4

NEURAL NETWORK APPLICATIONS TO SPEECH

by A.H. Waibel and J.B. Hampshire II,
School of Computer Science, Carnegie Mellon University

1 Overview

Research in the field of connectionist speech processing is moving at a tremendous pace: the advent of inexpensive supercomputing has provided the computational resources necessary for large scale neural network simulations in many disciplines. Those involving complex pattern classification tasks have sponsored some of the most vigorous connectionist research; speech processing — one of the most notable of such disciplines — has been particularly active in connectionist developments over the past three years. In this chapter we provide a summary of this research. We categorize neural network speech processing efforts in the following manner:

- Phoneme Recognition Networks
  - Temporally Static Networks
  - Temporally Dynamic Networks
- Extensions
  - Modularity and Scaling
  - CFM objective function for backpropagation
- Word Recognition Networks
  - Temporally Static Full-Word Networks
  - Temporally Dynamic Full-Word Networks
  - Hybrid Networks
- Networks with Other Applications to Speech Processing
  - Noise Suppression
  - Speech Coding
  - Text-to-speech Transcription

Many of the studies that we review proffer significant results for different levels of the speech recognition task. For this reason the reader will occasionally see the same work mentioned in a few sections of this chapter. Much of the research presented is based on the Backpropagation neural network model of Rumelhart, McClelland, and the PDP Research group — the reader interested in a detailed formulation of this paradigm should refer to [1,2]. Finally, our aim is to provide the reader with a general and representative overview of connectionist speech processing today. We cannot hope to cover all the important results within the bounds of a single chapter. We recommend the works by Richard Lippmann [3] and [4] as additional, detailed references to those seeking an alternative perspective. Throughout the chapter recognition rates apply to speaker-dependent experiments unless specifically stated.

2 Introduction

Human speech presents a formidable pattern classification task to recognition systems. Indeed speech recognition research has been active for more than three decades, yet the very best systems today have recognition capabilities well below those of a child. This is because the speech signal is extraordinarily complex. In very general terms, humans recognize speech by recognizing several types of cues — the predominant cues are acoustic, but there are many non-acoustic cues (e.g., visual and contextual) as well. Chief among the acoustic cues are the frequency content of the speech waveform, and the time-dependent changes in that frequency content. Thus, in its most simplistic form speech can be viewed as a stochastic process involving two principal dimensions — time and frequency. The complexity of the speech recognition task lies in the fact that a given utterance can be represented by an effectively infinite number of time-frequency patterns. A human speech signal is produced by moving the vocal-tract articulators towards target positions that characterize a particular sound. Since these articulatory motions are subject to physical constraints that vary from subject to subject and since they are stochastic in nature (i.e., the motions do not follow precisely the same trajectory each time they are performed) they do not produce consistently clean identifiable phonetic targets in the train of speech. Instead these articulations form acoustic-phonetic trajectories that have a high degree of variability in both the time and frequency domains. Effective recognition systems must therefore capture the dynamic "motion" of these acoustic-phonetic trajectories, scanning them for sequences and co-occurrences of cues necessary for robust recognition. Connectionist speech processing and recognition systems are well suited to this task because they are particularly effective at *learning and subsequently representing* the salient features of speech.

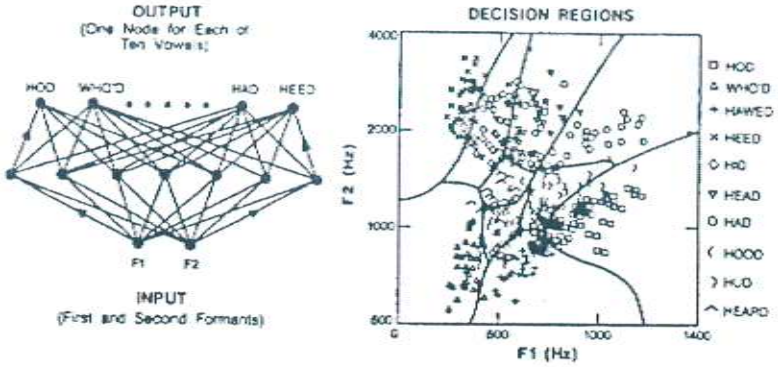

Figure 1: A 3-layer backpropagation network used to form classification boundaries on the formants F_1 and F_2 for vowels[1].

3 Phoneme recognition

In this section we review connectionist architectures that have been developed to recognize the acoustic-phonetic building blocks of speech: *phonemes*. These networks can be divided into two major groups: 1) those that require precise temporal alignment of input tokens for accurate recognition performance (making them temporally static, shift-variant classifiers) and 2) those that do not require precise temporal alignment of input tokens (making them temporally dynamic or shift-invariant classifiers).

3.1 Temporally static networks

Figure 1 serves as an illustrative introduction to the application of neural networks to speech recognition tasks. Huang and Lippmann [5] applied a 3-layer backpropagation network to the task of forming non-linear classification boundaries for the formants F_1 and F_2 using data obtained by Peterson and Barney [6] in studies of adult and child male and female subjects[2]. The network had 50 units in the hidden layer, and was trained for 50,000 trials, resulting in inter-formant boundaries comparable to those one would draw by hand and those formed by more traditional classification techniques such as k-nearest neighbor classifiers [8,9]. The network provides a particularly good example of the non-linear classification power of the neural network structure applied to a highly non-linear classification task.

Elman and Zipser performed phoneme classification experiments for the voiced-stop consonants /b, d, g/ (followed by the vowels /a, i, u/) [10]. 505 tokens of the nine discrete voiced-stop syllables were parsed from recordings of a single male

[1]Figure from Huang and Lippmann [5].

[2]Rabiner and Schafer [7] also provide a detailed analysis of this data.

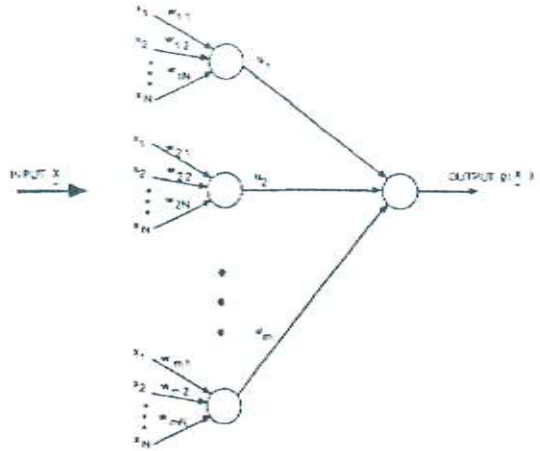

Figure 2: Niranjan and Fallside's RBF net[3].

speaker using a 10 kHz sampling rate applied to 3.5 kHz low-pass filtered speech. Twenty 16-coefficient DFTs were computed at overlapping 3.2 msec intervals to form the input of a 3-layer backpropagation 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 3 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. They drew the important conclusion that the noise source tended to obscure features of the training tokens that were "idiosyncratic" and not representative of all tokens for a given syllable or phoneme.

