MULTI-SPEAKER/SPEAKER-INDEPENDENT ARCHITECTURES FOR THE MULTI-STATE TIME DELAY NEURAL NETWORK

Hermann Hild and Alex Waibel

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3891, USA

ABSTRACT

In this paper we present an improved Multi-State Time Delay Neural Network (MS-TDNN) for speaker-independent, connected letter recognition which outperforms an HMM based system (SPHINX) and previous MS-TDNNs [2], and explore new network architectures with "internal speaker models". Four different architectures characterized by an increasing number of speaker-specific parameters are introduced. The speaker-specific parameters can be adjusted by "automatic speaker identification" or by speaker adaptation, allowing for "tuning-in" to a new speaker. Both methods lead to significant improvements over the straightforward speaker-independent architecture. Similar as described in [1], even unsupervised "tuning-in" (speech is unlabeled) works astonishingly well.

1. INTRODUCTION

The Multi-State Time Delay Neural Network (MS-TDNN) [2, 5] integrates the time-shift invariant architecture of a TDNN [7] and a nonlinear time alignment procedure (DTW) into a high accuracy word-level classifier. Figure 1 shows an MS-TDNN in the process of recognizing the excerpted word 'B', represented by 16 melscale FFT coefficients at a 10-msec frame rate. The first three layers constitute a standard TDNN, which uses sliding windows with time delayed connections to compute a score for each phoneme (state) for every frame, these are the activations in the "Phoneme Layer". In the "DTW Layer", each word to be recognized is modeled by a sequence of phonemes. The corresponding activations are simply copied from the Phoneme Layer into the word models of the DTW Layer, where an optimal alignment path is found for each word. The activations along these paths are then collected in the word output units. All units in the DTW and Word Layer are linear and have no biases. 15 (25 to 100) hidden units per frame were used for speaker-dependent (-independent) experiments, the entire 26 letter network has approximately 5200 (8600 to 34500) parameters.

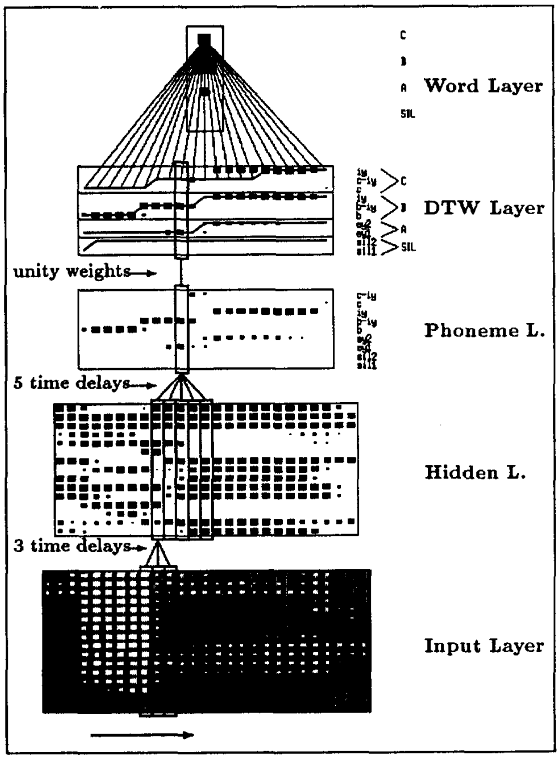

Figure 1: The MS-TDNN recognizing the excerpted word 'B'. Only the activations for the words 'SIL', 'A', 'B', and 'C' are shown.

Training starts with "bootstrapping", during which only the front-end TDNN is used with fixed phoneme boundaries as targets. In a second phase, training is performed with word level targets. Phoneme boundaries are freely aligned within given word boundaries in the DTW Layer. The error derivatives are backpropagated from the word units through the alignment path and the front-end TDNN. Choosing sensible objectives function is important; we are using "McClelland-Error" (similar to cross entropy) on the phoneme level and the "Classification Figure of Merit" [4] on the word level.

0-7803-0946-4/93 $3.00 © 1993 IEEE

2. IMPROVED CONTINUOUS RESULTS

Speaker Dependent (CMU Alph Data) 600/3000 train, 400/2000 test sentences/words			
speaker	SPHINX[2]	MS-TDNN[2]	our MS-TDNN
mjmt	96.0	97.5	**98.5**
mdbs	83.9	89.7	**91.1**
maem	–	–	**94.6**
fcaw	–	–	**98.8**
flgt	–	–	**86.9**
fee	–	–	**91.0**
Speaker Independent (Res. Manag. Spell-mode) 109/11000 train, 11/900) test speaker/words.			
SPHINX[6]		our MS-TDNN	
	+ Senone		gender specific
88.7	90.4	**90.8**	**92.0**

Table 1: Word accuracy (in % on the test set) on speaker dependent and speaker independent connected letter tasks.

