Advances in Speech Signal Processing

edited by

Sadaoki Furui
NTT Human Interface Laboratories
Tokyo, Japan

M. Mohan Sondhi
AT&T Bell Laboratories
Murray Hill, New Jersey

Marcel Dekker, Inc. New York • Basel • Hong Kong

18

Neural Network Approaches for Speech Recognition

ALEXANDER WAIBEL Carnegie-Mellon University, Pittsburgh, Pennsylvania

1. INTRODUCTION

In the light of three decades of activity aimed at systems capable of understanding natural human speech, one might wonder if the problem has been solved and, if not, what might be required to achieve this challenging goal. To be sure, the field has seen impressive advances and several very important lessons have been learned. Despite these inroads, however, present-day systems still fall short of the ease and reliability with which humans can communicate with each other by speech. This gap has indeed led to a continuing search for new models and techniques that might bring us closer to machines that have this ability, but also to a better understanding of cognitive processes such as speech perception and understanding.

Neural networks are the most recent development in this search for new models of speech understanding. Most of what will be described is relatively recent work and only experiments on parts of the problem have been completed to date. Although very promising work in this direction has begun, no fully integrated speaker-independent large-vocabulary speech understanding systems based on neural networks alone yet exist. What fuels the excitement and what is the promise behind "neural networks," "connectionism,"

or "parallel distributed processing," as these models have often been termed interchangeably?

A partial list of attractive properties of connectionism is given in the following. Note that some of these properties are not necessarily unique to the connectionist approach. Indeed, some of the important ones are shared with other recognition schemes and are partially responsible for their success.

- *Massive parallelism*: A connectionist net is composed of many simple computing units, and computation is performed in parallel and in a distributed fashion by many interconnected computing elements. Potential advantages resulting from this are speed, regularity (for hardware implementation), and fault tolerance.
- *Constraint satisfaction*: Processing in connectionist nets is not performed sequentially and does not depend on the performance of any one single computing element, but on many processing elements' joint evaluation of numerous interrelated constraints.
- *Learning*: Massively parallel architectures cannot be easily programmed and parallel distributed processing models must rely on automatic learning algorithms. A number of such algorithms now exist, including training techniques for multilayer perceptrons (back propagation), Boltzmann machines, learning vector quantization, and associative nets (for reviews see Lippmann, 1987; Rumelhart and McClelland, 1986; Hinton, 1987; Kohonen, 1988). These learning algorithms optimize local computing elements to jointly improve some more global overall objective.
- *Stochastic modeling, uncertainty, variability, fuzziness*: Connectionist models deal with variability and noise by finding suitable probabilistic generalizations. Probabilities are encoded in such a network predominantly as patterns of activity across its elements, instead of single scalar values. Connectionist networks do not assume any particular statistical distributions, and hence no parametric assumptions need to be made.
- *Nonlinear modeling*: Connectionist networks are nonlinear models that can implement nonlinear classifiers and mapping functions and represent multimodel distributions and complex relationships. This may lead to better performance than linear models in various classification, mapping, and interpolation tasks.
- *Discovery of "hidden" knowledge*: Connectionist networks generate hidden knowledge, abstractions, and generalizations in the process of solving a more complex problem. In multilayer perceptrons, this hidden knowledge is often encoded in the connection weights learned by so-called hidden units (Rumelhart and McClelland, 1986). If knowledge can be extracted from or encoded into these networks effectively, this may provide mechanisms for bridging the gap between knowledge-based approaches and stochastic models.
- *Uniformity*: Computation in connectionist nets is performed by simple underlying computing elements and the interaction between them. The computing steps performed by a particular unit (usually simple multiplications and additions) are generally independent of the task that the network is trying to solve. This is very useful for hardware implementations as the units are simple (cheap) and the same units can be used for a variety of tasks. Uniformity is also attractive as a means of achieving sensory fusion, i.e., the potential combination of different signals or input information (in speech, for example, phonetic and visual cues, or syntactic, semantic, and prosodic cues, etc.), possibly at varying levels of processing.
- *Speed—learning vs. recognition*: By virtue of massive parallel computation, connectionist nets can run very efficiently. Some connectionist models, however, do require considerable training to be performed.
- *Brain-style computation (?)*: Connectionist models attempt to simulate the style of computation as it is performed in the nervous system. Although similarities do exist, this analogy should not be carried too far. Our understanding of "brain-style computation" is still too fragmentary and our present engineering efforts too limited to warrant such comparisons. It is also necessary to replicate the brain accurately to build useful computer speech understanding systems. The human brain is, however, an existence proof that intelligent communication via speech is possible and insights and intuitions gleaned from its computational mechanisms could* usefully inspire new models and ideas for practical design.

Our goal in the following is to present a review of connectionist speech recognition models to date. We will omit a general introduction and assume that the reader is familiar with some of the more common neural network models. For more detailed introductions to neural networks in general, we recommend tutorial papers by Lippmann (1987) and Hinton (1987). Fundamentals of connectionist models and learning algorithms (e.g., back propagation, Boltzmann machines, associative nets, LVQ) will also not

*Along with many other scientific disciplines, of course.

be covered here. Important background material may be found in Rumelhart and McClelland (1986), Rumelhart et al. (1986), Hinton and Sejnowski (1986), and Kohonen (1988). We will limit our discussion to a review of connectionist advances specifically applied to speech recognition and understanding. So much has been written on this subject recently that we cannot hope to review all the activity in this area. We will limit ourselves therefore to a representative subset of important current activity, along with presentation of some of our own recent work. Three important levels of the speech understanding problem will be addressed:

- The phonemic level
  - Temporally static networks
  - Temporally dynamic networks
  - Modularity, scaling
  - Phoneme spotting
  - Speaker independence
- The word level
  - Static full-word models
  - Dynamic full-word models
  - Dynamic large-vocabulary models
  - Word model enhancements
- The language level
  - Word category prediction and disambiguation
  - Recurrent networks
  - Parsing

2. THE PHONEMIC LEVEL

Much of the earliest work in connectionist speech recognition has focused on phoneme recognition. This choice is motivated largely by the fact that phoneme recognition is (compared to the full-speech understanding problem) a tractable subproblem, difficult enough to be interesting, and also a useful focus, as large-vocabulary systems need to make use of atomic subunits or submodels of speech. Phoneme classification networks can be divided into two groups: (1) those that require precise temporal alignment of input tokens for accurate recognition performance (making them temporally static classifiers) and (2) those that do not (making them temporally dynamic or shift invariant).

2.1. Temporally Static Networks

In a static model, a speech pattern or feature vector centered around some boundary or target point is used as input to a neural network used as a classifier with outputs representing each of the phoneme categories in a language.

Figure 1 serves as an illustrative example of an early static application of neural networks to phoneme recognition tasks. Here, Huang and Lippmann (1988; Lippmann and Gold, 1987), apply a three-layer back-propagation network to vowel classification. The back-propagation training procedure was used and successfully learned to form nonlinear classification regions around vowel classes. For input features formant frequencies F_1 and F_2, as measured by Peterson and Barney (1952) in studies of adult and child male and female subjects, were used. The network has 50 units in its hidden layer and was trained for 50,000 trials, resulting in interformant boundaries comparable to those one would draw by hand and those formed by more traditional nonlinear classification techniques such as k-nearest neighbor classifiers (Duda and Hart, 1973; Makhoul et al., 1985). No parametric assumptions needed to be made, and the network provided a good and early demonstration of the usefulness of such networks as nonparametric nonlinear classifiers. The input features, however, were obtained by a human experimenter measuring formants from actual speech spectrograms, indeed a cognitive task in itself.

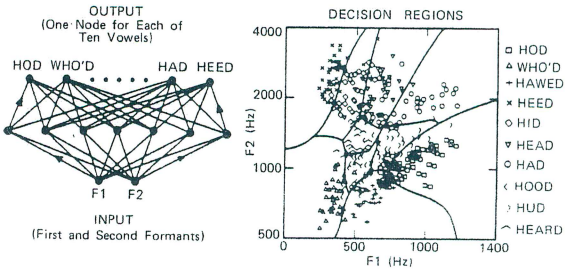

Figure 1 A three-layer back-propagation network used to form classification boundaries on the formants F_1 and F_2 for vowels.

Elman and Zipser (1987) reported phoneme classification based on experiments using actual speech patterns for the voiced-stop consonants /b, d, g/ (followed by the vowels /a, i, u/). 505 tokens of the nine discrete voiced-stop syllables were parsed from recordings of a single male speaker using a 10-kHz sampling rate applied to 3.5-kHz low-pass filtered speech. Twenty 16-coefficient discrete Fourier transforms (DFTs) are computed at overlapping 3.2-ms intervals to form the input of a three-layer back-propagation network. In a series of experiments, hidden layer and output layer node counts were varied. In one case, nine output nodes corresponding to the nine possible syllables were used; in two other cases an output node count of three corresponding to the three voiced-stop phonemes /b, d, g/ was used. More than 100,000 training passes were run for each experiment, using approximately half of the tokens as training exemplars. Recognition rates for the disjoint test data set were 84% for whole syllables, 98.5% for vowels, and 92.1% for voiced-stop consonants. Elman and Zipser found that introducing uniformly distributed white noise to training tokens at the input layer improved recognition rates to 90%, 99.7%, and 95%, respectively. The noise source tended to obscure idiosyncrasies of the training data and improve the networks' ability to generalize to unseen test data.

Both of these models used the back-propagation training procedure, but other methods have been proposed with similar success. Among them, good results were reported by Niranjan and Fallside (1988) using radial basis functions, by Kohonen and colleagues using phonotopic maps and learning vector quantization (LVQ) (Kohonen, 1988; Kohonen et al., 1988), and by Prager et al. (1986) using Boltzmann machines.

