Communicated by Richard Lippmann

Modular Construction of Time-Delay Neural Networks for Speech Recognition

Alex Waibel
Computer Science Department, Carnegie Mellon University,
Pittsburgh, PA 15213, USA
and
ATR Interpreting Telephony Research Laboratories,
Twin 21 MiD Tower, Osaka, 540, Japan

Several strategies are described that overcome limitations of basic network models as steps towards the design of large connectionist speech recognition systems. The two major areas of concern are the problem of time and the problem of scaling. Speech signals continuously vary over time and encode and transmit enormous amounts of human knowledge. To decode these signals, neural networks must be able to use appropriate representations of time and it must be possible to extend these nets to almost arbitrary sizes and complexity within finite resources. The problem of time is addressed by the development of a ***Time-Delay Neural Network,*** **the problem of scaling by** ***Modularity and Incremental Design*** **of large nets based on smaller subcomponent nets. It is shown that small networks trained to perform limited tasks develop time invariant, hidden abstractions that can subsequently be exploited to train larger, more complex nets efficiently. Using these techniques, phoneme recognition networks of increasing complexity can be constructed that all achieve superior recognition performance.**

1 Introduction

Numerous studies have recently demonstrated powerful pattern recognition capabilities emerging from connectionist models or "artificial neural 联结主义模型 networks" (Rumelhart and McClelland 1986; Lippmann 1987). Most are trained on mere presentations of suitable sets of input/output training data pairs. Most commonly these networks learn to perform tasks by effective use of hidden units as intermediate abstractions or decisions in an attempt to create complex, non-linear, decision functions. While these properties are indeed elegant and useful, they are, in their most simple form, not easily applicable to decoding human speech.

Neural Computation **1, 39–46** (1989)

2 Temporal Processing

One problem in speech recognition is the problem of time. A human speech signal is produced by moving the articulators towards target positions that characterize a particular sound. Since these articulatory motions are subject to physical constraints, they commonly don't reach clean identifiable phonetic targets and hence describe trajectories or signatures rather than a sequence of well defined phonetic units. Properly representing and capturing the dynamic motion of such signatures, rather than trying to classify momentary snapshots of sounds, must therefore be a goal for suitable models of speech.

Another consequence of the dynamic nature of speech is the general absence of any unambiguous acoustic cue that indicates when a particular sound occurs. As a solution to this problem, segmentation algorithms have been proposed that presegment the signal before classification is carried out. Segmentation, however, is an errorful classification problem in itself and, when in error, sets up subsequent recognition procedures for recognition failure. To overcome this problem, a suitable model of speech should instead simply scan the input for useful acoustic clues and base its overall decision on the sequence and co-occurrence of a sufficient set of detected lower level clues. This then presumes the existence of translation invariant feature detectors, i.e., detectors that recognize an acoustic event independent of its precise location in time.

A "Time Delay Neural Network" (TDNN) (Lang 1987; Waibel et al. 1987) possesses both of these properties. It consists of TDNN-units that, in addition to computing the weighted sum of their current input features, also consider the history of these features. This is done by introducing varying delays on each of the inputs and processing (weighting) each of these delayed versions of a feature with a separate weight. In this fashion each unit can learn the dynamic properties of a set of moving inputs. The second property, *"translation invariance"* is implemented by TDNN-units that scan an input token over time, in search of important local acoustic clues, instead of applying one large network to the entire input pattern. Translation invariant learning in these units is achieved by forcing the network to develop useful hidden units regardless of position in the utterance. In our implementation this was done by linking the weights of time shifted instantiations of the net during a scan through the input token (thus removing relative timing information). Figure 1 illustrates a TDNN trained to perform the discrimination task between the voiced stop consonants /b, d, g/ (Waibel et al. 1987) for a more detailed description of its operation).

