S8.7

LARGE VOCABULARY RECOGNITION USING LINKED PREDICTIVE NEURAL NETWORKS

Joe Tebelskis *Alex Waibel*

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

ABSTRACT

We present a large vocabulary isolated word recognition system based on Linked Predictive Neural Networks (LPNN's). In this system, neural networks are employed as predictors of speech frames, enabling a pool of such networks to serve as phoneme models. Higher level algorithms are used to organize these networks, linking them into sequences corresponding to the phonetic spellings of words, and to train and evaluate the networks for word recognition. By virtue of linking phonemic networks, the LPNN is vocabulary independent and can be applied to large vocabulary recognition. We obtained recognition rates of 94% for a 234-word Japanese vocabulary of acoustically similar words and 90% for a larger vocabulary of 924 words.

I. INTRODUCTION

Neural networks have attracted considerable interest in recent years due to their potential usefulness for speech processing. In speech recognition, neural networks have been successfully used for high-performance phoneme recognition [11,8,7] and for small vocabulary recognition [2,9]. Large vocabulary recognition, however, requires sequential control of subword units (e.g., phonemes); hence a logical and common line of current research is to employ a neural network as a phonetic classifier whose decision is fed into a more conventional alignment procedure (e.g., Dynamic Programming or Viterbi Alignment) for sequential control [3,4,8]. In addition to learning discrete classifications, however, neural networks can also learn non-linear mappings between real valued inputs and outputs. This can be exploited in speech for various signal mapping and coding applications, including noise suppression [10] and non-linear predictive coding [1]. The use of neural networks as non-linear signal predictors in speech recognition has recently been shown successfully by Iso [5] and Levin [6]; but both of these models have so far been limited to small vocabulary recognition tasks (i.e., digits). In this paper, we present an extension of the signal prediction based approach that – by virtue of using subword units – is applicable to large vocabulary recognition. By jointly optimizing time alignment and connection weights and by linking the weights (as in [11]) of network predictors corresponding to the same phoneme symbols, training and recognition can be performed without the need for segmentation. Our experiments with such Linked Predictive Neural Networks (LPNN's) indicate that good large vocabulary recognition performance can be achieved in a vocabulary independent fashion. This paper describes the basic LPNN system and several extensions, and evaluates the system's performance.

II. LINKED PREDICTIVE NEURAL NETWORKS

An LPNN network performs phoneme recognition not by classification, but by signal prediction. The idea is illustrated in Figure 1. A network, shown as a triangle, takes K contiguous frames of speech (we normally use K=2), passes these through a hidden layer of units, and attempts to predict the next frame of the speech signal. The predicted frame is then compared to the actual frame. If the error is small, the network is considered to be a good model for that segment of the speech signal. If one could teach the network to make accurate predictions only during segments corresponding to the phoneme "a" (for instance) and poor predictions elsewhere, then one would have an effective "a" phoneme recognizer, by virtue of its contrast with other phoneme models. The LPNN satisfies this condition, as explained below, so that we obtain a collection of phoneme recognizers, with one model per phoneme.

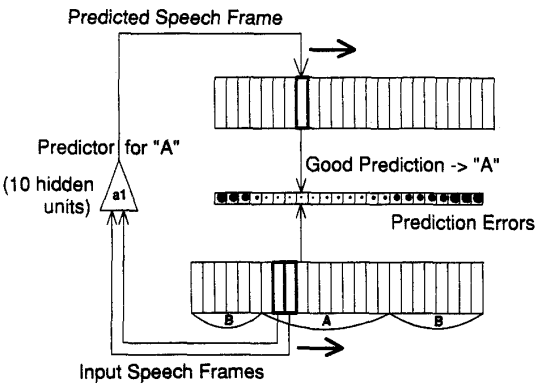

Figure 1: Modeling a phoneme by signal prediction.

CH2847-2/90/0000-0437 $1.00 © 1990 IEEE

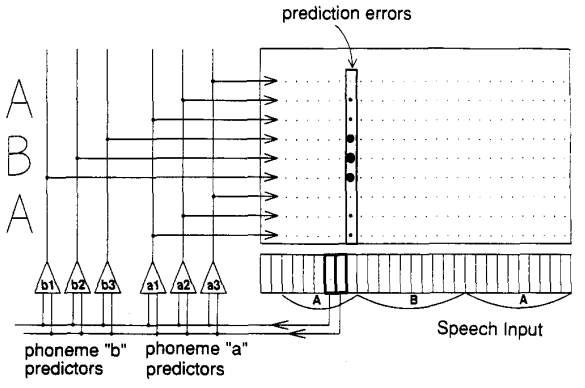

Figure 2: The forward pass during training.