2.2. Temporally Dynamic Networks

The experiments reported above achieved successful classification for patterns that were presegmented and extracted from the signal by hand or by some presegmentation procedure. In a full system design, however, such segmentation would eventually have to be performed automatically, and even the best automatic segmentation schemes are prone to make errors. These errors, in turn, generally result in higher error rates further along in the recognition process. A robust speech recognition system should therefore *scan* the speech signal for useful cues *without* relying on presegmentation, basing its overall recognition decision on the sequence and co-occurrence of a sufficient set of those cues. Neural networks, therefore, should be temporally dynamic or shift invariant (i.e., classification that is unaffected by temporal shifts of the input speech train). The experiments reported in the following all employed techniques aimed at yielding shift-invariant phoneme recognition.

2.2.1. *The Time Delay Neural Network (TDNN)*

One of the earlier models that demonstrated successfully that this can be done in a connectionist framework was reported by Waibel (1989), Waibel et al. (1989a,b), and Lang et al. (1990). Their time delay neural network (TDNN) architecture was aimed at high-accuracy phoneme recognition under varying conditions of phoneme duration and temporal location within the speech signal. Figure 2 illustrates the TDNN architecture. It consists of neural units that use time-delayed connections at each layer to capture varying amounts of contexts at the layer below. The sizes of the resulting temporal input windows increase with increasing layers to learn increasingly coarser abstractions at progressively higher layers. As lower layers produce firing patterns, higher layers observe the resulting patterns of activations. During training shift invariance was achieved by making time-shifted copies of the net and linking their corresponding weights. Thus, knowledge of position in time was removed and the network had to spot relevant features anywhere in the input range to assemble sufficient evidence in favor of one of the output phoneme classes. A set of /b, d, g/ tokens was extracted independent of context from a large-vocabulary data base of 5240 Japanese words (Sagisaka et al., 1987) spoken by three male speakers, resulting in approximately 200 training and 200 test tokens per speaker. Recognition results on test data yielded an average recognition rate of 98.5% across all speakers.* Cursory studies of the effects of temporal shifts of the test data input spectra with respect to vowel onset suggested that nominal shifts had little appreciable effect on recognition rates. Comparisons to a variety of hidden Markov models (HMMs) also showed that significantly higher recognition rates could by obtained using the TDNN for this discrimination task (Waibel et al., 1987, 1989a). Using the same speech data, the HMMs tried achieved an average recognition rate of 92.7%. Detailed analysis of the internal representations formed by the network also showed that a number of interesting, linguistically plausible features were "discovered." Hidden layer activations showed specific response to acoustic-phonetic features such as detectors for unvoiced speech, vowel onsets, and rising or falling formants.

2.2.2. *Recurrent Nets*

A recurrent architecture aimed at shift-invariant phoneme recognition was developed by Watrous (1988). His temporal flow model was evaluated on

*Higher recognition rates than those reported here have been observed in later training runs.

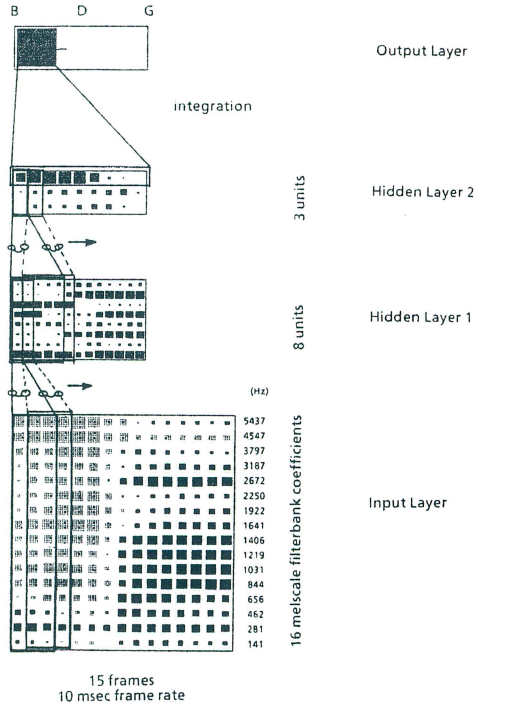

Figure 2 A time-delay neural network (TDNN) for stop consonants /b, d, g/. Eight hidden units in hidden layer 1 are fully interconnected with a set of 16 spectral coefficients and two delayed versions illustrated by the window over 48 input units. Each of these eight units in hidden layer 1 produces 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 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 ms) the /b, d, or g/ actually occurred.

consonant (/b, d, g/) and vowel (/i, a, u/) tasks and achieved similarly good performance figures. It employed recurrent connections, nonbinary output targets (the network is trained to produce a Gaussian-distributed activation across its output nodes), and time-delayed connections. The model was applied to hand-segmented speech from a single male speaker, yielding recognition rates of 99.2% for the /b, d, g/ task and 100% for the /i, a, u/ task. Extensions of these ideas to connected multispeaker recognition of four English letters, B, D, E, and V, have also been successfully carried out by Kuhn et al. (1989).

2.2.3. *Learning Vector Quantization (LVQ)*

McDermott and Katagiri (1989; McDermott et al., 1990) applied LVQ, proposed by Kohonen and collaborators (Kohonen 1988; Kohonen et al., 1988b), to the same Japanese consonant recognition tasks used by Waibel et al. The input layer structure and the shift-tolerant design were motivated by experience with the TDNN and allowed for comparisons (Waibel et al., 1989a). Figure 3 illustrates the moving spectral window used as input features to the LVQ classifier and its extension LVQ2. Hidden layer connections were initialized using a traditional k-means clustering algorithm (Makhoul et al., 1985). The network achieved a 99.2% recognition rate for the voiced-stop consonants /b, d, g/ excised from spoken isolated word utterances from a single male speaker. Performance for all stops, fricatives, and affricates for the same speaker, using a larger LVQ2 network, was 97.1 and 97.7%.* Network training time was somewhat less than training time for a comparable TDNN on the same task, at the cost of a three-fold increase in the total number of connections required in the network and an increase in the time required for posttraining recognition. Unlike feedforward multilayer networks, LVQ networks do not readily generate hidden abstractions of knowledge. On the other side, vectors are produced that provide an efficient encoding of speech that can be incorporated into traditional stochastic models. The potential advantage is that LVQ is a supervised training procedure and LVQ-trained vectors provide an encoding that possibly better represents phonetically relevant featural distinctions. The resulting encoding could therefore replace traditional (unsupervised) vector quantization and yield a crisper representation of speech for improved hidden Markov modeling. Work in this direction is in progress. (Iwamida et al., 1990; McDermott et al., 1990).

*Depending on the amount of training data used.

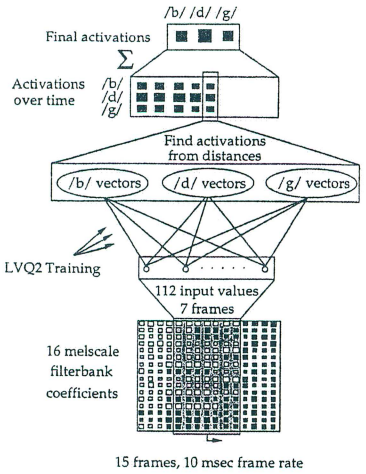

Figure 3 An LVQ network used for the /b, d, g/ recognition task.

2.3. Extensions

Several remaining problems, however, need to be addressed. The first is the problem of scaling and incremental learning, which plagues many neural network discriminant classifiers and can lead to prohibitively large amounts of training time and inflexible classifiers. It would be preferable if larger networks could be gradually built from smaller previously trained subcomponents. Second, can such networks be used to reliably *spot* phonemes in running speech, rather than to discriminate among them? Finally, how do these networks perform when faced with multispeaker or speaker-independent tasks? A number of extensions have been studied to answer these questions.

2.3.1. *Scaling*

Back-propagation training of large phonemic networks is an example of a training task that can require considerable computational resources. One approach to alleviating this training problem is the introduction of specially designed fast simulators [Haffner 88, Haffner 89]. Beyond speed improvements, however, it is clear that the full complexity of cognitive tasks requires incremental learning of new information and categories. Modular, incremental training was explored based on TDNNs to allow for more flexible, efficient, structured design of large neural network–based systems (Waibel, 1988, 1989; Waibel et al., 1989b). In doing so, one would certainly not want to lose the distributed, globally optimizing nature of connectionist learning. The underlying idea here is a compromise between these two extremes, namely to reuse the featural abstractions encoded in the *hidden* units. Hidden units from networks trained to perform smaller subtasks are linked into larger nets aiming at more complex tasks. A number of architectural schemes aimed at increasing the scale of phoneme recognition networks through an arrangement of interconnected modules were evaluated and found to yield performance as good as or better than that of monolithically trained nets. Figure 4 is an illustration of a modular stop-consonant TDNN that was trained by merging two submodules using "connectionist glue." Here connections to hidden units at the first hidden layer of a /b, d, g/-discriminating net and to the first hidden layer of a /p, t, k/ are frozen, while connections from these units to common higher layers are trained to join the two nets. The "connectionist glue" units are additional units that are free to learn missing features when these separate networks are merged (Waibel et al., 1989b). Figure 5 shows a network that was constructed in a modular fashion from subnetworks to recognize all phonemes in a Japanese large-vocabulary isolated word data base. Consonant discrimination results well in excess of 96% were achieved using such networks.

