CONTINUOUS SPEECH RECOGNITION USING LINKED PREDICTIVE NEURAL NETWORKS

Joe Tebelskis *Alex Waibel* *Bojan Petek*[*] *Otto Schmidbauer*[†]

School of Computer Science, Carnegie Mellon University
Pittsburgh, Pennsylvania 15213-3890, USA

ABSTRACT

We present a large vocabulary, continuous speech recognition system based on Linked Predictive Neural Networks (LPNN's). The system is a straightforward extension, from isolated to continuous speech, of the LPNN system presented last year. The system uses neural networks as predictors of speech frames, yielding distortion measures which can be used by the One Stage DTW algorithm to perform continuous speech recognition. The system currently achieves 95%, 58%, and 39% word accuracy on tasks with perplexity 7, 111, and 402 respectively, outperforming several simple HMMs that we tested. We also found that the accuracy and speed of the LPNN can be slightly improved by the judicious use of hidden control inputs. We conclude by discussing the strengths and weaknesses of the predictive approach.

I. INTRODUCTION

Neural networks are proving to be useful for difficult tasks such as speech recognition, because they can easily be trained to compute smooth, nonlinear, nonparametric functions from any input space to any output space. In speech recognition, the function most often computed by networks is *classification*, in which spectral frames are mapped into a finite set of classes, such as phonemes. In theory, classification networks approximate the optimal Bayesian discriminant function [1], and in practice they have yielded very high accuracy [2, 3, 4]. However, integrating a phoneme classifier into a speech recognition system is nontrivial, since classification decisions tend to be binary, and binary phoneme-level errors tend to confound word-level hypotheses. To circumvent this problem, neural network training must be carefully integrated into word level training [1, 5]. An alternative function which can be computed by networks is *prediction*, where spectral frames are mapped into predicted spectral frames. This provides a simple way to get non-binary distortion measures, with straightforward integration into a speech recognition system. Predictive networks have been used successfully for small vocabulary [6, 7] and large vocabulary [8] speech recognition systems. In this paper we extend our large vocabulary isolated word recognition system [8] to continuous speech. We describe the results of our experiments, and discuss the strengths and weaknesses of our current approach.

II. LINKED PREDICTIVE NEURAL NETWORKS

Linked Predictive Neural Networks have been described in detail in [8]. Here we present a brief review.

The LPNN system is based on canonical phoneme models, which can be logically concatenated in any order (using a "linkage pattern") to create templates for different words; this makes the LPNN suitable for large vocabulary recognition.

Each canonical phoneme is modeled by a topology of states, as in an HMM. And each state (i.e., phone model) is implemented by a predictive neural network. Each of these networks is trained to accurately predict the next frame of speech, within segments of speech corresponding to its phone model; the predictions will be less accurate in uncorrelated segments of speech. Hence, phonemes are "recognized" indirectly, by virtue of the relative accuracies of the different predictive networks in a given segment of speech. Note, however, that phonemes are not classified at the frame level. Instead, continuous scores (prediction errors) are accumulated for various word candidates, and a decision is made only at the word level, where it is finally appropriate.

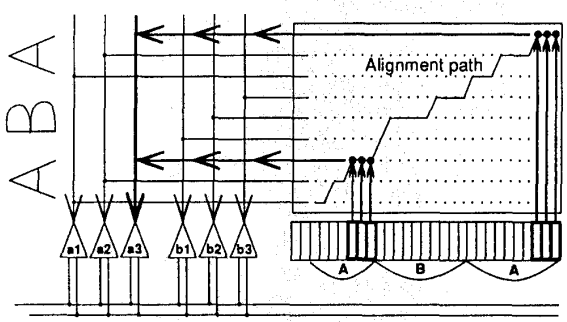

Figure 1: The backward pass during training.

The LPNN training algorithm consists of three steps: a forward pass, an alignment step, and a backward pass. Suppose we are given a training utterance and its phonetic spelling (or linkage pattern). Figure 1 shows /ABA/ as a typical example. In the forward pass, all the networks make their predictions in parallel

[*] B. Petek is a visiting researcher from the University of Ljubljana, Republic of Slovenia, Yugoslavia.

[†] O. Schmidbauer is a visiting researcher from Siemens Research Labs, Otto Hahn Ring 6, 8000 Munich 83, Germany.