Our MS-TDNN achieved excellent performance on both speaker dependent and independent tasks. For **speaker dependent** testing, we used the CMU "Alph-Data", with 1000 sentences (i. e. continuously spelled strings of letters in our context) from each of 3 male and 3 female speakers. **Speaker-independent** performance was measured on the DARPA Resource Management Spell-mode data, consisting of a total of 1680 spelled words from 120 speakers. Table 1 indicates the usage of training and test sets. For the CMU Alph data, 100 of the 600 training words were set aside for cross-validation. For the RM-spell data, one sentence from all 109 speakers and all sentences from 6 speakers were set aside for cross-validation.

In addition to the base-line system as introduced above, several techniques aimed at improving continuous recognition were used, including free alignment across word boundaries, word duration modeling and error backpropagation on the sentence rather than the word level, as described in more detail in [5].

3. ARCHITECTURES FOR SPEAKER MODELING

Selection of "Internal Speaker Models". The idea of the architectures presented is to have submodules in a network, each of which is specialized on one particular speaker (or a group of speakers). In other words, the system contains "internal speaker models" (ISMs) for a set of prototype speakers. When an unknown speaker is presented, somehow one or a (normalized) mixture of appropriate submodule(s) has to be selected. This is done by "internal speaker model selection units" (ISM-SUs, one for each ISM), which influence the network as shown in figure 4 and explained below. We explored two different mechanisms (figure 2): (**a**) An additional "speaker identification net" is trained to control the internal speaker models, i. e. the activations of the ISM-SUs are computed by this net each time before an utterance is recognized, and (**b**) a "tuning-in" process, in which a small set of speech samples from an unknown speaker is used to adapt the selection of the speaker-specific parameters for this speaker. "Tuning-in" is relatively straightforward for labeled samples, the "mixture parameters", i. e. the activations of the ISM-SUs, are found via gradient descent (error is backpropagated into the ISM-SUs, all other weights are frozen), where the objective function is to maximize the performance on the adaptation data. The so found mixture of ISM's is then used for recognition on the entire rest of the test data. "Tuning-in" can also be applied in an unsupervised fashion [1], in which case "phantom targets" (derived from the actual net output) are used.

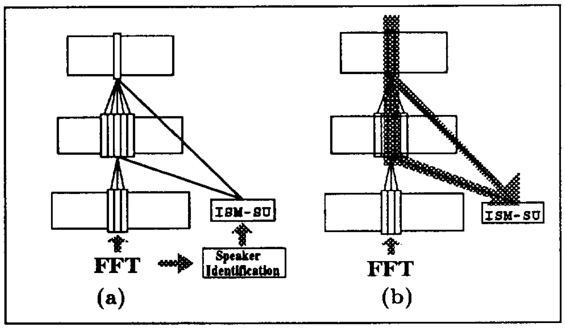

Figure 2: Two methods to adjust the internal speaker models, shown for the BIASED architecture: "Speaker Identification" (a), and "Tuning in" (b).

Four Architectures containing an increasing number of speaker-specific parameters, from no speaker-specific (POOLED) to only speaker-specific parameters (INDIVIDUAL), are shown in Figure 4. In the POOLED net, all 6 speakers are trained into the same single net with no speaker-specific parameters at all. In the BIASED net, an additional layer with ISM-SUs is fully connected into the Hidden and Phoneme Layer, providing a speaker-specific bias for these layers. In the SHARED net, the time-delay connections between the Input and the Hidden Layer are speaker-specific and gated via multiplicative connections by the ISM-SU, i. e. the effective weights between these two layers are a normalized linear combination of the speaker-specific weights. The connections into the Phoneme Layer are shared by all speakers. In the INDIVIDUAL net, every speaker has its own specific TDNN, and the individual ISMs are combined at the phoneme level, i. e. the effective phoneme activations are a linear combination of the speaker-specific phoneme activations, similar to Hampshire's[3] Meta-Pi architecture.

4. EXPERIMENTAL RESULTS

4.1 CMU Alph Data

Multi-Speaker. The 3 male and 3 female speakers listed in Table 1 were used to train and test the four different architectures. The results (% correct, excerpted words, averaged over all 6 speakers) are summarized in table 2. In the "speaker known" column, speaker identity is given and the corresponding ISMs are directly selected. In the "Speaker identified" case, the selection of ISMs is determined automatically with an additional network (figure 2a), which leads to a slight drop in performance, but is still a significant improvement over the POOLED architecture. For the speaker adaptation ("tune-in", figure 2b), 10 spelled words from the test set are used to determine the mixture parameters, which are then used to recognize the remaining test set. Obviously, introducing more speaker specific parameters helps, although the SHARED outperforms the INDIVIDUAL architecture in one case.