To accomodate the sometimes complex temporal structure of phonemes, each phoneme is actually modelled by a sequence of 3 networks, corresponding to the beginning, middle, and end of the phoneme. For example, the phoneme "b" is modeled by the sequence of networks: b1, b2, b3. The sequentiality of the networks is enforced by the LPNN's implementation.

A word is represented by a logically constrained sequence of models corresponding to the phonetic spelling of the word. This sequence is called the "linkage pattern" for the word. For example, the word "aba" (with phonetic spelling "a-b-a") is represented by the network linkage pattern: a1, a2, a3, b1, b2, b3, a1, a2, a3, as illustrated on the left side of Figure 2. Multiple occurrences of networks (such as a1, a2, and a3) are linked together, as there is only one canonical model for each phoneme. These links enable the LPNN to model phonemes from varying contexts and to recognize words that were not present in the training vocabulary.

IIa. Training the LPNN

Training the LPNN on a word proceeds in three steps: a forward pass, an alignment step, and a backward pass. The purpose of the alignment step is to identify an optimal alignment between the forward pass predictions and the speech signal; this alignment is then used to force specialization in the phoneme models during the backward pass. We now describe the training algorithm in detail.

The first step is the forward pass, illustrated in Figure 2. For each frame of input speech at time t, we feed frame(t-1) and frame(t-2) in parallel into all the networks which are linked into this word, for example the networks a1, a2, a3, b1, b2, and b3 for the word "aba". Each network makes a prediction of frame(t), and its Euclidean distance from the actual frame(t) is computed. These scalar errors are routed through the linkage pattern and stored in column(t) of a prediction error matrix. This is repeated for each frame until the entire matrix has been computed.

The second step finds the optimal alignment between the speech signal and the phoneme models. This is accomplished by applying a standard dynamic programming (DP) algorithm to the prediction error matrix. The DP algorithm finds a monotonically advancing diagonal path through the word's error matrix, such that this path has the lowest possible cumulative error. The constraint of monotonicity ensures the proper sequencing of networks, corresponding to the progression of phonemes in the word. Figure 3 shows some typical alignments (the jagged diagonal lines in either of the prediction error matrices shown).

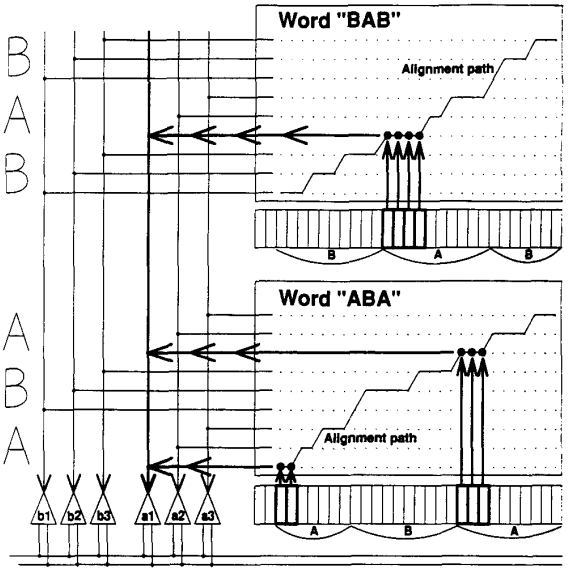

Figure 3: The backward pass during training.

The final step of training is the backward pass, illustrated in the bottom half of Figure 3. In this step, we back-propagate error at each point along the alignment path. In other words, for each frame we propagate error backwards into a single network, namely the one which best predicted that frame according to the alignment; its backpropagated error is simply the difference between this network's prediction and the actual frame. A series of frames may backpropagate error into the same network, as shown. Error is accumulated in the networks until the last frame of the word, at which time all the weights are updated.

This completes the training for a single word. The same algorithm is repeated for all the words in the training set.

The LPNN's training algorithm is remarkably synergistic. The alignment-controlled backpropagation causes each network to specialize on a different section of speech; consequently we obtain a full repertoire of individual phoneme models. This individuation in turn improves the accuracy of future alignments, in a self-correcting cycle. To bootstrap the LPNN on the first iteration (when weights have random values), it has proven useful to force an initial alignment based on average phoneme durations. However, the LPNN quickly outgrows the need for such hints: on subsequent iterations the LPNN itself segments speech on the basis of the increasingly accurate alignments. Furthermore, as illustrated in Figure 3, the same phoneme models are shared by all words, so a network such as a1 learns to predict an "a" sound

in any context. The system can thus generalize to contexts and words it has not seen before.

