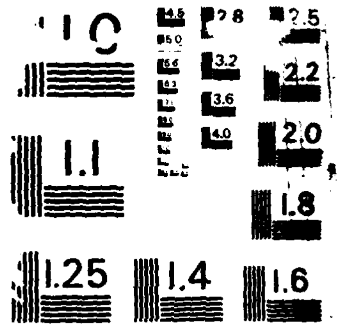
AD-A191 588 NEURAL NETWORKS FOR SPEECH APPLICATION(U) NAVAL OCEAN SYSTEMS CENTER SAN DIEGO CA S A LUSE NOV 87 1/1

UNCLASSIFIED F/G 25/4 NL



COPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

DTIC FILE COPY

REPORT DOCUMENTATION PAGE

AD-A191 588

1b. RESTRICTIVE MARKINGS	
3. DISTRIBUTION/AVAILABILITY OF REPORT	Approved for public release; distribution is unlimited.
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	
5. MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION	Naval Ocean Systems Center
6b. OFFICE SYMBOL (If applicable)	NOSC
7a. NAME OF MONITORING ORGANIZATION	Naval Ocean Systems Center
6c. ADDRESS (City, State and ZIP Code)	San Diego, California 92152-5000
7b. ADDRESS (City, State and ZIP Code)	San Diego, California 92152-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	
8b. OFFICE SYMBOL (If applicable)	
9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State and ZIP Code)	

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	AGENCY ACCESSION NO.

11. TITLE (Include Security Classification)

Neural Networks for Speech Application

12. PERSONAL AUTHOR(S)

S.A. Luse

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Professional paper	FROM Oct 1987 TO Oct 1987	December 1987	

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

brainlike paradigm
Artificial Neural Systems (ANS)
mapping

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This is a general introduction to the reemerging technology called neural networks, and how these networks may provide an important alternative to traditional forms of computing in speech applications. Neural networks, sometimes called Artificial Neural Systems (ANS), have shown promise for solving problems that traditional algorithmic and AI (artificial intelligence) approaches have found difficult. The world's greatest super-computer calculates Pi to thousands of decimal places in seconds using algorithmic techniques, but it may not ever be able to recognize a smiling human face when only a non-smiling version of this face is available for comparison.

One reason for this is that computer process information serially, and an incredibly large number of serial steps are required to perform such a task. Therefore, even with the fastest computer, developing algorithms that can ignore unimportant differences in images and match-stored patterns with acceptable time delays is not an easy feat. The brain, on the other hand, processes information in a parallel fashion, distributing information and processing tasks throughout many neurons and their interconnections. ANS processors mimic this parallel structure and are able to outperform serial processors for certain tasks. They can also learn from their environment and are highly tolerant of internal failures.

Published in Speech Technology Vol. 4, Oct/November 1987.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS
21. ABSTRACT SECURITY CLASSIFICATION	UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL	S.A. Luse
22b. TELEPHONE (Include Area Code)	619-553-3652
22c. OFFICE SYMBOL	Code 441

DD FORM 1473, 84 JAN

83 APR EDITION MAY BE USED UNTIL EXHAUSTED
ALL OTHER EDITIONS ARE OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

Neural Networks for Speech Applications

A "brainlike" paradigm offers computing abilities not available to standard algorithmic techniques, and shows potential for building high-quality speech recognizers.

Stephen A. Luse
Senior Engineer
Naval Ocean Systems Center
San Diego, CA

THIS IS A GENERAL INTRODUCTION to the re-emerging technology called neural networks, and how these networks may provide an important alternative to traditional forms of computing in speech applications. Neural networks, sometimes called Artificial Neural Systems (ANS), have shown promise for solving problems that traditional algorithmic and AI (artificial intelligence) approaches have found difficult. The world's greatest super-computer calculates Pi to thousands of decimal places in seconds using algorithmic techniques, but it may not ever be able to recognize a smiling human face when only a non-smiling version of this face is available for comparison!

