Fifth IEEE International Conference on Fuzzy Systems (VOL. 3)

Proceedings of the

Fifth IEEE International Conference on Fuzzy Systems

FUZZ-IEEE '96

September 8-11, 1996
Hyatt Regency Hotel
New Orleans, Louisiana

Sponsored by:

The IEEE Neural Networks Council

Proceedings of the Fifth International Conference on Fuzzy Systems (FUZZ-IEEE '96)

September 8-11, 1996 - New Orleans, Louisiana

IEEE Catalog Number 96CH35998
ISBN 0-7803-3645-3 (softbound)
ISBN 0-7803-3646-1 (casebound)
ISBN 0-7803-3647-X (microfiche)

Additional copies of this publication are available from:

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331

Phone: 1-800-678-IEEE

Greetings from the Conference Chair

On behalf of the Institute of Electrical and Electronic Engineers and the Neural Network Council, I am pleased to welcome you to the Fifth IEEE International Conference on Fuzzy Systems. I also extend you a warm welcome to the city of New Orleans. Bienvenue a la Nouvelle Orleans.

Our program will be described in more detail by the program chair, but I would like to recognize the strength and scope of the FUZZ-IEEE conferences. This year we have had submissions of over 440 papers from 33 countries representing every continent. The previous FUZZ-IEEE conference was in Japan and the next will be in Spain further illustrating the increasing international scope of this conference and the fuzzy set research community.

I must thank all the persons who have assisted me in so many ways in organizing this conference. In particular I wish to thank the outstanding organizing committee for this year's conference, especially the time and effort of the program chair, Don Kraft. I have received excellent advice from several previous chairs including Jim Bezdek, Enrique Ruspini and Piero Bonissone. Although I did not not always follow their advice, I generally found later that I wished I had!

I hope you will enjoy the technical program and the other activities at the conference, especially our special banquet at the Aquarium of the Americas. I also hope you will find time to enjoy the many aspects of New Orleans and its unique cultures; be sure to: **Laissez les bon ton rouler !**

Message from the Program Chair

We believe we have organized an outstanding program for the technical part of this year's FUZZ-IEEE conference. Over the three days of the conference there are five tracks of oral presentations totaling 51 separate regular sessions, 7 invited sessions and one special panel session. Additionally we have two outstanding plenary sessions and 6 poster sessions. There were a total of 443 papers received for the conference. After complete reviews and final evaluation by the program committee, 218 papers were scheduled for full oral presentation and 91 were additionally placed into poster sessions.

I would especially like to thank the program committee for their efforts in ensuring the quality of the conference. Additionally I would like to acknowledge numerous other referees who were so generous of their time in producing thoughtful reviews. Finally I must thank all the researchers whose papers submitted for presentation in the next three days have made this a successful conference for all participants.

Frederick E. Petry
General Chair FUZZ-IEEE '96

Donald H. Kraft
Program Chair FUZZ-IEEE '96

FIFTH IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS (FUZZ-IEEE ' 96)

Hyatt-Regency New Orleans, New Orleans Louisiana
September 8 - 11, 1996

Sponsored by the IEEE Neural Networks Council

GENERAL CHAIRPERSON: **Frederick E. Petry**
Tulane University

PROGRAM CHAIRPERSON: **Donald H. Kraft**
Louisiana State University

HONORARY CHAIR

Lotfi Zadeh

ADVISORY COMMITTEE

J. Bezdek
P. Bonissone
D. Dubois
G. Klir
H. Prade
E. Ruspini
E. Sanchez
P. Smets
M. Sugeno
T. Terano
E. Trillas
M. Zemankova
H. Zimmermann

TECHNICAL PROGRAM COMMITTEE

K. Asai
T. Fukuda
A. Kandel
S. Miyamoto
M. Roubens
I. Turksen
T. Sudkamp
H. Nguyen
T. Whalen
M. Mukaidono
M. Delgado
D. Ralescu
J. Baldwin
M.Gupta
J. Keller
C. Negoita
E. Triantaphyllou
M. Umano
Y. Liu
F. Gomide
R. Kruse
R. Lopez de Mantaras
K. Oh
E. Klement
B. Bouchon-Meunier
K. Hirota
L. Koczy
W. Pedrycz
P. Wang
B. Schott
Z. Bien
L. Hall
A. Rocha
J. Kacprzyk
T. Yamakawa
G. Bordogna
J. Buckley
Z. Feng
E. Mamdani
A. Ralescu
H. Tanaka
N. Green-Hall
E. Kerre
R. Yager
P. Bosc
S. Pal
G. Pasi
E. Walker