Niranjan and Fallside [11] employed a connectionist implementation of the Radial Basis Function (RBF) classifier [12] to the task of speaker-independent vowel recognition. The RBF network was based upon nodes called Spherically Graded Units [13]. It used a perceptron-like architecture employing SGUs at the input layer. Instead of performing the usual thresholding function, these input units $u_1 \ldots u_n$ computed RBFs which were fed to an output unit implementing the RBF interpolation function $g(X)$. The network is illustrated in Figure 2. In their experiment single utterances of 5 vowels were obtained from 20 speakers (10 male, 10 female). Speech data was 5 kHz low-pass filtered, sampled at 10 kHz, and used as input to a

[3]Figure from Niranjan and Fallside[11].

12th-order LPC spectral estimator using the autocorrelation method. Ten logarithmic spectral parameters were obtained from the LPC reflection coefficients, pre-processed through a sigmoid and used as input to a standard 3-layer backpropagation network as well as the RBF network. 70 of the 100 tokens were used for training and 30 were reserved for testing. The backpropagation network achieved an 80% (24/30) recognition rate, while the RBF network achieved a 93% (28/30) recognition rate.

3.2 Temporally dynamic networks

As mentioned in the introduction, the principal goal of an effective speech recognition system is to capture the dynamic nature of the acoustic-phonetic trajectory of the speech signal. The temporal aspect of this task is particularly challenging. Some speech recognition systems attempt to parse or segment speech into discrete units roughly corresponding to phonemes. However, the best segmentation schemes are highly susceptible to errors; these errors, in turn, result in higher error rates further along in the recognition process. As a result, a robust speech recognition system should simply scan the speech signal for useful cues *without* relying on pre-segmentation, basing its over-all recognition decision on the sequence and co-occurrence of a sufficient set of those cues. This, in turn, suggests a system that is temporally dynamic or "shift-invariant" (i.e., a system whose recognition performance is unaffected by temporal shifts of the input speech train). The experiments detailed above used utterances that were precisely parsed from the speech signal, obviating the need for shift-invariant performance in the network. The following series of experiments all employed techniques aimed at yielding shift-invariant phoneme recognition.

Waibel and colleagues [14,15] and Lang and Hinton [16] developed variants of the Time-Delay Neural Network (TDNN) — an architecture designed to perform high accuracy phoneme recognition under varying conditions of phoneme duration and temporal location within the speech signal. Figure 3 illustrates the TDNN architecture of [14] in block diagram form, and helps to explain the way in which it achieves shift-invariant recognition. The input layer of the network was fed by 15 16-point Melscale frequency spectra, representing the speech waveform sampled at 10 msec intervals. These input spectra were fully connected in groups of 3 to reduced abstract spectrum-like node structures in the first hidden layer. The connection strengths between the group of 3 input spectra and its first hidden layer counterpart were identical among all groups. Thus, the TDNN focused on 30 msec "windows" of speech, looking for the same features in each 30 msec window. The abstract spectra of the first hidden layer were bound in groups of 5 that mapped to corresponding phoneme classification node structures in the second hidden layer. Again, connection strengths were identical from group to group, so the network looked for the same abstract features across each 5 time-slice segment of abstracted speech in the first hidden layer. Finally, all second hidden layer classification nodes corresponding to the same phoneme were linked with equal connection strengths to a single phoneme

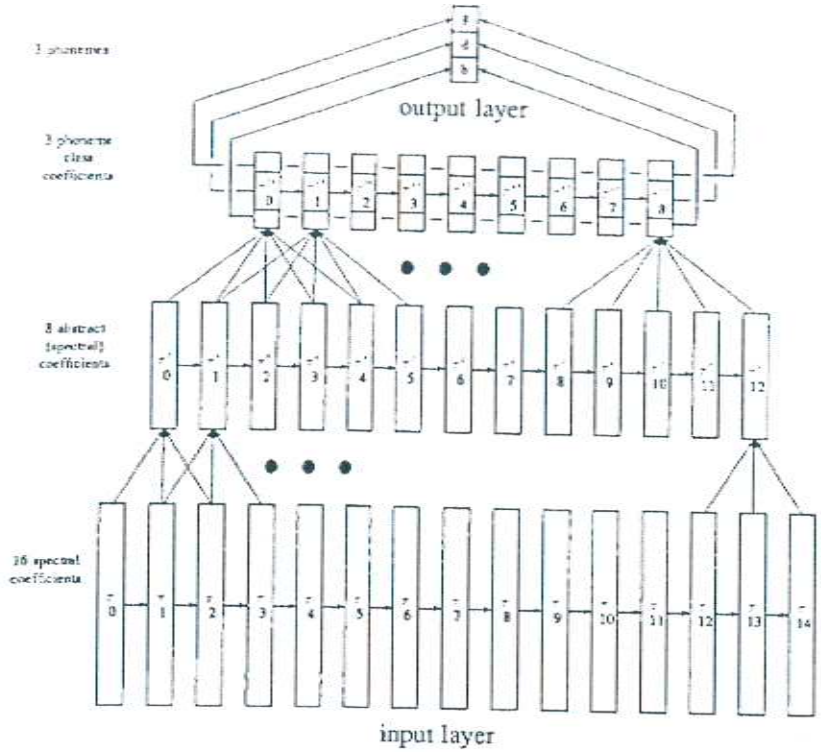

Figure 3: A Time-Delay Neural Network (TDNN) for the three voiced stop phonemes /b, d, g/.

classification node at the output layer. Figure 4 shows actual activation states for a TDNN trained to recognize the voiced stop consonants /b, d, g/. In the diagram, the background is white (indicating no activation), negative activation (input layer only) is depicted as grey, and positive activation is depicted as black. The level of activation for a given node is proportional to the size of its corresponding rectangle in the figure. The TDNN variants developed by Lang and Hinton [16] had slightly modified structures, but were conceptually identical to the model described above.