The three-category TDNN shown here (Fig. 1) has been evaluated over a large number of phonetic tokens (/b,d,g/). These tokens were generated by extracting the 150 msec intervals around pertinent phonemes from a phonetically handlabeled, large vocabulary database of isolated

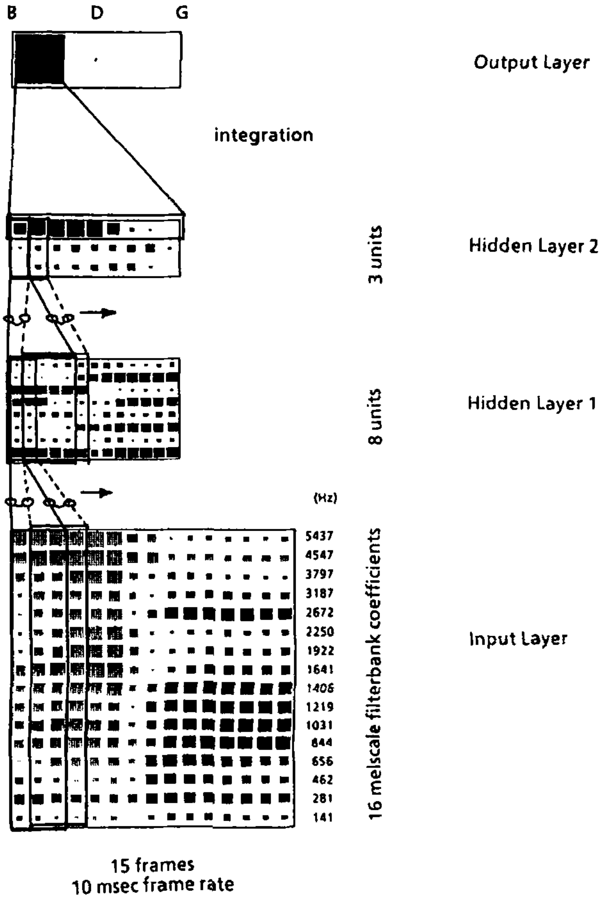

Figure 1: The TDNN architecture (input: "BA"). Eight hidden units in hidden layer 1 are fully interconnected with a set of 16 spectral coefficients and two delayed versions illustrated by the window over 48 input units. Each of these eight units in hidden layer 1 produces patterns of activation as the window moves through input speech. A five frame window scanning these activation patterns over time then activates each of three units in hidden layer 2. These activations over time in turn are then integrated into one single output decision. Note that the final decision is based on the combined acoustic evidence, independent of *where* in the given input interval (15 frames or 150 msecs) the /b, d or g/ actually occurred.

Japanese utterances (Waibel et al. 1987). While isolated pronunciation provided relatively well articulated tokens, the data nevertheless included significant variability due to different phonetic contexts (e.g., "DO" vs. "DI") and position in the utterance. Recognition experiments with three different male speakers showed that discrimination scores between 97.5% and 99.1% could be obtained.[1] These scores compare favorably with those obtained using several standard implementations of Hidden Markov Model speech recognition algorithms (Waibel et al. 1987).

To understand the operation of the TDNNs, the weights and activation patterns of trained /b,d,g/-nets have been extensively evaluated (Waibel et al. 1987). Several interesting properties were observed:

1. The TDNNs developed linguistically plausible features in the hidden units, such as movement detectors for first and second formants, segment boundary detectors, etc.
2. The TDNN has developed alternate internal representations that can link quite different acoustic realizations to the same higher level concept (here: phoneme). This is possible due to the multilayer arrangement used.
3. The hidden units fire in synchrony with detected lower layer events. These units therefore operate independent of precise time alignment or segmentation and could lead to translation invariant phoneme recognition.

Our results suggest that the TDNN has most of the desirable properties needed for robust speech recognition performance.

3 The Problem of Scaling

Encouraged by the good performance and the desirable properties of the model, we wanted to extend TDNNs to the design of large scale connectionist speech recognition systems. Some simple preliminary considerations, however, raise serious questions about the extendibility of connectionist design: Is it feasible, within limited resources and time, to build and train ever larger neural networks? Is it possible to add new knowledge to existing networks? With speech being one of the most complex and all encompassing human cognitive abilities, this question of scaling must be addressed.

As a first step, let us consider the problem of extending the scope of our networks from tackling the three category task of all voiced stops (/b,d,g/) to the task of dealing with all stop consonants (/b,d,g,p,t,k/). The first row in table 1 shows the recognition scores of two individually

[1] All recognition scores in this paper were obtained from evaluation on test data that was not included in training.