One reason for this is that computers process information serially, and an incredibly large number of serial steps are required to perform such a task. Therefore, even with the fastest computer, developing algorithms that can ignore unimportant differences in images and match-stored patterns with acceptable time delays is not an easy feat. The brain, on the other hand, processes information in a parallel fashion, distributing information and processing tasks throughout many neurons and their interconnections. ANS processors mimic this parallel structure and are able to outperform serial processors for certain tasks. They can also learn from their environment and are highly tolerant of internal failures.

—

Text-to-Speech Conversion: A Mapping Problem

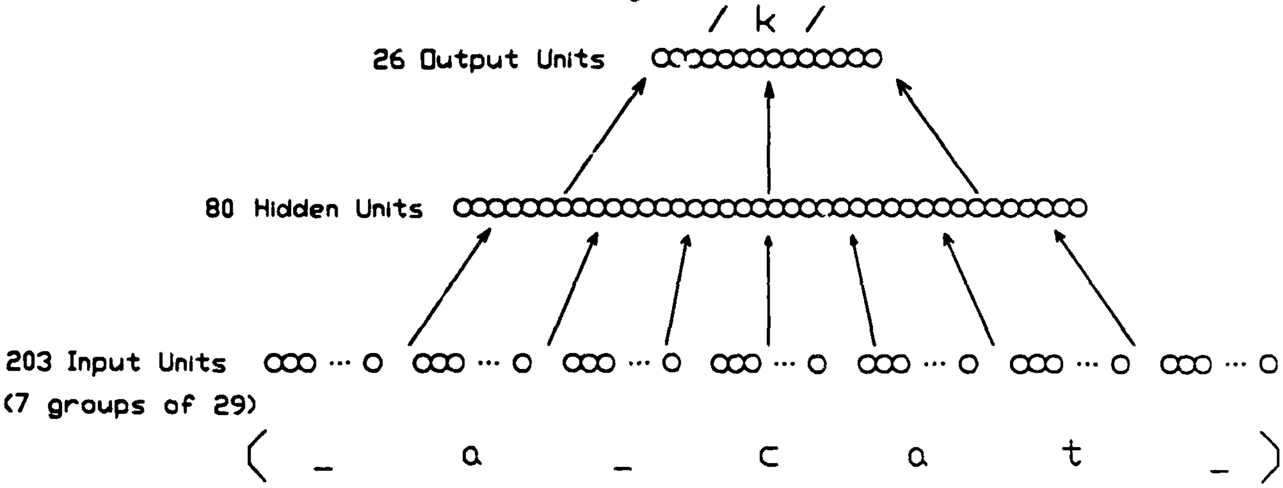
Neural networks are paradigms, or models, that "compute" in a brainlike fashion. While neural networks do not emulate the brain's full complexity, their structure is "brain-inspired" in that they consist of groups of interconnected "neurons" called processing elements. In practice, these processing elements are interconnected in different ways to form different processing architectures with different functionalities. For example, some networks can perform associative memory functions that map arbitrary input and output pairs while others can group inputs into classes, or even self-organize and perform feature extraction on data presented to them. Other architectures reconstruct full sets of data when given only parts or corrupted versions of what is stored (content addressable memory functions). In other words, they take an incomplete input and supply the missing parts. Still other networks have the ability to generalize and combine knowledge in a way that allows extraction of consensus and off-setting of differing opinions among experts in a specific domain. More complex networks can model the concepts of memory and motor control in biological systems to show possible mechanisms by which organisms learn to interact with an environment.

With all of this functionality, neural networks seem to have at least some potential for building high-quality speech recognizers with superior capabilities for speaker independence and continuous speech. Current speech-recognition implementations suffer from large data storage requirements and relatively long processing times, due to the algorithmic processes involved. These requirements make real-time speech-recognition systems difficult to achieve. As ANS technology and applications become refined and true neural network computers become available, we may see neural networks solving problems related to syntax and knowledge extraction of recognized speech without the limitations imposed by algorithmic approaches.