2.3.2. *Phoneme Spotting*

A related problem here is whether a network is able to spot phonemes in running speech rather than discriminate among them if a range of suitable input data is provided. Various experiments that enhance a TDNN's ability to model shift-invariant features of speech have shown that this can indeed be done. By introducing training patterns with varying time shifts around a particular phoneme boundary and counterexamples, Lang (1989) and Miyatake et al. (1990) showed that excellent letter- and phoneme-spotting performance can be achieved. The latter study reported 98% phoneme-spotting performance over running Japanese isolated word utterances. Word recognition

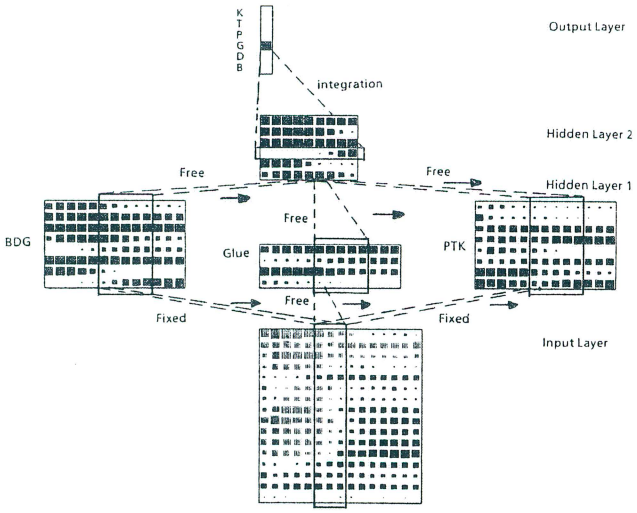

Figure 4 Incremental training: from /b, d, g/ to /b, d, g, p, t, k/.

based on these results can indeed be carried out, as we shall see in the next section.

2.3.3. *Speaker-Independent/Multispeaker Recognition*

Multispeaker and speaker-independent recognition has also been attempted by a number of investigators. As with other recognition techniques, speaker independence can be achieved by training a connectionist network using training data from many different speakers. This has been demonstrated by Leung (1988) and Leung and Zue (1988) for phoneme recognition in continuous English speech from the TIMIT speech data base. Phoneme recognition performance of up to 64% was achieved for a 16-vowel discrimination task, given excised frames of speech. If other sources of information (both numeric

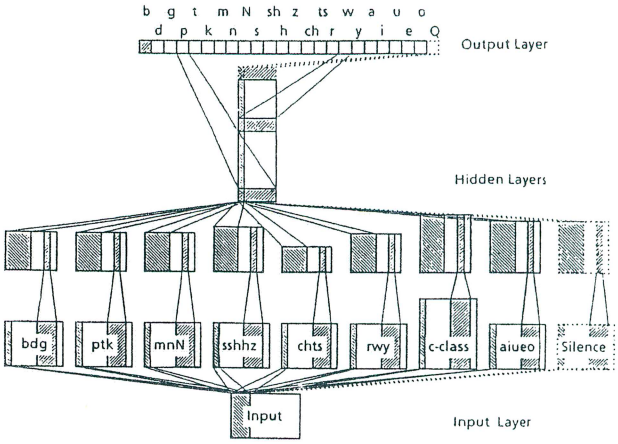

Figure 5 Modular all-phoneme TDNN.

and symbolic), such as durational and contextual information, are provided to the network, performance improves to up to 77%. The study illustrates how connectionist models lend themselves to fusing heterogeneous sources of information and knowledge gracefully.

Another model aimed at multispeaker and speaker-independent recognition was proposed by Hampshire and Waibel. It was motivated by the desire to mimic humans' ability to adapt rapidly within a syllable or two (Kato and Kakehi, 1987) to a speaker's voice. Their model seeks to represent speaker differences explicitly and provide the networks with a mechanism that rapidly focuses on a suitable speaker or set of speakers. The Meta-Pi architecture is the mechanism by which this was done. It is a hierarchical connectionist phoneme classifier that performs multispeaker phoneme discrimination at speaker-*de*pendent rates. The overall network is composed of speaker-dependent submodules (i.e., TDNNs trained to classify the phonemes of a particular speaker) that are linked together by an integrating superstructure. The Meta-Pi superstructure is itself a TDNN. The

outputs of the superstructure act as connection weights, gating the speaker-dependent classification decisions of all the modules to a global classification decision. Thus, the global classification decision is a linear combination of the constituent speaker-dependent classification decisions. The connections of the Meta-Pi superstructure are trained so that the resulting combination of speaker-conditional module outputs produces a correct global classification. Individual speaker-dependent nets are trained individually, as before, as regular TDNN's. Figure 6 shows how the error signal obtained from the global classification output then back-propagates through multiplicative con-

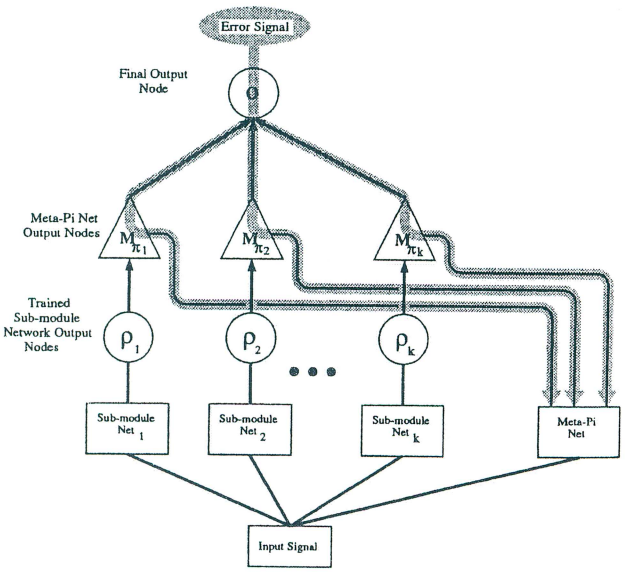

Figure 6 Error back propagation in a Meta-Pi architecture.

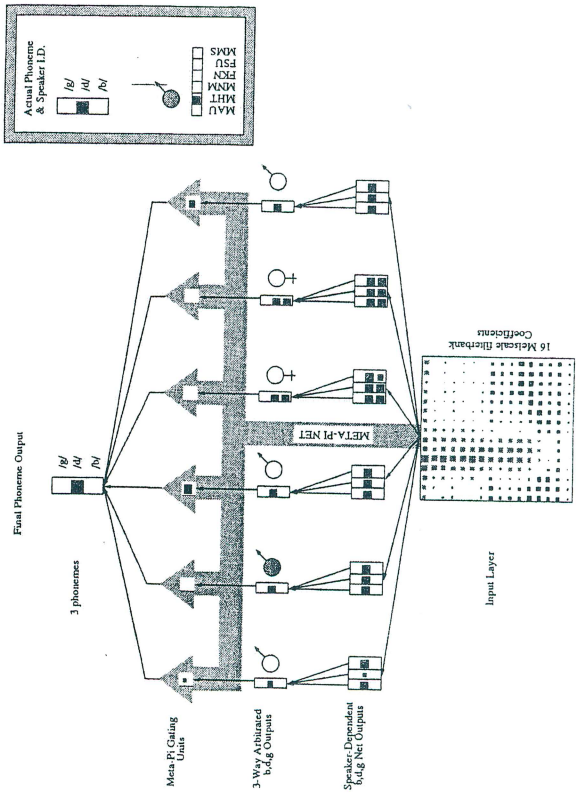

Figure 7 Typical activations in a Meta-Pi network for multispeaker phoneme discrimination.

nections into the Meta-Pi superstructure to optimize the integrating Meta-Pi net. Figure 7 illustrates the fully trained Meta-Pi architecture performing the /b, d, g/ recognition task on speech from six speakers (four male, two female). Here a /d/ token from speaker MHT is input. The Meta-Pi structure gates a mixture of classification decisions from three male speakers' subnetworks to produce the correct overall classification decision. A Bayesian analysis of the Meta-Pi architecture indicates that the classifier performs robust multispeaker phoneme discrimination (98.4%) by learning to identify and separate relevant speaker types associated with particular signal patterns. This result approaches the mean speaker-dependent rate of 98.7% and significantly improves the discrimination rate of 95.9% obtained for a single TDNN trained on all six speakers' training data. This unconstrained form of rapid speaker adaptation could be extensible to speaker independence, where speaker-specific subnetworks could be replaced by nets responsible for groups of speakers or where new prototypical speaker (group) specific nets are incrementally added only if this should help to improve performance.

Many other improvements in phoneme level recognition, particularly in view of speaker independence, have been attempted, including the use of improved objective functions (Hampshire and Waibel, 1990), phonetic features motivated by speech knowledge (Bengio et al., 1989), and different input representations (Kamm, 1989).

3. THE WORD LEVEL

Although good results have been achieved using neural networks for phonemic patterns, the question remains whether this technology can be used effectively for word recognition as well. An early set of experiments simply extended the classification capabilities of these networks by applying an entire word's coefficient matrix to the inputs of static full-word networks with output units for each word to be classified. Good results were achieved, but time alignment and word end point detection are problems that limit this approach. Similarly limiting is the fact that only small vocabularies can be handled in this fashion, because network size and training time become prohibitively large with increasing vocabulary size. To overcome the former limitations, networks that model time alignment and/or shift invariance internally have been developed for small-vocabulary recognition. For large-vocabulary recognition, subword units such as phonemes or syllables must be employed. A number of novel techniques are emerging that attempt integration of connectionist subword models into words and sentences. A majority of them could be characterized as hybrid techniques, that is, techniques that seek to combine the perceived strengths of neural networks at the pattern recognition level with the strengths at modeling stochastic sequences of conventional methods such as hidden Markov models, Viterbi decoding, or dynamic programming.

