Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 6 (2011) 390–395

Procedia Computer Science

Complex Adaptive Systems, Volume 1
Cihan H. Dagli, Editor in Chief
Conference Organized by Missouri University of Science and Technology
2011- Chicago, IL

Parallel and distributed computations of maximum independent set by a Hopfield neural net embedded into a wireless sensor network

Gursel Serpen, Jiakai Li

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

Abstract

This paper, as the first one in a three-paper sequence, presents a proposed framework to employ a wireless sensor network as a hardware computation platform for fully parallel and distributed computation of maximum independent set of a given graph through a Hopfield neural network. Theoretical and mathematical foundations of the proposed framework will be discussed. Mapping the maximum independent set problem to Hopfield neural network dynamics is presented. This is followed by the demonstration of embedding the Hopfield neural network as a static optimizer into the wireless sensor network in fully parallel and distributed mode. The outcome is a wireless sensor network operating as a parallel and distributed computing hardware platform for a Hopfield neural network configured to solve a static optimization problem. The nesC-TinyOS model of the proposed computational framework and the corresponding simulation study are deferred to the second and third papers, respectively, in the three-paper sequence.

Keywords: wireless sensor network, artificial neural network, Hopfield network, maximum independent set, parallal and distributed computation, static optimization, adaptation, computational intelligence

1. Introduction

Wireless sensor networks (WSN) are an emerging technology due to recent advancements in very small-scale manufacturability and high-scale integration of various electronic components in a single packaging. A typical sensor node (or mote) is a standalone package of electronics necessary to hold a number of sensors, a processor-based miniaturized computing platform, a power unit that has limited capacity which may or may not be renewable, and a radio trans-receiver at its core. Typical size of a sensor node is anywhere from a matchbox to a coin at the present time, but is expected to shrink dramatically in the next decade with the exciting promise of micro electro-mechanical systems or nanotechnology manufacturing and fabrication.

Current and projected applications of wireless sensor networks encompass a wide variety of domains which have been traditionally challenging to access due to many reasons including potential harm to humans, being at remote sites or distributed over very large areas, and being subject to harsh geo or meteorological circumstances among

doi:10.1016/j.procs.2011.08.073

others. Monitoring environment (pollution in a lake or river), forest fires, volcanoes, battlefield troop movements, human body, monitoring structural health of high-rise buildings or bridges, smart home automation, and the last but not the least, smart renewable energy grid monitoring and control are among the countless potential applications [1].

Given the nature of applications for wireless sensor networks, a typical deployment scenario in many cases entails scattered random placement of hundreds or thousands of such nodes in a given geographic area through either dropping aerially from a flying craft or spreading from a moving land vehicle. Resultantly, the set of sensor nodes form an ad hoc computer network. Often such nodes and the network are expected to operate for a period of at least one to two years using the on-board power source depending on the nature of the application without any outside maintenance or repair access since such access may simply be not feasible or practical.

Wireless sensor networks are conceived to be deployed and expected to operate autonomously and particularly in non-hospitable environments without human involvement. Various factors including geography, climate, and human-induced intentional or non-intentional interference in the electromagnetic spectrum will adversely affect the deployed network, and hence be requiring a good level of adaptability to changing circumstances. A wireless sensor network is a dynamic system in the sense that it goes through changes over time, which have important consequences on the operation and requirements of the network. Some of these changes may include revisions to mission or functionality at different scales; changes in static or dynamic node composition; energy consumption profile of nodes and the network over time; destruction or death of certain nodes; and transient effects that may temporarily hinder a node, a cluster of nodes, or a sub-network to function within its normal operating framework.

The goal of this paper is to explore feasibility of introducing ability for "adaptation" and facilities for "computational intelligence" to the wireless sensor networks for significantly-enhanced autonomous behaviour and operation. The desirable adaptation capability can be introduced through embedding an artificial neural network (ANN), which can be instantiated to any specialized form such as feedforward, self-organizing or recurrent, in the fully parallel and distributed mode within the wireless sensor network. The expectation from an adaptive or intelligent WSN is that it can readily take into consideration the changes dictated by dynamic nature of operational and application aspects, and accordingly adapt to changing conditions, circumstances, missions, and operational demands following the deployment. The scope of the work being reported in this paper is limited to that of feasibility only, and hence a single case study will be employed due to space limitations, in which a Hopfield recurrent neural network in static optimization mode configured for the maximum independent set problem will be embedded in fully parallel and distributed mode within a TinyOS-based WSN.

There are a number of attempts in the literature that strive to bring together the WSN and ANN technologies [2-12]. 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 sensor node. In other cases, a gateway node (often another name for a laptop-grade computing platform) calculates a centralized (non-distributed) solution through a neural network algorithm using global information and the solution is transmitted to the sensor network nodes afterwards. Existing studies do not view the WSN as a hardware implementation platform for an ANN for massively parallel and fully distributed computation.

2. Distributed and Parallel Neural Computation on Wireless Sensor Networks

Each sensor node or mote in a wireless sensor network (WSN) possesses computational power due to the on-board microcontroller or an equivalent digital system, and can operate independent of other motes for asynchronous processing or in time synchronization with the rest of the network as needed. Distributed programs can be embedded within the local storage or memory of each mote either during the initial manufacturing phase or after deployment and through the wireless channel of the on-board radio trans-receiver. Motes can exchange their computations with other motes over the air through their radio trans-receivers. Typically, reach of each antenna for uni/multi/broadcasting and reception is constrained to a small geographic neighbourhood of each mote for a number of reasons including the inverse power law that dominates the radio transmission power consumption and the need to reduce the interference and crowding in a given channel for the purpose of medium access control among others. Routing protocols are implemented to facilitate exchange of data and information among the motes themselves and with the gateway mote which would typically be interfaced to a powerful laptop-grade computer. It is not hard to imagine thousands or hundreds of thousands of motes forming a single WSN for certain applications. Considering an algorithm that implements a task that can be decomposed into numerous subtasks with parallel executability, mapping it to such a WSN for truly and massively parallel and distributed computations, will facilitate real time computations for large-scale problems.

Wireless sensor networks are topologically similar to artificial neural networks. A WSN is constituted from hundreds or thousands of sensor nodes or motes. Similarly, an artificial neural network (ANN) is composed of hundreds or thousands of (computational) nodes or neurons, each of which is assumed to possess only very limited computational processing capability. This similarity can be the basis to benefit the adaptation and operational aspects of WSNs through leveraging the existing neural network theory in its entirety for all practical purposes. Since a wireless sensor network with thousands of motes or nodes is a highly distributed system with parallel computing ability, a fusion of yet another highly parallel and distributed system, the artificial neural network, is in order. In fact there is a one-to-one correspondence, in that, a sensor mote can act like or implement a neural network neuron or node, while wireless links among the motes are analogous to the connections among neurons. Fusing WSNs with ANNs sets the stage for WSNs to suddenly possess considerably substantial computational intelligence to be able to address a very comprehensive portfolio of problems both at the protocol and application layers.

3. Hopfield Neural Network and Maximum Independent Set Problem

Static optimization and more specifically the combinatorial optimization have been subject to rigorous empirical studies in numerous domains including hard and soft sciences, engineering, economics, business, and finance to list a few during the past decades particularly after the widespread emergence of mainframe computing in early 70s. Most notable solutions leveraged direct search techniques in graph theory, heuristic search in artificial intelligence, and dynamic programming in operations research among others. Reasonably large-scale problem instances were successfully addressed for near-optimal solutions by these algorithms even though in some cases computational complexity, i.e. both spatial and temporal, turned out to be prohibitive for a real time context. Artificial neural network based optimizers on the other hand might be positioned to address this very same challenge; scalability with respect to real time computation requirements as the problem size increases to real life dimensions. This premise is based on the fact that neural network optimizers do offer the prospect of hardware realization for truly distributed computation upon appropriate technological advances and accordingly are poised to fully leverage the massive parallelism inherent to neural optimization algorithms. If in fact adequate technological advances can be realized, then the neural optimizers can compute solutions of hard optimization problems practically in constant time regardless of the problem size. This premise is inherently lacking in non-neural approaches: no other non-neural algorithm appears to offer an ability to leverage such a massive degree of parallelism so that it can be promising for a real time context.

3.1 Hopfield Neural Network Dynamics

A Hopfield neural network is a nonlinear dynamical system, whereby the definition of the continuous Hopfield neural network is as follows [13]. Let z_i represent a neuron output and $z_i \in [0.0, 1.0]$ with $i = 1,2,\ldots,K,$ where K is the number of neurons in the Hopfield network. Then,

$$E(\mathbf{z}) = -\frac{1}{2}\sum_{i=1}^{K}\sum_{j=1}^{K} w_{ij} z_i z_j + \frac{1}{\lambda}\sum_{i=1}^{K}\int_{0}^{z_i} f^{-1}(z)dz - \sum_{i=1}^{K} b_i z_i \tag{1}$$

is a Liapunov function for the system of equations defined by

$$\frac{du_i(t)}{dt} = -u_i(t) + \sum_{j=1}^{K} w_{ij} z_j(t) + b_i \text{ and } z_i = f(u_i), \tag{2}$$

where w_{ij} is the weight between neurons z_i and z_j subject to $w_{ij} = w_{ji}$ and $w_{ii} = 0$, b_i is the external bias input for node z_i, and $f(\cdot)$ is a nonlinearity - typically the sigmoid function with positive slope steepness value represented by λ. Note that the second term in the Liapunov function vanishes for very large positive values of the parameter λ for cases where the activation function is sigmoid shaped.

3.2 Maximum Independent Set Problem and Mapping to Hopfield Network

Clustering is an important topology control option for wireless sensor networks. A number of problems due to a densely-deployed network, such as nodes interfering with each other, too many routing paths to choose among, excessive transmission power usage to communicate with distant nodes directly and the consequential limitations on the wireless bandwidth, and oversensitivity to recomputed routes due to small movements of the motes in the network. The main idea in clustering is to partition the original network into smaller sub-networks or clusters each of which also has a cluster head. Every mote in a cluster is at one hop distance from the cluster head. It is also often of interest to minimize the average number of motes in each cluster, which is equivalent to searching for a maximum independent set (MIS) in the graph equivalent model of the sensor network.

Assume a graph has a set of N vertices, $V_i, i = 1,2,\ldots, N$, and up to $K\ (= N^2)$ edges, $e_{ij}, i, j = 1,2,\ldots,N$, where some of the edges may not exist. Consider a neural network with N neurons where outputs of neurons are represented by $z_1,\ldots,z_N$. Each neuron in the neural network will be mapped or correspond to a vertex in the graph. Maximum independent set (MIS) is defined as a maximum cardinality subset of V such that there is no edge, e_{ij}=0, between any two vertices in the maximum cardinality subset (which is a hard constraint). Assuming that the cardinality of set V is N, a Hopfield network with N neurons can be used to map the MIS problem. Assume unipolar value range for neuron outputs and let an active neuron, z_i=1.0, indicate that the corresponding vertex, V_i, is selected for inclusion in the MIS. A second constraint is that the number of vertices that are not in the subset (inactive neuron count) needs to be minimized. Accordingly, the following error function can be utilized to represent the problem constraints:

$$E = \frac{1}{2} g_a \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j\neq i}}^{N} e_{ij} z_i z_j + \frac{1}{2} g_b \sum_{i=1}^{N} (1 - z_i) \tag{3}$$

where it holds that $g_a, g_b \in R^+$. This energy function has the globally minimum value if the first term is zero and the second term is a positive value that is as small as possible. The first term is zero if no two adjacent vertices are active while the second term is minimum if a minimum number of neurons are inactive (corresponding to the case for the smallest possible number of vertices being excluded from the maximum cardinality subset). Again associating this error function with the Hopfield network Liapunov function in Equation 1 will facilitate definition of weights, biases and thresholds for the neural dynamics, effectively mapping the problem to the Hopfield network dynamics. The weights and neuron biases of the Hopfield network for the MIS problem are as follows:

$$w_{ij} = -2 g_a e_{ij} \quad \text{and} \quad b_i = g_b \tag{4}$$

According to the Equation 4, the weight values are determined by g_a and e_{ij} and the bias is determined by g_b. Finding values for the constraint weighting parameters (such as g_a and g_b) so that the Hopfield neural network converges to a state that is a solution for a static optimization problem is described elsewhere [14]. Derivation of values for these parameters for the MIS problem for N-neuron Hopfield network is shown next. For an inactive neuron in a solution state of the MIS problem mapped to Hopfield network, we know that $net_i < 0$. Therefore, the following holds:

$$\sum_{j=1}^{N} w_{ij} z_j + b_i < 0, \text{ which leads to } -2 g_a \sum_{j=1}^{N} e_{ij} z_j + g_b < 0$$

if the value of w_{ij} in Equation 4 is substituted. Finally, the algebraic manipulation results in the following form:

$$\sum_{j=1}^{N} e_{ij} z_j > \frac{g_b}{2 g_a}.$$

The sum term on the left has an upper bound which is the maximum number of neighbors for any given node in the graph that models the sensor network or equivalently the neural network. Hence assuming that M (which is a positive integer value) is the maximum connectivity value, then the above equation yields

$$2M g_a > g_b. \tag{5}$$

A similar derivation for an active neuron in a solution of MIS problem mapped to Hopfield network does not yield a different value, when compared to values through Equation 5, for the two constraint weighting parameters g_a and g_b.

The set of differential equations for the neurons in Hopfield network as given by Equation 2 are discretized for simulation or realization on a digital system. There is extensive literature on simulation of Hopfield networks dynamics and its stochastic variants [15] which can readily be leveraged. Asynchronous update of neuron outputs, i.e. one neuron at a time, which is a natural fit for wireless sensor network context, is desirable for convergence to a fixed point rather than a limit cycle.

4. Embedding and Operation of Hopfield Net Optimizer in Wireless Sensor Network

Assume a wireless sensor network (WSN) with *N* motes is given. Each WSN mote is assigned a single (Hopfield net) neuron: each mote computationally implements a single neuron (i.e. calculates the *k*-th iteration value for the discrete-time equivalent of the dynamics equations given in Equation 2) along with the storage needed for the weight vector for the neuron, bias and threshold, nonlinearity slope, and others. A recurrent neural network and its mapping to wireless sensor network under the "one neuron per mote" dictum are presented in Figure 1. The weight vector, and bias and threshold terms for a given neuron residing on a given mote are initialized to the values obtained through resolving Equations 1 with 3 to determine values for weights, biases and thresholds as shown in Equation 4. In general, any given neuron can talk to any other neuron in the network (through multi-hop communications over the WSN in many cases), and thus establishing the required connectivity of the Hopfield neural network as dictated by the specific optimization problem error or energy function. Connections to neurons on one-hop neighbor motes as dictated by the current trans-receiver range settings will be direct or without any intermediaries. Connections to neurons residing on motes that are not one-hop neighbors of the current mote will be over multiple hops. In the case of the MIS problem, each neuron will need to receive inputs from those neurons residing on motes that are one-hop neighbors for the mote that is the host as indicated by the error function formulation for the MIS problem in Equation 3.

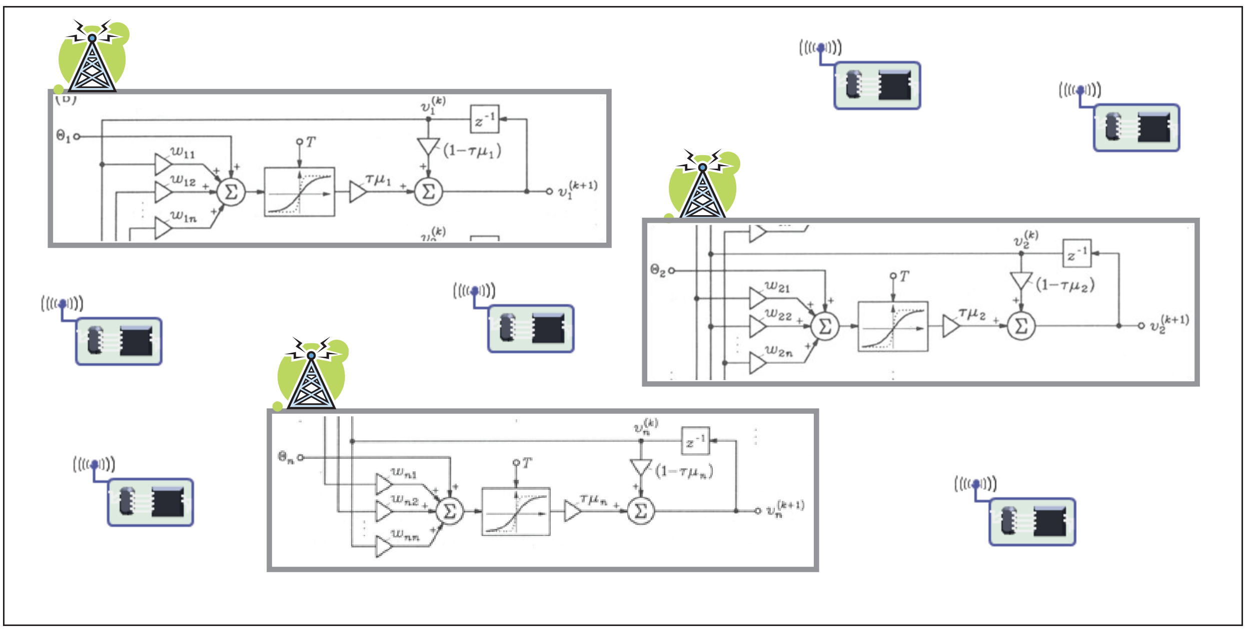

Figure 1. WSN with Hopfield neuron model [16] embedded within a mote.

5. Conclusions

This paper proposed a methodology to map a Hopfield recurrent neural network configured for a static optimization problem, the maximum independent set, to a wireless sensor network as a hardware computing platform for fully parallel and distributed computation. A wireless sensor network (WSN) was identified and proposed as a hardware platform for fully parallel and distributed computing of neural network algorithms. Utilization of a WSN as a parallel and distributed hardware platform was demonstrated for a Hopfield recurrent neural network configured for the maximum independent set problem.

The presentation in this paper as the first publication in a three-paper sequence, which demonstrated the feasibility of such a mapping on a case study, is easily generalizable to other neural network algorithms and a wide selection of application and protocol-level problems for wireless sensor networks. The second paper [17] in the three-paper sequence will discuss the nesC-TinyOS model for the Hopfield network configured for the maximum independent set problem and embedded into a wireless sensor network. TOSSIM simulation of the nesC-TinyOS model for the Hopfield neural network configured to solve the maximum independent set problem will be discussed in the third and the final paper [18] of the sequence.

References

[1] Stankovic, John A., Wood, Anthony D. and He, Tian, Realistic Applications for Wireless Sensor Networks in Theoretical Aspects of Distributed Computing in Sensor Networks, Monographs in Theoretical Computer Science. An EATCS Series, Eds. Nikoletseas, Sotiris and Rolim, José D.P., Springer Berlin Heidelberg, pp. 835-863, 2011. Doi: 10.1007/978-3-642-14849-1_25

[2] Barbancho, J. Leon, C. Molina, J. and Barbancho, A. “Using artificial intelligence in wireless sensor routing protocols,” Computer Communications,Vol.30, pp. 2802-2811, 2006.

[3] Ergut, S.; Rao, R.R.; Dural, O.; Sahinoglu, Z., "Localization via TDOA in a UWB Sensor Network using Neural Networks", IEEE International Conference on Communications (ICC), ISBN: 978-1-4244-2075-9, pp. 2398-2403, 2008.

[4] Guan, P. and Li, X. “Minimizing Distribution Cost of Distributed Neural Networks in Wireless Sensor Networks”, GLOBECOM, pp.790-794, 2007.

[5] Hosseingholizadeh, A. and Abhari, A.” A neural network approach for wireless sensor network power management”, in Proc. 28th IEEE Inter. Symp. on Reliable Distributed Systems, Niagara Falls, NY, USA. 2009.

[6] He, H. Zhu, Z. Mäkinen, E. “A neural network model to minimize the connected dominating set for self-configuration of wireless sensor networks,” IEEE Transactions on Neural Networks. Vol. 20(6), pp. 973-982, 2009.

[7] Jankowska, A. Schut, M. C. Ferreira-Schut, N. "A Wireless Actuator-Sensor Neural Network for Evacuation Routing," International Conference on Sensor Technologies and Applications, Athens, Greece. pp. 139-144, 2009.

[8] Kulakov, A. and Davcev, D. “Data mining in wireless sensor networks based on artificial neural network algorithms”, in Proc SIAM International Conference on Data Mining, Newport Beach, CA 2005.

[9] Paladina, L. Biundo, A. Scarpa, M. and Puliafito, A. Artificial Intelligence and Synchronization in Wireless Sensor Networks, Journal of Networks, Vol. 4, No. 6, pp 382-391, August 2009.

[10] Szymanski, K. and Chen, G.-G. “Computing with time: from neural networks to sensor networks,” The Computer Journal, Vol. 51(4), pp. 511-522, 2008.

[11] Xu, Y. Ford, J. Becker, E. and Makedon, F. “BP-Neural Network Improvement to Hop-counting for Localization in Wireless Sensor Networks,” in Tools and Applications with Artificial Intelligence, Springer Berlin - Heidelberg, 2009, pp.11-23.

[12] Youssef, W. and Younis, M. “A cognitive scheme for gateway protection in wireless sensor network,” Applied Intelligence, Vol. 29, No.3, pp216-227, 2008.

[13] Hopfield J. J., and Tank, D. W. “Neural Computation of Decision in Optimization Problems,” *Biological Cybernetics,* Vol. 52, pp. 141-152, 1985.

[14] Serpen, G., and Livingston, D. L., “Determination of Weights for Relaxation Recurrent Neural. Networks,” Neurocomputing, Vol. 34, pp. 145-168, 2000.

[15] Smith, K. Potvin, J-Y. and Kwok, T. "Neural network models for combinatorial optimization: deterministic, stochastic and chaotic approaches", *Control and Cybernetics*, vol. 31, no. 2, pp. 183-216, 2002.

[16] Cichocki, A. and Unbehauen, R. *Neural Networks for Optimization and Signal Processing*, John Wiley & Sons Ltd. & B. G. Teubner, Stuttgart, 1994.

[17] Li, J. and Serpen, G., “A nesC-TinyOS model for parallel and distributed computations of max independent set by Hopfield network on a wireless sensor network,” Proc. of Complex Adaptive Systems Conference, Chicago, 2011.

[18] Li, J. and Serpen, G., “TOSSIM simulation of wireless sensor network serving as hardware platform for Hopfield neural net configured for max independent set,” Proc. of Complex and Adaptive Systems Conference, Chicago, 2011.