An experiment performed by Terrence Sejnowski and Charles Rosenberg, at Johns Hopkins University, shows the potential for neural network applications in the speech field. They used a neural network called the back-propagation model (see sidebar: The Back-Propagation Network) to address the problem of synthesizing speech from text. They call their network NETtalk.

Sejnowski and Rosenberg realized that text-to-speech conversion is a mapping problem and assumed that some function exists that can perform the necessary mapping. The abilities of the back-propagation network made it unnecessary to know the form of this mapping function. The back-propagation network has the ability to compare a set of inputs with desired outputs and generalize the implied relationship, developing its own internal representation of the function.

Fig. 1



NETtalk Architecture

The Architecture

NETtalk's architecture consists of 203 input units, 80 hidden units and 26 output units (see Fig. 1). The input layer is a cluster of seven groups. Each processing element in each group encodes one letter of input text (A through Z, two punctuation characters and a space character used to separate word boundaries). Seven groups allow the network to view a window of seven characters at one time. The desired output from the network is the correct phoneme associated with the center, or fourth, letter of this window. The three letters on either side of the center letter provide partial context for determination of the proper phoneme.

The output units of the network encode phonemic information represented by twenty-one articulatory features such as point of articulation, voicing, vowel height, and so on. Five additional units encode stress and syllable boundaries for a total of 26 output units. These output features drive a DECtalk phoneme synthesizer bypassing its internal text-to-speech algorithm.

The Network Makes Rules

As a training set, 1024 words of transcribed natural speech from a six-year-old child were used. Each input word was assigned a set of output phonemes and stresses that made the word sound natural when played through DECtalk. The network learned the relationship between the input words and output phonemes by stepping each word through the seven-character window, shifting one character at a time. After 10 passes through the training set, the synthesized text was understandable, and after 50 passes, the error rate was less than 5%.

To determine if the network merely memorized the training words or whether it actually determined its own internal rules for pronunciation, Sejnowski presented the network with 439 different words. The performance on these words was 78%, which showed that the network did in fact make generalized rules with considerable success. The network performed quite well on novel (untrained) words and accurately transformed them to appropriate phonemic representations.

Enhancements to this experiment have demonstrated interesting properties of the back-propagation network. Plotting learning curves of the network as error rates, Sejnowski noticed that learning follows a power law which is characteristic of human learning. The network also exhibits resistance to damage. Random perturbations of the network connection weights had little effect on the performance of a highly trained network, and retraining of the damaged network occurred much quicker than original training.

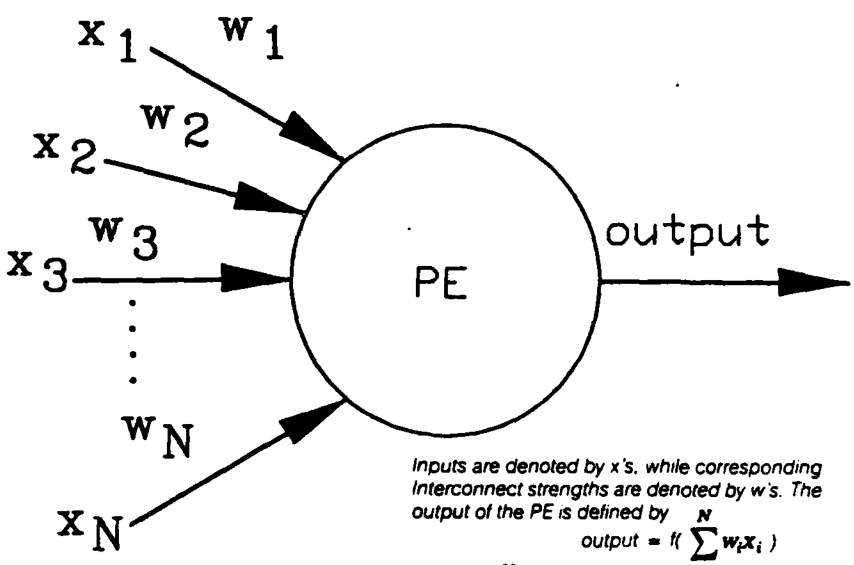
Potential and Applications of the Concept

Sejnowski's experimental synthesizer is far from becoming a commercial product because the neural network is simulated on a digital computer and cannot run in real time, but he has shown through this experiment that neural networks provide hope for solving complex problems in man-weeks or months rather than man-years, as required by traditional algorithmic approaches.

