Building Blocks for Speech

Modular neural networks are a new approach to high-performance speech recognition

Alex Waibel and John Hampshire

Some speech-recognition abilities that we take for granted—understanding a conversation involving several different speakers over lots of extraneous noise, for instance—are still beyond the reach of even the most powerful supercomputer. This may seem strange, since the human brain can't hope to match the arithmetical performance of a pocket calculator, but it does indicate the complexity of automatic speech recognition. Modular neural networks, however, might hold the key to achieving rapid and more-reliable machine-based speech recognition.

We recognize speech by applying an enormous body of knowledge to rapidly interpret the audio signals from the world around us. This knowledge ranges from low-level acoustic features to high-level facts about the world and the speaker's intent. These features and facts are heavily interrelated. No piece of the speech-recognition puzzle can be considered by itself, nor can pieces be evaluated sequentially. Rather, each provides a constraint that, together with many other facts and constraints, forms a total picture.

Neural Nets in Speech Recognition

The limited ability of current computer models to absorb and apply a large body of facts restricts efforts to achieve automatic recognition of human speech. Effective models must determine, maintain, and program all necessary facts and rules of speech into a system. They must then integrate the massive number of interrelationships between these facts and rules to rapidly interpret the spoken word. If speech-recognition systems could learn important speech knowledge automatically and represent this knowledge in a parallel distributed fashion for rapid evaluation, they would then be able to overcome the deficiencies of current systems. Such a system would mimic the functions of the human brain, which consists of several billion simple, inaccurate, and slow processors that perform reliable speech recognition.

The development of parallel distributed processing (PDP) or neural-network models and the development of automatic learning algorithms (see reference 1) are two very important steps in the development of reliable speech-recognition systems. You can implement algorithms that simulate PDP learning models on anything from a microcomputer to a supercomputer (see reference 2). These algorithms are even available commercially.

Two major problems have to be addressed, however, before neural-network models become useful for speech recognition: time and scaling. 

continued
Figure 1: The left side (in red) shows the time-delay feature of the network. Three 10-millisecond input slices are combined to create activations in the first hidden layer (a). Activations in the second hidden layer (b) are created by combining five slices from the first hidden layer. The right side (in blue) shows the connections from the input layer to node 4 of the first hidden layer. When an input (c) matches the pattern of the connections (d), the node is activated strongly (e).
Speech and Time

Speech is a dynamic signal, and a speech-recognition system must be able to classify sounds without knowing when a particular sound will occur. It must also be able to capture the time-varying properties—the *signature*—of speech in feature space rather than simply taking static “snapshots” of the signal. These requirements are addressed by the Time-Delay Neural Network (TDNN) (see reference 3).

Rather than trying to decide whether a particular sound is, for example, a letter *b* (the speech signal may not contain useful information at certain points in time), the TDNN scans the input for clues that provide the evidence it needs to construct an overall recognition decision. Using this method, the TDNN has demonstrated performance superior to that of other speech-recognition models in small but difficult recognition tasks.

The TDNN shown in figure 1a is designed to discriminate the voice-stop consonants *b*, *d*, and *g* as they occur in a large database of isolated spoken words. At the output, three units represent each of the three *phoneme* categories. (Phonemes are the unique sounds of a spoken language; they form the acoustic-phonetic building blocks of speech.) The input layer of the network consists of 15 time slices of speech. Each one of these time slices is a frequency spectrum representing 10 milliseconds of the speech waveform—a 10-ms voiceprint of the speaker. Each spectrum, in turn, consists of 16 coefficients representing frequencies ranging from the lower limit of hearing (about 20 Hz) to over 5 kHz.

In many neural networks, each node in a given layer is connected to all the nodes in the next layer. This is not the case, however, for the TDNN. The reasons for this are related to the temporal complexity of human speech.

Windows to the Spoken Word

Rapid changes in human speech occur over several tens of milliseconds. Therefore, a 30-ms “window” of speech (or an overlapping series of such windows) can capture the local acoustic-phonetic events that act as identifying features of a particular phoneme. The TDNN groups three 10-ms time slices from the input layer into a 30-ms window. Each coefficient in this window connects to eight nodes in the first hidden layer of the TDNN. Each of these nodes forms a condensed feature representing important cues that the network looks for in the input. The network shifts the window one time slice at a time across the input (a range of 150 ms of speech), creating 13 distinct firings at the eight nodes of the first hidden layer. The grouping scheme in the first hidden layer and its connections to the second hidden layer are analogous to the input layer’s groupings and connections to the first hidden layer. The firing patterns of the eight nodes in the first hidden layer over a five-time-slice window form the input to each of three nodes in the second hidden layer. As this window sweeps over the activation patterns in hidden layer 1, it generates activations at the three nodes in hidden layer 2. These form preliminary votes for one of the output’s three phoneme categories.