3.1. Static Full-Word Models

Among the early static full-word models, Lippmann and Gold studied a number of back-propagation network architectures applied to the task of isolated digit recognition (Lippmann and Gold, 1987). Seven isolated monosyllabic digits were obtained from the TI Isolated Word Database representing speech from 16 different speakers. The speech data were sampled at 12 kHz, windowed, and a discrete Fourier transform performed; preprocessing produced 15-coefficient Mel-scale spectra at 10-ms intervals. These spectra were used to develop two 11-point cepstra offset by 30 ms in time; the cepstrum was taken from the maximum acoustic energy segment of each digit. These cepstra served as input to a series of networks all having 22 input layer nodes and 7 output layer nodes (corresponding to the 7 digits). Seventy training and 112 testing tokens were obtained for each speaker, and networks were trained and tested for single speakers only. A three-layer network yielded the best connectionist recognition performance of 92.3%, averaged over all 16 speakers.

Peeling and Moore (1987) also ran experiments with isolated digit recognition. They used a three-layer network with 50 hidden-layer nodes. Sixty 19-coefficient spectra taken at 20-ms intervals were used as input in order to capture the longest duration utterances. Shorter utterances were zero padded and time shifted randomly in the network input "window." Isolated digit speech data were taken from the 40-speaker Royal Speech and Radar Establishment (RSRE) data base. Speaker-dependent recognition under these conditions was 99.7%.

Burr (1988a) conducted a series of experiments in isolated E-set and polysyllabic word recognition using a single-layer perceptron. The network input comprised twenty 64-coefficient spectra; in separate experiments these spectra were computed using smoothed DFT and linear predictive coding (LPC) techniques. Input tokens were temporally aligned in the spectral "window" using a DP time alignment procedure. Five tokens of 20 polysyllabic words containing three to five syllables were recorded from a single male speaker. Training tokens were also used as testing tokens in this experiment—under these conditions, recognition rates were, not surprisingly, nearly 100%. Burr also ran recognition experiments on single-syllable words recorded from a single male speaker. Twenty tokens of each of the nine single-syllable E-set

words were obtained. Half of the tokens were reserved for training and half for testing. Recognition accuracy under these conditions was 91.4%. Word recognition was increased to 98.2% following modifications to the network's input layer structure and spectral estimation methods; these modifications focused network activity on the first 40% of each word.

3.2. Dynamic Full-Word Models

Word recognition of static classifiers is sensitive to time alignment and needs to rely on end point detection, as each connection represents a specific portion of an utterance. During fast and slow speech the relative position of acoustic features will generally be distorted, and a static net would only be able to blur its internal representation to compensate for such distortions. Word end points cannot always be determined reliably (particularly in noise and in continuous speech). Several connectionist models described here seek to overcome some of these problems.

3.2.1. A Word Level TDNN

Bottou (1988) used a large TDNN for recognition of small vocabularies and a novel time-warping approach to increase the temporal variance of isolated words and achieve shift-invariant speaker-independent word recognition on five consonant-vowel French words (Bottou et al., 1989). Single exemplars of each word were obtained from six speakers. Speech from four speakers was used for training and speech from the remaining two speakers constituted testing data. The data were sampled at 10 kHz and used to compute 256-point DFTs at 12.8-ms intervals. These spectra were reduced to 16 spectral mel-scale coefficients covering a frequency range of 100 Hz to 5 kHz. These formed the input to a 65-time-frame TDNN input layer. The TDNN architecture was in principle similar to the one described above except that the higher-layer units undersampled (skipped every other) activations from lower-layer outputs for greater efficiency. Bottou took the original 20-token training set and created a total of 400 additional training tokens by time warping the original set independent of phonetic structure. The extent of warping ranged from warping 80% of the word into 50% of the TDNN input spectra to warping 50% of the word into 80% of the input spectra. Occasionally, warping was so extreme that it eliminated consonant portions of words. The TDNN was trained on the original 20 tokens, plus these 400 "synthesized" versions. After training, Bottou achieved 100% recognition on all 20 original

training tokens and 94% recognition on the 400 warped tokens.* The recognition rate on test data was 90% using this technique of artificially expanding the training set by means of temporal warping. In a separate experiment involving word recognition on the TI 20-word data base, Kammerer and Kupper (1988) realized a 30% reduction in the number of classification errors on test data by using a similar time-warping technique to artificially increase the size and variance of their training token set. Their recognition results were 99.6% for speaker-dependent experiments and 97.3% for a speaker-independent trial.

3.2.2. Tank–Hopfield Time Delays

Tank and Hopfield (1987) developed an analog neural network model for recognizing particular stimulus sequences (comprising letters of a word) that were slightly distorted and embedded in larger sequences. The network employed a series of detectors $D_A \ldots D_X$ for single elements of a stimulus sequence; each of these detectors was replicated over a series of time delays, allowing the network to detect a single element of the sequence of interest across a range of time segments $f_1(t) \ldots f_k(t)$. Appropriate combinations of these time-shifted detectors fed a recognition unit V corresponding to the precise sequence to be detected. Inhibitory connections between recognition units minimized network output for stimulus sequences not closely matching the desired sequence. The network was very effective in locating distorted letter sequences embedded in larger sequences. In follow-on work, Unikrishnan et al. (1988) used this same network paradigm to achieve a 99.3% recognition rate on random sequences of digits.

3.2.3. The Dynamic Neural Net

Sakoe (1987) and Sakoe et al. (1989) developed a dynamic programming neural network (DNN) for speaker-independent word recognition. This network employed a three-layer back-propagation architecture capable of dynamically warping its input. The input layer consisted of a series of 10-coefficient Mel-scale spectra taken at 16-ms intervals. These spectra were linked in groups of two frames to single groups of four hidden units; each hidden unit group represented a temporal shift from its predecessor. All hidden layer unit groups were fully connected to a decision output unit corresponding to one of ten spoken digits. Speech from 50 speakers was used to train the networks in

*The relatively low rate for the warped training set was due to the extreme warping performed on a small number of those tokens.

two ways. In a temporally prewarped training method called "fixed time alignment," all training tokens for a particular word were time warped to a standard pattern prior to training. In an alternative training procedure called "adaptive time alignment," each token of a word was interactively warped in order to produce the maximum output activation of the network. Once the adaptive alignment was complete, the back-propagation iteration for that token was performed. Recognition performance was tested on tokens obtained from 57 speakers (none of whom were used for training). Recognition rates were 97.5% for networks trained with the fixed time alignment procedure and 99.3% for networks trained using the adaptive time alignment procedure. The added computational cost of the recognition improvement afforded by the adaptive time alignment training procedure was substantial.

3.2.4. *Predictive Neural Nets*

Most connectionist models that we have discussed so far apply neural nets as classifiers of either word patterns or subpatterns. For classification, the input usually consists of a coefficient matrix and the output approximates a bit pattern representing the classification results. In addition to learning discrete classifications, however, neural networks can also learn nonlinear mapping functions between real-valued inputs and outputs. This can be exploited in speech for various signal mapping and coding applications, including noise suppression (Tamura and Waibel, 1988) and nonlinear predictive coding (Lapedes and Farber, 1987). The neural networks have been used successfully in the neural prediction model proposed by Iso and Watanabe (1990) and the hidden control neural network proposed by Levin (1990). Both of these models have so far been limited to small-vocabulary recognition tasks (i.e., digits), but they appear to yield high speaker-independent recognition performance. Extensions are also possible, as we shall see later.

The basic idea is illustrated in Fig. 8. A two-frame window of input coefficients is input into a multilayer feedforward net trained to produce at its output a frame of coefficients that is as close as possible to the next (future) speech frame. The distance between this predicted frame and the actual next speech can be measured as a prediction error or distortion, and this distortion is used as an error criterion for back-propagation training. Given a set of predictor networks, one can imagine training each predictor for a separate region of an utterance. Each predictor net becomes specialized to best predict this portion of an utterance, so that the prediction error is likely to be lowest in these regions. A word is then represented by the *sequence* of predictor nets that best predicts the actual observed speech. Dynamic programming is used as a mechanism to apply each predictor sequentially over time to best ap-

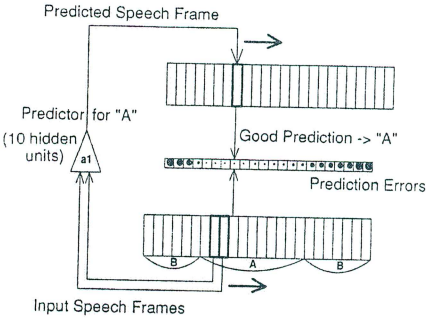

Figure 8 Modeling a phoneme by signal prediction.

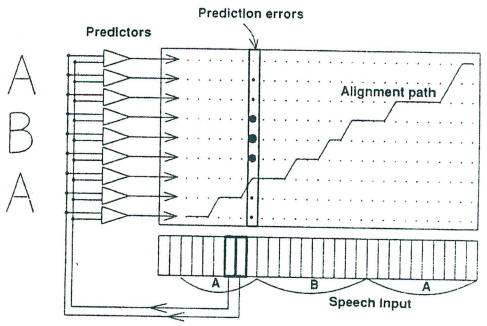

Figure 9 A neural prediction model.

proximate the actual signal. Figure 9 shows this alignment step based on the matrix of distances between actual speech frames and predicted frames. During training an alignment path is determined by dynamic programming. Each predictor is then trained to minimize the error between its output and the speech frames that it was assigned to predict by the DP alignment path. During recognition the word whose sequence of predictors minimizes the error between predicted frames and actual signal frames is chosen. Iso and Watanabe (1990) used 10-mel-scale cepstral coefficients and amplitude change as inputs to their networks. The number of predictors used depended on the utterance and ranged (for Japanese digits) between 9 and 14. Each predictor net has three layers, an input layer of two 11-coefficient frames, 9 hidden units, and 11 predicted output coefficients. Excellent performance (0.2% error) was reported for a Japanese speaker-independent isolated digit recognition task uttered over telephone lines. This result compared favorably with other techniques—0.7% for the DNN (Sakoe, 1987; Sakoe et al., 1989) and 1.1% for DP matching (Sakoe and Chiba, 1978) tested on the same data.

The model proposed by Levin is similar to the one described above and is illustrated in Fig. 10. As before, it uses nonlinear prediction by neural nets to measure a model's fit to the input data. Unlike the neural prediction model, however, it uses only one single predictor for an entire word and a sequence of varying input flags or "control units" that switch the predictor into alternate

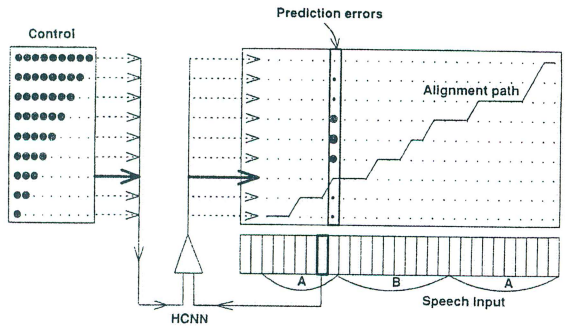

Figure 10 The hidden control neural network.

modes of operation as time progresses. Similar to "counter nodes"* (Kukich, 1988), these units are used to control the sequential state of the network. The predictor network used by Levin had 24 speech inputs (12 cepstral and 12 deltacepstal parameters), 30 hidden units, 24 predicted outputs, and 8 control input units. The control units turn on sequentially when appropriate and remain on as additional bits are activated ("thermometer" representation). The correspondence between an input speech frame and a control transition (when a new bit is turned on) is determined by Viterbi alignment. During training, the Viterbi algorithm determines the state of control unit settings for each speech input frame given a trained predictor net. Based on this state, then, the prediction error produced by the predictor net is reduced by back-propagation learning, given each frame of input speech and its corresponding control unit setting. The network was tested on connected digits from the TI-digit data base (using male speakers only). Using independent test data but from the same speakers used in training, a word recognition rate of 99.3% was achieved.

3.3. Dynamic Large-Vocabulary Models

Given the encouraging results for the connectionist models reviewed in the previous section, we would now like to explore possible extensions to large-vocabulary recognition. The most significant difference here is that large-vocabulary word recognition models must rely on units smaller than the word (such as syllables or phonemes) to decompose the large number of words into a limited set of atomic subunits that can be trained and optimized for use in (ideally) any vocabulary. How can connectionist models contribute usefully to this problem?

Most popular at present are so-called hybrid models, which seek to combine the perceived strengths of connectionist models with those of more classical recognition techniques such as dynamic programming or hidden Markov modeling. In this approach, connectionist models are viewed as high-performance nonlinear classifiers or predictors that could replace distance metrics, or vector quantization steps commonly found at the front end of most typical recognizers. Dynamic programming, Viterbi alignment, and/or hidden Markov models are then viewed as mechanisms for providing the additional constraints that phonemes must be of acceptable order, duration, and likelihood to produce a legal word.

*Proposed to control state information in networks used for spelling correction.

3.3.1. *TDNN-LR-DP*

Based on the Japanese large-vocabulary isolated word data base described before (Sagisaka et al., 1987; Waibel et al., 1987, 1989a), a number of experiments were carried out to extend TDNNs to large-vocabulary recognition. In this approach a TDNN is trained as before to classify input speech into one of several phoneme output categories over running speech (in this case entire words spoken in isolation). As the original TDNNs were trained on excised phoneme tokens only, several enhancements were introduced. First, the original excised phoneme training patterns were artificially misaligned in time by various offsets. This was particularly effective for phoneme *spotting* as opposed to discrimination, as it enforced shift-invariant phoneme classification even in transitory regions between phonemes. In doing so, the phoneme spotting rate was improved from 95.8 to 98.0% and, more important, the false alarm rate* decreased from 62.2 to 23.2%.† For word recognition a silence category was necessary, which was added by modular design to the existing net (Miyatake et al., 1990).

A total of 24 phonemes (5 vowels, 18 consonants, and silence) were spotted in this fashion by TDNNs shifted across time providing the front end for phoneme-based word recognition (Fig. 11 shows typical phoneme output unit firings over time). To do so, the output categories were considered to be output probability estimates of the likelihood of each phoneme occurring in a particular position in time. An LR parser provided top-down predictions of the set of phonemes that are legal under a given dictionary. The likelihood of a given phoneme predicted by the LR parser is evaluated at the outputs of the TDNNs at each time frame. Dynamic time warning (DTW) is then applied to find an optimal alignment between TDNN outputs and the predicted phoneme states. For duration control, each phoneme state was expanded to the average number of frames of that phoneme before DTW was carried out. Recognition experiments on various vocabularies were carried out with this system. All experiments were independent of vocabulary‡ and were performed on independent test data (phonemes not used for training). For a 500-word test vocabulary, first choice accuracy of 98% was achieved. For a large vocabulary of 5000 words, recognition rates as high as 92.6% were obtained. Second and fifth choice rates for the latter vocabulary were 97.6 and

*Presumably due to previously undefined transitory regions.
†All recognition tests were run on independent test data from the same speaker.
‡The phonemes used for training were extracted from words of a different vocabulary than the one used for testing.

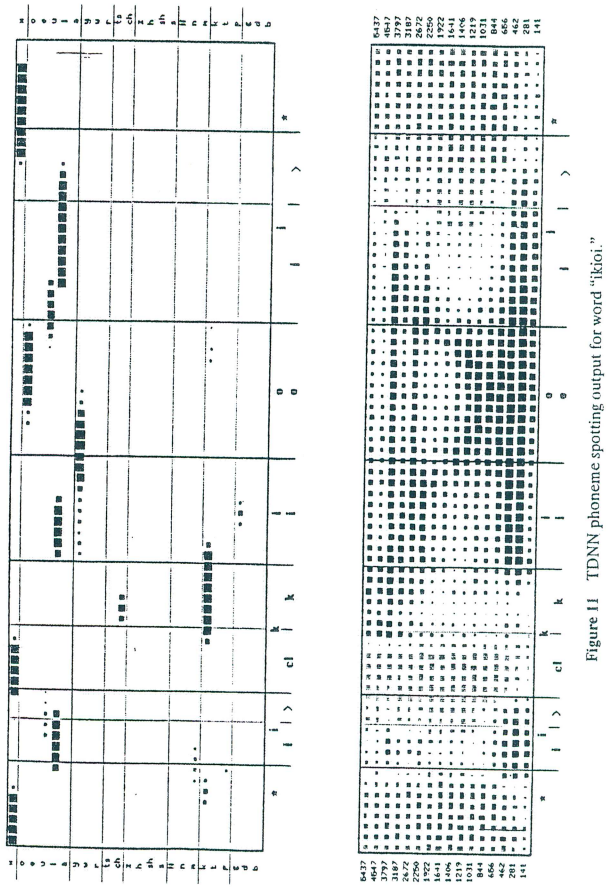

Figure 11 TDNN phoneme spotting output for word "ikioi."

99.1%, respectively, indicating that most confusions occurred among a small group of acoustically similar words (e.g., "itai" → "ittai").

3.3.2. *Neural Nets and Hidden Markov Models*

Some of the earliest attempts to combine the strengths of neural net classifiers with traditional word modeling techniques were developed by Bourlard and Wellekens (1988, 1989) and Morgan and Bourlard (1990). In theoretical and experimental work they have shown that the outputs of a multilayer perceptron (feedforward network) trained by back propagation of a mean square error may be considered to be estimates of the maximum a posteriori probabilities of a corresponding class. They have since built on this notion to construct hidden Markov model chains where the output activations of a local multilayer perceptron (MLP) are used as output probabilities for the states in a traditional HMM. Viterbi alignment is performed to assign the framewise MLP firings to corresponding states and to compute an overall word output probability. Not unlike HMM systems, an iterative optimization procedure was then introduced [Bourlard 90] that combines Viterbi alignment with back-propagation learning. This procedure performs a forward pass on the MLPs over time and assigns these outputs to their respective best matching states by way of Viterbi alignment. Back propagation of errors is then performed to improve the outputs of the MLPs given this assignment and the procedure iterates. In many of their experiments the MLP consisted of a three-layer feedforward network with nine binary 132-bit input vectors encoding the input spectrum over nine frames, and 50 outputs representing each phoneme. A variable number of hidden units was used. Several additional modifications have been added to achieve good performance (Morgan and Bourlard, 1990). First, the output a posteriori probability estimates produced by the MLPs were normalized by their respective prior probabilities to eliminate a bias due to uneven distributions in number of tokens per phoneme. Second, extra word transition penalties were introduced to reduce insertion errors. Third, a cross-validation learning scheme was applied that improved generalization considerably. Here the performance of the trained networks was checked against a cross-validation "test" set during each iteration to determine at what point test performance started to degrade (i.e., overfitting to the training data). Training was then halted. In speaker-dependent German large-vocabulary (918 words) continuous speech recognition experiments, word recognition performance currently approaches 60%. In the absence of a language model, this compared favorably with an HMM evaluated over the same data under similar conditions (Morgan and Bourlard, 1990).

Franzini et al. (1989) have investigated two similar methods for combining neural nets with hidden Markov models to perform continuous speech recognition. These methods were applied to speaker-independent recognition of continuous digit strings. In the first method, a network is trained to identify the word and phone to which a given frame of speech belongs. A four-layer recurrent network is used with LPC cepstrum coefficients over 70 ms of speech as input and one output unit corresponding to each word and each phone. In order to generate word and phone labels to use in training the network, a conventional discrete HMM is trained on the task, and a forced Viterbi alignment is performed with the HMM. The network is then trained on the labels generated by the alignment. During recognition, the network's input window is shifted, frame-by-frame, across an entire sentence, generating a vector of outputs for each input frame. Subsequently, an HMM is used to combine these output values into a final sentence hypothesis. The output values are treated as output probabilities associated with transitions in the HMM, and the recognition is performed as in conventional HMM systems, using a Viterbi search.

The second method (Franzini et al., 1990) also used a four-layer neural net with HMM postprocessing, but optimization of the two components is more tightly coupled. This method, connectionist Viterbi training (CVT), is a variant of the Viterbi training (or segmental k-means) method for training HMMs. As before, a discrete HMM is trained and an initial force alignment performed using this HMM. The network is then iteratively optimized by repeated application of back-propagation training, Viterbi alignment, and reestimation of transition probabilities. A cross-validation set was also used as a halting criterion. A second important difference from the previous method is that the network's output nodes model output probabilities corresponding to states in an HMM phone model. Such phone models can therefore be used to construct different words, and the system is extensible to large-vocabulary recognition. Both of the methods described, although still being investigated, have produced good results. The first method achieved 97% word accuracy on the Texas Instruments continuous digits data base, and the CVT procedure has reached 99% word accuracy on the TI digits.

3.3.3. *Linked Predictive Neural Networks*

Another word level model is an extension of work discussed in the previous section for small-vocabulary recognition, i.e., the use of neural nets as nonlinear predictors of speech. For use in large-vocabulary recognition, words must here again be decomposed into subword units such as phonemes or syllables and an optimal model for these units must be trained. Work by Tebelskis and

Waibel (1990) has demonstrated that this can be done without the need for segmentation by jointly optimizing time alignment and connection weights and by linking the weights (as in Waibel et al., 1989a) of sets of network predictors corresponding to the same phoneme symbols. Experiments with the linked predictive neural network (LPNN) resulted in 94% recognition performance for speaker-dependent isolated word recognition over a data base of 234 confusable Japanese words and 90% over a confusable 1000-word vocabulary.*

The operation and training of the LPNN are shown in Fig. 12. As in the neural prediction model, a set of predictors are assigned to different portions of a word. Here these portions are defined to be phonemes and each occurrence of the same phoneme is modeled by the same set of predictors. In Fig. 12, for example, two words "BAB" and "ABA" may consist of the same phonemes in different order and position. Time alignment over the sequence of predictors is done as before, but all prediction errors assigned to the same phoneme (or portion thereof) train the same predictor net by way of a linkage pattern that defines the legal phoneme sequence of a word. A number of enhancements of this basic scheme have been found effective. A set of parallel predictors was added to each phoneme model to allow the LPNN to better represent alternate pronunciations and context dependences. An assignment of each alternate was not predetermined, but the system decided itself which alternate to use based on the prediction errors produced by each alternate. Significant improvements were also obtained when phoneme pairs that are distinguishable only on the basis of duration (e.g., in Japanese: "k" vs. "kk") were represented by different sets of predictors. Figure 13 shows an example of processing in the LPNN. Here the prediction errors obtained by each phoneme state predictor during the word "shikisai" and the corresponding alignment path are shown.

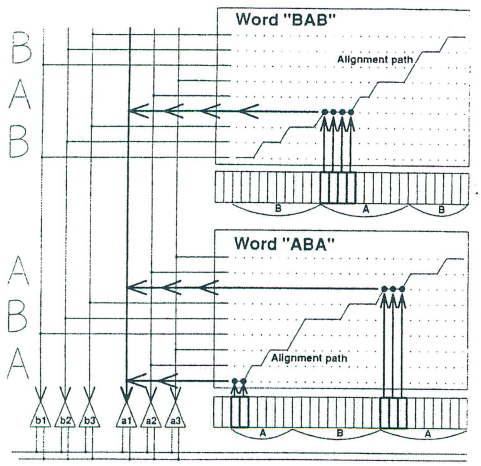

Figure 12 LPNN training.

*The data in this evaluation were a confusable subset from the Japanese large-vocabulary isolated word data base used in other experiments discussed above.

3.4. Connectionist Word Level Enhancements

In addition to the connectionist word models described so far, other connectionist solutions to the word recognition problem and enhancements have been reported. These include connectionist alignment strategies and connectionist postprocessors (to enhance discrimination performance at the word level). Further advances are likely to emerge from theoretical work (in progress) aiming at finding unified formulations for connectionist and stochastic models of speech (Bridle, 1990; Young, 1990).

3.4.1. Viterbi Net

A connectionist solution to the problem of alignment and sequential control was proposed by Lippmann and Gold (1989) and is called the Viterbi net. Figure 14 shows this network. The triangular nodes of the network corresponded to single nodes in an HMM word model; each of these nodes performed a thresholding and time delay function. Input layer nodes accepted mel and differential mel cepstra updated at 10-ms intervals. Connection strengths between input and HMM nodes were set to values obtained by conventional HMMs. The small subnetworks feeding input to the HMM nodes were used to select the maximum of two competing inputs. This network achieved a 99.4% word recognition rate—virtually identical to that achieved by nonconnectionist HMM recognizers. The network might, however, be of

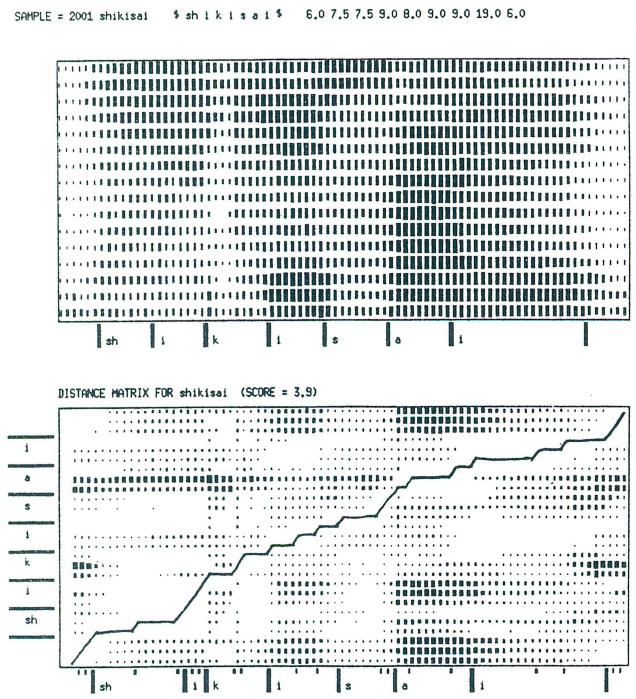

Figure 13 Prediction errors for different phoneme predictors during the word "shikisai."

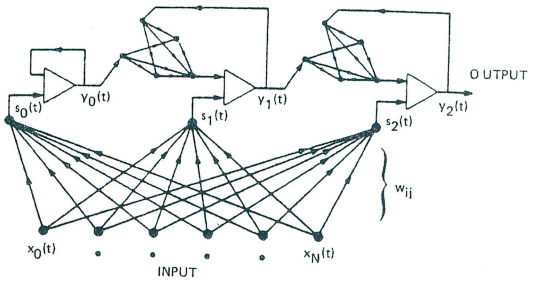

Figure 14 Lippmann and Gold's Viterbi net.

interest for hardware implementations as well as provide a basis for trainable connectionist extensions of Viterbi decoding.

3.4.2. Connectionist Postprocessing

Another enhancement employed successfully by several investigators is to use connectionist models as postprocessors. In this approach recognition is first carried out by DTW or HMMs. A connectionist classifier is then applied to discriminate between potentially confusable pairs before a final recognition decision is made. The study previously reported by Burr (1988b) is one example. Prealignment was done by DTW before a connectionist classifier was applied. In a similar vein, (sometimes considerable) improvements were achieved by connectionist postprocessors (multilayer perceptrons or LVQs) following a hidden Markov model-based recognition pass (Huang et al., 1988; Katagiri and Lee, 1990).

4. THE LANGUAGE LEVEL

Beyond recognition of words, connectionist models have also been applied to language models and natural language processing. The attempts are driven by the desire to develop more robust and perhaps cognitively plausible models of language. Indeed, some recent work that we review here suggests productive uses for the development of spoken language systems. Of interest in this

regard are the possibility of representing complex syntactic and semantic relationships stochastically and the hope to learn them automatically from text (or speech). Moreover, the uniformity of connectionist processing may allow for complex codes that include nonsyntactic information that may be difficult to incorporate by traditional means (e.g., pragmatic or prosodic information).

4.1. Word Prediction, Coding, and Disambiguation

A very direct approach to language modeling which has been explored is connectionist N-gram modeling. Straightforward statistical approaches to N-gram models become intractable quickly as N grows large. The number of parameters to estimate grows exponentially, and the requirements for sample data grow excessive. Nakamura and Shikano (1989) have proposed the NETgram, a connectionist network architecture which learns N-gram models efficiently. The basic bigram network had a localist input-output representation for word category, and the network was trained to predict the category of the next word given the current word by using error back propagation. A NETgram architecture for a particular value of N words is constructed by augmenting a trained NETgram for $N-1$ words. In tests on the Brown corpus, the architecture had comparable performance to the traditional statistical method, but the number of parameters in a NETgram does not increase exponentially with N.

Rather then predicting the next word given a sequence of previous words (or word categories), networks have also been used as autoassociators or disambiguators to learn suitable codes for language. An example of the former is PARSNIP (Hanson and Kegl, 1987), a three-layer feedforward back-propagation network that was trained to reproduce an input word category sequence at its outputs by way of a set of hidden units. The units develop during training a code for language that can usefully incorporate information that extends beyond the first one or two previous words. An extension of this idea that could perhaps be applied more directly to speech was proposed by Benello et al. (1989). Here a multilayer feedforward net was trained to produce an unambiguous word category as output, given an input that consisted of current ambiguous word categories and several unambiguous (known) preceding word categories. After training, the network could correctly disambiguate 95% of the words in previously unanalyzed text.

4.2. Recurrent Networks

Recursive network architectures have also been shown to learn fairly difficult syntactic relationships. The explicit purpose for their recurrent connections

was to provide a network with state sequence information. Early connectionist research focusing on the design and training of recurrent networks to do this was done by Jordan (1986) (see Fig. 15), and his work spawned several other papers on the subject. Various forms of recurrence were used to achieve temporal sequencing and shift invariance for time-varying signals including robotic control, speech production, and phoneme and word recognition, but several applications to modeling syntax now also exist. Elman and Zipser (1988) developed a three-layer network with "context" units that formed a feedback mechanism between the hidden and input layers of the network. Using this structure (very similar to that illustrated in Fig. 16), they ran a series of experiments to assess the network's ability to represent temporally sequential relationships in the input data. Network performance was judged on its ability to predict future input states, given present input state and former internal (hidden) state. In effect, the network was tasked with learning discrete state-space trajectories. They successfully trained the network to predict follow-on states for a set of three discrete trajectories in one experiment. In a more complex task they trained a similar network with 200 variable-length sentences generated from a 15-word lexicon. The training was conducted with the objective of correctly predicting the next letter of the sequence representing a given word in the lexicon. The trained network performed the task consistently; prediction errors were typically high for the first letter of each word and dropped rapidly (indicating high-confidence predictions) as the letter stream corresponding to the word was processed.

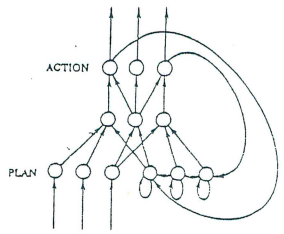

Figure 15 Recurrent network used by Jordan (1986) to generate an unfolding output sequence given a static input plan.

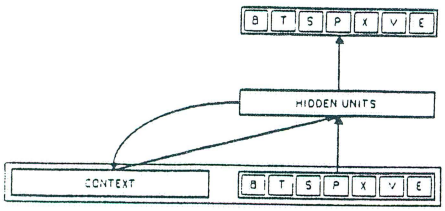

Figure 16 The recurrent network form used by Elman and Servan-Schreiber in their word recognition experiments.

Servan-Schreiber et al. (1988a,b) expanded on Elman's work using the same recurrent connection paradigm. They trained a recurrent network with 200,000 strings of varying length drawn from a finite-state grammar. After training, the network was tested with 20,000 strings drawn randomly from the 200,000-string training set. Since substrings of different full strings could be identical—thereby leading to different predictions for next state—performance measures accounted for multiple predictions of follow-on states. Under these criteria, the network predicted next states flawlessly for all 20,000 "test" strings. When tested with 130,000 strings, only 0.2% of which were consistent with the finite-state grammar, the network rejected all 99.8% nongrammatical strings while it correctly processed all grammatical strings.

The results of Elman and Zipser and Servan-Schreiber et al. illustrated the effectiveness of capturing temporal context with representations of sequential state. Extensions to continuous outputs have also been proposed for control systems application (Pearlmutter, 1988). These networks follow continuous state space trajectories, in contrast to sequences of discrete states, and might be useful for speech applications as well.

4.3. Parsing

Another approach to the problem of temporal context has been taken by Jain (1989) and Jain and Waibel (1990a) in work on connectionist parsing. Instead of requiring networks to learn to capture arbitrarily complex and distant temporal context information through recurrence, the process of capturing context is explicitly built into the task to be learned. This work extends the standard back-propagation paradigm to allow the construction of well-behaved storage buffers within networks which operate on sequential input. The parsing networks are constructed in a hierarchical fashion from separately trained modules. Each module performs some transformation of either input data or substructures built by other modules. Arbitrarily complex information structures can be built up over time in this manner. Jain and Waibel have successfully trained a network to parse grammatically complex sentences including passive constructions and center embedded clauses. The dynamic parsing behavior is predictive; the trained network produces hypotheses about sentence structure at every moment in time and confirms or revises hypotheses as input words are processed. In Fig. 17, the temporal activation patterns corresponding to the semantic role description for a single clause unit are plotted for a passive sentence. The initial hypothesis of

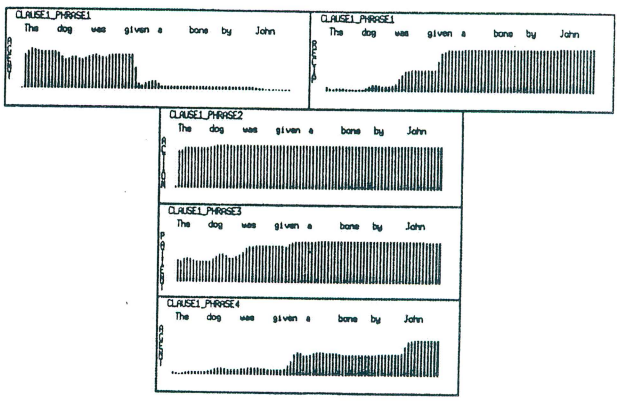

Figure 17 Dynamic role assignment behavior for "The dog was given a bone by John." The phrase numbers correspond to the following phrases: "[The dog] [was given] [a bone] [by John]." The network begins with an agent/action/patient assignment and finishes with the correct recipient/action/patient/agent assignment. All roles are predicted before their respective phrases are processed.

the common agent/action/patient role structure is quickly revised when the passive construction is detected.

The parser has the advantage that its structure is learned automatically from text data and not programmed. This should allow extensions to parsers that compensate for word recognizer confusions and syntactically ill-formed spontaneous speech. In preliminary evaluations the present version of the parser was found to be tolerant of ungrammatical sentences and various other types of degradations (Jain and Waibel, 1990a,b). A trainable parser may also easily incorporate other nonsymbolic information, such as prosodic cues, intonation, stress, intensity, and rhythm, that have so far been ignored in most language models of speech.

5. CONCLUSION

In this chapter we have provided a review of recent research on applying neural networks to speech recognition. At this writing, research in the field is rapidly expanding and new models for phoneme, word, and language modeling continue to emerge. This creative search for novel solutions to the speech recognition problem is likely to stimulate new insights and intuitions beyond the connectionist approach that should lead to a better understanding of speech recognition system design.

ACKNOWLEDGMENT

The author gratefully acknowledges his collaborators, Mike Franzini, John Hampshire, Ajay Jain, and Joe Tebelskis, for numerous discussions and help in preparing this chapter.

REFERENCES

Benello, J., Mackie, A. W., and Anderson, J. A. (1989). Syntactic category disambiguation with neural networks. *Comput. Speech Language 3*:203–217.

Bengio, Y., Cardin, R., Cosi, P., and DeMori, R. (1989). Speech coding with multilayer networks. *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, May, pp. 164–167.

Bottou, L.-Y. (1988). Reconnaissance de la Parole par Reseaux multi-couches. *Proc. Neuro-Nimes 88*, November.

Bottou, L., Fogelman-Soulie, F., Blanchet, P., and Lienard, J. S. (1989). Experiments with time-delay networks and dynamic time warping for speaker independent isolated digits recognition. *Proc. Eurospeech*, September.

Bourlard, H., and Wellekens, C. J. (1989). Links between Markov models and multilayer perceptrons. In *Advances in Neural Network Information Processing Systems*. D. S. Touretzky (ed.) Morgan Kaufman, San Mateo, pp. 502–510.

Bourlard, H., and Wellekens, C. J. (1989). Speech pattern discrimination and multilayer perceptrons. *Comput. Speech Language 3*:1–19.

Bridle, J. S. (1990). Alpha-nets: A recurrent neural network architecture with a hidden Markov model interpretation. *Speech Commun.*, in press.

Burr, D. J. (1988a). Speech recognition experiments with perceptrons. In *Advances in Neural Information Processing Systems*. Morgan Kaufmann.

Burr, D. J. (1988b). Experiments on neural net recognition of spoken and written text. *IEEE Trans. Acoust. Speech Signal Process 36*:1162–1168.

Duda, R. O., and Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. Wiley, New York.

Elman, J. L. (1988). Finding structure in time. Technical Report CRL 8801, University of California, San Diego.

Elman, J. L., and Zipser, D. (1987). Learning the hidden structure of speech. Technical Report, University of California, San Diego, February.

Franzini, M., Witbrock, M., and Lee, K. F. (1989). A connectionist approach to continuous speech recognition. *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, May.

Franzini, M. A., Lee, K. F. and Waibel, A. H. (1990). Connectionist Viterbi training: A new hybrid method for continuous speech recognition. *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, April.

Haffner, P., Waibel, A., Sawai, H., and Shikano, K. (1988). Fast back-propagation learning methods for neural networks in speech. Technical Report TR-I-0058, ATR Interpreting Telephony Research Laboratories, November.

