Preface

Purpose and Goals

As artificial neural network (ANN) applications move from laboratories to the practical world, the concern about reducing the time for the learning and recall phases becomes very real. Parallel implementation schemes offer an attractive way of speeding up both these phases of ANNs. Although much progress has taken place on the subject of parallel implementation of various ANN paradigms, the work relating to this has not been been explored comprehensively in a single publication; rather it is scattered in various conference proceedings and journal publications, some of which are even outside the field of neural networks. This motivated us to present the parallel implementation aspects of all major ANN models in a single book, which can then serve as a good reference to all those involved in the research and application of this area of neural networks.

The book also aims to provide details of implementations on various processor architectures (ring, torus, and others) built on different hardware platforms ranging from large general-purpose parallel computers to custom-built multi-instructions multi-data (MIMD) machines using transputers and digital signal processors (DSPs). We believe that such a book will promote fruitful research in the interdisciplinary area of parallel processing and neural networks.

The book consists of several self-contained chapters, each authored by experts who have actually performed the implementations covered in their respective chapters. The authors provide results of their work, which in some cases are as yet unpublished.

The book is aimed at graduate students and researchers working in the area of artificial neural networks and parallel computing. It can be used by educators at the graduate level to illustrate the use and methods of parallel computing for ANN simulation. Because the mathematical analyses are lucid we feel it can also serve as a good reference book for the practitioners in this field.

An Overview

Because the aim of the book is to provide both analysis and experimental results of parallel implementations (of different ANN models) on various hardware platforms, we have divided the book broadly into three parts each providing analysis and implementations on a class of related hardware. As parallel simulation of ANN is the topic of the book, a detailed discussion on the parallel processing aspects of neural networks is given in Chapter 1. This chapter also describes the different ANN models used in the rest of the book for parallel implementation.

As multilayer feedforward network with backpropagation (BP) learning is the most popular of all ANN models, Chapter 2, preceding the three parts, provides a comprehensive review of the various forms of parallelism and a survey of different parallel implementations of BP.

Part I of the book mainly concentrates on the theoretical analysis of parallel implementation schemes on MIMD message-passing machines. It consists of four chapters providing detailed analysis of parallelism for BP, real-time recurrent learning (RTRL), and adaptive resonance theory (ART1).

Part II describes the details of the parallel implementation of BP neural networks on a general-purpose, large, parallel computer. Analysis and results from extensive studies on the Fujitsu AP1000 parallel computer are discussed.

Part III consists of four chapters each describing a specific special-purpose parallel neural computer configuration. They also provide application case studies on practical implementation of some of the ANN models discussed in Chapter 1. Our overall conclusions and predictions of future trends are presented in Chapter 12.

Organization of the Book

The first two chapters of Part I, Chapters 3 and 4, deal with parallel mapping of multilayer feed-forward neural networks with BP learning on a homogeneous and heterogeneous processor array in a ring topology. The authors, Dr. P. Saratchandran, Dr. N. Sundararajan, Mr. R. Arularasan, and Mr. Foo Shou King, from Nanyang Technological University, Singapore, have addressed the issue of finding the optimal mapping (to find the minimal training time) *using theoretical means* without actually connecting the hardware and running simulations. By developing rigorous mathematical models of the parallel BP running on the hardware, the authors find the optimal solution to the mapping problem for both network-based and training-set parallelism. Using benchmark problems, they have also validated their theoretical models with experimental results using an array of transputers.

Chapter 5 deals with parallel implementation of recurrent neural networks. Written by Prof. Elias Manolakos from the Northeastern University, Boston, this chapter covers in detail the parallel implementation of a fully recurrent neural network on a transputer-based multiprocessor system with a real-time recurrent learning algorithm used for training. This chapter shows that the computationally intensive sequential RTRL algorithm can be transformed into an equivalent parallel algorithm realized in a ring topology that can be matched to a variety of target architectures, ranging from application specific very large scale integration (VLSI) arrays to general-purpose multiprocessor systems. Efficient implementation of the RTRL algorithm on a ring of 19 transputers is discussed and the speedup verified both analytically and experimentally.

Chapter 6, also by Prof. Manolakos and his coworkers from Boston, contains some recently developed results on parallel implementation of ART1 networks. They describe a newly developed parallel algorithm for ART1 that can be mapped on MIMD processor networks and is also suitable for VLSI implementation.