Method	bdg	ptk	bdgptk
Individual TDNNs	98.3%	98.7%	
TDNN: Max. Activation			60.5%
Retrain BDGPTK			98.3%
Retrain Combined Higher Layers			98.1%
Retrain with V/UV-units			98.4%
Retrain with Glue			98.4%
All-net Fine Tuning			98.6%

Table 1: From /b,d,g/ to /b,d,g,p,t,k/; Modular Scaling Methods.

trained three category nets, one trained on the voiced stop consonant discrimination task (/b,d,g/) and the other on the voiceless stop consonant discrimination task (/p,t,k/). A naive attempt of combining these two nets by simply choosing the maximally activated output unit from these two separately trained nets resulted in failure as seen by the low recognition score (60.5%) in the second row. This is to be expected, since neither network was trained using other phonetic categories, and independent output decisions minimize the error for only small subsets of the task. A larger network (/b,d,g,p,t,k/-net) with six output units was therefore trained. Twenty hidden units (instead of eight) were used in Hidden layer 1 and six in hidden layer 2. Good performance could now be achieved (98.3%), but significantly more processing had to be expended to train this larger net. While task size was only doubled, the number of connections to be trained actually tripled. To make matters worse, more training data is generally needed to achieve good generalization in larger networks and the search complexity in a higher dimensional weight space increases dramatically as well. Even without increasing the number of training tokens in proportion to the number of connections, the complete /b,d,g,p,t,k/-net training run still required 18 days on a 4-processor Alliant supermini and had to be restarted several times before an acceptable solution had been found. The original /b,d,g/-net, by comparison, took only three days. It is clear that learning time increases more than linearly with task size, not to mention practical limitations such as available training data and computational capabilities. In summary, the dilemma between performance and resource limitations must be addressed if Neural Networks are to be applied to large real world tasks.

Our proposed solutions are based on three observations:

1. Networks trained to perform a smaller task may not produce outputs that are useful for solving a more complex task, but the knowledge and internal abstractions developed in the process may indeed be valuable.
2. Learning complex concepts in (developmental) stages based on previously learned knowledge is a plausible model of human learning and should be applied in connectionist systems.
3. To increase competence, connectionist learning strategies should *build* on existing distributed knowledge rather than trying to undo, ignore or relearn such knowledge.

Four experiments have been performed:

1. The previously learned hidden abstractions from the first layer of a /b,d,g/-net and a /p,t,k/-net were frozen by keeping their connections to the input fixed. Only connections from these two hidden layers 1 to a combined hidden layer 2 and to the output layer were retrained. While only modest (a few hours of) additional training was necessary at the higher layers, the recognition performance (98.1%) was found to be almost as good as for the monolithically trained /b,d,g,p,t,k/-net (see table 1). The small difference in performance might have been caused by the absence of features needed to merge the two subnets (here, for example, the voicing feature distinguishing voiced from voiceless stops).
2. Hidden features from hidden layer 1 are fixed as in the previous experiment, but four additional class-distinctive features are incorporated at the first hidden layer. These four units were excised from a net that was exclusively trained to perform voiced/unvoiced discrimination. The voiced/unvoiced net could be trained in little more than one day and combination training at the higher layers was accomplished in a few hours. A high recognition rate of 98.4% was achieved.
3. The hidden units from hidden layer 1 are fixed as before, and four additional *free* units are incorporated. These free units are called *connectionist glue,* since they are intended to fit or glue together two distinct, previously trained nets. This network is shown in figure 2. The four glue units can be seen to have free connections to the input that are trained along with the higher layer combinations. In this fashion they can discover additional features that are needed to accurately perform the larger task. In addition to training the original /b,d,g/- and /p,t,k/-nets, combination training using glue

units was accomplished in two days. The resulting net achieved a recognition rate of 98.4%.

4. All-net fine tuning was performed on the previous network. Here, *all* connections of the entire net were freed once again for several hours of learning to perform small additional weight adjustments. While each of these learning iterations was indeed very slow, only few iterations were needed to fine tune the entire network for best performance of 98.6%.

Only modest additional training beyond that required to train the subcomponent nets was necessary in these experiments. Performance, however, was as good or better than that provided by a monolithically trained net and as high as the performance of the original smaller subcomponent nets.