Waibel and colleagues performed the /b, d, g/ recognition task with the TDNN [14], using a large vocabulary database of 5240 Japanese words [17]. Approximately 200 training and 200 disjoint testing tokens were obtained for each voiced-stop

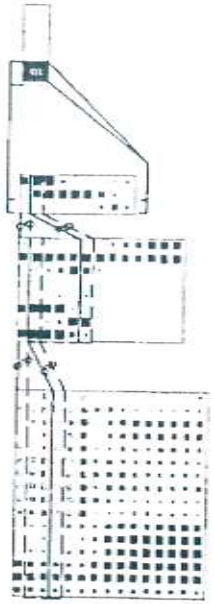

Figure 4: A Time-Delay Neural Network.

consonant from the 5240-word database produced by each of three male speakers. As a result, the tokens contained a high degree of phonetic variability. These tokens were parsed from the speech signals of entire words, and 150 msec spectral windows were centered about vowel onset. Recognition results on test data yielded an average recognition rate of 98.5% for all phonemes 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. Waibel and colleagues also found that the TDNN achieved significantly higher recognition rates than the best Hidden Markov Models currently used for the same task [15]. Using the same speech data, they found that HMMs achieved an average recognition rate of 92.7%. Thus, the TDNN reduced by more than a factor of four the HMM recognition error rate (i.e. 6.3% reduced to 1.5%). They also described in detail the internal representations formed by the network, corresponding to a number of dynamic features of the speech signal. Hidden layer activations showed specific responses to features such as unvoiced speech, vowel onset and rising formants.

Lang and Hinton ran a series of TDNN developmental experiments on the four syllabic words "bee", "dee", "ee", and "vee" (/B, D, E, V/) [16]. The 800 tokens for these tests were obtained from speech originally recorded by the IBM speech group. 100 male speakers were used to generate the data comprising 144 msec segments

of speech parsed on the basis of vowel onset. Approximately 670 tokens were used to train the TDNN, while 100 tokens were used for testing. Recognition results for test data were 93% for a TDNN very similar to that of Figure 3. A multi-resolution training procedure which involved an alteration of the temporal scale of the TDNN between two training phases — a narrow time-window TDNN was first trained and used to set the initial connection strengths of a wide time-window TDNN — increased recognition performance to 94%. When the ratio of testing to training data was reduced to 1:1, recognition rates averaged 91.4%. Lang and Hinton also ran a series of experiments aimed at larger scale shift-invariance wherein the TDNN was used to scan entire words. This involved training with 216 msec speech segments as well as some adjustments to the temporal scale of the TDNN. Following training involving these changes, the use of multi-resolution techniques, and "counter-examples" to suppress false classifications, they achieved a 94.1% recognition rate on TDNN-scanned unsegmented speech for the /B, D, E, V/ set.

Rossen and colleagues took a modular approach to temporal shift-invariance [18]. Their network used a series of five input modules which fed a single hidden layer module. This hidden layer module then fed an output layer module. 810 tokens of speech data obtained from three male speakers representing the six stops /b, d, g, p, t, k/ followed by the three vowels /i, a, u/ (18 possible combinations) were obtained from discrete syllabic utterances. This data was bandpass filtered (70 Hz - 9.6 kHz) and sampled at 20 kHz. Log DFT spectra were computed at 5 msec intervals from Hamming windowed speech signals. These spectra were smoothed to produce 32-coefficient spectra evenly spaced between 0 and 9.9 kHz. A series of these smoothed spectra formed the input to four of the five input modules. The spectral sequences of modules 2 – 4 were time shifted by -5, -10, and -15 msec respectively from the spectral sequence of module one. The fifth input module contained cepstral information of the speech signal. Output layer nodes could have auto-associative connections (i.e. feedback connections to their own inputs), and target output patterns were multi-node (as opposed to 1-of-n) binary activation patterns equidistantly spaced in activity space. This architecture achieved a 94% recognition rate for consonants and a 93% recognition rate for vowels when trained and tested on 2-speaker data. The network achieved an 80% recognition rate for consonants obtained from the third speaker not used to train the network.

Watrous pursued shift-invariant phoneme recognition for the /b, d, g/ and /i, a, u/ tasks using a connectionist structure called the Temporal Flow Model [19]. This architecture had many similarities to the model described above. It employed recurrent connections at the output layer, non-binary output targets (i.e. the network was trained to produce a Gaussian-distributed activation across its output nodes) and temporal representations through the use of delay links between processing sub-units. The Temporal Flow Model architecture 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.

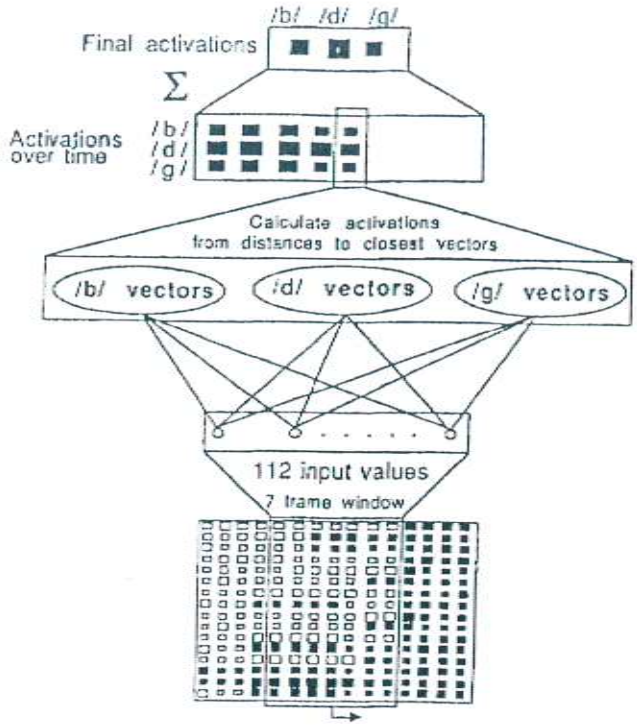

Figure 5: an LVQ network used for the /b, d, g/ recognition task[4].