Other researchers have also shown the potential power of the neural network concept. Dr. Jeff Elman, University of California at San Diego, is using neural networks as a tool for studying speech perception. Using the network's ability to internally organize information, he is gaining an understanding of how humans may organize and classify speech information.

Teuvo Kohonen, Helsinki University of Technology, is developing self-organizing and content-addressable memory networks. Using these networks, he has developed a prototype speech recognizer that self-organizes phonemes into a two-dimensional map (This map is interesting because the network groups phonemes according to high-level and low-level features much as humans have done in speech analysis.) He then processes spoken speech so that a map of a word is drawn on this two-dimensional phoneme

Fig. 2



The Processing Element

Inputs are denoted by x's, while corresponding interconnect strengths are denoted by w's. The output of the PE is defined by

$$output = f(\sum_i^N w_i x_i)$$

where $f(\sum_i^N w_i x_i)$ *is the activation function.*

map. A word spoken multiple times by the same speaker will traverse the map in a consistent manner due to the network's power to generalize. Another network uses the word map as input and is able to match a particular map to the proper corresponding word.

Neural networks are showing promise in other fields as well. A demonstration was provided at the International Conference on Neural Networks that took place recently in San Diego. Using a personal computer, a TV camera, and a neural network plug-in coprocessor board, the operator could store images of many faces in the network. A person could then disguise himself with wigs, glasses, mustaches, etc. A new image is taken and input to the network. The network recalls the original image of the person with enough accuracy to demonstrate the use of neural networks as a recognition technique.

Another demonstration showed a character recognizer that was reasonable at de-

A Brief History of Neural Networks

■ The theory of neural networks has evolved from bio-physiological and neuroscience theories of how neurons process information in the brain. Early studies by McCulloch and Pitts during the forties led to the development of the first neural network model in 1943. This model showed that a neural network could compute and process information in a manner different for current standard computing devices. However, the model was extremely limited.

In 1949, Hebb introduced the concept of learning to neural networks. From observations of how biological neurons learn, he developed the famous Hebbian Learning Rule which says that the weight of a synaptic connection is modified in proportion to the difference between the target and actual output of a neuron. This modification always tends to move the neuron (processing element) output in a direction closer to the desired output values.

Lashley's studies of the mind (1950) led to a formalization that the mind has a distributed knowledge representation. That is, an idea or concept is not stored locally, but is stored in a distributed fashion throughout many cells. This added reinforcement to the original ideas established by McCulloch and Pitts.

The field of neural systems initially became popular in 1957 when Rosenblatt developed a neural model which he called the Perceptron. The Perceptron proved many useful concepts and was the first model developed that had the ability to learn patterns and make generalizations.

Interest in the field subsided when Minsky and Papert published their book *Perceptrons: An Introduction to Computational Geometry*. In this book they demonstrated Perceptron computational limitations by showing it couldn't perform basic functions as simple as an exclusive-or (XOR).

Even though neural network research was slowed as a result of Minsky and Papert's book, it still continued in several areas through the 1960s. Wilshaw extended the mathematical analysis of distributed memory models and found properties associated with various modeling schemes. Stephen Grossberg used neural models to explain his concepts of memory and motor control. His theories are providing powerful insights into the workings of the mind and laying a foundation for an overall mathematical theory for neural network operation and design.