FUZZ-IEEE' 96 ORGANIZING COMMITEE

FINANCE : Sujeet Shenoi - University of Tulsa

PUBLICITY : Roy George & R. Srikanth - Clark-Atlanta University
Jim Keller - University of Missouri

EXHIBITS : Valerie Cross - Miami University of Ohio
V. Ganesh - Allied-Signal

PROCEEDINGS: Padmini Srinivasan - North Carolina State University

TUTORIALS : John Yen - Texas A&M University

STUDENT SUPPORT & AWARDS: Evangelos Triantaphyllou - Louisiana State University

ICNN 1996

From the Proceedings of the

International Conference on Neural Networks

June 3 - 6, 1996

SPECIAL SESSION AND PANEL PAPERS

TECHNICAL SESSION PAPER

Studies of Inference Rule Creation using LAPART

Thomas P. Caudell† and Michael J. Healy††

† *Interdisciplinary Computational Systems Group*
Department of Electrical and Computer Engineering
University of New Mexico
Albuquerque, N.M. 87131

†† *Boeing Information Services*
MS 7L-66
P.O. Box 24347
Seattle, WA

Abstract

The logical neural architecture LAPART is used in a mode that allows through learning the easy creation and extraction of IF-THEN inference rules from data. This paper first describes ART1 and the complement coded stack input binary representations. Next we present a more detailed discussion of LAPART. Then we show how rules are learned and extracted from the memory templates of the ART1s. We present a pedagogical example of rules extracted from a simple data set. Finally, we note that a fundamental difference between LAPART rule-based systems and regular rule-based systems is the existence of a "rule attractor" that can enhance system generalization in a controlled manner.

1.0 Introduction

Throughout the history of knowledge-based systems, the acquisition of "knowledge" or rules from humans and data sets has been one of the larger technical challenges. A number of methodologies have been developed in the field of Artificial Intelligence and Machine Learning to address this challenge. More recently, feedforward neural networks have been used to learn rules through training with the Backpropagation algorithm. The primary technical issue with is work involves the extraction of the rules from the feedforward weights in the network. Since all weights are typically used in the process of mapping an input space into an output space, the individual rules are not easily teased apart.

Other neural architectures are now being used for rule learning that drastically simplify the extraction process[1,2,3,4]. In this paper, we focus an architecture that is based on the self-organizing model of ART1[5] called LAPART. LAPART[1] is a neural network architecture that is composed of two ART modules set side-by-side and allowed to meddle in each other's workings. It is capable of clustering complex patterns and learning associations between these clusters in different domains.

The next Section briefly describes the algorithmic form of the ART1 neural networks which make up LAPART plus supporting concepts. Section 3 gives a more detailed description of LAPART. Section 4 discusses rule extraction and rule attractors, and illustrates the idea with an example. Section 5 concludes the paper.

2.0 The ART1 and Input Representations

In this Section, we discuss the ART1 algorithm, stack numeral representations of analog input values, complement coding of input patterns, and the interpretation of the learned ART1 templates as hyberboxes.

2.1 Algorithmic form of ART1

The ART1 is a <u>binary in / binary out</u> neural network model that is canonically represented by a coupled set of ordinary nonlinear differential equations[5]. If appropriate restrictions are made on the relationship between the dynamical time constants, the learning rates, and the length of time the input pattern is stable on the networks input nodes, then this system of equations reduces to a procedural algorithm[6]. The dynamical time constants are required to be much smaller than the learning rate, which is in turn much smaller than the stable presentation time. These restrictions lead to a "fast learning" mode of operation where the learned weights in the system reach asymptotic stability before a new input pattern is presented. The procedural algorithm autonomously places binary input patterns into clusters. Each cluster is represented in neural memory by an abstraction called a template prototype. A template is a logical conjunction of all the patterns in the cluster, and are formed and modified during the learning (training) process. If the bits in the binary input pattern represent logical predicates, then the template codes what is logically TRUE of all of the input patterns in the cluster. The number of clusters discovered by the network is determined in general by the underlying structure or semantics of the training set of input patterns, the order of presentation of input patterns, and a system level threshold called the vigilance factor. During testing of the trained network, neural memories or templates are directly recalled when examples from known clusters are presented. On the other hand, the learning process remains forever plastic in this model, even after training, allowing new or novel input patterns to be learned without disturbing old memories

2.2 Stack Interval Representations

A common criticism of binary networks is their inability to process analog inputs. This can be easily overcome by the appropriate choice of an input representation, such as binary coded decimal (BCD). In this sub-section we describe a representation call stack numerals[7] that has high utility in this regard. In particular, they represent analog numbers in binary patterns in a way relevant to ART1 clustering; patterns similar as measured by the ART1 metrics correspond to numbers which are also similar in magnitude. This is not true of the BCD format used in digital computers, in which 0 and 1 are coefficients of powers of 2.