IIb. Testing LPNN Performance

Testing is performed by "matching" an unidentified speech signal against each word in the vocabulary. This process involves taking the linkage pattern for each word, tentatively assuming that it corresponds to the speech signal, and performing training steps 1 and 2 with it (i.e., a forward pass and alignment with the speech signal). If a linkage pattern does not correspond to the speech signal, its optimal alignment score will be very poor (high cumulative error). Thus, the vocabulary word with the lowest alignment score is identified as the best recognition match. If desired, next-best matches can be determined just by comparing scores.

IIc. Extensions to the System

The basic LPNN architecture is elegant and achieves good performance. We quickly found, however, that its performance can be improved by two simple extensions. These two extensions were used for all the experiments reported in this paper.

The first extension was the use of duration constraints. We applied two types of duration constraints during recognition: 1) hard constraints, where any candidate word whose average duration differed by more than 20% from the given sample was rejected; and 2) soft constraints, where the optimal alignment score of a candidate word was penalized for discrepancies between the alignment-determined durations of its constituent phonemes and the known average duration of those same phonemes.

The second extension was a simple heuristic to sharpen word boundaries. For convenience, we include a "silence" phoneme in all our phoneme sets; this phoneme is linked in at the beginning and end of each isolated word, representing the background silence. Word boundaries were sharpened by artificially penalizing the prediction error for this "silence" phoneme whenever the signal exceeded the background noise level.

Besides these "standard" extensions to the system, we also explored several others variations, including alternate networks, expanded phoneme sets, variable numbers of networks per phoneme, discriminatory learning, and power information. The first two of these will be discussed in detail in the next section.

III. RECOGNITION EXPERIMENTS

The experiments reported here have been carried out on a Japanese isolated word database recorded by one male native Japanese speaker (MAU). All utterances were recorded in a soundproof booth and digitized at a 12 kHz sampling rate. A Hamming window and an FFT were applied to the input data to produce 16 melscale spectral coefficients every 10 msec. From this database, two sets of isolated word samples were extracted for experimentation.

The first set contained almost 300 samples representing 234 unique vocabulary words, all limited to the seven phonemes a,i,u,o,k,s,sh (plus an eighth phoneme for silence). This set was divided into a training set of 229 words and a testing set of 70 words; the testing set included 50 homophones of training samples, and 20 completely unique words. Using homophones in the testing set allowed us to test generalization to new samples of known words, while the unique words allowed us to test generalization to novel words (i.e., vocabulary independence).

A second, larger subset of the Japanese database was selected for further experimentation. This set contained 1078 samples representing 924 unique vocabulary words, all limited to the 13 phonemes a,i,u,e,o,k,r,s,t,kk,sh,ts,tt (plus a 14th phoneme for silence). As before, the utterances were divided into a training set of 900 words and a testing set of 178 words; the testing set included 118 homophones of training samples, and 60 novel words.

Our initial experiments on the 234 word vocabulary used a three-network model for each of the eight phonemes. After training for 200 iterations, recognition performance was perfect for the 20 novel words, and 45/50 (90%) correct for the homophones in the testing set. The fact that novel words were recognized better than new samples of familiar words is due to the fact that most homophones are short confusable words (e.g., "kau" vs. "kao", or "kooshi" vs. "koshi"). By way of comparison, the recognition rate was 95% for the training set.

IIIa. Variations

One successful variation on the standard LPNN architecture was to allow a limited number of "alternate" models for each phoneme. Since phonemes have different characteristics in different contexts, the LPNN's phoneme modeling accuracy can be improved if an independent sequence of networks is allocated for each type of context to be modeled. Rather than assigning an explicit context for each alternate model, however, we let the system itself decide which alternate to use in a given context, by trying each alternate and linking in whichever one yields the lowest alignment score. When errors are backpropagated, the "winning" alternate is reinforced with backpropagated error in that context, while competing alternates remain unchanged.

We evaluated networks with as many as three alternate models per phoneme. As we expected, the alternates successfully distributed themselves over different contexts. For example, the three "k" alternates became specialized for the context of an initial "ki", other initial "k"s, and internal "k"s, respectively. We found that the addition of more alternates consistently improves performance on training data, as a result of crisper internal representations, but generalization to the test set eventually deteriorates as the amount of training data per alternate diminishes. The use of two alternates was generally found to be the best compromise in the experiments reported here.

Significant improvements were also obtained by expanding the set of phoneme models to explicitly represent consonants that in Japanese are only distinguishable by the duration of their stop closure (e.g., "k" versus "kk"). However, allocating new phoneme models to represent diphthongs (e.g., "au") did not improve results, presumably due to insufficient training data.