In the early '70s Amari made contributions in Boolean network theory while Anderson developed the concept of completely distributed linear associative memories. Kohonen successfully explained the concept of self-organizing associative memory, which organizes and stores information similar to mechanisms expected to be operating in the brain.

In 1977, the field of Cognitive Psychology began using the concept of neural networks as cognitive and learning models for speech understanding. This movement was initiated by Rumelhart, at the University of California at San Diego, and by McClelland, at Carnegie-Mellon University, whose work was inspired by the HEARSAY Speech Understanding System at Stanford University. They are responsible for the term "parallel distributed processing," or PDP models, as cognitive neural networks are often called today.

About this same time commercial applications research was conducted by Robert Hecht-Nielsen at TRW. While working at TRW he developed the commercially available Mark III and Mark IV neurocomputers that model neural networks and run simulations quickly.

The recent resurgence of interest in ANS technology is mostly attributed to biophysicist John Hopfield, California Institute of Technology. In 1982 he published a paper which presented the subject in the broader context of classical mechanics and statistical physics as well as electrical engineering and information processing. This highlighted neural network potential to researchers in many fields.

Since that time many interesting developments and contributions have occurred. One of these came from Bart Kosko of Verac Corporation. He integrated the concepts of fuzzy logic and fuzzy data with neural networks through fuzzy cognitive maps and mapping networks that he developed. These techniques and networks allowed the individual inputs of many experts in a specific field to be combined and applied without constraining the experts to form a consensus of opinion. This work shows promise in applications that are now being done by Expert Systems.

termining which digit, zero through nine, any user writes on a digitizing pad. The character recognizer works with a degree of accuracy high enough to demonstrate potential, but what makes this simple demonstration interesting is that it was developed with less than three man-weeks of work.

Attendees at the IEEE first annual International Conference on Neural Networks (ICNN) demonstrated a high degree of interest in the field as evidenced by the larger than expected attendance. Talks and tutorials at the conference showed that neural networks provide powerful concepts for solutions to problems of pattern recognition, classification, image processing, speech processing and knowledge processing. Although demonstrations at the conference showed potential for neural network applications in the near future, they were only simulations on standard computers limiting their overall usefulness.

Current real-time applications will initially be limited to small networks for signal and image processing. Experts agree that for neural networks to conquer the difficult problems such as independent recognition of large vocabularies, knowledge representation, and cognition, they

Processing Elements and Neural Networks

■ Neural Networks consist of discrete independent units called Processing Elements (PE) which are analogous to neurons in the brain.

In its simplest form, the PE has multiple inputs and one output (see Fig. 2). Associated with each input is a weight which determines how much influence the input has on the PE. Inputs to a PE can be from the real world or from other PEs in the network.

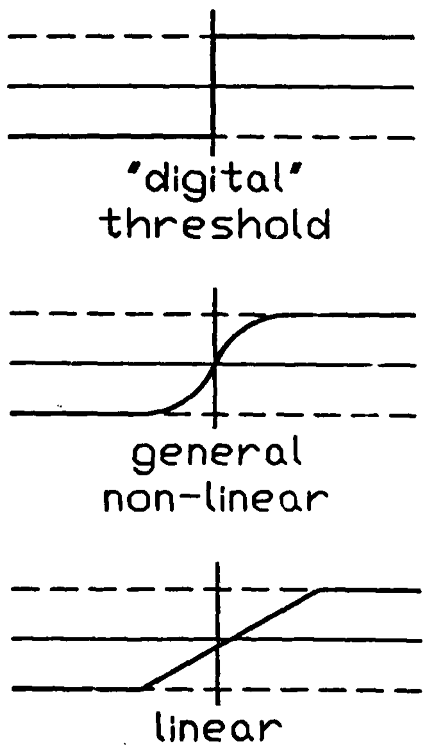
The forms of the inputs are application dependent. If the inputs are from the real world, they can be real numbers or in binary form (0,1); in some cases it is useful to use bipolar form (-1, +1). If the inputs are from other processing elements, they are real numbers with magnitude less than or equal to one.