A stack is defined as a binary vector with N components where an analog value is represented by the number of ON bits "stacked" consecutively from one end of the vector, with the remaining bits in the vector set to OFF. Another term for this is Thermometer Coding. The analog value is affine transformed to fit within a fixed minimum and maximum interval, defined as all OFFs and all ONs respectively. The precision with which the value is coded depends on N. For example, to code a minimum/maximum bounded value to one percent precision over this interval would require a stack with $N=100$. When an analog value is represented in this fashion, it is referred to as a stack numeral. Stack numerals are a form of course coding that address the precision/noise tradeoff directly. When a value is transformed into the interval and quantized into bits, signal smoothing automatically occurs that performs low pass filtering.

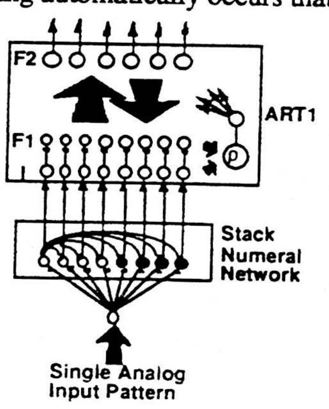

Figure 1. An eight component Stack Numeral representation of a single analog value being input to an ART1 neural network.

Two stack numeral binary patterns are similar in the ART1 sense when they fall into a cluster with the same template pattern. If the corresponding values were coded in BCD, this equivalence would not hold. For example, the values 122 and 128 represented in powers of two, with low-order binary digits to the right, yield the dissimilar patterns 01111010 and 10000000, respectively, even though they are in fact close in magnitude. By contrast, a stack numeral representation requires a minimum of 128 vector components to represent the numbers 0- 128, but the two stacks will be very close as judged by the ART1. Figure 1 illustrates a stack numeral input to an ART1 module. (A simple neural network has been devised which converts an analog value into stacks[7].) If the number of analog inputs m is greater than one, stack numeral representations of each analog value are concatenated together to form the composite input vector for the ART1.

2.3 Complement Coding

ART1 networks learn through repeated exposure to a set of training data. Georgiopoulos et al [8] proved that in an ART1 system processing N -bit patterns I of constant size $|I|$, template recoding, and therefore learning, will cease after the first pass through a fixed set of input patterns which are presented again, arbitrarily reordered, in subsequent passes. This property can be achieved for any arbitrary binary input pattern encoding through the use of "complement coding"[3,7]. Let Y denote a binary input pattern intended for ART1. In complement coding, the input pattern actually input to the ART system is compose of two parts concatenated together

$$I = (Y, Y^c) \quad ,$$

where

$$Y^c = 1 - Y \quad .$$

that is, the one's complement. Figure 2 illustrates complement coding by displaying the stack vectors and their complements next to each other.. Notice that the ART1 system now processes the set of binary patterns $\{Y_i\}$ with $2Nm$ total components, where the original input space consisted of N binary components and m analog values.

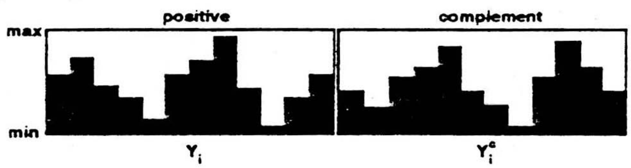

Figure 2. An illustration of a complement coded stack numeral representation of a set of binary values $\{Y_i\}$. Each analog component stack numeral is plotted as a black bar where the height is proportional to its magnitude. Note that the values must be normalized such that each component ranges between a max and min value. Taken together as one input vector with twice the components, this defines the complement code.

2.4 Cluster Templates as Hyberboxes.

During ART1 learning with complement coded stack numeral input representations, each cluster template adjusts to code the minimum and maximum analog values of all the patterns that are associated with its cluster. This is the result of the conjunctive nature of template learning in ART1. The minimum portion is coded in the positive part of the template, while the maximum is found in the complemented. The template minimums and maximums for each variable can be extracted out of their respective portions of the composite learned templates. As illustrated for a 2D case in Figure 3, the minimum and maximum points, taken as coordinates, form boxes in the plane. For higher numbers $m > 2$ of variables, these become hyperboxes. The hyperboxes form around clusters of points that are similar as judged by the ART1 network [3,7].

Because of the use of complement coding, ART1 and LAPART learning converges very rapidly, usually in a single pass through the training data. After learning has stabilized, learning can be "turned off" and the system used as a logical inferencer using the learned rules based on associations of the hyperboxes in different domains.

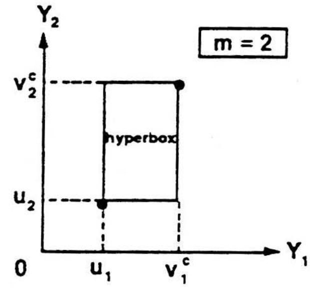

Figure 3. Hyperboxes formed by ART1 templates when the 2D analog input patterns were complement coded stacks. The labels indicate the min and max analog values decoded from the composite template for a single cluster. There exists one such hyperbox for each learned template or cluster. These boxes begin to tessellate the input feature space and are the bases for learned rules.

3.0 LAPART Architecture

LAPART[1] is a neural network architecture that is composed of two ART1 modules set side-by-side and allowed to meddle in each other's workings. It is capable of clustering complex patterns and learning the associations between learned clusters in different domains. LAPART is similar to ARTMAP in many ways[2]. This architecture defines a class of networks that can be trained to associate classes of patterns appearing on the inputs of each of its constituent self-organizing modules. The basis for the inference function of the LAPART architecture is the coupling of two pattern organization networks through a system of lateral interconnects, or a map field[1,2]. The interconnects implement a dual system of inference rules. (see Fig. 4)

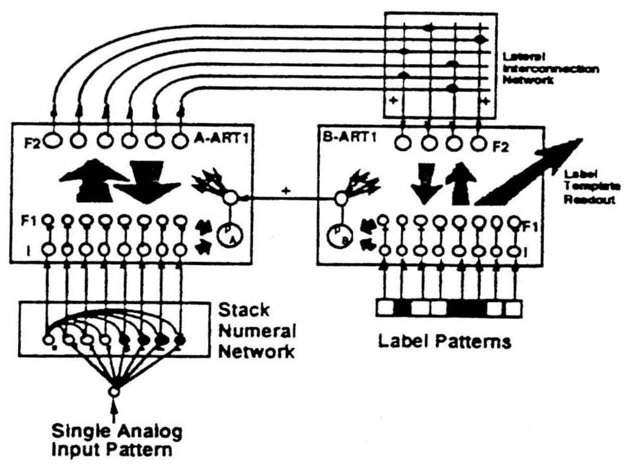

Figure 4. A diagram showing one instance of a LAPART, where a stack numeral is input to the A module and a class label is input to the B module. The lateral interconnections are shown above the B module.

The LAPART system learns the inferences during presentation of training pairs of patterns, through 1) ART pattern clustering, involving synaptic learning within each self-organizing network, and 2) synaptic learning of the class-to-class inferences through the lateral interconnects. A LAPART system can remain adaptive following training, continuing to learn from observed patterns. Distinctly novel inputs are automatically classified separately from those encountered in the past. Loosely speaking, the network "knows" when it has encountered a pattern association that lies outside its trained generalization capability. LAPART has many potential applications, including pattern recognition, function approximation, and explicit learning of rules.

4.0 Rule Extraction and Rule Attractors

A rule consists of a antecedent and a consequence. For example, "If A_1=True and A_2=True then C_1=True", where the logical expressions "A_1=True and A_2=True" is the antecedent and "C_1" is the consequence. Using LAPART, we can learn such rules by representing a combined representation of variables A_1 and A_2 to the A-ART1 module, and representing C_1 to the B-ART1 module[4]. If complement coded stacks are used for input representations, as discussed earlier, each ART1 will separately learn hyperbox templates, and the lateral interconnects will learn a association between them. That is, each hyperbox in the A-ART1 will imply a hyperbox in the B-ART1. After learning, when a new input pattern is clustered into an A-ART1 hyperbox, the lateral connections from it will signal the B-ART1 to readout its associated hyperbox. This constitutes an inference and therefore represents rules.

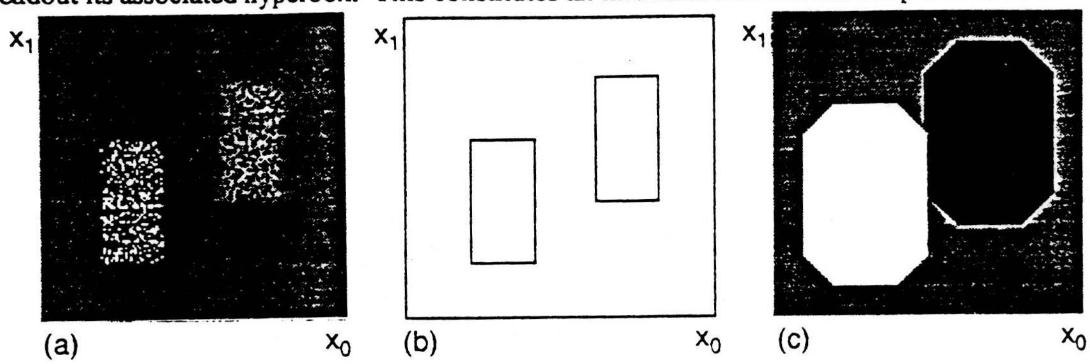

Figure 5. a) A scatter plot of data in two classes. b) the hyperboxes learned for the rules, c) the rules attractors, white=Class 1, black=Class2, gray=Unknown..

For an illustrative example, consider the data for a two class pattern recognition problem seen plotted in an m=2 dimensional analog input space in Fig. 5a. Fig. 5b plots the learned hyperboxes and the lateral interconnections to the two class labels. Figure 5c plots the "rule attractor" for this problem. The attractor is defined as the volume of input feature space around the hyperbox in which a new input pattern will be associate, in an ART1 sense, with the cluster. Note that this is a distinctly larger volume than would be found for a conventional rule-based system, which requires a point be within the hyperbox to be considered TRUE.

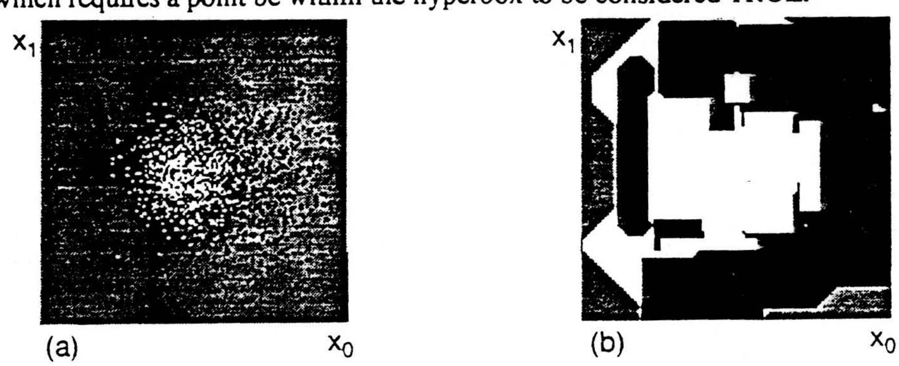

Figure 6. a) Two overlapping normal distributions with different variances, b) a set of rule attractors for the two classes, with ART-A vigilance =0.7, white=Class 1, black=Class2, gray=Unknown.

As a second example, LAPART was trained on 2000 samples drawn from two normal distributions with means of (0,0) and (1,0), and variances of 1 and 4 respectively. The distributions are seen in Fig 6a. After training, the network was tested on 32,000 points to determine classification error. Using a grid search in vigilance during training, the highest performance was found to be 78% at a vigilance of 0.53, to be compared to the Bayesian optimal of 82% [9]. A composite of rule attractors for vigilance of 0.7 is shown in Fig 6b. Again, the basins of attraction for the rules can extend beyond their hyperboxes, effecting the generalization power of the network.

5.0 Conclusions

With the use of ART1 modules and complement coded stack numerals, LAPART provides a rapid rule learner for symbolic (logical), analog, or combined data sets. In addition, the extraction of the rules is a simple as decoding the learned templates. The antecedents and consequences of the rules are represented as hyperboxes in the input and output feature spaces. The rules can attract test patterns to themselves from regions outside the learned hyberboxes called rule attractors. Conventional rule-based systems "fire" a rule only if a test pattern falls within the hyperbox. The general effect of neural rule attractors on generalization when compared to conventional rule-based systems is not known at this time. It is expected there will be important differences in the two approaches. Still, the LAPART system is being used in real-world applications of target recognition, process monitoring, and weather parameter prediction[10]. Studies continue to understand the implications of the rule attractors to the general inferencing problem.

6.0 Acknowledgments

The first author would like to thank Kathryn Chalfan of the Boeing Company for continued support of this work, and Dr. Pete Soliz for promoting the application of this technology.

7.0 References

[1] Healy, M. J., Caudell, T. P., and Smith S. D. G. "A neural architecture for pattern sequence verification through inferencing", IEEE Transactions on Neural Networks, Jan, 1993.

[2] Carpenter, G., Grossberg, S., and Reynolds, J., "ARTMAP: Supervised Real-Time Learning and Classification of Nonstationary Data by a Self-Organizing Neural Network", Neural Networks, Vol. 4, pp. 565-588, 1991.

[3] G. A. Carpenter, S. Grossberg, D. B. Rosen, "Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System", Neural Networks 4, No. 6, pp. 759-772 (1991).

[4] M. Healy and T. P. Caudell, "Rule Extraction in the Logical Neural Architecture LAPART", in preparation (1995).

[5] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," in Computer Vision, Graphics, and Image Processing, No. 37, (Academic Press, 1987), pp. 54-115.

[6] B. Moore, "ART1 and Pattern Clustering," Proceedings of the 1988 Connectionist Summer School , Touretzky and Hinton, ed., (Morgan Kaufman, Carnegie-Mellon University, 1989).

[7] M. J. Healy and T. P. Caudell, "Stack Representations and Fuzzy ART1", Proceedings of the World Congress on Neural Networks, Portland, June, 1993

[8] M. Georgiopoulos, G. L. Heileman, and J. Huang, "Properties of Learning Related to Pattern Diversity in ART1", Neural Networks 4, No. 6, pp. 751-758 (1991).

[9] "Neural Networks, a Comprehensive Foundation", S. Haykin, Macmillan College Publishing, NY, pp. 165-176 (1994).

[10] P. Soliz and T. P. Caudell, "Application of Artificial Neural Networks to Battlefield Atmospherics", Proceedings of the White Sands Conference on Battlefield Atmospherics (Dec 1995).

NEURAL NETWORKS AND EMERGING TECHNOLOGIES

Gail Erten
IC Tech. Inc.
ictech@mcimail.com

Mary Lou Padgett
Auburn University
mpadgett@auburn.edu

Fathi M. Salam
Michigan State University
salam@egr.msu.edu

Participants

This panel was organized by the authors and chaired by Dr. Gail Erten. The panelists are listed below in alphabetical order:

H. P. Graf, *AT&T Bell Labs*
Thomas McKenna, *The Office of Naval Research*
Robert W. Newcomb, *University of Maryland*
Mary Lou Padgett, *Auburn University*
Benjamin Wah, *University of Illinois*

Overview

Neural networks and related soft computing algorithms and architectures represent the next step in the evolution of information processing. Some of us involved in this field are trying to put eyes and ears on computers so they may not only look but see, not only hear but listen. Some are trying to have computers learn to do things such as walking or talking simply by observing how it is done. Some are testing their performance with seemingly impossible manipulations and modeling poorly understood systems. Neural networks, models of the brain, procedures that define human reasoning are natural sources of inspiration and enlightenment towards these goals.

Two categories of approaches define the domains of implementation. One approach is to apply the concept of conventional processing to emulate neural algorithms, perhaps through interlinked and concurrent processors. Another is to create networks of experimental computer chips, called silicon neurons, that mimic data-processing functions of brain cells.

The year 2000 is right around the corner and represents an inspection point at which our world will be judged by many future generations. The threshold of the third millennium is likely to become such an instance in the history of computing, as well. Several emerging technologies pertaining to advances in materials, automotive and transportation systems, information, communication, and computation, as well as manufacturing, are opening new frontiers in science and technology which demand novel computing paradigms. The role of neural networks in redefining and expanding computing, intelligent processing, and multimedia integration are explored by this panel. The goal of the forum is to present an overview or a snapshot of examples of the status and role of neural networks and related fields. Panelists with broad backgrounds in the field have been invited to discuss directions and aims as well as to present their latest views.

Dr. Hans Peter Graf

Recognition is one of the key technologies driving advances in many industrial applications. For example, image recognition is now an integral part in document processing, inspection of manufactured parts and surveillance. Moreover such important technologies as image compression and computer-user interfaces depend more and more on image recognition. Neural nets have become a widely accepted approach for recognition tasks. Typically, neural nets do not provide complete solution for a problem, yet they represent a powerful suit of algorithms that can be used in combination with other, 'conventional', algorithms. Key for a successful application of recognition techniques is (still) to find good representations that the algorithms, neural or others, will find proper solutions.

Dr. Thomas McKenna

If neural technology is to evolve beyond a tool for pattern recognition, it must incorporate and exploit advances in related fields. Examples of such emerging multidisciplinary approaches will be provided for computational neuroscience, mechanics, and material failure analysis, hydrodynamics, autonomous vehicle control, chemical engineering, and molecular genetics.

Professor Robert W. Newcomb

Hardware realization of neural networks is important since it represents a practical means to implement real time signal processing with artificial neural networks. Discussion will center around significant research areas where future progress can be foreseen. These include multiple valued processors, adaptive-adjoint methods of implementing back propagation, biologically motivated ANNs based upon D. K. Hartline's models, Kemp echo based neural type speech recognition systems, and the design of chips which include live neuron signal processors.

Dr. Mary Lou Padgett

Neural networks research should be directed into areas which can help integrate neural networks with supportive intelligent system capabilities. Exploring the full use of fuzzy systems and genetic/evolutionary systems in a virtual environment should provide ideas for the implementation of intelligent neural networks development environments. Providing a systematic way to track the heuristics and hunches that are part of model development can help to formalize, repeat and extend valid ideas. It can also help check assumptions and reveal unexpected or erroneous responses to new algorithms or ideas. Working on the statistical theory and stability theorems to facilitate meshing neural and fuzzy systems should pay dividends.

In addition to development of intelligent working environments for neural systems researchers, new research endeavors should be encouraged in the integration of fuzzy and genetic components into working neural systems. Internal incorporation of these strengths is an asset. So is linking neural systems to the external system and its overall goals by using fuzzy and evolutionary / genetic systems. Areas of concern include: hardware capable of such integration, stability and signal to noise ratio in multi-modular or complex systems. There is much to be gained from exploring these avenues.

Professor Benjamin Wah

There are two important problems in the hardware support of neural network learning: (a) parallel processing and VLSI designs to support learning and application of neural networks, and (b) hardware/software supports for generating/emulating an environment necessary for learning. The first problem has been studied extensively by many researchers, but the latter is a less glamorous problem that has been largely overlooked. A related question to be addressed is whether the environment generated for learning is realistic or not.

For instance, to design a neural network to control jet engine fire, it may be necessary to create an environment in which a jet engine can catch fire, and the neural network can be tested to put out the fire. In addition, the color of the fire may indicate the type of fire, which may need to be considered in the design. Creating such a testbed, either in hardware or in software, is a nontrivial task. As another application, it may seem easy to design a neural network to load balance jobs in a network of workstations. However, creating an environment in which realistic multiprocessing workload can be repeated in the network of workstations is nontrivial. In general, a lot of the research and applications of neural networks is hampered by the difficulty of creating a realistic environment for learning and testing. Emerging technologies will help in the development of faster and more robust learning algorithms but may not simplify the design of the learning environment.

Biographic Sketches of Panelists

Hans Peter Graf, a Distinguished Member of Technical Staff at AT&T Bell Laboratories in Holmdel, NJ, is conducting research on image recognition and on the implementation of machine vision algorithms on massively parallel processors. Since 1984 he has been working on neural net models, designing micro-electronic processors and leading the implementation of vision systems for industrial applications. Among Dr. Graf's theoretical work are algorithms for the hierarchical decomposition of complex images into elementary shapes. These algorithms are being used for such applications as analyzing bank checks or finding the location and identity of people in complex images. Mr. Graf received a Diploma in physics in 1976 and a Ph.D. in physics in 1981, both from the Swiss Federal Institute of Technology in Zurich, Switzerland. He is a Fellow of the IEEE and a member of the American Physical Society. He is author and co-author of more than 90 articles on image recognition and neural networks, acted as guest editor for IEEE Micro and for the J. VLSI Signal Processing and is associate editor of IEEE CAS.

Thomas M. McKenna received his B.S. degree from Massachusetts Institute of Technology (Biology) and his Ph.D. from the University of North Carolina at Chapel Hill (Physiology-Neurophisiology) in 1979. From 1971 to 1975 he was employed at the Harvard Medical School where he conducted research on primate cortico-cortical connections, and neuropharmacological electrophysiological investigations of brain stem control in sleep states. His dissertation research and subsequent post doctoral research at the University of North Carolina were concerned with the functional organization of the somatosensory cortex, and stochastic neuron models. From 1982 to 1988 he was Assistant Research Neurobiologist in the department of Physics and the Center for the Neurobiology of Learning and Memory at the University of California - Irvine. During this period he conducted experimental research on neuronal coding in the auditory cortex and theoretical neurobiology on biological neural nets and single neuron computations. Dr. McKenna is currently a Program Officer in the Division of Cognitive and Neural Science and Technology, of the Office of Naval Research and is responsible for the development, funding, and management of basic research programs in neural computation, computational neuroscience, nonlinear dynamics of neural systems, hybrid architectures combining neural nets with physical models for mechanical and hydrodynamic systems, computational vision, and sensory-motor systems, including control of legged robots. These programs produce novel computational architectures and VLSI electronic implementations of neural systems. He has developed a major new thrust in neural networks for mechanical diagnosis based on vibration signals. He serves as a member of the overall management of the Accelerated Capability Initiative in Condition Based Maintenance and is team leader in the Machinery Diagnostics and Prognostics thrust. He also served as primary agent for the ARPA program in artificial neural networks. In that capacity he managed research designed to evaluate the comparative performance of artificial neural networks on classification of sonar signals and machine vision.

Robert W. Newcomb was raised in Southern California where he was born in June 1933. He obtained the BSEE from Purdue in 1955, the MS from Stanford in 1957, and the Ph.D. from Berkeley in 1960 while on the teaching faculty. He later joined the tenured faculty of Stanford and then the University of Maryland to establish the graduate program in electrical engineering and where he now directs the Microsystems Laboratory devoted to analog, biomedical, and VLSI electronic circuit theory. His recent research includes that of modeling live neurons on a chip and a scattering theory for Kemp echoes for noninvasive study of the inner ear. Associated with this latter the Robert Wayne Newcomb Oral Communication Laboratory has been named in his honor in Madrid, Spain. He is most proud of his students who are spread and known world wide.

Mary Lou Padgett has a broad-based background in academia and in industry. She has held teaching and research positions in nine different departments of Auburn University over the past twenty-six years. She has consulted with aerospace industry, engineering consortia and government laboratories since 1982. Currently working as a Senior Research Associate of Electrical Engineering at Auburn University, she draws upon prior experience in quantitative genetics, topology and analysis, computer simulation and engineering statistics, decision theory, biomedicine and neural anatomy, and graphics design for commercial applications. Supplementing these academic and industrial experiences, she interacts with a number of professional society groups in order to facilitate distribution of new information among her colleagues. Padgett is a member of the IEEE Standards Board, IEEE-SB Liaison to the IEEE-Educational Activities Board for 1996, and a member of the NNC Advisory Committee. She is a past Vice President for North America of the Society for Computer Simulation, International, and has edited numerous conference proceedings in the Computational Intelligence arena. She is a member of Eta Kappa Nu, Tau Beta Pi and Phi Kappa Phi.

Benjamin W. Wah received his Ph.D. degree in computer science from the University of California, Berkeley, CA, in 1979. He is currently a Professor in the Department of Electrical and Computer Engineering and the Coordinated Science Laboratory of the University of Illinois at Urbana-Champaign, Urbana, IL. Previously, he had served on the faculty of Purdue University (1979-85), as a Program Director at the National Science Foundation (1988-89), as Fujitsu Visiting Chair Professor of Intelligence Engineering, University of Tokyo (1992), and McKay Visiting Professor of Electrical Engineering and Computer Science, University of California, Berkeley (1994) He is currently serving in the IEEE Computer Society as a member of its Governing Board, Publications Board, and Press Activities Board, and as a vice-chair of its Fellows Committee. He is a Fellow of the IEEE. Dr. Wah's current research interests are in the areas of parallel and distributed processing, knowledge engineering, neural networks and nonlinear optimization. In the area of neural networks, he is interested in the design of hardware/software systems to support learning and the design of parallel learning algorithms.