



*Journal of Advances in
Science and Technology*

*Vol. IV, No. VII, November-
2012, ISSN 2230-9659*

STUDY ON VARIOUS RELATED PROCESSING ARCHITECTURES

Study on Various Related Processing Architectures

Remya Balakrishnan¹, Dr. Radhey Shyam Jha²

¹Research Scholar, Monad University, Pilkhwa (Hapur), Ghaziabad

²Principal, B.M. College of Technology & Management, Farukhnagar, Haryana

Abstract – *The processing units and NN are similar. In both cases, the processor units are multi-input, dynamical systems, and the behavior of the overall systems is driven primarily through the weights of the processing unit's linear interconnect. The main discriminator is that in , processors, connections are made locally, whereas in ANN, connections are global. For example neurons in one layer are fully connected to another layer in feed-forward NN and all the neurons are fully interconnected in Hopfield networks. In ANN, it the weights contain information on the processing system's previous state or feedback, but in, processors, the weights are used to determine the dynamics of the system. Furthermore, due to the high interconnectivity of ANN, they tend not exploit locality in either the data set or the processing and as a result, they usually are highly redundant systems that allow for robust, fault-tolerant behavior without catastrophic errors. A cross between an ANN and a , processor is a Ratio Memory , (RM.). In RM, processors, the cell interconnect is local and topologically invariant, but the weights are used to store previous states and not to control dynamics. The weights of the cells are modified during some learning state creating long-term memory.*

Key words: *Hopfield networks, interconnectivity, dynamics, fault-tolerant, long-term memory*

INTRODUCTION

The definition of a system is a collection of independent, interacting entities forming an integrated whole, whose behavior is distinct and qualitatively greater than its entities. Although connections are local, information exchange can happen globally through diffusion. In this sense, , processors are systems because their dynamics is derived from the interaction between the processing units and not within processing units, and as a result, they exhibit emergent, collective behavior. Mathematically, the relationship between a cell and its neighbors, located within an area of influence, can be defined by a coupling law, and this is what primarily determines the behavior of the processor. When the coupling laws are modeled by fuzzy logic, it is fuzzy. When these laws are modeled by computational verb logic, it becomes computational verb, (verb,)Both fuzzy and verb, s are useful to modelling social networks when the local couplings are achieved by linguistic terms.

For those that have already had differential equations, the Laplace transform equivalent will be presented as an alternative while focusing on phasors and calculus.

This research work will expect the reader to have a firm understanding of Calculus specifically, and will not stop to explain the fundamental topics in Calculus.

The result is a linear analysis experience that is general in nature but skips Laplace and Fourier transforms.

REVIEW OF LITERATURE

There are several overviews of processors. One of the better references is a paper, "Cellular Neural Networks: A Review" written for the Neural Nets WIRN Vietri 1993, by Valerio Cimagalli and Marco Balsi. This paper is beneficial because it provides definitions, types, dynamics, implementations, and applications in a relatively small, readable document. There is also a book, "Cellular Neural Networks and Visual Computing Foundations and Applications", written by Leon Chua and Tamas Roska. This reference is valuable because it provides examples and exercise to help illustrates points, which is uncommon in papers and journal articles. This book covers many different aspects of processors and can serve as a textbook for a Masters or PhD course. These two references are invaluable since they manage to organize the vast amount of literature into a coherent framework.

The best place for literature is from the "International Workshop on Cellular Neural Networks and Their Applications" proceedings. The proceedings are available online, via IEEE Xplore, for conferences held in 1990, 1992, 1994, 1996, 1998, 2000, 2002, 2005, and 2006. There is also a workshop being held

on July 14–16 in Santiago de Compostela, Spain. Topics include theory, design, applications, algorithms, physical implementations, and programming/training methods. For an understanding of the analog semiconductor based technology, AnaLogic Computers has their product line, in addition to their published articles available on their homepage and their publication list. They also have information on other technologies such as optical computing. Many of the commonly used functions have already been implemented using processors. A good reference point for some of these can be found in image processing libraries for based visual computers such as Analogic's , based systems.

MATERIAL AND METHOD

Processors could be thought hybrid between ANN and CA (Continuous Automata). The processing units and NN are similar. In both cases, the processor units are multi-input, dynamical systems, and the behavior of the overall systems is driven primarily through the weights of the processing unit's linear interconnect. The main discriminator is that in processors, connections are made locally, whereas in ANN, connections are global. For example neurons in one layer are fully connected to another layer in feed-forward NN and all the neurons are fully interconnected in Hopfield networks. In ANN, it the weights contain information on the processing system's previous state or feedback, but in processors, the weights are used to determine the dynamics of the system. Furthermore, due to the high interconnectivity of ANN, they tend not exploit locality in either the data set or the processing and as a result, they usually are highly redundant systems that allow for robust, fault-tolerant behavior without catastrophic errors. A cross between an ANN and a processor is a Ratio Memory, (RM,). In RM, processors, the cell interconnect is local and topologically invariant, but the weights are used to store previous states and not to control dynamics. The weights of the cells are modified during some learning state creating long-term memory.

