos, P.J., (1990) *Backpropagation Through Time: What It Does and How To Do It*, Proc. IEEE, Vol. 78, pp 1550-1560.