McDermott and Katagiri applied the LVQ network of Kohonen and colleagues to the consonant recognition task [20]. Their experiments used a single speaker from the same Japanese database used by Waibel and colleagues, and the input layer structure of the LVQ network was identical to that of [14]. Figure 5 illustrates the moving spectral window used to feed the hidden layer that effected a connectionist implementation of Kohonen's LVQ2 algorithm [21]. Hidden layer connections were initialized using a traditional k-means clustering algorithm [9]. This network achieved a 99.6% recognition rate for all stop consonants obtained from hand-segmented data taken from a single male speaker. Performance for all stops, fricatives and affricates for the same speaker, using a modular version of the LVQ network[5], was 97.3%. Network training time was somewhat less than training time for a comparable TDNN

[4]Figure from McDermott and Katagiri [20]

[5]Section 4 details modular network architectures.

on the same task, at the cost of a 3-fold increase in the total number of connections required in the network and an increase in the time required for post-training recognition.

4 Extensions

In relative terms, all of the networks described in Sections 3.1 and 3.2 were applied to highly restricted speech recognition tasks. A natural question that follows from positive results on a limited task is how one might adapt the experimental apparatus to handle larger, more general tasks.

Waibel, Sawai, and Shikano addressed the issue of scaling the speaker-dependent /b, d, g/ TDNN to the larger problem of recognizing a combination of stops, fricatives, affricates, and nasals for a single speaker [22,23,24]. They began by investigating an expanded version of the TDNN in Figure 3. The expanded TDNN had twenty nodes per abstract spectrum in the first hidden layer, 6 phoneme class node groups in the second hidden layer, and 6 output nodes, corresponding to the voiced and unvoiced stops /b, d, g, p, t, k/. This network eventually achieved a 98.3% recognition rate on a 1613 token test set, but the amount of training required to achieve this performance level was extraordinarily high. As a result, Waibel, Sawai and Shikano investigated a number of architectural schemes aimed at increasing the scale of phoneme recognition networks through an interconnected series of sub-modules constituting a considerably larger total network. Figure 6 is an illustration of the modular all-consonant TDNN they developed. As in Figure 3, negative node activations are shown in gray against a background of white and positive activations are black. Individual TDNNs were trained for consonant sub-groups, and a TDNN designed to identify the type of articulation was trained — the training of all modules was done using the same training set. After all the sub-modules were trained they were essentially connected in parallel. Initially connections from the common input layer to all the modules' first hidden layers and connections between the modules' first and second hidden layers were constrained to their individually-trained values, but connections between second hidden layers and the output layer were re-trained using backpropagation. In the final phase of training, all connections were "freed" (i.e., allowed to be re-trained in the context of the entire network architecture). This fine-tuning resulted in an all-consonant recognition rate of 96%. Among their conclusions Waibel, Sawai, and Shikano found that sub-networks developed internal (i.e. hidden layer) abstractions that were valuable in forming distributed representations of more complex recognition tasks across the entire network. In contrast, it was not clear whether sub-network output layers could be combined as effectively — evidence suggested that these outputs simply contained insufficient information for high accuracy recognition within a modular system. These findings supported the notion of connectionist learning strategies based on distributed modular representations of knowledge.

Hampshire and Waibel investigated the use of replicated TDNNs trained on iden-

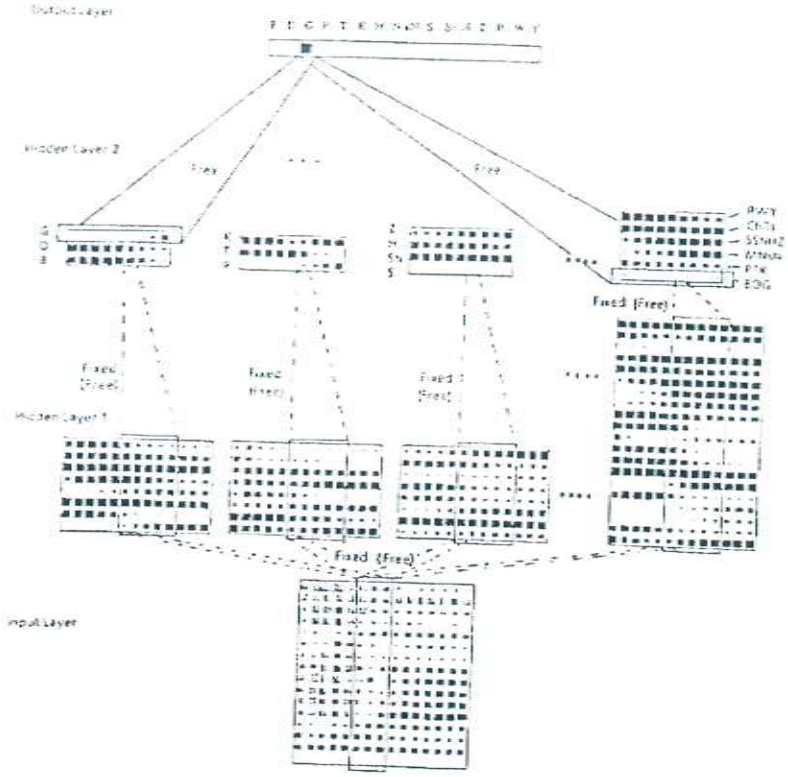

Figure 6: Waibel, Sawai and Shikano's modular all-consonant TDNN.

tical data using different objective functions for the backpropagation gradient search. In addition to the Mean-squared-error (MSE) objective function typically used in backpropagation learning, they developed an alternative objective function which they termed the Classification Figure-of-Merit (CFM) [25]. The CFM was developed as a more effective objective function for networks employed as classifiers. In their experiments, they found that the CFM classifier did not exhibit the over-learning tendency often displayed by its MSE counterpart. Additionally they found that a simple combination of both classifiers typically reduced by 24% the number of misclassifications made by the MSE classifier alone, while it "flagged" 70% percent of the remaining errors as probable misclassifications. This had the effect of

increasing single-speaker /b, d, g/ recognition rates that were on the order of 97.5% for MSE classifier networks to rates on the order of 98.5% for combined MSE/CFM classifier networks. Using these same techniques, Hampshire and Waibel improved the recognition performance of a single TDNN trained with 3 speakers on the /b, d, g/ task from an MSE classifier rate of 97.3% to a combined MSE/CFM rate of 98.1%. They surmised that the technique of flagging probable phoneme misclassifications might reduce the complexity of resolving ambiguities at higher levels of the speech recognition task.

5 Word Recognition

Section 3 highlights a number of connectionist structures applied to the task of phoneme recognition. In this section we review a number of experiments involving word recognition. We start by citing developments of temporally static word recognition networks, and follow this with a discussion of temporally dynamic word recognition architectures. As is the case for phoneme recognition networks, temporally static word recognition networks require precise temporal alignment of the word, while temporally dynamic networks do not.