The topology and dynamics of processors closely resembles that of CA. Like most processors, CA consists of a fixed-number of identical processors that are spatially discrete and topologically uniform. The difference is that most processors are continuous-valued whereas CA has discrete-values. Furthermore the processor cell's behavior is defined via some non-linear function whereas CA processor cells are defined by some state machine. However, there are some exceptions. Continuous Valued Cellular Automata or Continuous Automata are CA with continuous resolution. Depending on how the Continuous Automata is specified, it can also be a. There is also Continuous Spatial Automata, which consists of an infinite number of spatially continuous, continuous-valued automata. There is considerable work being performed in this field since continuous spaces are easier to mathematically model than discrete spaces, thus allowing a more quantitative approach as

opposed to an empirical approach taken by some researchers of cellular automata. Continuous Spatial Automata processors can be physically realized though an unconventional information processing platform such as chemical computers. Furthermore, it is conceivable that large processors compared to the resolution of the input and output can be modeled as a Continuous Spatial Automata.

MODEL OF COMPUTATION

The dynamical behaviors of processors can be expressed mathematically as a series of ordinary differential equations, where each equation represents the state of an individual processing unit. The behavior of the entire processor is defined by its initial conditions, the inputs, the cell interconnect (topology and weights), and the cells themselves. One possible use of processors is to generate and respond to signals of specific dynamical properties. For example, processors have been used to generate multi-scroll chaos, synchronize with chaotic systems, and exhibit multi-level hysteresis. , processors are designed specifically to solve local, low-level, processor intensive problems expressed as a function of space and time. For example processors can be used to implement high-pass and low-pass filters and morphological operators. They can also be used to approximate a wide range of Partial Differential Equations (PDE) such as heat dissipation and wave propagation.

Processors can be used as Reaction-Diffusion (RD) processors. RD processors are spatially invariant, topologically invariant, analog, parallel processors characterized by reactions, where two agents can combine to create a third agent, and diffusions, the spreading of agents. RD processors are typically implemented through chemicals in a Petri dish (processor), light (input), and a camera (output) however RD processors can also be implemented through a multilayer processor RD processors can be used to create Voronoi diagrams and perform skeletonisation. The main difference between the chemical implementation and the implementation is that implementations are considerably faster than their chemical counterparts and chemical processors are spatially continuous whereas the processors are spatially discrete. The most researched RD processor, Belousov-Zhabotinsky (BZ) processors, has already been simulated using a four-layer , processors and has been implemented in a semiconductor.

Like CA, computations can be performed through the generation and propagation of signals that either grow or change over time. Computations can occur within a signal or can occur through the interaction between signals. One type of processing, which uses signals and is gaining momentum is wave processing, which involves the generation, expanding, and eventual collision of waves. Wave processing can be used to measure distances and find optimal paths.

Computations can also occur through particles, gliders, solitons, and filterons localized structures that maintain their shape and velocity. Given how these structures interact/collide with each other and with static signals, they can be used to store information as states and implement different Boolean functions. Computations can also occur between complex, potentially growing or evolving localized behavior through worms, ladders, and pixel-snakes. In addition to storing states and performing Boolean functions, these structures can interact, create, and destroy static structures.

Although, processors are primarily intended for analog calculations, certain types of processors can implement any Boolean function, allowing simulating CA. Since some CA are Universal Turing machines (UTM), capable of simulating any algorithm can be performed on processors based on the von Neumann architecture, that makes this type of , processors, universal , a UTM. One architecture consists of an additional layer, similar to the ANN solution to the problem stated by Marvin Minsky years ago. , processors have resulted in the simplest realization of Conway's Game of Life and Wolfram's Rule 110, the simplest, known universal Turing Machine. This unique, dynamical representation of an old systems, allows researchers to apply techniques and hardware developed for, to better understand important CA. Furthermore, the continuous state space of processors, with slight modifications that have no equivalent in Cellular Automata, creates emergent behavior never seen before.

CONCLUSION:

Any information processing platform that allows the construction of arbitrary Boolean functions is called universal, and as result, this class processors are commonly referred to as universal processors. The original processors can only perform linearly separable Boolean functions. This is essentially the same problem Marvin Minsky introduced with respect to the perceptions of the first neural networks. In either case, by translating functions from digital logic or look-up table domains into the , domain, some functions can be considerably simplified. For example, the nine-bit, odd parity generation logic, which is typically implemented by eight nested exclusive-or gates, can also be represented by a sum function and four nested absolute value functions. Not only is there a reduction in the function complexity, but the implementation parameters can be represented in the continuous, real-number domain.