Because their weights are fixed across time shifts, the connections between the layers allow the network to find key features of the speech waveform despite the fact that these features may be spread across time or shifted along the time axis. Figure 1a illustrates the activation of a TDNN when given the voiced consonant *d* in the syllable *do*. In this figure, negative node activations in the input layer are gray, and positive node activations throughout the network are black. The degree of node activation is proportional to the size of the rectangle depicting a given node.

In figure 1c, connections from the input-layer window to node 4 of the first hidden-layer time slice are shown to the side of the TDNN. (Unlike activations, positive connections are white and negative connections are black; the background is gray.) The activation level of node 4 in the first hidden layer at a given time slice is obtained by taking the activation of each of the 48 nodes in the input layer window, multiplying this node activation by the strength of its connection to node 4, and adding up these 48 products. This sum forms the input to node 4, which uses a thresholding (or “squashing”) function to produce the output activation shown.

Note that the connections from the input layer to node 4 of the first hidden layer are positive for midrange frequencies in the input that rise or fall over time. The positive (white) connections that slope downward over time provide a strong input stimulus to node 4 when they detect a downward-sloping spectrum over time in the input layer. The arrow in figure 3 marks the onset of the *u* sound in *do*. Beginning at this point, the nodes in the input layer corresponding to frequencies from 800 Hz to 1600 Hz show the downward-sloping activation pattern over time indicative of a falling formant. (A formant is a quality of sound representative of vowels.) This results in a strong firing of node 4 in the first hidden layer.

Falling midrange frequencies are characteristic of the utterance *do* shown in figure 1c. There is a great deal of experimental evidence showing that humans rely heavily on the perception of this acoustic event (a *formant transition*) for accurate speech recognition. The positive connections in the figure that slope upward over time detect rising formant transitions, which are also vital to understanding human speech. Clearly, the TDNN has learned—without any explicit supervision—the importance of both rising and falling formant transitions for accurate discrimination of the *b*, *d*, and *g* phonemes.

Because the TDNN scans across the input speech signal, it is relatively insensitive to the timing of vowel onset for the voice stops *b*, *d*, and *g*. A version of the same utterance shown in figure 1c shifted forward in time results in the same strong output activation indicating the detection of the *d* phoneme. The advance of vowel onset merely causes the hidden units to fire earlier, in synchrony with events in the input. The combined accumulated evidence from these firings still allows the network to recognize the utterance as *a d*, as opposed to *a b* or *a g*.

The TDNN has been experimentally evaluated on a number of small phonemic discrimination tasks and has achieved excellent recognition performance. The voiced consonants *b*, *d*, and *g*, for example, can be detected in more than 98 percent of the trials with a TDNN trained on data from a single
Accelerated Learning

Coincident with the development of modular design techniques, recent advances in neural-network learning strategies and hardware and software implementations have led to dramatic improvements in network processing speeds. Learning speeds are also accelerating. These can be increased by improving the metrics that a network uses to measure how well it classifies training data.

Speed can also be increased significantly by improving the numerical search techniques that form the basis of network learning. Research in this area has resulted in learning procedures that converge to near-optimal results much more rapidly than before (see references 4, 5, and 6). Indeed, improvements in learning algorithms have brought the training time for a typical TDNN task down from three days of run time on a supercomputer to 8 minutes of CPU time on a high-end engineering workstation.

High-speed computing capabilities for neural-network training are becoming more accessible to personal computer and workstation users. Several manufacturers now offer plug-in floating-point accelerator boards for microcomputers that yield speeds of more than one million floating-point operations per second, while workstation manufacturers are producing desktop machines that rival super-minicomputers produced just a few years ago. Massively parallel connectionist hardware designs are also under development in various laboratories (see reference 7).

Speech recognition using modular neural networks is progressing rapidly. What seemed impossible a short time ago will soon be done on a personal computer. Advances in system-design techniques, learning software, and underlying hardware are creating the computing power required for very-large-scale neural-network tasks. All these advances bring connectionist design for speech and signal interpretation within reach of commercially available and affordable technology.

REFERENCES

ACKNOWLEDGMENT
We gratefully acknowledge K. Lang, G. Hinton, K. Shikano, T. Hanazawa, H. Sawai, and P. Haffner for their significant contributions, and ATR Interpreting Telephony Research Laboratories, Bell Communications Research, and the National Science Foundation for supporting this research.