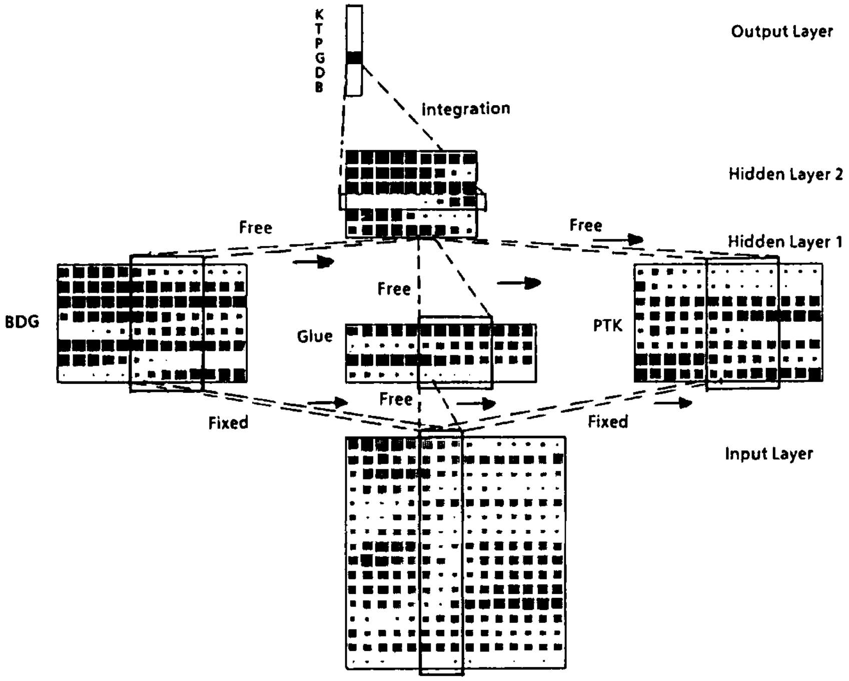

Figure 2: Combination of a /b,d,g/-net and a /p,t,k/-net using 4 additional units in hidden layer 1 as free "Connectionist Glue."

4 Conclusion

We have described connectionist networks with delays that can represent the dynamic nature of speech and demonstrated techniques to scale these networks up in size for increasingly large recognition tasks. Our results suggest that it is possible to train larger neural nets in a modular, incremental fashion from smaller subcomponent nets without loss in recognition performance. These techniques have been applied successfully to the design of neural networks capable of discriminating all consonants in spoken isolated utterances (Waibel et al. 1988). With recognition rates of 96%, these nets were found to compare very favorably (Waibel et al. 1988) with competing recognition techniques in use today.

Acknowledgments

The author would like to acknowledge his collaborators: Toshiyuki Hanazawa, Geoffrey Hinton, Kevin Lang, Hidefumi Sawai, and Kiyohiro Shikano, for their contributions to the work described in this paper. This research would also not have been possible without the enthusiastic support of Akira Kurematsu, President of ATR Interpreting Telephony Research Laboratories.

References

Lang, K. 1987. *Connectionist Speech Recognition*. Ph.D thesis proposal, Carnegie Mellon University.

Lippmann, R.P. 1987. An Introduction to Computing with Neural Nets. *IEEE ASSP Magazine*, 4–22.

Rumelhart, D.E. and J.L. McClelland. 1986. *Parallel Distributed Processing; Explorations in the Microstructure of Cognition*. Cambridge, MA: MIT Press.

Waibel, A., T. Hanazawa, G. Hinton, K. Shikano, and K. Lang. 1987. *Phoneme Recognition Using Time-Delay Neural Networks*. Technical Report TR-1-0006, ART Interpreting Telephony Research Laboratories. Also scheduled to appear in *IEEE Transactions on Acoustics, Speech and Signal Processing*, March 1989.

Waibel, A., H. Sawai, and K. Shikano. 1988. *Modularity and Scaling in Large Phonemic Neural Networks*. Technical Report TR-I-0034, ATR Interpreting Telephony Research Laboratories; *IEEE Transactions on Acoustics, Speech and Signal Processing*, to appear.

Received 6 November; accepted 8 December 1988.