REFERENCES:

- A. Adamatzky, B. Costello, T. Asai "Reaction-Diffusion Computers", 2005.
- F. Gollas and R. Tetzlaff, "Modeling Complex Systems by Reaction-Diffusion Cellular Nonlinear Networks with Polynomial Weight-Functions", Int'l Workshop on Cellular Neural Networks and Their Applications, 2005.
- A. Selikhov, "mL-: A , Model for Reaction Diffusion Processes in m Component Systems", Int'l Workshop on Cellular Neural Networks and Their Applications, 2005.
- B. Shi and T. Luo, "Spatial Pattern Formation via Reaction-Diffusion Dynamics in 32x32x4 , Chip", IEEE Trans. On Circuits and Systems-I, 51(5):939-947, 2004.
- E. Gomez-Ramirez, G. Pazienza, X. Vilasis-Cardona, "Polynomial Discrete Time Cellular Neural Networks to solve the XOR Problem", Int'l Workshop on Cellular Neural Networks and Their Applications, 2006.
- F. Chen, G. He, X. Xu1, and G. Chen, "Implementation of Arbitrary Boolean Functions via ,", Int'l Workshop on Cellular Neural Networks and Their Applications, 2006.
- R. Doguru and L. Chua, ", Genes for One-Dimensional Cellular Automata: A Multi-Nested Piecewise-Linear Approach", Int'l Journal of Bifurcation and Chaos, 8(10):1987-2001, 1998.
- R. Dogaru and L. Chua, "Universal Cells", Int'l Journal of Bifurcations and Chaos, 9(1):1-48, 1999.
- R. Dogaru and L. O. Chua, "Emergence of Unicellular Organisms from a Simple Generalized Cellular Automata", Int'l Journal of Bifurcations and Chaos, 9(6):1219-1236, 1999.
- T. Yang, L. Chua, "Implementing Back-Propagation-Through-Time Learning Algorithm Using Cellular Neural Networks", Int'l Journal of Bifurcations and Chaos, 9(6):1041-1074, 1999.
- T. Kozek, T. Roska, and L. Chua, "Genetic Algorithms for Template Learning," IEEE Trans. on Circuits and Systems I, 40(6):392-402, 1993.
- G. Pazienza, E. Gomez-Ramirez and X. Vilasis-Cardona, "Genetic Programming for the -UM", Int'l Workshop on Cellular Neural Networks and Their Applications, 2006.
- J. Nossek, G. Seiler, T. Roska, and L. Chua, "Cellular Neural Networks: Theory and Circuit

Design," Int'l Journal of Circuit Theory and Applications, 20: 533-553, 1998.

- K. Wiehler, M. Perezowsky, R. Grigat, "A Detailed Analysis of Different Implementations for a Real-Time Image Processing System", Int'l Workshop on Cellular Neural Networks and Their Applications, 2000.
- A. Zarandry, S. Espejo, P. Foldesy, L. Kek, G. Linan, C. Rekeczky, A. Rodriguez-Vazquez, T. Roska, I. Szatmari, T. Sziranyi and P. Szolgay, "Technology in Action", Int'l Workshop on Cellular Neural Networks and Their Applications, 2000.
- L. Chua, L. Yang, and K. R. Krieg, "Signal Processing Using Cellular Neural Networks", Journal of VLSI Signal Processing, 3:25-51, 1991.
- T. Roska, L. Chua, "The Universal Machine: An Analogic Array Computer", IEEE Trans. on Circuits and Systems-II, 40(3): 163-172, 1993.
- T. Roska and A. Rodriguez-Vazquez, "Review of CMOS Implementations of the Universal Machine-Type Visual Microprocessors", International Symposium on Circuits and Systems, 2000
- A. Rodríguez-Vázquez, G. Liñán-Cembrano, L. Carranza, E. Roca-Moreno, R. Carmona-Galán, F. Jiménez-Garrido, R. Domínguez-Castro, and S. Meana, "ACE16k: The Third Generation of Mixed-Signal SIMD-, ACE Chips Toward VSoCs," IEEE Trans. on Circuits and Systems - I, 51(5): 851-863, 2004.
- T. Roska, "Cellular Wave Computers and Technology – a SoC architecture with xK Processors and Sensor Arrays", Int'l Conference on Computer Aided Design Accepted Paper, 2005.
- K. Karahaliloglu, P. Gans, N. Schemm, and S. Balkir, "Optical sensor integrated for Real-time Computational Applications", IEEE Int'l Symposium on Circuits and Systems, pp. 21–24, 2006.