In Part II, Chapter 7 covers the implementations of BP neural networks on large parallel computer systems. Dr. Jim Torresen of Norwegian University of Science and Technology, Norway, and Shinji Tomita of Kyoto University, Japan, have been working in this area using the big Fujitsu AP1000 machine for their studies. They consider the problem of mapping the backpropagation training of real neural applications onto large parallel systems, and they discuss a parallel mapping scheme of neural training that adapts the configuration to the neural application. Also, a heuristic for selecting the best mapping scheme, combining all degrees of parallelism, which minimizes the total training time, is developed in this chapter. The mapping scheme is tested on the general-purpose, parallel computer AP1000, which is a message passing MIMD computer with a two-dimensional torus network of processing elements containing a maximum of 512 processing elements. The results indicate that the training speed can be reduced from hours to minutes if the parallel system is used

instead of a single-processor workstation. Moreover, the flexible mapping, which adjusts the configuration to the given neural application, may result in a training speed several hundred percent faster than a fixed mapping.

Part III of the book discusses parallel implementations on special-purpose processor architectures and computing devices that are dedicated for ANN simulation.

Chapter 8 in Part III discusses special architectures called the toroidal lattice architecture (TLA) and the planar lattice architecture (PLA) as massively parallel architectures of neurocomputers for large-scale neural network simulations. Proposed by Prof. Yoshiji Fujimoto from Ryukoko University, Japan, the discussion indicates that the performance of these architectures are almost proportional to the number of node processors and they adopt the most efficient two-dimensional processor connections to be implemented by the wafer scale integration (WSI) technology. Prof. Fujimoto also discusses the implementation of the TLA using transputers for BP as well as Hopfield networks, including the details of parallel processor configurations and the load balance algorithm and evaluation of its performance on problems like the traveling salesman problem (TSP) and the identity mapping problem (IM).

Chapter 9 deals with an implementation method that exploits the maximum amount of the parallelism form of neural computation without enforcing stringent conditions on the neural network interconnection structures for achieving high implementation efficiency. In this chapter, Prof. Jean-Luc Gaudiot of the University of Southern California and Dr. Soheil Shams of the Hughes Research Laboratory, California, propose a new reconfigurable parallel processing architecture called the dynamically reconfigurable extended array multiprocessor (DREAM) Machine and an associated mapping method for implementing neural networks with regular interconnection structures. Examples of BP and Hopfield networks are used to demonstrate the efficiency of the mapping method and also of the DREAM Machine architecture on implementing diverse interconnection structures.

In Chapter 10, Patrick Spiess and his group from ETH, Switzerland, discuss the architecture, implementations, and applications of a DSP-based parallel high-performance computer system named MUSIC (multiprocessor system with intelligent communication). This system consists of an array of up to 63 processing elements, distributed memory, and a ring communication network. The performance for backpropagation learning featuring continuous weight update (on-line learning) is up to 330 million connection updates per second, which outperforms even the fastest conventional supercomputers.

The last chapter in Part III, Chapter 11, describes a special-purpose parallel neural computer called SPERT-II developed by Krste Asanovic and his coworkers at the International Computer Science Institute, Berkeley, California. SPERT-II is based on T0, a custom fixed-point vector microprocessor and serves as an attached processor to a standard workstation host. These systems have been used primarily to speed neural network training for speech recognition work, and results based on this work are presented in this chapter.

The book concludes with Chapter 12, which summarizes the work reported in the earlier chapters and provides some overall conclusions and future directions in this field.

Acknowledgments

Undertaking and completing a project like this would not have been possible without the support and encouragement of many individuals. First and foremost, we wish to thank all the contributors, who readily agreed to participate in this work, for providing the material in a timely manner. Without their support this book would not have materialized.

Next, we want to thank our doctoral student Foo Shou King who helped us to organize the contributions from authors all over the world with different word processing software

and different hardware platforms from PC to Macintosh. He brought them together to the nice world of LaTeX so as to bring a common standard to this volume. We also hereby acknowledge the countless hours he spent with us in compiling this volume while doing his research work.

We wish to record our thanks to Dr. Cham Tao Soon, president of Nanyang Technological University, for providing an excellent academic environment in which we could undertake and succeed in an endeavor like this.

We are grateful to Prof. Er Meng Hwa, dean of the School of Electrical and Electronic Engineering, for his support and encouragement during this project. We also wish to thank Prof. Soh Yeng Chai, head of the Division of Control and Instrumentation, for his support during this project.

We owe a debt of gratitude to the anonymous reviewers who provided valuable comments in the preliminary stages of this project, most of which have been incorporated in the book.

Finally, we extend our thanks to Mr. Bill Sanders and Ms. Cheryl Baltes of the IEEE Computer Society Press, California, for their constant support and help during this project.