Table 1 shows the recognition performance of our two best LPNN's, for the 234 and 924 word vocabularies, respectively. Both of these LPNN's used all of the above optimizations. Their

Vocab size	Tolerance	Testing Set		Training Set
		Homophones	Novel words	
234	1	47/50 (94%)	19/20 (95%)	228/229 (99%)
	2	49/50 (98%)	20/20 (100%)	229/229 (100%)
	3	50/50 (100%)	20/20 (100%)	229/229 (100%)
924	1	105/118 (89%)	55/60 (92%)	855/900 (95%)
	2	116/118 (98%)	58/60 (97%)	886/900 (98%)
	3	117/118 (99%)	60/60 (100%)	891/900 (99%)

Table 1: Word recognition performance.

performance is shown for a range of "tolerances", where a tolerance of K means a word is considered correctly recognized if it appears among the best K candidates.

For the 234 word vocabulary, we achieved an overall recognition rate of 94% on test data using an exact match criterion, or 99% or 100% recognition within the top two or three candidates, respectively. For the 924 word vocabulary, our best results so far on the test data are 90% using an exact match criterion, or 97.7% or 99.4% recognition within the top two or three candidates, respectively. Among all the errors made for the 924 word vocabulary (training and testing sets), approximately 15% were due to duration problems, such as confusing "sei" and "seii"; another 12% were due to confusing "t" with "k", as in "tariru" versus "kariru"; and another 11% were due to missing or inserted "r" phonemes, such as "sureru" versus "sueru". On the basis of the systematicity of these errors, we believe that recognition can be further improved by using better duration constraints, and by the better use of techniques such as discriminatory learning and power information.

IV. CONCLUSION

We have presented a large vocabulary recognition system based on neural networks used as predictors. By virtue of using a phonemic representation, the system is applicable to large vocabularies and can recognize words outside its training vocabulary. Recognition performance on test data was 94% and 90% for a 234 and a 924 word vocabulary, respectively. Remaining errors consisted of highly confusable words and recognition within the top two or three candidates was near perfect.

Our experience with the LPNN so far suggests that good recognition performance can be achieved for large vocabularies using predictive neural networks. The predictive networks presented here appear to be particularly well suited to tasks where strong sequential top-down constraints (e.g., phonetic spellings) are available. By contrast, fine phonetic distinctions and confusions between similar sounding words appear to be more easily resolved in bottom-up classification based models[8]. Future research should attempt to address these conflicting limitations.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the support of ATR Interpreting Telephony Research Laboratories, NEC Corporation, and the National Science Foundation.

REFERENCES

[1] Lapedes A. and Farber R. *Nonlinear Signal Processing Using Neural Networks; Prediction and System Modeling.* Technical Report LA-UR-87-2662, Los Alamos National Laboratory, 1987.

[2] Fogelman-Soulie F. Blanchet P. Lienard J.S. Bottou, L. Experiments with time-delay networks and dynamic time warping for speaker independent isolated digits recognition. In *Proceedings of the Eurospeech*, September 1989.

[3] H. Bourlard and C.J. Wellekens. Links between markov models and multilayer perceptrons. In *Advances in Neural Network Information Processing Systems*, Morgan Kaufmann, 1988.

[4] A. Hirai and A. Waibel. *Phoneme-Based Word Recognition by Neural Network -A Step Toward Large Vocabulary Recognition.* Technical Report, Carnegie Mellon University, August 1989.

[5] K. Iso and T. Watanabe. Speaker-independent word recognition using a neural prediction model. In *IEEE International Conference on Acoustics,Speech, and Signal Processing*, IEEE, April 1990.

[6] E. Levin. Speech recognition using hidden control neural network architecture. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, IEEE, April 1990.

[7] E. McDermott and S. Katagiri. Shift-invariant, multi-category phoneme recognition using kohonen's lvq2. In *IEEE International Conference on Acoustics,Speech, and Signal Processing*, page 9.S3.1, IEEE, May 1989.

[8] Sawai H. Shikano K. Miyatake, M. Integrated training for spotting japanese phonemes using large phonemic time-delay neural networks. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 1990.

[9] Isotani R. Yoshida K. Iso K. Sakoe, H. and T. Watanabe. Speaker-independent word recognition using dynamic programming neural networks. In *IEEE International Conference on Acoustics,Speech, and Signal Processing*, pages 29–32, May 1989.

[10] S. Tamura and Waibel A. Noise reduction using connectionist models. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, page S12.7, April 1988.

[11] Hanazawa T. Hinton G. Shikano K. Waibel, A. and Lang K. Phoneme recognition using time-delay neural networks. *IEEE, Transactions on Acoustics, Speech and Signal Processing*, March 1989.