5.1 Temporally Static Full-word networks

Lippmann and Gold studied a number of backpropagation network architectures applied to the task of isolated digit recognition [26]. Seven isolated monosyllabic digits were obtained from the TI Isolated Word Database representing speech from 16 different speakers. The speech data was sampled at 12 kHz, windowed, and discrete Fourier transformed; post-processing produced 15-coefficient Melscale spectra at 10 msec intervals. These spectra were used to develop two 11-point cepstra offset by 30 msec in time; the latter 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 3-layer network (i.e. a 2-layer perceptron) yielded the best connectionist recognition performance of 92.3%, averaged over all 16 speakers.

Peeling and Moore also ran experiments with isolated digit recognition [27]. They used a 3-layer network with 50 hidden-layer nodes. 60 19-coefficient spectra taken at 20 msec intervals formed the network 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 was taken from the 40-speaker Royal Speech and Radar Establishment (RSRE) database. Speaker-dependent recognition under these conditions was 99.7%.

Burr conducted a series of experiments in isolated E-set and polysyllabic word recognition using a single-layer perceptron [28]. The network input comprised 20

64-coefficient spectra: in separate experiments these spectra were computed using smoothed DFT and LPC techniques. Speech signals were sampled at 10 kHz. For DFT processing, 64 spectral coefficients were computed from Hamming windowed time series transformed to 128 point spectra; these spectra were moving-average (MA) filtered to form the smoothed 64-point spectra. For LPC processing, 300-sample Hamming windowed data formed the input to a tenth-order autocorrelation LPC estimator employing Levinson-Durbin recursion. 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.

5.2 Temporally Dynamic Full-word networks

The preceding word recognition results suggest that recognition accuracy at the word level is quite sensitive to temporal alignment of the word within the processing "window" of the classifier — as was true for the phoneme recognition task. The following experiments used a number of novel techniques to achieve shift-invariant recognition at the word level. Some of these experiments were not, strictly speaking, speech *recognition* research; nevertheless, they were all speech related, and they illustrate a number of connectionist paradigms that may prove useful in future research at the word recognition level.

Bottou used a large TDNN (see Figure 3) 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 [29]. Single exemplars of each word were obtained from 6 speakers. Speech from four speakers was used for training and speech from the remaining two speakers constituted testing data. The data was sampled at 10 kHz and used to compute 256-point DFTs at 12.8 msec intervals. These spectra were reduced to 16 spectral coefficients covering a frequency range of 100 Hz to 5 kHz — again separated by 12.8 msec intervals. Low frequency coefficients were linearly spaced, while high-frequency coefficients were logarithmically spaced. These formed the input to a 65 time-frame TDNN input layer. Three input spectra connected to 3 units in the first hidden layer; the 3 spectra "window" of the input layer was shifted 2 time delays for each first hidden layer 3-node group. Seven first hidden layer 3-node groups were combined to form input to 3-node second hidden layer groups. These, in turn, were fully connected to the

5-node output layer, corresponding to the 5 words to be recognized. 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 (he surmised that this relatively low rate for the warped training set was due to the extreme warping performed on a small number of those 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 database, Kammerer and Kupper 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 [30]. Their recognition results were 99.6% for speaker-dependent experiments and 97.3% for a speaker-independent trial.

Sakoe, Isotani, and Iso developed a Dynamic Programming Neural Network (DNN) for speaker-independent word recognition [31]. This network employed a 3-layer backpropagation architecture capable of dynamically warping its input. The input layer comprised a series of 10-coefficient Melscale spectra taken at 16 msec intervals. These spectra were linked in groups of 2 to single groups of 4 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 two ways. In a temporally pre-warped training method called "fixed time alignment", all training tokens for a particular word were time warped to a standard temporal 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.

Tank and Hopfield 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 [32]. The network, illustrated in Figure 7, 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 $f_1(\tau) \ldots f_n(\tau)$. Appropriate combinations of these time-shifted detec-

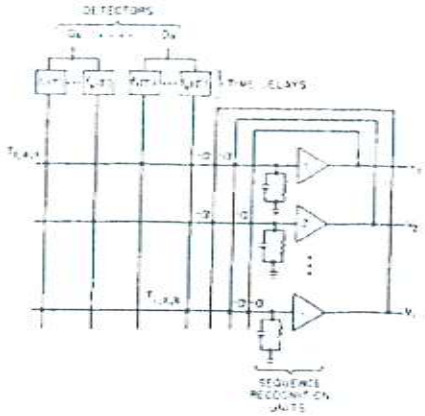

Figure 7: Tank & Hopfield's analog network for detecting nominally distorted sequences embedded in larger sequences[6].

tors 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, Unnikrishnan, Hopfield, and Tank used this same network paradigm to achieve a 99.3% recognition rate on random sequences of digits [33].

Another interesting approach to temporally dynamic word recognition has involved the use of recurrent (i.e., feedback) connections in networks to capture the sequential features of speech. Section 3.2 mentioned a number of phoneme recognition networks employing recurrent connections: Watrous, Shastri, and Waibel used recurrent connections in the output layer of a 3-layer network used to recognized the voiced stops /b, d, g/; similarly, Rossen and colleagues employed recurrence in their phoneme recognition network [18]. Prager, Harrison, and Fallside used recurrent Boltzmann machine architectures based on a first-order Markov model [35]. The explicit purpose of all these recurrent connectionist structures was to provide the network with state sequence information. The first connectionist research primarily focusing on the design and training of recurrent networks was done by Jordan [36], and that work spawned several other papers on the subject. The following networks employed various forms of recurrence to achieve temporal shift-invariant word recognition. The networks described below employed recurrence as a means of capturing higher-level sequential features of speech involving syntax.

Elman developed a 3-layer network with "contextual" units that formed a feedback mechanism between the hidden and input layers of the network [37]. Using this structure (very similar to that illustrated in Figure 8), he ran a series of experiments to assess the network's ability to form general temporal representations of input data. Network performance was judged on its ability to predict future input

[6]Figure from Tank and Hopfield [32].