Alex Waibel holds a Ph.D. in computer science from Carnegie Mellon University, where he is a research computer scientist. John Hampshire is a Ph.D. candidate at Carnegie Mellon in electrical and computer engineering. The authors can be reached on BIX c/o "editors."

Neural Networks are Solving Real Problems

Here's What NeuroShell Users are Doing:

- Circuit board problem diagnosis
- Psychiatric evaluations
- Stock market predictions
- Sales forecasts
- Oil exploration
- Optimizing biological experiment results
- Price forecasts
- Analysis of medical tests
- Optimizing scheduled machine maintenance
- Predicting student performance
- Horse racing picks
- Factory and shop problem analysis
- Optimizing raw material orders
- Spectral analysis
- Selection of criminal investigation targets
- Employee selection
- Process control
- And much, much more.

Since NeuroShell learns by example, handles fuzzy logic, can give tight data fits, and doesn't try to capture knowledge in rules, it is also being used as an alternative in many cases to expert systems, the ID3 algorithm, and regression analysis.

NeuroShell is ready to use for real problems on your IBM PC or compatible, and still only $195. Math coprocessor recommended. No programming or Ph.D. required! Shipping free by mail in US, Canada, and Mexico ($9 elsewhere). Add 5% tax in MD.

Ward Systems Group, Inc.
228 West Patrick St. / Frederick, MD 21701
TEL (301) 662-7950  FAX (301) 663-6656

NeuroShell is a trademark of Ward Systems Group, Inc. IBM PC is a registered trademark of International Business Machines
Neural Networks: Theory and Practice

For most of their existence, neural networks and neural-network simulations have been solely objects of university-based research. In the last few years, however, researchers and others have founded companies dedicated to producing commercial products based on neural-network technology. To reflect both the academic and commercial aspects of the technology, this resource guide consists of two parts. The In Theory section lists books and articles you can read to learn more about neural networks. The In Practice section lists some of the available neural-network hardware and software products, listed alphabetically by company name.

IN THEORY


IN PRACTICE

The Brain Simulator $99
Runs under MS-DOS
Tutorial software for neural circuit design
Abbott, Foster & Hauserman
44 Montgomery, Fifth Floor
San Francisco, CA 94014
(800) 562-0225
(415) 955-2711
Inquiry 1181.

N-NET
MS-DOS version $895
VAX/VMS version... starting at $2995
Integrated neural-network development system; uses functional link net architecture
AI Ware, Inc.
11000 Cedar Ave., Suite 212
Cleveland, OH 44106
(216) 421-2380
Inquiry 1182.

BrainMaker
Runs under MS-DOS
Neural-network simulation software; supports five types of nodes and can process up to 500,000 connections per second
California Scientific Software
160 East Montecito, Suite B
Sierra Madre, CA 91024
(818) 355-1094
Inquiry 1183.

Cognitron
MS-DOS Windows and Mac versions... $600
INMOS transputer version... $1800
Neural-network/parallel-processing prototyping and delivery system
Cognitive Software, Inc.
703 East 30th St.
Indianapolis, IN 46205
(317) 924-9988
Inquiry 1184.

Connections $87.95
Runs under MS-DOS
A traveling-salesman demo modeled after Hopfield networks
NetWurkz ........................................ $79.95
Runs under MS-DOS
Elementary introduction to neural networks
DAIR Computer Systems
3440 Kenned Dr.
Palo Alto, CA 94303
(415) 498-7081
Inquiry 1185.

Savvy Text Retrieval System
Savvy Signal Recognition System
Savvy Vision Recognition System
(Call for pricing)
Run under VAX/VMS, MS-DOS, and Unix
Libraries of C subroutines that use
neural technology to solve real-world
problems
Excalibur Technologies
2300 Buena Vista SE
Albuquerque, NM 87106
(505) 764-0081
Inquiry 1186.

ANZA ........................................... from $7000
At-compatible neural-network
coprocessors; includes software and
programming interface
ANZA Plus ................................. from $12,500
At-compatible neural-network
coprocessors
ANZA Plus/VME ............................ $24,950
Neural-network coprocessor for Sun
workstations
AXON ....................................... $1950
A neural-network description language
Neural Network Development
Toolkit ........................................ $3950
For ANZA Plus and ANZA Plus/VME
systems
Ports C programs into ANZA Plus and
ANZA Plus/VME formats; includes
AXON
ExploreNet
MS-DOS version ....................... $995
Sun version ......................... $995
Stand-alone neural-network software
HNC, Inc.
5501 Oberlin Dr.
San Diego, CA 92121
(619) 546-8877
Inquiry 1187.

