Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Parallel and distributed neurocomputing with wireless sensor networks

Gursel Serpen, Linqian Liu

Electrical Engineering and Computer Science, University of Toledo, Toledo, OH 43606, USA

ARTICLE INFO

Article history:
Received 20 December 2012
Received in revised form
27 August 2015
Accepted 28 August 2015
Available online 7 September 2015

Keywords:
Neurocomputing
Artificial neural network
Wireless sensor network
Parallel and distributed processing
Parallel hardware
Kohonen's SOM neural network

ABSTRACT

This paper proposes a novel hardware computing platform for fully parallel and distributed computation of artificial neural network (ANN) algorithms. The proposed idea entails leveraging the existing wireless sensor networks (WSN) technology to serve as a parallel and distributed hardware platform to implement computations for artificial neural network algorithms. Feasibility of the proposed neurocomputing architecture is demonstrated through a simulation-based case study, which uses Kohonen's self-organizing map as the neural network algorithm. MATLAB-based PROWLER, which is a protocol and application level simulator for wireless sensor networks, is employed for the simulation study. Findings demonstrated that the proposed neurocomputing architecture was able to train the SOM neural network with competitive accuracy values for the unsupervised clustering task. Conclusions of the simulation study suggest that the WSN-based neurocomputing architecture is a feasible alternative for realizing parallel and distributed computation of artificial neural network algorithms.

1. Introduction

A truly parallel and distributed hardware realization of artificial neural network (ANN) algorithms has been an on-going quest of researchers [1,2]. Many attempts to realize the hardware implementation in silicon proved to be too challenging or nearly impossible particularly when the scale of the implementation increased to dimensions at par with the real life problems. Numerous attempts were made through VLSI-based technology with limited success [3–6]. In majority of cases hybrid systems with specially designed accelerator hardware boards were interfaced with von Neumann based systems (standalone supercomputers or even parallel computing machines) to achieve acceptable speed of training and execution [7–12]. The results were mixed. Some really large-scale ANN algorithms could be trained and executed on these platforms. On the other hand, the main hindrance was the lack of availability of such platforms for ready access other than those in specialized research and development facilities or labs. Software-only implementation of artificial neural networks was also attempted primarily for smaller scale or non real-time problems due to apparent time complexity of such realizations. Again, larger scale problems would require supercomputing-grade platforms for software-only realizations and proved to be simply not practical in general.

Following several sections present a survey of prominent studies reported in the literature for VLSI, software-only, and hybrid (i.e. combination of hardware and software) implementations of artificial neural network algorithms to date.

1.1. VLSI for ANN implementations

The very large scale integration (VLSI) technology has been unable to deliver a truly parallel and distributed realization of artificial neural networks algorithms at large scales. This section briefly presents the state of the art in hardware-only realization of artificial neural network (ANN) algorithms through the VLSI technology. One of a few larger scale VLSI technology based hardware realization of neural networks is discussed in [5] from purely an academic perspective. Authors report a neural network design with 6144 spiking neurons and 1.57 million synapses. In another recent study reported in [3], a neural network with up to 10k neurons has reportedly been realized and plans for implementing a neural network with 100k neurons and 100 million synapses was discussed as part of the future plans. A comprehensive review of commercial, and yet mainly experimental, VLSI integrated circuits (IC) (analog, digital or hybrid) implementing ANN algorithms is presented with detailed characterization of number of neurons (nodes or processing elements), on-chip learning capability, type of neural network algorithm etc. in [1] . Their paper concludes "*Moreover, there is no clear consensus on how to exploit the currently available VLSI and even ultra-large-scale integration (ULSI) technological capabilities for massively parallel*

E-mail addresses: gursel.serpen@utoledo.edu (G. Serpen), linqian.liu@utoledo.edu (L. Liu).

http://dx.doi.org/10.1016/j.neucom.2015.08.074
0925-2312/© 2015 Elsevier B.V. All rights reserved.

neural network hardware implementations." There are also other studies that report essentially very small-scale VLSI hardware realizations of artificial neural networks [2,6].

As to actual VLSI hardware that made into the marketplace, the news is relatively old and discouraging. Some of the well-known hardware implementations are Intel ETANN, CNAPS, Synaptics Silicon Retina, NeuraLogix NLX-420, HNC 100-NAP, Hitachi WSI, Nestor/Intel NI1000, Siemens MA-16, SAND/1, MCE MT19003, AT&T ANNA among others [1,2]. Many of these hardware realizations are small-scale and geared towards highly specific neural network algorithms. The most recent example of a custom VLSI IC that implements a neuromorphic chip was reported in September 2011 by a team of researchers with IBM. This IC houses 256 neurons and 64k binary synapses [4]. The authors claimed that the proposed system is scalable, and yet no reports of an actual prototype being produced to date materialized.