Haffner, P., Waibel, A., Sawai, H., and Shikano, K. (1989). Fast back-propagation learning methods for large phonemic neural networks. *Proc. Eurospeech*, September, pp. 553–556.

Hampshire, J., and Waibel, A. (1990). A novel objective function for improved phoneme recognition using time delay neural networks. *IEEE Trans. Neural Networks*, June, *1*:216–228.

Hanson, S. J., and Kegl, J. (1987). PARSNIP: A connectionist network that learns natural language grammar from exposure to natural language sentences. *Proc. Ninth Annual Conf. Cognitive Science Society*, pp. 106–119.

Hinton, G. E. (1989). Connectionist learning procedures. *Artif. Intell. 40*:185–234.

Hinton, G. E., and Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. In *Parallel Distributed Processing; Explorations in the Microstructure of Cognition*, J. L. McClelland and D. E. Rumelhart (Eds.). MIT Press, Cambridge, pp. 282–317.

Huang, W., and Lippmann, R. (1988). Neural net and traditional classifiers. In *Neural Information Processing Systems*, D. Anderson (Ed.). American Institute of Physics, New York, pp. 387–396.

Huang, W. M., Lippmann, R. P., and Nguyen, T. (1988). Neural nets for speech recognition. *Conf. Acoustical Society of America*, May.

Iso, K., and Watanabe, T. (1990). Speaker-independent word recognition using a neural prediction model. *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, April.

Iwamida, H., Katagiri, S., McDermott, E., and Tohkura, Y. (1990). A hybrid speech recognition system using HMMs with an LVQ-trained codebook. *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, April, p. 35.S10.1.

Jain, A. (1989). A connectionist architecture for sequential symbolic domains. Technical Report CMU-CS-89-187, Carnegie Mellon University, December.

Jain, A., and Waibel, A. (1990a). Robust connectionist parsing of spoken language. *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, April.

Jain, A., and Waibel, A. (1990b). Incremental parsing by modular recurrent connectionist networks. In *Advances in Neural Information Processing Systems*, D. S. Tovretzky (ed.) Morgan Kaufman, San Mateo, pp. 364–371.

Jordan, M. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. In *Proceedings of the 1986 Cognitive Science Conference*. L. Erlbaum, Hillsdale, N.J.

Kamm, C. A. (1989). Effects of input representation on speaker-independent digit recognition using neural networks. in *Proceedings of Speech Tech '89*, Media Dimensions, May, pp. 301–303.

Kammerer, B., and Kupper, W. (1988). Experiments for isolated word recognition with single and multi-layer perceptrons. In *Proceedings of the First International Conference of the Neural Network Society*. INNS, Boston, p. 302.

Katagiri, S., and Lee, C. H. (1990). A new HMM/LVQ hybrid algorithm for speech recognition. *Proc. GlobeCom '90*, November.

Kato, K., and Kakehi, K. (1987). Listener adaptability to individual speaker differences. *J. Acoust. Soc. Jpn.*, in press.

Kohonen, T. (1988). *Self-Organization and Associative Memory*, 2nd ed. Springer, Berlin.

Kohonen, T., Torkkola, K., Shozakai, M., Kangas, J. and Venta, O. (1988a). Phonetic typewriter for Finnish and Japanese. *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, April, pp. 607–610.

Kohonen, T., Makisara, K., and Saramaki, K. (1988b). Statistical pattern recognition with neural networks: Benchmarking studies. *IEEE Proc. 2nd Annual Int. Conf. Neural Networks*, July.

Kuhn, G., Watrous, R. L., and Ladendorf, B. (1989). Connected recognition with a recurrent network. *Proc. Neurospeech '89*, May.

Kukich, K. (1988). Back-propagation topologies for sequence generation. *IEEE Int. Conf. Neural Networks*, pp. 301–308.

Lang, K. (1989). A time delay neural network architecture for speech recognition. Ph.D. thesis, Carnegie Mellon University.

Lang, K. J., Hinton, G. E., and Waibel, A. H. (1990). A time-delay neural network architecture for isolated word recognition. *Neural Networks J. 3*:23–44.

Lapedes, A., and Farber, R. (1987). Nonlinear signal processing using neural networks; prediction and system modeling. Technical report LA-UR-87-2662, Los Alamos National Laboratory.

Leung, H. C. (1988). Some phonetic recognition experiments using artificial neural nets. *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, April.

Leung, H. C and Zue, V. W. (1989). Applications of error back-propagation to phonetic classification In Advances in Neural Information Processing Systems, D. S. Touretzky (ed.) Morgan Kaufman, San Mateo, pp. 206–214.

Levin, E. (1990). Speech recognition using hidden control neural network architecture. *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, April.

Lippmann, R. P. (1987). An introduction to computing with neural nets. *IEEE ASSP Mag.* April, pp. 4–22.

Lippmann, R. P., and Gold, B. (1987). Neural-net classifiers useful for speech recognition. *IEEE Int. Conf. Neural Networks*, June.

Makhoul, J. I., Roucos, S., and Gish, H. (1985). Vector quantization in speech coding. *IEEE Proc. 73*:1551–1588, November.

McDermott, E., and Katagiri, S. (1989). Shift-invariant, multi-category phoneme recognition using Kohonen's LVQ2. *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, May, p. 9.S3.1.

McDermott, E., Iwamida, H., Katagiri, S., and Tohkura, Y. (1990). Shift-tolerant LVQ and hybrid LVQ-HMM for phoneme recognition. In *Readings in Speech Recognition*, A. Waibel and K. F. Lee (eds.) Morgan Kaufmann, San Mateo, pp. 425–438.

Morgan, N., and Bourlard, H. (1990). Continuous speech recognition using multilayer perceptrons with hidden Markov models. *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 26.S8.1.

Nakamura, M., and Shikano, K. (1989). A study of English word category prediction based on neural networks. *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, May, p. 731–734.

Niranjan, M., and Fallside, F. (1988). Neural networks and radial basis functions in classifying static speech patterns. Technical report CUED/F-INFENG/TR-22, Cambridge University Engineering Department.

Pearlmutter, B. (1988). Learning state space trajectories in recurrent neural networks. Technical Report CMU-CS-88-191, Carnegie Mellon University, December.

Peeling, S., and Moore, R. (1987). Experiments in isolated digit recognition using the multi-layer perceptron. Technical Report 4073, Royal Speech and Radar Establishment (RSRE), December.

Peterson, G. and Barney, H. (1952). Control methods used in a study of vowels. *J. Acoust. Soc. Am. 24*:175–184, March.

Prager, R. W., Harrison, T. D., and Fallside, F. (1986). Boltzmann machines for speech recognition. *Comput. Speech Language 1*:3–27, March.

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

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature 323*:533–536.

Sagisaka, Y., Takeda, K., Katagiri, S., and Kuwabara, H. (1987). Japanese speech database with fine acoustic-phonetic transcriptions. Technical Report, ATR Interpreting Telephony Research Laboratories, May.

Sakoe, H. (1987). Dynamic neural network—A new speech recognition model based on dynamic programming and neural network. IEICE Technical Report, December.

Sakoe, H. and Chiba, A. (1978). Dynamic programming optimization for spoken word recognition. *IEEE Trans. Acoust. Speech Signal Process. ASSP-26*(1):43–49.

Sakoe, H., Isotani, R., Yoshida, K., Iso, K., and Watanabe, T. (1989). Speaker-independent word recognition using dynamic programming neural networks. *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, May., pp. 29–32.

Miyatake, M., Sawai, H., and Shikano, K. (1990). Integrated training for spotting Japanese phonemes using large phonemic time-delay neural networks. *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, May.

Servan-Schreiber, D., Cleeremans, A., and McClelland, J. L. (1988a). Learning sequential structure in simple recurrent networks. *Proc. Second IEEE Conf. Neural Information Processing Systems.*

Servan-Schreiber, D., Cleeremans, A., and McClelland, J. L. (1988b). Encoding sequential structure in simple recurrent networks. Technical Report CMU-CS-88-183, Carnegie Mellon University, November.

Tamura, S., and Waibel, A. (1988). Noise reduction using connectionist models. *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, April, p. S12.7.

Tank, D. W., and Hopfield, J. J. (1987). Neural computation by concentrating information in time. *Proc. Natl. Acad. Sci. USA* 1896–1900. April.

Tebelskis, J., and Waibel, A. (1990). Large vocabulary recognition using linked predictive neural networks. *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, April.

Unikrishnan, K., Hopfield, J., and Tank, D. (1988). Learning time delayed connections in a speech recognition circuit. Snowbird Conference, March.

Waibel, A. (1989). Consonant recognition by modular construction of large phonemic time-delay neural networks. In *Advances in Neural Network Information Processing Systems*. D. S. Touretzky (ed.) Morgan Kaufmann, San Mateo, pp. 215–223.

Waibel, A. (1989). Modular construction of time-delay neural networks for speech recognition. *Neural Computation 1*:39–46.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang K. (1987). Phoneme recognition using time-delay neural networks. Technical Report TR-1-0006, ATR Interpreting Telephony Research Laboratories, October.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang K. (1989a). Phoneme recognition using time-delay neural networks. *IEEE Trans. Acoust. Speech Signal Process. 37*:328–339, March.

Waibel, A., Sawai, H., and Shikano, K. (1989b). Modularity and scaling in large phonemic neural networks. *IEEE Trans. Acoust. Speech Signal Process. 37*: 1888–1898, December.

Watrous, R. (1988). Speech recognition using connectionist networks. Ph.D. thesis, University of Pennsylvania.

Young, S. J. (1990). Competitive training: A connectionist approach to the discriminative training of hidden Markov models. Technical Report CUED/F-INFENG/TR.41, Cambridge University, March.