MD/210 Fuzzy Set
Comparator .................................. $38
Hardware implementation of Hopfield
neurons
Micro Devices
5695 Beggi Rd.
Orlando, FL 32810
(407) 299-0211
Inquiry 1188.

N1000
Neural-network development tools for
signal and image processing applications
Including 80386 computer
........................................ from $19,000
Support package only for Sun-4,
Sun-3, IBM PC AT, and PS/2
Model 50 ............................... from $7955

NS90 ........................................... from $495
Runs on IBM PC AT and PS/2 Model 50
Single-user NCE network software
Nestor, Inc.
1 Richmond Sq.
Providence, RI 02906
(401) 331-9640
Inquiry 1189.

Awareness ................................ $275
Runs under MS-DOS
Introduction to four types of neural
network paradigms
Genesis ............................. $1095
Runs under MS-DOS
Neural-network development
environment
Neural Systems
2827 West 43rd Ave.
Vancouver, BC Canada V6N 3H9
(604) 263-3667
Inquiry 1190.

NeuralWorks Explorer .............. $299
Runs under MS-DOS
An introduction and tutorial on neural
networks
NeuralWorks Professional II 
MS-DOS and Macintosh
versions ............................ $1495
Sun 3 and Sun 4 versions ............ $2995
NeXt and INMOS transducer
versions ............................... call for pricing
Neural-network development system
NeuralWorks Designer Pack ........ $1995
MS-DOS and Sun versions
Links Professional II networks with C
programs
Neuware, Inc.
103 Buckskin Court
Sewickley, PA 15143
(412) 741-5659
Inquiry 1191.

MacBrain ................................. $995
Runs on Macintosh
Lets you prototype and deliver neural
network applications
HyperBrain
(Comes with MacBrain)
Toolkit allows you to build neural
network applications within HyperCard
Neurom, Inc.
1 Kendall Sq., Suite 2200
Cambridge, MA 02139
(617) 577-1202
Inquiry 1192.

Owl I, II, III ...................... from $349
Libraries of modules for IBM and
compatible that lets you define and
access 10 different neural networks
Extension Pack ....................... $149
Three additional networks
Olmsted & Watkins
2411 East Valley Pkwy., Suite 294
Escondido, CA 92025
(619) 746-2765
Inquiry 1193.

Intelligent Pattern Recognition
Chips ................................ $590
Stores a 1000-by-64 matrix of weights and
multiplies it with an input vector
Oxford Computer
39 Old Good Hill Rd.
Oxford, CT 06483
(203) 881-0891
Inquiry 1194.

ANSim 2 .................................... $495
Runs under MS-DOS
13 neural-network models
ANSkit ................................ $950
Runs under MS-DOS
Neural-network development system
ANSpec ................................ $2995
Runs under MS-DOS
Neural-network specification language
Delta Floating Point
Processor ................................ $24,950
Runs on IBM PC, AT, PS/2, and
Sun386
Neural-network accelerator boards
Sigma Neurocomputer
Workstations ........................ from $1,500
80386-based systems with Delta
Processor, ANSkit, Delta C, Delta
Macro, and ANSpec
SAG
10260 Campus Point Dr.
Mail Stop 71
San Diego, CA 92121
(619) 546-6290
Inquiry 1195.

DENDROS-1 ......................... $35
Neural-network chip that produces the
firing pattern of the inputs and the
connection weights of 22 synapses
DENDROS-1 Evaluation Board .... $695
Uses eight DENDROS-1 chips to create
a hardware-based neural network
Syntomics Systems, Inc.
20790 Northwest Quail Hollow Dr.
Portland, OR 97229
(503) 293-8167
Inquiry 1196.

TRW Mark V Neural Processor
Write for pricing information
Runs on VAX/VMS
MC68020-based parallel-processing
system includes tools for neural-network
applications
TRW
Military Electronics & Avionics Div.
One Rancho Carmel
San Diego, CA 92128
(619) 592-3482
Inquiry 1197.

NeuroShell .................. $195
Runs under MS-DOS
Creates neural-network applications using
a modified back-propagation
Ward Systems Group, Inc.
228 West Patrick St.
Frederick, MD 21701
(301) 662-7950
Inquiry 1198.

Inclusion in the resource guide does not indicate that BYTE endorses or recommends either the
product or the company. In addition, BYTE is not responsible for any omissions, changes, or errors in the information listed.