Another technological direction of exploration pursued the synthesis of quantum computing and artificial neural networks, which led to the study of so-called quantum neural networks (QNNs). A prognosis for the field of QNNs was presented in a 2011 survey [47]. The said survey did not portray a promising picture at the time. It reported that many papers that contributed to QNNs mainly appeared during a decade-long period that accelerated in mid-nineties [48–57]. Following this period, however, the survey claimed that most authors appeared to have ceased to publish on the topic. During this same period or publication activity, reported studies constructed a substantial portion of the theoretical and simulation-based empirical foundations, and yet there was no demonstrable actual hardware realization of the proposed QNN designs.

The realization of neural algorithms through VLSI on silicon hardware as integrated circuits has been essentially a mainly academic exploration and failed to score any notable real-life or commercial success. The VLSI hardware implementation of a neural network, where neurons are physically discrete entities within a given IC, has not been able to deliver large-scale neural network realizations. The number of neurons, which is direct indication of the computational power, in a VLSI-based neural network realization is limited to not more than 10,000 neurons under the best and most optimistic scenarios (and that is typically in an academic or research lab setting). The VLSI approach further suffers from inflexibility, and the curse of exploding parameter space (too many parameters to set, adjust, tune or adapt). The VLSI realizations of ANNs are in almost all cases closely aligned and optimized for a specific artificial neural network algorithm without further flexibility to accommodate another even slightly different neural algorithm. The size of networks that can be implemented in VLSI ICs is still small by any measure.

1.2. Simulations of ANNs

Simulations on even parallel architectures (in traditional computer architecture sense) fail to scale with the size of the neural network since both time and space complexities quickly reach a level that is beyond what is affordable. Even if multiple processors of a parallel computing platform update a multiplicity of neurons in a given neural network and specialized concurrency techniques perform or facilitate certain operations in parallel (i.e. matrix algebra), the spatio-temporal cost of pure simulation is still insurmountable. For instance, approximately several tera bytes of memory are required for the weight matrix of a Hopfield neural network configured to search a 200-vertex graph. Lack of simulations for truly large-scale neural networks is a testimony to the fact that the option of pure simulation is severely constrained for any practical utility. The only exception is reported in [9], which apparently has been, at least at the time of the writing this manuscript, highly controversial for the validity of its findings [13].

1.3. Hybrid (software and hardware) implementations

There have been various attempts to build specialized computing platforms based on a mix of hardware and software components. The resultant computing systems were byproducts of different techniques drawn from software or hardware domains to essentially speed up or accelerate computations. One approach entails software models running on high-end supercomputer grade platforms like the Blue Brain [9] or Beowulf cluster [10]. Blue Brain project reportedly aims to simulate sections of the brain and uses the IBM Blue Gene/L supercomputer (360 Tera flops through 8192 PowerPC™ CPUs). The computing platform is claimed to be able to simulate 100k neurons with very complex biological models and 100 million neurons with simple biological models. The focus of this project was simulation of parts of brain through realistic and accurate models of a biological neural system. In the case Beowulf cluster, which is a 27-processor machine, simulation of a Thalamocortical model for one second of activity required 10^{11} neurons and 10^{15} synapses, and took nearly two months to complete. Such an approach exhibits high flexibility but requires hard-to-access and very expensive hardware. Field programmable gate array (FPGA)-based approach forms the basis of a second approach where primary software routines are implemented in hardware for significantly accelerated computation. Although FPGA-based approach offers great flexibility, practitioners often struggle to establish the correct system balance between processing and memory while also dealing with a harder programming aspect compared to the software-only approach. The third approach is the custom-built hardware. This has been tried many times without notable success owing mainly to the fundamental problems which application specific integrated circuits (ASIC) possess. It has proven to be a major challenge, as evidenced by the lack of an operational system deployed in the field. Dealing with the issue of deciding how much of the neural network functionality should be realized through hardware required challenging compromises. It typically results in the optimization of performance but the loss of flexibility.

There are some recent examples of projects that attempted to implement hybrid paradigms. The Synaptic Plasticity in Spiking Neural Networks (SP2INN) project [12] envisioned custom hardware design and prototyping for a neural network with one million neurons alongside of several million synaptic connections. The outcome is not a success and this project has been reportedly abandoned. The follow-up project, SEE, attempted to leverage FPGA-based approach with certain level of success – it was claimed that about a half million neurons each with up to 1.5k synaptic connections could be modeled [7]. However, there are no traces of this work in the technological marketplace either. The SpiNNaker project [8] aimed at development of a massively parallel computing platform based on essentially a modified and highly-tuned system-on-a-chip (SOC) technology. It was conceived to serve a neural network realization with up to a billion spiking neurons and intended to explore the potential of spiking neuron based systems in engineered systems. However, a successful outcome apparently does not appear to have been achieved [8].

1.4. Neural nets and wireless sensor networks

There are a number of attempts in the literature [14–28] that strive to bring together the wireless sensor network (WSN) and ANN technologies. In some cases, what has been done is to simply embed an entire neural network, say a Kohonen's self-organizing map or multilayer perceptron network, within each and every