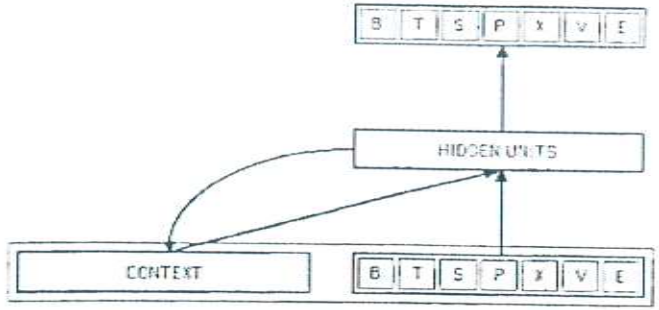

Figure 8: The recurrent network form used by Elman and Servan-Schreiber in their word recognition experiments[7].

states, given present input state and former internal (hidden) state. In effect, the network was tasked with learning discrete state-space trajectories. Elman successfully trained the network to predicted follow-on states for a set of 3 discrete trajectories in one experiment. In a more complex task he trained a similar network with 200 variable word-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 input letter of a word, and dropped rapidly (indicating high-confidence predictions) as the letter stream corresponding to the word was processed.

Servan-Schreiber, Cleermans, and McClelland expanded upon Elman's work using the same recurrent connection paradigm [38]. In their work they trained a recurrent network with 200,000 strings of varying length ($\mu = 6$, $\sigma \approx 7$) 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 sub-strings 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% non-grammatical strings while it correctly processed all grammatical strings.

Both Elman's and Servan-Schreiber, Cleermans, and McClelland's research results illustrated the effectiveness of capturing temporal context with representations of sequential state. The parallels of these works with classical linear auto-regressive (AR) signal processing theory are clear, and promise further developments of connectionist systems employing recurrent architectures. Pearlmutter is particularly active

[7]Figure from Servan-Schreiber, Cleermans, and McClelland [38].

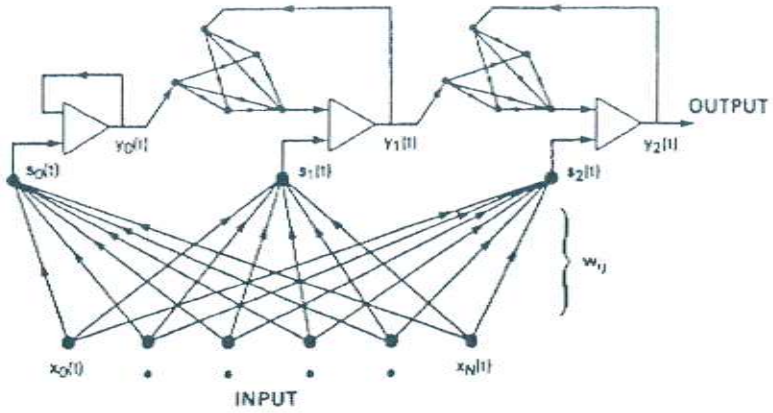

Figure 9: Lippmann and Gold's Viterbi net[8].

in developing first-order recurrent networks for control systems [39]. These networks follow continuous (i.e., non-discrete) state space trajectories in contrast to the systems described above, which follow discrete "clocked" trajectories. His research will, no doubt, prove valuable in speech-related applications of recurrent networks.

5.3 Hybrid Networks

A number of researchers have used connectionism to perform computations generally associated with more traditional forms of temporally dynamic word recognition. Lippmann and Gold developed a hybrid network called the Viterbi net [26] to perform the Viterbi algorithm [40] in a connectionist structure. Figure9 illustrates this network. The triangular-shaped 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 represented mel and differential mel cepstra (updated at 10 msec intervals). Connection strengths between input and HMM nodes were set to values obtained by conventional HMM computational techniques. The small sub-networks 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 non-connectionist HMM recognizers.

Bourland and Wellekens developed a 3-layer network to compute distance scores between input allophones and target phoneme models; these scores were fed to a traditional dynamic time warping (DTW) phoneme/word recognizer [41]. Recognition performance of this system on 10 German digits obtained from a single speaker was

[8]Figure from Lippmann and Gold[26].

100%. These same researchers wrote extensively on connectionist implementations of Hidden Markov Models [42]. These implementations included the use of recurrent connections and context sensitivity.

6 Networks with Other Applications to Speech Processing

The networks we have reviewed so far have all taken a purely connectionist approach to speech recognition — at the phoneme, word and sentence levels. In this section we review a number of different networks that have proven effective in more diverse speech processing applications.

6.1 Noise Suppression

Tamura and Waibel used a 4-layer backpropagation network to perform noise reduction on Japanese speech [17] that had been corrupted with both stationary white noise and non-stationary "colored" (computer room) noise. 216 words were corrupted with computer room noise and used as training input to a 4-layer feed-forward network with 60 nodes in each layer. All successive layers were fully interconnected. Actual speech time series formed network input and output — noisy speech was presented at the input and noise-free speech was presented at the output. The network scanned successive 60-point samples of the speech waveform comprising each of the 216 words processed. Training encompassed approximately 200 passes through the training word set. Noise reduction results for words not included in the training set were evaluated subjectively against results obtained with traditional spectral subtraction noise reduction techniques. Subjects asked to evaluate the superior noise limited speech signal chose the connectionist processed versions over spectral subtraction processed versions by a margin of 57% to 46%. Tamura and Waibel found that although their connectionist noise reduction technique yielded higher signal-to-noise ratios than spectral subtraction, it did not result in a more intelligible speech signal (again, this finding was based on subjective evaluation). They suggested that intelligibility might be enhanced by focusing network learning on more important acoustic-phonetic features of the speech signal.

6.2 Speech Coding

Tamura and Waibel's research was, in a sense, an experiment involving the abstract coding of speech in order to map a noisy input signal to a noise-free output signal. Elman and Zipser [10] conducted a series of experiments on continuous speech signals specifically aimed at developing abstract representations of the speech signal, with an eye towards possible applications to speech encoding. The speech data used comprised 505 tokens of consonant-vowel syllables (the same data described in Section 3.1). Fifteen minutes of continuous speech data containing the digits 0 through

9, a prose passage, 100 phonetically balanced words and 500 frequently-used words obtained from a single male speaker under noiseless conditions was also used. A series of network architectures were trained in an "identity mapping" mode, whereby input and output patterns were identical, and hidden layer units were trained to form reduced abstractions of the input/output speech data. Elman and Zipser made a number of interesting findings. In one case, consonant-vowel (CV) syllables were passed through a network trained on the continuous speech corpus; hidden layer node activations produced by the CV inputs were used as input to a separate network. This network was trained to categorize the abstract representations of CV input into one of nine possible categories. Recognition results for this second network varied between 86.5% and 94.2%. In another case, a network was trained with a time-domain representation of 4 minutes of speech taken from the corpus (all previous networks had been trained with frequency domain representations of speech). Following extensive training, the network was presented with a sentence composed from words on which it had not been trained. The network output was a reasonably intelligible version of the input sentence despite the fact that the network had never been trained on the utterance. Among their many conclusions the authors suggested that multiple networks trained on the same data might provide a more robust representation of speech than any single network — a conclusion later supported in work by Hampshire and Waibel [25] (see Section 4).