Multi-Speaker	Speaker known	Speaker identified	tune-in supervised	tune-in unsuperv.
POOLED	92.7	n/a	n/a	n/a
BIASED	93.8	92.9	92.2	89.7
SHARED	95.9	94.5	95.0	78.5
INDIVID.	95.1	94.9	95.6	95.6

Table 2: In the Multi-Speaker case, the networks are tested with new data from the the same 6 training speakers.

New Speakers. 50 (250) sentences (words) from each of 5 female and 2 male speakers were available to test the system on *new* speakers, i. e. there are no ISMs for these speakers. The results, averaged over all 7 speakers, are shown in table 3. If a mixture of ISMs is automatically determined ("Speaker identified"), the performance improves consistently with a *decreasing* number of speaker-specific parameters, but never reaches the POOLED model (81.3%). However, if speaker adaptation by "tuning-in" to a new speaker is applied, the POOLED model can be outperformed by the BIASED and SHARED architecture, even with unsupervised "tuning-in" for the BIASED architecture. To summarize, for new speakers automatic selection fails but "tuning-in" works.

New Speakers	Speaker known	Speaker identified	tune-in supervised	tune-in unsuper.
POOLED	81.3	n/a	n/a	n/a
BIASED	n/a	79.5	83.2	82.2
SHARED	n/a	73.6	83.4	73.6
INDIVID.	n/a	66.2	72.3	69.3

Table 3: In the New Speaker case, the networks are tested with new data from new speakers.

4.2 RM Spell-mode Data

Since only 6 speakers were available for speaker-independent training of the CMU Alph Data, it is not unexpected that the new speakers perform relatively poor. To test the system on a real speaker-independent task, we performed experiments on the Resource Management Spell-mode data, which contain speech of 85 male and 35 female speakers.

Gender Specific Nets. In a first experiment, we divided the 120 speakers into the two obvious groups of male and female speakers, and trained the four architectures with the corresponding ISMs, as shown in figure 3 for the SHARED architecture. Since the gender identification network classifies almost 100% correct, the results (table 4) for known and automatically determined gender are basically the same.

RM spell	Gender known	Gender identified
POOLED	90.8	n/a
BIASED	90.4	90.4
SHARED	92.0	92.0
INDIVIDUAL	91.3	91.1

Table 4: Word accuracy (continuous speech) on the RM spell test set for the four different architectures

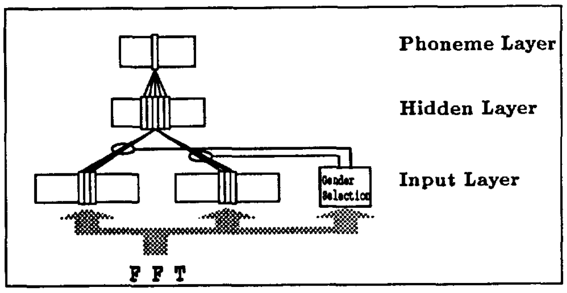

Figure 3: Gender-specific and shared connections in the SHARED architecture. Only the front-end TDNN is shown.

Speaker-Specific Phone Models. The RM spell database has many speakers (120) but little speech (15 spelled words) per speaker. To train one "internal speaker model" for each speaker is impractical, therefore we clustered all 74 male training speakers into 6 ISMs, using a k-means algorithm: a randomly selected speaker was tested on all 6 ISMs, assigned to the ISM on which he performed best, and then the system was retrained. This procedure was repeated until the performance started converging. While this method improved our results on the training data from 98.3% to 99.2% (excerpted words), no gain in performance was achievable by tuning-in to the new speakers of the test set. However, a finer granularity of the ISM-mixing was helpful: So far, when the system tuned-in to one particular ISM, it had to "accept" all its phonemes, i. e. there was no way to use (say) the vowels of one ISM and the consonants of another ISM. In a more flexible tuning-in scheme, an individual speaker-mixture can be selected for each phoneme independently, conceptionelly similar to the speaker-adaptive phoneme models in [?]. With this approach, the performance of the new speakers on the test set improved from 95.4% to 96.5% (excerpted words) with supervised tuning-in. The first 5 sequences of each speaker were used for tuning in, the remaining 10 for testing.

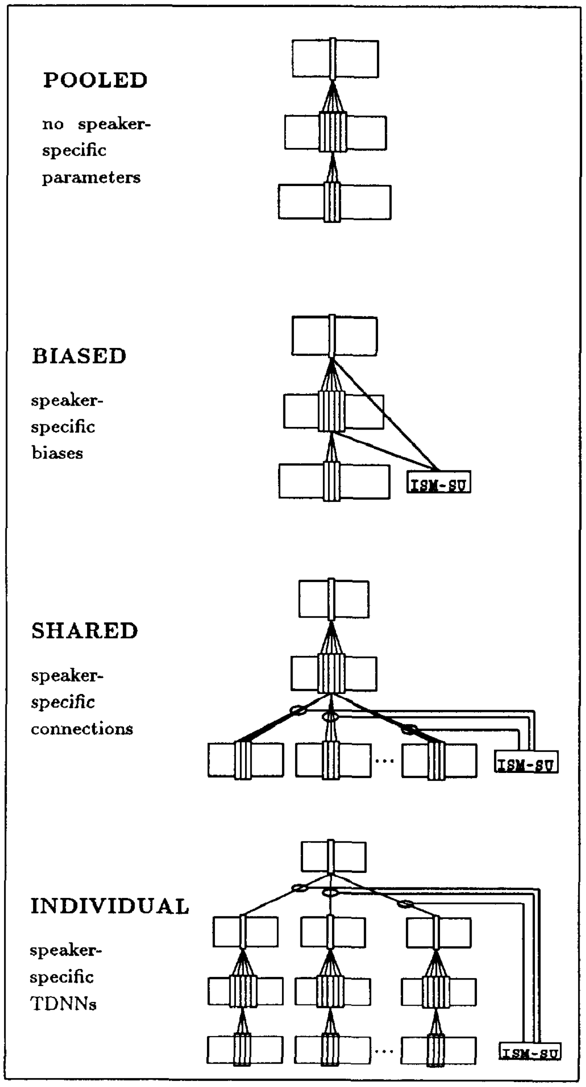

Figure 4: Four net architectures with an increasing degree of speaker specific parameters, which are controlled by the "Internal Speaker Model Selection Units" (ISM-SU).

5. CONCLUSIONS AND FUTURE WORK

We presented a state-of-the-art connectionist speech recognizer for connected letters, and explored several types of architectures for speaker-independent recognition. Our experiments show that introducing speaker-specific parameters improve recognition performance. The degree of desired specialization depends on the amount of available data; in our case, the SHARED architecture seemed to be the best compromise between specific and shared parameters. In a multi-speaker environment, the gating of the specialized parameters (ISMs) can be handled by a "Speaker-Identification Network" for which no additional adaptation data are required, but "tuning-in" on a small adaptation set is a more powerful method which in addition works well for *new* speakers. For the RM task, a finer "tuning-in granularity" (speaker-adaptive phoneme models) was necessary. In the future, we will try to include speaker-specific duration modeling into the adaptation process.

6. ACKNOWLEDGEMENTS

The authors gratefully acknowledge support by the National Science Foundation and DARPA.

7. REFERENCES

[1] J.S. Bridle and S. J. Cox. RecNorm: Simultaneous Normalisation and Classification applied to Speech Recognition. In *Adv. in Neural Network Information Processing Systems (NIPS-4-)*, Morgan Kaufmann.

[2] P. Haffner, M. Franzini, and A. Waibel. Integrating Time Alignment and Neural Networks for High Performance Continuous Speech Recognition. In *Proc. ICASSP*, IEEE, May 1991.

[3] J. Hampshire and A. Waibel. The Meta-Pi Network: Connectionist Rapid Adaptation for High-Performance Multi-Speaker Phoneme Recognition. In *Proc. ICASSP*. IEEE, April 1990.

[4] J. Hampshire and A. Waibel. A Novel Objective Function for Improved Phoneme Recognition Using Time Delay Neural Networks. *IEEE Transactions on Neural Networks*, June 1990.

[5] H. Hild and A. Waibel. Connected Letter Recognition with a Multi-State Time Delay Neural Network. To appear in: *Adv. in Neural Network Information Processing Systems (NIPS-5-)*. Morgan Kaufmann, 1993.

[6] M.Y. Hwang and X. Huang. Subphonetic Modeling with Markov States - Senone. In *Proc. ICASSP*, IEEE, March 1992.

[7] O. Schmidbauer and J. Tebelskis. An LVQ Based Reference Model for Speaker-Adaptive Speech Recognition. In *Proc. ICASSP*, IEEE, March 1992.

[8] 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.