A PE performs a simple operation. The output from the PE is the weighted sum of the inputs passed through a non-linear activation function. Activation functions are incorporated into neural-network models to emulate thresholding effects present in biological processes, and are necessary for effectively training the network.

A neural network is a set of interconnected PEs. There are many ways that PEs can be interconnected. A number of different interconnection architectures having different performance characteristics are popular in the literature. These have names like Counter-Propagation, Back-Propagation, Adaptive Resonance, Hopfield Networks, Kohonen Memories, and many more. Each architecture has properties that make it useful for different applications. Common to all architectures is the simple processing element with variable weight interconnections.

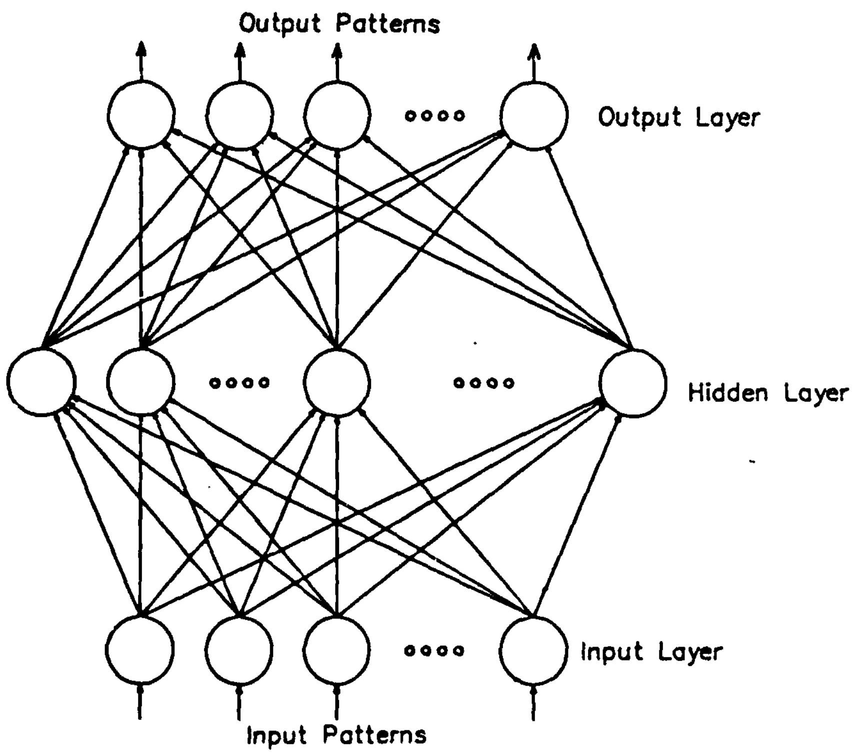
In each architecture, PE outputs are determined by a class of simple, monotonically increasing activation functions acting on the weighted sums of their inputs. These output functions range from "digital" threshold functions to totally linear functions with the most interesting networks having non-linear functions somewhere between these extremes (Fig. 3).

Fig. 3



PE Activation Functions

Fig. 4



The Back-Propagation Network

will require tremendous amounts of parallel processing that cannot be achieved on today's hardware. Optical processing holds promise for the future in this area.

STEPHEN LUSE *received his MSEE from San Diego State University in 1982 and his BSEE from the University of California at Santa Barbara in 1975. He is currently a senior engineer and systems analyst for the Human Factors and Speech Technology Branch at the Naval Ocean Systems Center in San Diego, CA, and is developing applications for speech technology in military environments.*

REFERENCES

Amari, S., "Characteristics of Randomly Connected Threshold-Element Networks and Network systems," *Proceedings of the IEEE*. Vol. 59, No. 1, Pgs. 35–47 (January 1971)

Elman J. and Zipser, D., "Learning the Hidden Structure of Speech," *University of California, San Diego Technical Report*. Department of Linguistics and Institute for Cognitive Science, University of California: San Diego (1987)