6.3 Text-to-speech transcription

Sejnowski and Rosenberg developed a backpropagation network named "NETtalk" to produce acoustic-phonetic transcriptions of a corpus of 1000 input words [44]. They used a 3-layer network. The input layer of the network comprised seven node groups; each of the seven groups represented a single letter of the alphabet or one of three punctuation markers. Twenty-six output units represented 21 articulatory features and five stress and syllable boundary conditions — all of which were used to model the various phonemes represented in the word corpus. The network had 120 hidden layer units fully connected to input and output layers. During training, text was stepped across the input layer's seven groups letter-by-letter, while the network output was matched against the ideal representation of the phoneme corresponding to the central element of the input letter sequence. Connections were modified using backpropagation to minimize the network output error. A correct transcription rate of 98% was achieved on the 1000 word corpus.

7 Epilogue

In this chapter we have provided a review of many of the recent significant research results in neural network applications to speech processing. At the writing of this chapter research in the field is burgeoning and the findings of many groups are both fascinating and encouraging. Clearly there is tremendous power in connectionist classifiers. Much of the challenge in applying them to speech lies in understanding how to interpret the abstract representations each network produces in order to learn more about the speech process *from the connectionist perspective*. We feel that this understanding, in turn, will lead to the development of more meaningful and more effective representations of human speech.

References

[1] Rumelhart, D., McClelland, J., and the PDP Research Group, *Parallel Distributed Processing, vol. 1*. Cambridge, MA: MIT Press, 1987, ch. 8, pp. 322 - 328.

[2] Rumelhart, D., Hinton, G., and Williams, R., "Learning Representations by Backpropagation Errors," *Nature*, vol. 323, pp. 533 - 536, October, 1986.

[3] Lippmann, R., "An Introduction to Computing with Neural Nets", *IEEE ASSP Magazine*, vol. 4, pp. 4 - 22, April, 1987.

[4] Lippmann, R., "Review of Neural Networks for Speech Recognition", *Neural Computation*, vol. 1, March, 1989.

[5] Huang, W., and Lippmann, R., "Neural Net and Traditional Classifiers", *Neural Information Processing Systems*, D. Anderson, ed., New York: American Institute of Physics, 1988, pp. 387 - 396.

[6] Peterson, G., and Barney, H., "Control Methods Used in a Study of Vowels", *Journal of the Acoustical Society of America*, vol. 24, pp. 175 - 184, March, 1952.

[7] Rabiner, L., and Schafer, R., *Digital Processing of Speech Signals*, Englewood Cliffs, NJ: Prentice-Hall, 1978, ch. 3.

[8] Duda, R., Hart, P., *Pattern Classification and Scene Analysis*, New York: John Wiley, 1973, pp. 103 - 105.

[9] Makhoul, J., Roucos, S., and Gish, H., "Vector Quantization in Speech Coding", *IEEE Proceedings*, vol. 73, pp. 1551 - 1588, November, 1985.

[10] Elman, J., and Zipser, D., "Learning the Hidden Structure of Speech", *UCSD Institute for Cognitive Sciences (ICS) report 8701*, February, 1987.

[11] Niranjan, M., and Fallside, F., "Neural Networks and Radial Basis Functions in Classifying Static Speech Patterns", *Cambridge University Engineering Department technical report CUED/F-INFENG/TR22*, 1988.

[12] Powell, M., "Radial Basis Functions for Multi-variable Interpolation", *Cambridge University Department of Applied Mathematics and Theoretical Physics technical report DAMTP/NA12*, 1985.

[13] Hansen, J., and Burr, D., "Knowledge Representation in Connectionist Networks", *Bell Communications Research technical report*, 1987.

[14] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K., "Phoneme Recognition Using Time-Delay Neural Networks," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. ASSP-37, March, 1989.

[15] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K., "Phoneme Recognition: Neural Networks vs. Hidden Markov Models," *Proceedings of the 1988 IEEE International Conference on Acoustics, Speech and Signal Processing, vol. S*, pp. 107 - 110, April, 1988.

[16] Lang, K., and Hinton, G., "A Time-Delay Neural Network Architecture for Speech Recognition", *Carnegie Mellon University technical report CMU-CS-88-152*, December, 1988.

[17] Sagisaka, Y., Takeda, K., Katagiri, S., and Kuwabara, H., "Japanese Speech Database with Fine Acoustic-Phonetic Transcriptions", *ATR Interpreting Telephony Research Laboratories technical report TR-I-0003/TR-A-0004*, May, 1987.

[18] Rossen, M., Niles, L., Tajchman, M., Bush, M., Anderson, J. and Blumstein, S., "A Connectionist Model for Consonant-Vowel Syllable Recognition", *Proceedings of the 1988 IEEE International Conference on Acoustics, Speech and Signal Processing, vol. S*, pp. 59 - 66, April, 1988.

[19] Watrous, R., *Speech Recognition Using Connectionist Networks*. PhD Thesis, University of Pennsylvania, November, 1988.

[20] McDermott, E., and Katagiri, S., "Phoneme Recognition Using Kohonen's Learning Vector Quantization", *Annual Meeting of the Acoustical Society of America*, Honolulu, November, 1988.

[21] Kohonen, T., Makisara, K., and Saramaki, T., "Statistical Pattern Recognition with Neural Networks: Benchmarking Studies", *IEEE Proceedings of the 2nd Annual International Conference on Neural Networks*, San Diego, July, 1988.

[22] Waibel, A., Sawai, H., and Shikano, K., "Modularity and Scaling in Large Phonemic Neural Networks", *ATR Interpreting Telephony Research Laboratories technical report TR-I-0034*, August, 1988.

[23] Waibel, A., "Modular Construction of Time Delay Neural Networks for Speech Recognition", *Neural Computation*, vol. 1, March, 1989.