CH2977-7/91/0000-0061 $1.00 © 1991 IEEE

for each frame of speech, and the prediction errors are routed through the linkage pattern, yielding a prediction error matrix. Next, the DTW algorithm is applied to this matrix, to find the optimal alignment path between the input speech and the sequence of predictors. Finally, in the backward pass (shown in Figure 1), error is backpropagated into the associated network at each point along the alignment path. Hence backpropagation causes the nets to become better predictors, and the alignment path induces specialization of the networks for different phonemes. During testing, isolated word recognition is performed by finding the word candidate with the minimum DTW score.

IIa. Extension to Continuous Speech

During the past year we extended the LPNN to deal with continuous speech. In the continuous LPNN, the training procedure is the same as before, since the phonetic spelling of the entire utterance is known. Testing, however, becomes more complicated, because word boundaries must be located while words are matched within those boundaries. To solve this compound problem, we use an efficient extension of DTW, the One Stage algorithm [9], which jointly optimizes the segmentation and the word matches, yielding a complete sentence hypothesis. Transitions between words can be further constrained by using a word-pair or bigram grammar, derived from the training corpus.

During training, the need for labeled data can be reduced or eliminated, by first bootstrapping the networks on a small amount of speech with forced phoneme boundaries, and then training on the whole database with looser alignment constraints, e.g., using only forced word boundaries, or no forced boundaries at all. Since less constraint implies more search, we usually strike a computational balance between these latter possibilities by training with forced alignment on "loose" word boundaries, located by dithering the word boundaries obtained from an automatic labeling procedure (based on Sphinx [10]), in order to optimize those word boundaries for the LPNN system.

III. RECOGNITION EXPERIMENTS

We have evaluated the LPNN system on a database of continuous speech recorded at CMU. The database consists of 204 English sentences using a vocabulary of 402 words, comprising 12 dialogs in the domain of conference registration. A typical sentence is "Okay, then I'll send you a registration form." The average sentence length is 8 words; the maximum is 15 words. Training and testing versions of this database were recorded with a close-speaking microphone in a quiet office by multiple speakers for speaker-dependent experiments. Recordings were digitized at a sampling rate of 16 KHz. A Hamming window and an FFT were computed, to produce 16 melscale spectral coefficients every 10 msec.

In our experiments we used 40 context-independent phoneme models (including one for silence), each of which had the topology shown in Figure 2. In this topology, similar to the one used in the SPICOS system [11], a phoneme model consists of 6 states, economically implemented by 3 networks covering 2 states each, with self-loops and a certain amount of state-skipping allowed. This arrangement of states and transitions provides a tight temporal framework for stationary and temporally well structured

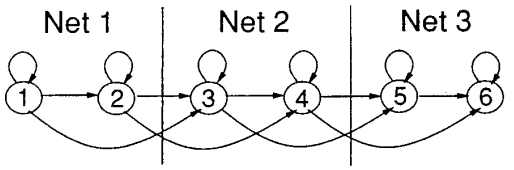

Figure 2: The LPNN phoneme model.

phones, as well as sufficient flexibility for highly variable phones. Because the average duration of a phoneme is about 6 frames, we imposed transition penalties to encourage the alignment path to go straight through the 6-state model. Transition penalties were set to the following values: zero for moving to the next state, s for remaining in a state, and 2s for skipping a state, where s was the average frame prediction error. Hence 120 neural networks were evaluated during each frame of speech. These predictors were given contextual inputs from two past frames as well as two future frames. Each network had 12 hidden units, and used sparse connectivity, since experiments showed that accuracy was unaffected while computation could be significantly reduced. The entire LPNN system had 41760 free parameters.

Since our database is not phonetically balanced, we normalized the learning rate for different networks by the relative frequency of the phonemes in the training set. During training the system was bootstrapped for one iteration using forced phoneme boundaries, and thereafter trained for 30 iterations using only loose word boundaries from canonical word pronunciations.

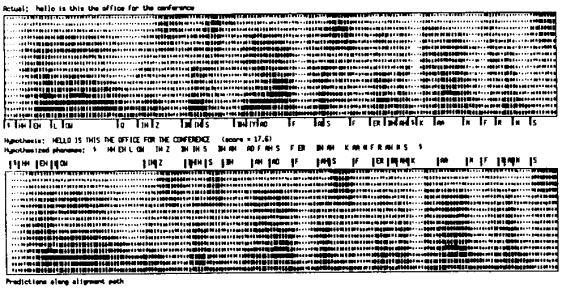

Figure 3: Actual and predicted spectrograms.

Figure 3 shows the result of testing the LPNN system on a typical sentence. The top portion is the actual spectrogram for this utterance; the bottom portion shows the frame-by-frame predictions made by the networks specified by each point along the optimal alignment path. The similarity of these two spectrograms

	speaker A			speaker B		
Perplexity	7	111	402	7	111	402
Substitutions	1%	28%	43%	4%	28%	46%
Deletions	1%	8%	10%	2%	12%	14%
Insertions	1%	4%	6%	0%	4%	3%
Word Accuracy	97%	60%	41%	94%	56%	37%

Table 1: LPNN performance on continuous speech.

indicates that the hypothesis forms a good acoustic model of the unknown utterance (in fact the hypothesis was correct in this case).

Speaker-dependent experiments were performed under the above conditions on two male speakers, using various task perplexities (7, 111, and 402). Results are summarized in Table 1.

IV. COMPARISON WITH HMMs

In order to confirm that the predictive networks were making a positive contribution to the overall system, we performed a set of comparisons between the LPNN and several pure HMM systems. In experiment E1 we replaced each predictive network by a univariate Gaussian whose mean was determined analytically from the labeled training data, and whose variance was unity, resulting in 16 free parameters per PDF. In experiment E2, the variances were also computed, resulting in 32 free parameters per PDF. In experiment E3, the mean was computed for delta coefficients as well (from t-2 and t+2), and the variances were again set to unity, resulting in 32 free parameters per PDF. Each experiment used speaker A, with task perplexity 111. Results are summarized in Table 2. As can be seen, each of these simple HMMs had a lower accuracy than the LPNN (which had 60% accuracy).

Experiment	E1	E2	E3
Substitutions	41%	35%	30%
Deletions	12%	16%	13%
Insertions	10%	5%	2%
Word Accuracy	37%	44%	55%

Table 2: HMM performance (speaker A, perplexity 111).

V. HIDDEN CONTROL EXPERIMENTS

In another series of experiments, conducted under other conditions, we varied the LPNN architecture by introducing hidden control inputs, as proposed by Levin [7]. Figure 4 shows three architectures used in our comparative study. The first is a basic LPNN (no hidden control), in which P × N = 80 networks are required to represent P = 40 phonemes with N = 2 states each. In (b), hidden control inputs were introduced such that only P = 40 networks are required for the same task: each phoneme is modeled by a single network modulated by N hidden control input bits which distinguish the state using a thermometer representation. In (c), the hidden control idea is taken to its limit: one big network is modulated by P × N hidden control inputs which specify both the phoneme and the state.

A theoretical advantage of hidden control architectures is that they reduce the number of free parameters in the system. As the number of networks is reduced, each one is exposed to more training data, and – up to a certain point – generalization may improve. The system can also run faster, since partial results of redundant forward pass computations can be saved. (Notice, however, that the total number of forward passes is unchanged.) Finally, the savings in memory can be significant.

Table 3 shows the results of our experiments. Speaker B was used in these tests; we repeat that these tests used a 2-state, rather than a 6-state, phoneme topology. Besides testing continuous

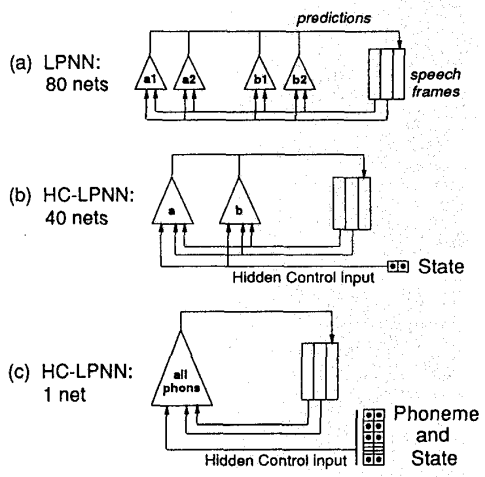

Figure 4: Architectures used in Hidden Control experiments.

speech recognition, we also tested excerpted word recognition, in which word boundaries within continuous speech are given; this allowed us to compare the acoustic discriminability of the three architectures more directly.

As the table shows, we observed minor differences in performance between architectures (a) and (b): the LPNN was slightly more discriminant, but the hidden control architecture generalized better and ran faster. Meanwhile, architecture (c) did very poorly, presumably because it had too much shared structure and too few free parameters, overloading the network and causing poor discrimination. Hence, hidden control may be useful, but only if it is used carefully.

Architecture	(a)	(b)	(c)
# free parameters (weights)	80960	42080	6466
Word accuracy:			
Excerpted words (P = 402)	70%	67%	39%
Continuous speech (P = 7)	91%	91%	n/a
Continuous speech (P = 402)	14%	20%	n/a

Table 3: Results of Hidden Control experiments.

VI. CURRENT LIMITATIONS OF PREDICTIVE NETWORKS

While the LPNN system is good at modeling the acoustics of speech, it presently tends to suffer from poor discrimination. In other words, for a given segment of speech, all of the phoneme models tend to make similarly good predictions, rendering all phoneme models fairly confusable. For example, Figure 5 shows an actual spectrogram and the frame-by-frame predictions made by the /eh/ model and the /z/ model. Disappointingly, both models are fairly accurate predictors for the entire utterance.

This problem arises because each predictor receives training in only a small region of input acoustic space (i.e., those frames corresponding to that phoneme). Consequently, when a predictor

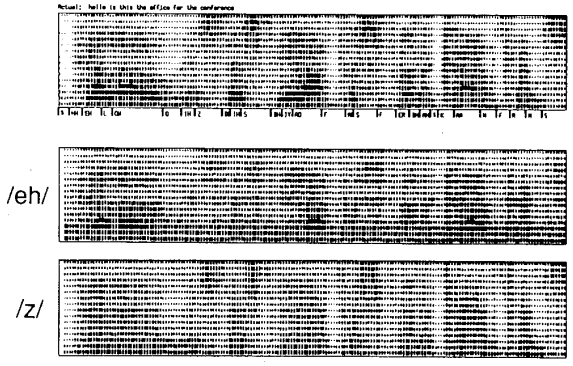

Figure 5: Actual spectrogram, and corresponding predictions by the /eh/ and /z/ phoneme models.

is shown any other input frames, it will compute an undefined output, which may overlap with the outputs of other predictors. In other words, the predictors are currently only trained on positive instances, because it is not obvious what predictive output target is meaningful for negative instances; and this leads to problematic "undefined regions" for the predictors. Clearly some type of discriminatory training technique should be introduced, to yield better performance in prediction based recognizers.

VII. CONCLUSION

We have studied the performance of Linked Predictive Neural Networks for large vocabulary, continuous speech recognition. Using a 6-state phoneme topology, without duration modeling or other optimizations, the LPNN achieved an average of 95%, 58%, and 39% accuracy on tasks with perplexity 7, 111, and 402, respectively. This was better than the performance of several simple HMMs that we tested. Further experiments revealed that the accuracy and speed of the LPNN system can be slightly improved by judicious use of hidden control inputs.

The main advantages of predictive networks are that they produce non-binary distortion measures in a simple and elegant way, and that by virtue of their nonlinearity they can model the dynamic properties of speech (e.g., curvature) better than linear predictive models [12]. Their main current weakness is that they have poor discrimination, since their strictly positive training causes them all to make confusably accurate predictions in any context. Future research should concentrate on improving the discriminatory power of the LPNN, by such techniques as corrective training, context dependent phoneme modeling, and function word modeling.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of DARPA, the National Science Foundation, ATR Interpreting Telephony Research Laboratories, and NEC Corporation. B. Petek also acknowledges support from the Research Council of Slovenia.

REFERENCES

[1] H. Bourlard and C. J. Wellekens. Links Between Markov Models and Multilayer Perceptrons. *Pattern Analysis and Machine Intelligence*, 12:12, December 1990.

[2] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang. Phoneme Recognition Using Time-Delay Neural Networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, March 1989.

[3] M. Miyatake, H. Sawai, and K. Shikano. Integrated Training for Spotting Japanese Phonemes Using Large Phonemic Time-Delay Neural Networks. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 1990.

[4] E. McDermott and S. Katagiri. Shift-Invariant, Multi-Category Phoneme Recognition using Kohonen's LVQ2. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 1989.

[5] P. Haffner, M. Franzini, and A. Waibel. Integrating Time Alignment and Connectionist Networks for High Performance Continuous Speech Recognition. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 1991.

[6] K. Iso and T. Watanabe. Speaker-Independent Word Recognition Using a Neural Prediction Model. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 1990.

[7] E. Levin. Speech Recognition Using Hidden Control Neural Network Architecture. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, April 1990.

[8] J. Tebelskis and A. Waibel. Large Vocabulary Recognition Using Linked Predictive Neural Networks. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 1990.

[9] 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, April 1984.

[10] K. F. Lee. Large Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System. PhD Thesis, Computer Science Department, Carnegie Mellon University, 1988.

[11] H. Ney, A. Noll. Phoneme Modeling Using Continuous Mixture Densities. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 1988.

[12] N. Tishby. A Dynamic Systems Approach to Speech Processing. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 1990.