Grossberg, S. and Kuperstein, M., **Neural Dynamics of Adaptive Sensory-Motor Control: Balistic Eye Movements.** Elsevier/North Holland: Amsterdam (1985)

Hebb, D., **Oranization of Behavior**, Wiley Press, New York (1949)

Hecht-Nielsen, R., "Performance of Optical, Electro-Optical, and Electronic Neurocomputers," *TRW Rancho Carmel Artificial Intelligence Center Technical Report* One Rancho Carmel, San Diego, CA 92128 (1986)

Hopfield, J., "Neural Networks and Physical Systems With Emergent Collective Computational Abilities," *Proceedings of the National Academy of Science USA* Vol. 81, Pgs. 3088–3092 (May 1984)

Kohonen, T., **Self-Organization and Associative Memory.** Springer-Verlag, Berlin (1984)

The Back-Propagation Network

The back-propagation network is popular because it is powerful, fairly easy to implement, and well understood. It has the ability to learn mappings by example, that is, it will generalize input-output pair relationships.

Most back-propagation models consist of three layers of processing elements (PEs) (see Fig. 4). The input layer collects inputs from the real world and distributes these to the hidden layer. Each output from the first layer forms an input to every PE in the hidden layer. Outputs from PEs in the hidden layer are distributed in the same fashion to the output layer. The outputs from this layer form real-world outputs

Inputs to the network can be real numbers or binary in form (1's and 0's) depending on the application. Outputs can also be real or thresholded to binary form.

Mappings

Since the back-propagation network has the ability to learn mappings, it is useful to discuss what is meant by a mapping. A mapping is some function that converts an input data set to an output data set. For example, the conversion of text-to-speech is a mapping. The complexity of this mapping is evidenced by the algorithms used in commercial speech synthesizers.

The interconnect weights to each processing element in the back-propagation network determine what mapping the network performs. Determining a set of weights that performs an optimal mapping would seem formidable, but we can take advantage of learning algorithms to automate this task. We can let the network determine the best weights by using an automated procedure that allows the network to improve performance through practice.

The goal of the learning procedure is to minimize the average squared error between the computed values of the network's outputs with the desired outputs. Each weight is modified in proportion to the amount of error that it contributes to the overall error for all patterns presented. Modification moves weight values in a direction that decreases the overall output error (a gradient descent algorithm).

Training

1. Start with a back-propagation network and a randomized set of starting weights.

2. Collect a large set of input-output pairs that are representative of the mapping to be performed.

3. Present one input to the network and determine the network's output.

4. Find the error between this output and the desired output.

5. "Back-propagate" this output error to weights of elements in the output layer as well as weights of the hidden layer. That is, modify the weights so as to decrease the error for this pattern. The amount of change necessary for each weight is governed by the learning rules for the back-propagation network which are based on a gradient descent algorithm.

Repeat steps 3 through 5 of the second input in the training set. Determine its associated error and again modify the weights. Obviously, how much the weights are modified will greatly affect how well the network weights can be adjusted to improve performance for the second input while not losing much performance for the first input. The trick is to determine the error and weight change that best accommodates both training inputs. Such a "statistical" feature can be built into the learning algorithm in several ways.

The presentation of inputs and "statistical" weight adjustment continues throughout the training set. After the complete training set has been presented, the overall performance error of the network is lower than prior to training. The overall error can be decreased toward an asymptotic minimum by repeating this training process. Indeed, it may be necessary to present the training set hundreds or even thousands of times before the overall error is low enough for a particular application.

Once the network is trained, it can map any input from the training set to its corresponding element in the output set with a relatively high degree of accuracy. The network has generalized and 'learned' the mapping implied by the input-output pairs. Note that it is not necessary to consider all possible input-output pairs, just enough to statistically represent the relationship between them. Novel input data can now be applied to the network and it will respond with an appropriate output based on its internal, self-generated representation of the implied mapping.

Kosko, B., "Fuzzy Entropy and Conditioning," *Information Sciences*. (1986)

Lashley, K., **Brain Mechanisms and Intelligence,** University of Chicago Press, Chicago (1929)

McCulloch, W. and Pitts, W., 'A Logical Calculus of the Ideas Immanent in Nervous Activity," **Bulletin of Mathematical Biophysics,** Vol. 7, Pgs. 115-133, (1943)

Minsky, M. and Pappert S., **Perceptrons,** MIT Press, Cambridge, MA (1969)

Rosenblatt, F., **Principles of Neuro-dynamics,** Spartan Books, Washington (1962)

Rumelhart, D. and McClelland, J. "Learning Internal Representations by Error Propagation," **Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume One: Foundations.** Pgs. 318-362. Cambridge, MA: Bradford Books/MIT Press (1986)

Sejnowski, T. and Rosenberg, C., "NETtalk: A Parallel Network that Learns to Read Aloud," *Johns Hopkins University Electrical Engineering and computer Science Technical Report JHU/EECS-86/01*. Johns Hopkins University, Baltimore, MD 21218 (1986)

Wilshaw, D. and Longuet-Higgins, H., "The Holophone—Recent Developments," **Cognitive Processes: Methods and Models.** Pgs. 349-357, University of Edinburgh (1969)

SUGGESTED READINGS

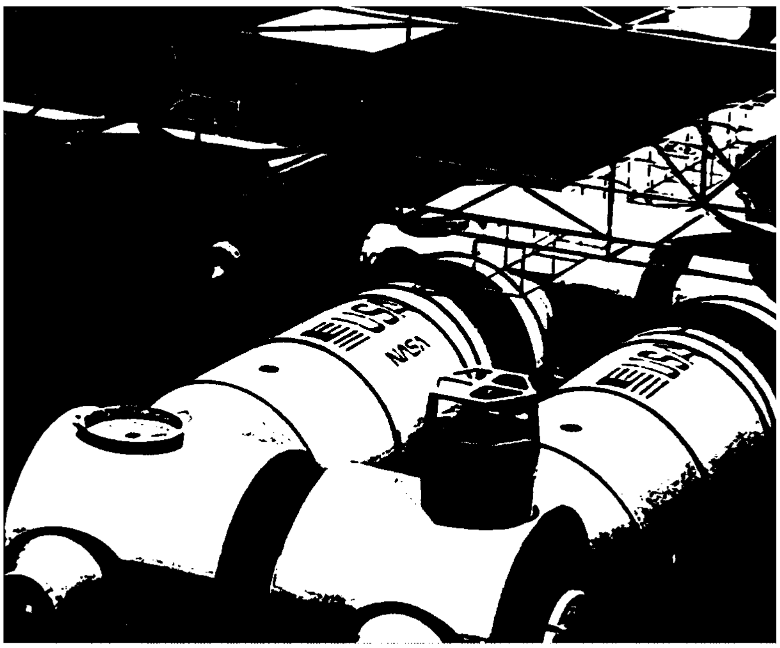
1. **T.J. Sejnowski & C.R. Rosenberg,** "NETtalk: A parallel network that learns to read aloud," Johns Hopkins University Department of Electrical Engineering and Computer Science Technical Report 86/01 (1986).
2. **P.K. Simpson,** 'A survey of Artificial Neural Systems," Naval Ocean Systems Center, Technical Document 1106 (June 1987).
3. **D.E. Rumelhart & J.L. McClelland,** *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press (1986).

ACKNOWLEDGEMENTS

The author would like to thank Paul Klimowitch of ARINC Research Corporation and Patrick Simpson of Verac Corporation for their valued contributions.

FOR MORE INFORMATION

Contact Stephen Luse, NOSC Code 441, San Diego, CA 92152. (619) 225-7372


USA
NASA
USA

END
DATE
FILMED
6-1988
DTIC