[24] Waibel, A., Sawai, H., and Shikano, K., "Consonant and Phoneme Recognition by Modular Construction of Large Phonemic Time-Delay Neural Networks", *Proceedings of the 1989 IEEE International Conference on Acoustics, Speech and Signal Processing, vol. S*, May, 1989.

[25] Hampshire, J., and Waibel, A., "A Novel Objective Function for Improved Phoneme Recognition Using Time-Delay Neural Networks", *Carnegie Mellon University technical report CMU-CS-89-118*, March, 1989.

[26] Lippmann, R., and Gold, B., "Neural Classifiers Useful for Speech Recognition", IEEE Proceedings of the First International Conference on Neural Networks, June, 1987.

[27] Peeling, S., and Moore, R., "Experiments in Isolated Digit Recognition Using the Multi-Layer Perceptron", *Royal Speech and Radar Establishment (RSRE) technical report 4073*, December, 1987.

[28] Burr, D. J., "Speech Recognition Experiments with Perceptrons", *Advances in Neural Information Processing Systems (NIP Proceedings)*, D. Touretzky, ed., San Diego: Morgan-Kaufmann, 1989.

[29] Bottou, L., "Reconnaissance de la Parole par Réseaux Multi-couches", *University of Paris 5, School of Advanced Studies in Information Theory, Paris 5 Artificial Intelligence Laboratory technical report*, presented at NEURO-NIMES 88, Nîmes, France, November, 1988.

[30] Kammerer, B., and Kupper, W., "Experiments for Isolated Word Recognition with Single and Multi-layer Perceptrons", *Neural Networks (Abstracts of the First INNS Meeting, Boston, Sep., 1988)*, vol. 1, sup. 1, pg. 302, 1988.

[31] Sakoe, R., Isotani, R., and Iso, K., "Speaker-Independent Word Recognition Using Dynamic Programming Neural Networks", *Proceedings of the 1989 IEEE International Conference on Acoustics, Speech and Signal Processing, vol. S*, May, 1989.

[32] Tank, D., and Hopfield, J., "Neural Computation by Concentrating Information in Time", *Proceedings of the National Academy of Science: Biophysics*, vol. 84, pp. 1896 - 1900, April, 1987.

[33] Unnikrishnan, K., Hopfield, J., Tank, D., "Learning Time-delayed Connections in a Speech Recognition Circuit", *Snowbird Conference*, 1988.

[34] Watrous, R., Shastri, L., Waibel, A., "Learned Phonetic Discrimination Using Connectionist Networks", *Proceedings of the European Conference on Speech Technology*, Edinburgh, September, 1987.

[35] Prager, R., Harrison, T., and Fallside, F., "Boltzmann Machines for Speech Recognition", *Computer Speech and Language*, vol. 1, pp. 2 -27, 1986.

[36] Jordan, M., "Attractor Dynamics and Parallelism in a Connectionist Sequential Machine", *Proceedings of the 1986 Cognitive Science Conference*, L. Erlbaum, ed., Hillsdale, NJ, 1986.

[37] Elman, J., "Finding Structure in Time", *UCSD Center for Research in Language technical report 8801*, April, 1988.

[38] Servan-Schreiber, D., Cleeremans, A., and McClelland, J., "Encoding Sequential Structure in Simple Recurrent Networks", *Carnegie Mellon University technical report CMU-CS-88-183*, November, 1988.

[39] Pearlmutter, B., "Learning State Space Trajectories in Recurrent Neural Networks", *Carnegie Mellon University technical report CMU-CS-88-191*, December, 1988.

[40] Forney, G., "The Viterbi Algorithm", *IEEE Proceedings*, vol. 61, pp. 268 - 278, March, 1973.

[41] Bourland, H., and Wellekens, C., "Speech Pattern Discrimination and Multilayer Perceptrons", *Philips Research Laboratory technical report M-211*, September, 1987.

[42] Bourland, H., and Wellekens, C., "Links Between Markov Models and Multilayer Perceptrons", *Philips Research Laboratory technical report M-263*, September, 1988.

[43] Tamura, S., and Waibel, A., "Noise Reduction Using Connectionist Models", *Proceedings of the 1988 IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 5*, pp. 553 - 556, April, 1988.

[44] Sejnowski, T., and Rosenberg, C., "Parallel Networks that Learn to Pronounce English Text", *Complex Systems*, vol. 1, pp. 145 - 168, 1987.

5

SIGNAL/IMAGE PROCESSING AND UNDERSTANDING WITH NEURAL NETWORKS

by O.K. Ersoy, School of Electrical Engineering, Purdue University

1. INTRODUCTION

Artificial neural networks (ANN's) are very useful in many applications of signal/image processing and understanding. A number of methods of signal processing can also be used in ANN's to improve performance, to reduce complexity and cost of implementation, to reduce learning and recall times, to generate truly parallel architectures, to achieve better generalization, and to come up with new learning techniques.

There are two main trends in applying ANN's to signal processing problems. The first trend is the representation of the signal processing problem as one of optimization with an energy function which matches the energy function of a particular neural network [1], [2], [3]. Processing with the ANN leads to the solution of the problem by minimizing the energy function. An exciting realization in this approach is that analog computations with binary stable outputs can be used to solve such problems, similar to the way biological neural networks do their computations. This is one reason why the input and the output signals in ANN's are often represented in binary codes.

The second trend is the application of ANN's to signal recognition problems, especially speech/image recognition and vision [4], [5], [6]. Both supervised and unsupervised learning techniques such as backpropagation and competitive learning have been used for this purpose.

In this chapter, we will discuss a number of special and important topics in the interaction between ANN's and signal/image processing/understanding problems and methodologies. Sec. 2 will cover Hopfield-like neural networks, related signal representations, analog implementations and mapping of inverse problems, which often occur in signal processing as well as allied fields, to Hopfield-like neural networks. Sec. 3 discusses special issues of neural networks based on the delta rule, such as autoassociative and heteroassociative memory, and delta rule for finding projection coefficients. Sec. 4 describes the interaction between fast transforms and neural networks; the topics of learning of fast transforms and spectral-domain neural computing, nonlinear matched-filter-based neural networks, hierarchical neural networks involving nonlinear spectral processing and a number of applications are discussed in detail. Section 5 is the conclusions.

2. HOPFIELD-LIKE NEURAL NETWORKS

A Hopfield-like neural network (HNN) will be defined as a network with a state vector $[X]$ and the following properties: