Letters

A Parallel Algorithm for Tiling Problems

YOSHIYASU TAKEFUJI AND KUO-CHUN LEE

Abstract—**A parallel algorithm for tiling with polyominoes is presented in this paper. The tiling problem is to pack polyominoes in a finite checkerboard. The algorithm using $l \times m \times n$ processing elements requires O(1) time where l is the number of different kinds of polyominoes on an $m \times n$ checkerboard. The algorithm can be used for placement of components or cells in a very large scale integrated circuit (VLSI) chip, designing and compacting printing circuit boards, and solving a variety of 2-D or 3-D packing problems.**

Introduction

The problem of tiling with polyominoes was introduced by Golomb in 1953 [1]. The tiling problem is to pack a checkerboard with polyominoes. Although Klarner [2], Golomb [3], Gardner [4], Klamkin and Liu [5], Golomb [6], and others have been working on the problem, few sequential algorithms have been reported. No parallel algorithm has been given during the last three decades. Akiyama *et al.* [7] have proposed the first parallel algorithm based on the stochastic neural network using noise to escape from the local minimum. However, the quality of the solution drastically degrades with the problem size although the experimented problem size is small. The complexity of solving their 5×5 checkerboard problem is to find a single solution among 3.3×10^6 possible candidates.

This paper introduces the new deterministic parallel tiling algorithm. The algorithm packs a checkerboard with polyominoes within O(1) time. The quality of the solution does not degrade with the problem size. The algorithm was verified by more than 1000 simulation runs solving a 7×7 checkerboard tiling problem with 10 polyominoes. Without rotation or reflection, the complexity of our problem is to find a single solution among 1.3×10^{14} ($\approx 25^6 \times 36 \times 30 \times 24 \times 21$) possible candidates where there exists one and only one solution in the problem.

The algorithm uses a three dimensional $10 \times 7 \times 7$ neural network array where the output of the ijkth neuron follows:

$$V_{ijk} = \begin{cases} 1 & \text{if } U_{ijk} > 0 \text{ and } U_{ijk} = \max(U_{ixy}) \\ & \text{for } x = 1, \cdots, 7 \text{ and } y = 1, \cdots, 7 \\ 0 & \text{otherwise.} \end{cases}$$

Fig. 1 shows ten polyominoes and their markers. A marker is used to locate a polyominoe on the checkerboard. A polyominoe requires a two-dimensional 7×7 neural network array so that this problem can be solved by the three-dimensional $10 \times 7 \times 7$ neural network array as shown in Fig. 2. One and only one marker neuron is forced to fire per 7×7 neural network array. Besides a 7×7 neural network array each polyominoe needs a tiling sideboard to represent its shape and violation conditions. In other words, the state of the sideboard indicates occupied tiles of a polyominoe on a checkerboard where the sideboard is composed of a 7×7 binary array where V'_{ijk} indicates the state of the (j, k) coordinate on the ith sideboard. The motion equation of the ijk neuron is given by:

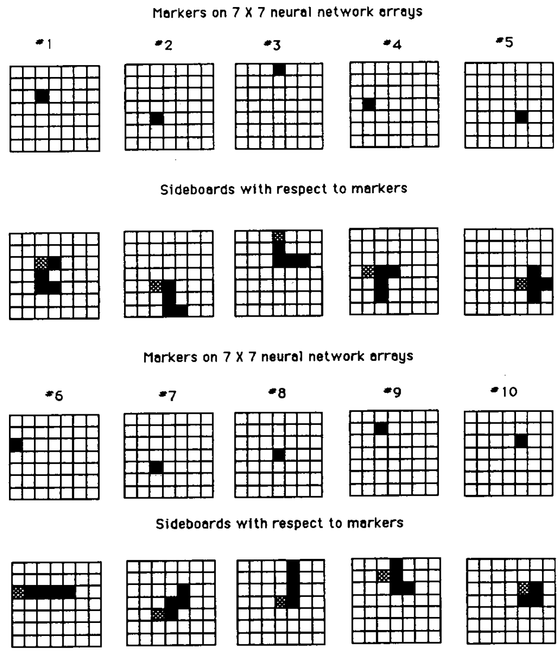

Fig. 1. Ten polyominoes and their markers.

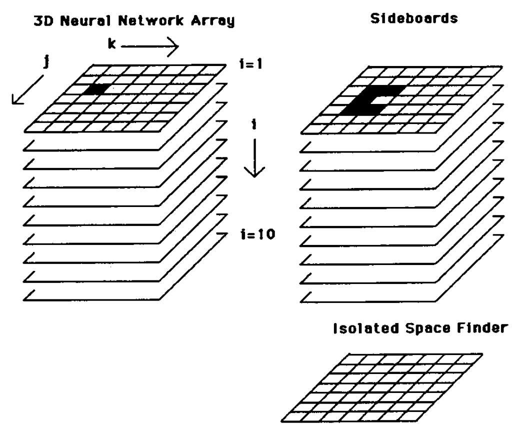

Fig. 2. Neural representation for the 7×7 tiling problem.

$$\frac{dU_{ijk}}{dt} = -A\left(\sum_{q=1}^{m}\sum_{r=1}^{n} V_{iqr} - 1\right) - B\left(\sum_{q=1}^{l} V_{qjk} - 1\right) - Cf(i) + Dh\left(\sum_{q=1}^{m}\sum_{r=1}^{n} V_{iqr}\right) \quad (2)$$

Manuscript received September 22, 1989; revised October 26, 1989.
The authors are with the Department of Electrical Engineering and Applied Physics, Center for Automation and Intelligent Systems Research, Case Western Reserve University, Cleveland, OH 44106.
IEEE Log Number 8933092.

1045-9227/90/0300-0143$01.00 © 1990 IEEE

where U_{ijk} is the input of the jkth neuron on the ith neural network array. In (2) the first term forces one and only one neuron to fire in the ith polyominoe neural network array. The second term forces no two markers to be placed in the same position on the checkerboard. The third term is always inhibitory which describes the overlap violation between polyominoes where the ten violation functions for ten polyominoes: $f(1)$ through $f(10)$ are given by, respectively:

$$f(m) = \sum_{\substack{q=1\\ q\neq m}}^{11} (V'_{qjk} + w(m))$$

$$w(1) = (V'_{q,j+1,k} + V'_{q,j,k+1} + V'_{q,j,k+2} + V'_{q,j+1,k+2})$$

$$w(2) = (V'_{q,j+1,k} + V'_{q,j+1,k+1} + V'_{q,j+1,k+2} + V'_{q,j+2,k+2})$$

$$w(3) = (V'_{q,j,k+1} + V'_{q,j,k+2} + V'_{q,j+1,k+2} + V'_{q,j+2,k+2})$$

$$w(4) = (V'_{q,j+1,k} + V'_{q,j+2,k} + V'_{q,j+1,k+1} + V'_{q,j+1,k+2})$$

$$w(5) = (V'_{q,j+1,k} + V'_{q,j+1,k-1} + V'_{q,j+1,k+1} + V'_{q,j+2,k})$$

$$w(6) = (V'_{q,j+1,k} + V'_{q,j+2,k} + V'_{q,j+3,k} + V'_{q,j+4,k})$$

$$w(7) = (V'_{q,j+1,k} + V'_{q,j+1,k-1} + V'_{q,j+2,k-1} + V'_{q,j+2,k-2})$$

$$w(8) = (V'_{q,j+1,k} + V'_{q,j+1,k-1} + V'_{q,j+1,k-2} + V'_{q,j+1,k-3})$$

$$w(9) = (V'_{q,j+1,k} + V'_{q,j+1,k-1} + V'_{q,j+1,k+1} + V'_{q,j+2,k+1})$$

$$w(10) = (V'_{q,j+1,k} + V'_{q,j,k+1} + V'_{q,j+1,k+1}).$$

Another extra sideboard #11 is used to destroy the isolated space on a checkerboard. Set $V'_{11j-1,k} = V'_{11j+1,k} = V'_{11j,k-1} = V'_{11j,k+1} = 1$ if the isolated space is found. The condition of the isolated space is given by:

$$\sum_{q=1}^{10} V'_{qjk} = 0, \quad \sum_{q=1}^{10} V'_{qj-1,k} = 1, \quad \sum_{q=1}^{10} V'_{qj+1,k} = 1$$

$$\sum_{q=1}^{10} V'_{qjk-1} = 1, \text{ and } \sum_{q=1}^{10} V'_{qjk+1} = 1.$$

The third term $-Cf(i)$ in (2) and $-Cf(i)V_{ijk}$ are alternatively used to converge to the global minimum. Related values of V'_{iqr} are set to one if the marker neuron V_{ijk} on the ith neural network array is fired where q and r are depending on the polyominoe's shape. For example, set $V'_{1jk} = V'_{1,j+1,k} = V'_{1,j,k+1} = V'_{1,j,k+2} = V'_{1,j+1,k+2} = 1$ if $V_{1jk} = 1$ for the #1 polyominoe in Fig. 1. The last term provides hill-climbing which allows the state of the system to escape from the local minimum and to converge to the global minimum. The function $h(x)$ is 1 if $x = 0$, 0 otherwise.

Parallel Algorithm for Tiling Problems

The following procedure describes the proposed algorithm based on the first-order Euler method.

0) Set $t = 0$ and $A = B = C = D = 1$.

1) The initial values of $U_{ijk}(t)$ for $i = 1, \cdots, 10$ $j = 1, \cdots, 7$ $k = 1, \cdots, 7$ are randomized.

2) Evaluate values of $V_{ijk}(t)$ based on the conditional binary function:

$$V_{ijk}(t) = 1 \text{ if } U_{ijk}(t) > 0 \text{ and } U_{ijk}(t) = \max\{U_{ixy}(t)\} \text{ for } x = 1, \cdots, 7 \text{ and } y = 1, \cdots, 7$$
$$= 0 \quad \text{otherwise.}$$

3) Values of $V'_{ijk}(t)$ are set to zero.

4) Set appropriate values of $V'_{ijk}(t)$ to one if the mark neuron is fired:

$$V'_{1jk} = V'_{1,j+1,k} = V'_{1,j,k+1} = V'_{1,j,k+2} = V'_{1,j+1,k+2} = 1 \text{ if } V_{1jk} = 1$$

$$V'_{2jk} = V'_{2,j+1,k} = V'_{2,j+1,k+1} = V'_{2,j+1,k+2} = V'_{2,j+2,k+2} = 1 \text{ if } V_{2jk} = 1$$

$$V'_{3jk} = V'_{3,j,k+1} = V'_{3,j,k+2} = V'_{3,j+1,k+2} = V'_{3,j+2,k+2} = 1 \text{ if } V_{3jk} = 1$$

$$V'_{4jk} = V'_{4,j+1,k} = V'_{4,j+2,k} = V'_{4,j+1,k+1} = V'_{4,j+1,k+2} = 1 \text{ if } V_{4jk} = 1$$

$$V'_{5jk} = V'_{5,j+1,k} = V'_{5,j+1,k-1} = V'_{5,j+1,k+1} = V'_{5,j+2,k} = 1 \text{ if } V_{5jk} = 1$$

$$V'_{6jk} = V'_{6,j+1,k} = V'_{6,j+2,k} = V'_{6,j+3,k} = V'_{6,j+4,k} = 1 \text{ if } V_{6jk} = 1$$

$$V'_{7jk} = V'_{7,j+1,k} = V'_{7,j+1,k-1} = V'_{7,j+2,k-1} = V'_{7,j+2,k-2} = 1 \text{ if } V_{7jk} = 1$$

$$V'_{8jk} = V'_{8,j+1,k} = V'_{8,j+1,k-1} = V'_{8,j+1,k-2} = V'_{8,j+1,k-3} = 1 \text{ if } V_{8jk} = 1$$

$$V'_{9jk} = V'_{9,j+1,k} = V'_{9,j+1,k-1} = V'_{9,j+1,k+1} = V'_{9,j+2,k+1} = 1 \text{ if } V_{9jk} = 1$$

$$V'_{10jk} = V'_{10,j+1,k} = V'_{10,j,k+1} = V'_{10,j+1,k+1} = 1 \text{ if } V_{10jk} = 1.$$

5) Set

$$V'_{11jk} = 1 \text{ if } \sum_{q=1}^{10} V'_{qjk} = 0, \quad \sum_{q=1}^{10} V'_{qj-1,k} = 1, \quad \sum_{q=1}^{10} V'_{qj+1,k} = 1, \quad \sum_{q=1}^{10} V'_{qjk-1} = 1, \text{ and } \sum_{q=1}^{10} V'_{qjk+1} = 1.$$

6) Use the motion equation in (2) to compute $\Delta U_{ijk}(t)$. If $(t \bmod 10) < \omega$ then:

$$\Delta U_{ijk}(t) = -A\left(\sum_{q=1}^{m}\sum_{r=1}^{n} V_{iqr}(t) - 1\right) - B\left(\sum_{q=1}^{1} V_{qjk}(t) - 1\right) - Cf(i)\,V_{ijk}(t) + Dh\left(\sum_{q=1}^{m}\sum_{r=1}^{n} V_{iqr}(t)\right)$$

else

$$\Delta U_{ijk}(t) = -A\left(\sum_{q=1}^{m}\sum_{r=1}^{n} V_{iqr}(t) - 1\right) - B\left(\sum_{q=1}^{1} V_{qjk}(t) - 1\right) - Cf(i) + Dh\left(\sum_{q=1}^{m}\sum_{r=1}^{n} V_{iqr}(t)\right).$$

7) Compute $U_{ijk}(t+1)$ based on the first-order Euler method:

$$U_{ijk}(t+1) = U_{ijk}(t) + \Delta U_{ijk}(t)$$
$$i = 1, \cdots, 10, \quad j = 1, \cdots, 7, \quad k = 1, \cdots, 7.$$

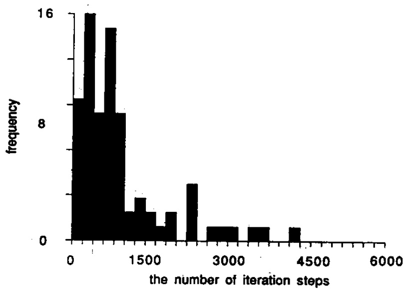

Fig. 3. The relationship between the frequency and the number of iteration steps to converge to the global minimum.

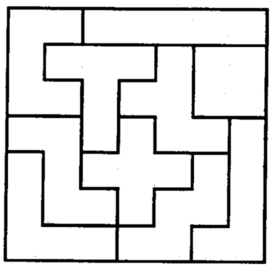

Fig. 4. The solution for a 7 × 7 checkerboard tiling problem.

TABLE I
FREQUENCY OF CONVERGENCE TO THE SOLUTION

ω	No. of placed polyominoes	
	<10	10
5	55%	45%
6	21%	79%
7	0%	100%
8	3%	97%
9	No Convergence	

8) Increment t by 1. If $t = T$ terminate this procedure, or go to step 2.

More than 1000 simulation runs were performed to observe the frequency of the system to converge to the global minimum with varying ω as shown in procedure 6. Table I shows the result. When $\omega = 7$ was used, the state of the system always converged to the global minimum within 5000 iteration steps. The relationship between the frequency and the number of iteration steps to converge to the global minimum was observed and the result is shown in Fig. 3. The average number of iteration steps for the proposed system to solve the 7 × 7 checkerboard tiling problem without rotation and reflection is 930. Fig. 4 shows one and only one solution of the problem without rotation and reflection.

REFERENCES

[1] M. Gardner, "Mathematical games," *Sci. Amer.*, Aug. 1975.
[2] D. A. Klarner, "Packing a rectangle with congruent N-ominoes," *J. Combinat. Theory*, vol. 7, pp. 107–115, 1969.
[3] S. W. Golomb, "Tiling with sets of polyominoes," *J. Combinat. Theory*, vol. 9, pp. 60–71, 1970.
[4] M. Gardner, "Mathematical games," *Sci. Amer.*, vol. 217, pp. 115–116, July 1967.
[5] M. S. Klamkin and A. Liu, "Polyominoes on the infinite checkerboard," *J. Combinat. Theory*, vol. A 28, pp. 7–16, 1980.
[6] S. W. Golomb, "Polyominoes which tile rectangles," *J. Combinat. Theory*, vol. A 51, pp. 117–124, 1989.
[7] Y. Akiyama, A. Yamashita, M. Kajiura, and H. Aiso, "Combinatorial optimization with Gaussian machines," in *Proc. Int. Joint Conf. Neural Networks*, 1989.

Neural Networks in Communication

EDWARD C. POSNER, FELLOW, IEEE

***Abstract*—This letter summarizes material from eight papers on neural networks from the November 1989 special issue of the IEEE COMMUNICATIONS MAGAZINE.**

INTRODUCTION

The "Guest Editorial" reprinted below is from the November 1989 IEEE COMMUNICATIONS MAGAZINE,[1] a special issue on neural networks in communication. The function of Guest Editor was assumed by me. The appearance of this and other special issues and special sections on neural networks in various existing IEEE publications in the last several years attests to the importance of neural networks to the interests of the IEEE. As the editorial points out, communications is a natural field of application for neural networks. Although the eight papers are briefly summarized in the editorial, the entire issue itself is worth having if you are interested in neural networks but are not a Communications Society member. Perhaps the editorial will encourage you to dig deeper into the November issue.

GUEST EDITORIAL

This timely special issue of the IEEE COMMUNICATIONS MAGAZINE[1] is devoted to neural networks in communication. We all feel we know very well what communication is. One of this special issue's goals is to make communications engineers aware of what neural networks are and how they could be applied to communication systems in the future. We will also see how ideas from communication, specifically information theory, can help elucidate what is going on in neural networks.

First, a good temporary working definition to adopt before getting deeply into this special issue is to think of a neural network as a richly connected set of interacting devices that produces and distributes outputs based on simple functions of their inputs, such as mildly distorted sums of products and simple combinations of sums of products. Even a correlator can be considered a very special neural network, although the power of neural networks resides in

Manuscript received October 17, 1989.
The author is with the Department of Electrical Engineering and Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91125.
IEEE Log Number 8932985.
[1] *IEEE Commun. Mag.*, vol. 27, no. 10, Nov. 1989.

1045-9227/90/0300-0145$01.00 © 1990 IEEE

their decision-making capabilities. In fact, the individual components (i.e., synthetic neurons) are by themselves too simple, special, and nongeneral-programmable to be called computers. The computation a neural network does is often best thought of as residing in the connections and their strengths, rather than in the functions performed by the individual neurons. For this reason, representation of data is a quite subtle issue in neural networks, still under active investigation. Other architectural issues to look for in the articles are the presence or absence of feedback, the modifiability of connection (synapse) strengths, and synchronous versus asynchronous operation (natural neural networks are asynchronous).

Because synthetic and natural neural networks perform, or promise to perform, very demanding feats of decision, recognition, and computation based on their network structure, it is not surprising that ideas from communications help analyze and understand neural networks. In the opposite direction, we ask, why do we expect neural networks to be especially useful in the communications industry? We think there are many good reasons.

1) Specialized forms of neural networks with linear neurons are already in use in communications, for adaptive antenna arrays and adaptive feeds, and, in the commercial sector, as adaptive equalizers for computer communications modems.

2) Neural networks, both natural and synthetic, seem able to perform the pattern recognition and optimization tasks so useful in transmission, switching, memory, and the human/system interface. (Threshold decoding could be listed as another example, but the codes are not competitive with other error-correcting coding schemes.)

3) The extensive algorithm analysis in communications, in particular, error-correcting coding, data compression, switching, and queuing, preadapt communication theorists to think about neural network algorithms.

4) Information theory, which considers probability distributions on long sequences, is similar in spirit, and some techniques, to what is needed to understand large synthetic neural associative memories.

5) The potential for learning can reduce the life-cycle costs of major implementations by reducing sustaining engineering and customization costs.

6) The large installed base of communication equipment around the world makes even small improvements in performance valuable, for example, picture quality in high-definition television, which someday could be enhanced by neural-style decoders.

7) The communications industry has traditionally invented, designed, and manufactured many of its own electrical and electronic components for more than 100 years, including such things as the hybrid transformer, the step-by-step switch, the feedback amplifier, the transistor, the CCD camera, and signal-processing chips.

8) The revenue stream produced by the common carriers can support many of the developments needed to apply neural networks to communications.

9) Many communications laboratories are already supporting neural network research and development, living up to their traditions, such as Bell Labs, Bellcore, Lincoln Lab, JPL, Texas Instruments, Hughes, NTT in Japan, Philips in Europe, and others in the civil sector, as well as the U.S. Department of Defense.

Of course, neural networks are also being studied for other applications, in particular robotics. The applications tend to overlap, though. For example, a robotic teleservicer for a communications satellite is part of a communication system. Robotic applications are not specifically discussed here, although many others are.

This issue has eight articles, which will now be highlighted. The first article, "Information Theory, Complexity, and Neural Networks," by Yaser Abu-Mostafa of Caltech, describes what a neural network is from an abstract information complexity standpoint. The author explains how neural networks could be used as associative memories, pattern discriminators, and routing optimizers, and what the theoretical limitations on the ability of neural networks to perform these functions may be. The article also introduces the concept of learning for a synthetic neural network, which, if done offline, can be thought of as very high level programming. The potential ability to learn or generalize is what makes synthetic neural networks so attractive in many classes of applications. For example, a human/system interface, with learning, can potentially tailor itself to the idiosyncrasies of individual users.

The second article, "Neural-Style Microsystems that Learn," by Joshua Alspector, describes progress at Bellcore with an actual prototype VLSI chip that can be taught by example. Natural and sythetic neurons are contrasted so that we can see how far away we are from the connectivity of vertebrate neural nets, and yet we can begin to do interesting tasks with networks we can build. (We can already build some invertebrate-sized synthetic neural networks, but we don't use them as well as the creatures do.) If we try to perform neural network tasks purely in software, we may lose most of the advantage of the high connectivity of simple elements, and just wind up with another clever, but not blindingly fast, computer algorithm. It is specially designed neural hardware that holds the promise of decisive applications in communications; analog or hybrid analog/digital VLSI and optoelectronics seem especially promising for this.

Along these lines, the third article, "Optoelectronic Implementations of Neural Networks," by Demetri Psaltis of Caltech and five members of his research group, provides an alternative or supplement to purely electronic neural chips. Optical interconnections are especially attractive because of their high interconnectivity potential. Natural neural networks are slow in speed compared with gallium arsenide or even with conventional silicon VLSI, but they make up for it by very numerous and often seemingly complex interconnections. Learning is accommodatable with optical neural networks via dynamic holographic media. Pattern recognition applications are demonstrated in the article, with actual pictures recognized shown along with the explanation and block diagrams.

Speaking of pattern recognition, the fourth article, "Handwritten Digit Recognition: Application of Neural Net Chips and Automatic Learning," by Y. Le Cun, L. D. Jackel, and seven colleagues, all from AT&T Bell Laboratories, demonstrates an effective application of neural networks, without and with learning, to handwritten digit recognition, i.e., zip codes. The error and erasure rates obtained are as good as, if not better than, other demonstrated handwriting recognition methods. The advantages come with less system size and cost, due to a special-purpose neural chip in the nonlearning approach, or to a fast signal processing chip with fast alogrithm in the slower learning approach. This foreshadows the future use of neural networks to facilitate the human/system interface.

A related article is the fifth, "Pattern Classification using Neural Networks," by Richard P. Lippmann of the M.I.T. Lincoln Laboratory. The author surveys many neural and neural-inspired nonparametric pattern classifiers. The author compares them as to training complexity and memory requirements, including the performance of the famous back-propagation learning technique for multilevel feed-forward neural networks. A very valuable and extensive bibliography is provided. Classification is a function that often has to be performed by a communication system, for functions such as quantization and data compression, decoding, on-line fault isolation, interference rejection, link quality monitoring, signal detection, and, as in the preceding paper, in the human/system interface, such as for handwritten or voice input to a communication system.

The human/system interface comes in article six, "Integration of Acoustic and Visual Speech Signals using Neural Networks," by B. P. Yuhas and M. H. Goldstein, Jr., of Johns Hopkins, and T. J. Sejnowski of the Salk Institute and the University of California, San Diego. The authors show the progress that has been made in one very important classification problem in communications, that of speech recognition in a noisy environment. Here the "integration" means using the visual signals from the lips and perhaps from other facial expressions to augment the decisions from sound alone. Great improvement is found by incorporating the visual signals ("read my lips"), using neural networks with both speech and visual inputs. Linguistic theory was used in designing the layered feedforward neural network, which was simulated on a RISC computer. Back-propagation learning was used, and great improvement was found over sound input alone at signal-to-noise ratios below 15 dB.

The previous applications have used mostly neurons that distort sums of products. The seventh article, "Neural Networks for Switching," by Timothy X Brown of the California Institute of Technology, shows how neural networks with inhibition (same term used in biology and engineering) can be applied to a central function of common-carrier communication, that of finding or assigning a path through a switch. This is an example of an optimization or assignment problem at which neural networks, natural and synthetic, have been found to be useful. The particular switch studied here is a rearrangeable switch, but the general nonlayered architecture that makes heavy use of inhibitory connections, as natural neural networks often do, is more broadly applicable to switching and routing. Other communications applications of the inhibitory architecture could be found in communication routing problems, such as call routing or packet routing in wide-area networks, bandwidth allocation in space and ground networks, and frequency assignment or reassignment in cellular radio. The article builds on the earlier work of M. C. Paull and D. Slepian on call rearrangement, 25 or more years ago, but neural networks were not part of these early algorithms.

The eighth and final article, "Defense Applications of Neural Networks," by Jasper Lupo of the Defense Advanced Research Projects Agency (DARPA), shows how neural networks can help provide functions such as machine vision, speech recognition, and data structuring for efficient utilization by humans. These are operations which occur in many places in the national defense in sensing and communication systems. The computational capabilities of living creatures, of neural network simulators, and of potential special-purpose neural hardware are reviewed, with projected technological capabilities. Potential applications of neural networks to high data-rate sensors in the national defense are presented with figures and estimated requirements on interconnects and interconnects/second. These applications motivated the new DARPA program in neural networks.

The eight articles, all invited by the Guest Editor, were also all refereed. I thank the referees, who are listed here in alphabetical order: Philip Alvelda, Pierre Baldi, Allen Gersho, Rodney M. Goodman, Fernando Pineda, Jawad Salehi, Bernard H. Soffer, and Eyal Yair. Thanks are also due to Stephen B. Weinstein, who suggested this special issue, and to Carol M. Lof, publisher of IEEE COMMUNICATIONS MAGAZINE, who ably handled the marked-up submissions. The work of the Guest Editor for this issue was supported by NASA and Pacific Bell.

Neural Networks for Circuits and Systems

NEVINE EL-LEITHY AND ROBERT W. NEWCOMB,[1] FELLOW, IEEE

Abstract—**This letter summarizes material from 16 papers on neural networks from the May 1989 special issue of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS.**

The purpose of this letter is to let readers know about the special issue on neural networks published in the May 1989 IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS.[2] The issue contains the following papers.

1) M. A. C. Maher, S. P. DeWeerth, M. A. Mahawold, and C. A. Mead, "Implementing Neural Architectures Using Analog VLSI Circuits," pp. 643-652. This paper discusses a methodology for building artificial neural networks in CMOS VLSI.

2) D. K. Hartline, "Simulation of Restricted Neural Networks with Reprogrammable Neurons," pp. 653-660. Network models for the SYNETSIM program are presented, these being based upon electrobiochemical data.

3) G. Mirchandani and W. Cao, "On Hidden Nodes for Neural Nets," pp. 661-664. A proof is given that the maximum number of separable regions of the input space is a function of both the number of hidden nodes and the input space dimension.

4) M. L. Brady, R. Raghaven, and J. Slawny, "Back Propagation Fails to Separate Where Perceptrons Succeed," pp. 665-674. Counterexamples are presented to show that the limitations of perceptrons are not overcome by the back propagation algorithm.

5) D. L. Standley and J. L. Wyatt, Jr., "Stability Criterion for Lateral Inhibition and Related Networks that is Robust in the Presence of Integrated Circuit Parasitics," pp. 675-681. A design approach is presented which guarantees that lateral inhibition networks will remain stable in the presence of parasitics.

6) A. R. Stubberud and R. J. Thomas, "Associative Recall Using a Contraction Operator," pp. 682-686. An associative memory transformation is introduced which has rapid convergence, noise rejection, and some learning.

7) N. J. Dimopoulos, "A Study of the Asymptotic Behavior of Neural Networks," pp. 687-694. Neural network nonlinear differential equations are discussed and topologies, including cerebellum type, are established which exhibit asymptotic behavior.

8) A. D. Culhane, M. C. Peckerar, and C. R. K. Marrian, "A Neural Net Approach to Discrete Hartley and Fourier Transforms," pp. 695-703. An electronic circuit based on a multiply connected neural net is presented to compute the discrete Hartley and Fourier transforms.

9) O. K. Ersoy and C.-H. Chen, "Learning of Fast Transforms and Spectral Domain Neural Computing," pp. 704-712. The interaction between neural networks and fast transforms is presented with emphasis upon the use of learning algorithms.

10) J.-H. Li, A. N. Michel, and W. Porod, "Analysis and Synthesis of a Class of Neural Networks: Variable Structure Systems with Infinite Gain," pp. 713-731. The theory of ordinary differential equations with discontinuities is used to set up analysis and design procedures for neural networks with infinite gain.

11) D. E. Van den Bout and T. K. Miller, III, "A Digital Architecture Employing Stochasticism for the Simulation of Hopfield Neural Nets," pp. 732-738. A digital architecture which uses stochastic logic for simulating the behavior of Hopfield neural networks is described.

12) M. K. Habib and H. Akel, "A Digital Neuron-Type Processor and Its VLSI Design," pp. 739-746. A set of neuron-type circuit elements based on logic gate circuits with multi-input multi-fan output capability and realizable in VLSI is presented.

13) K. A. Boahen, P. O. Pouliquen, A. G. Andreou, and, R. E. Jenkins, "A Heteroassociative Memory Using Current-Mode MOS Analog VLSI Circuits," pp. 747-755. Use is made of analog current-mode circuits operating in subthreshold conduction to achieve a scalable architecture for the implementation of neural networks with low power consumption.

14) B. Linares-Barranco, E. Sánchez-Sinencio, A. Rodríguez-Vázquez, and J. L. Huertas, "A Programmable Neural Oscillator Cell," pp. 756-761. Using operational transconductance amplifiers a programmable neural oscillator based upon hysteresis is presented.

15) M. Verleysen, B. Sirletti, A. Vandemeulebroecke, and P. G. A. Jespers, "A High-Storage Capacity Content-Addressable

Manuscript received August 11, 1989.

The authors are with the Microsystems Laboratory, Electrical Engineering Department, University of Maryland, College Park, MD 20742.

IEEE Log Number 8932744.

[1]N. El-Leithy and R. W. Newcomb were Guest Editors of the Special Issue on Neural Networks, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, May 1989.

[2]*IEEE Trans. Circuits Syst.*, vol. 36, no. 5, May 1989.