

Building Distributed Soft Sensors

S. G. Soares¹, T. B. Takáo², A. F. da Rocha³, R. A. de M. Araújo¹ and T. M. G. de A. Barbosa²

¹ ISR-Institute of Systems and Robotics, and DEEC-Department of Electrical and Computer Engineering, University of Coimbra, Coimbra, Portugal.
symonesoares@isr.uc.pt and rui@isr.uc.pt

² Department of Computing, Pontifical Catholic University of Goiás, Goiânia, Brazil
eng.thales@gmail.com and talles@ucg.br

³ Department of Electrical Engineering, University of Brasília, Brasília, Brazil
adson@unb.br

Abstract: Advances in digital electronics have enable the development of low-cost, low-power, multifunctional sensor nodes that are small in size and communicate in short distances. These tiny sensor nodes consist of sensing, data processing, and communication components, leverage the idea of Wireless Sensor Networks (WSN) based on collaborative effort of a large number of nodes. Despite the high number of publications dealing with WSN applications, there are still some potential to be explored in WSN development and maintenance. Recent contributions of Intelligence Modeling have made possible the construction of Smart Sensors, reducing maintenance costs and time prototyping. In this scenario, Neural Networks (NNs) have found many successful applications in nonlinear system identification and control, digital communication, pattern recognition, pattern classification, etc. Many similarities between NN and WSN can be found and explored to improve WSN application process by reducing the development costs. For example, the sensor node itself can be seen as an artificial neuron, since the WSN application shows characteristics such as distributed representation and processing, massive parallelism, learning generalization ability, adaptively, inherent contextual information processing, fault tolerance and low computation. This paper examines the hybridization with NN and WSN into a Smart Home application, called Smart Table. Preliminary prototypal results have shown that Multilayer Perceptron is good candidate for using into low-cost System-on-a-Chip (SoC) such as PIC microcontrollers.

Keywords: Wireless Sensor Network, Neural Network, Smart Home, Soft Sensors.

I. Introduction

Wireless Sensor Networks (WSNs) have become an attractive technology for the research community, particularly with the proliferation in Micro-Electro-Mechanical Systems technology which has facilitated the development of smart sensors [1]. Typically, a WSN is a distributed system that is composed of autonomous units with sensing capabilities (sensor nodes), interconnected by wireless communication system. This network offers potentially low-cost solution to several problems including military target tracking [2], health care monitoring [3],

environment control systems [4], animal monitoring [5], and Smart Homes [6].

In this scenario, a smart sensor is expected to have capability that functionality and architecture as well as raw data acquisition are based on the existence of micro-processing unit. The purposes of development of this type of sensors are network and scalable, energy-saving, smart and programmable, capable of fast data acquisition, reliable and accurate over the long term, cost little to purchase and easy-to-install [7].

As the demand for smart sensor increases, one cannot expect that the necessary data or domain knowledge will always be available to support. Consequently applications of WSN provide an especially strong motivation for the study of learning-theoretic approach [10], since distributed intelligence systems have a proven ability to autonomously assess their environment, learn from their experience and communicate their knowledge, conclusions, and reasoning to others.

This technique is also known as “cognitive sensor network”. Cognitive techniques and tools of artificial intelligence are being increasingly used in WSNs to circumvent the limitation imposed by existing techniques. “Cognition” refers to the process of knowing through perception, reasoning, knowledge and intuition. When nodes with cognitive capabilities are introduced into an entire network of communicating in sensor nodes, it gives rise to exciting new opportunities in sensor network research that could overcome the limitations imposed by current design techniques [8]. Therefore, merging intelligence with WSN technology looks promising for the revolutionary development of intelligent system theory and applications as well as networking technology.

In intelligent learning system, Neural Networks (NNs) are a powerful tool in solving complex problems, because of their universal approximation capabilities. The NNs have found many successful application in nonlinear system identification and control, digital, communications, pattern recognition, pattern classification, etc. Several applications of NN-based techniques have been proposed [8][9]. Due to their partly parallel network topology, NN could feasibly distribute its operation over computer networks. Most

applications in this field are based on supervised learning such as Multilayer Perceptron (MLP), and unsupervised learning such as Self-Organization Maps (SOMs).

There are many similarities between WSN and NN. For example, the sensor node itself can be seen as an artificial neuron since the WSN application show characteristics such as distributed representation and processing, massive parallelism, learning generalization ability, adaptively, inherent contextual information processing, fault tolerance and low computation. Thus, NN can be easily embedded into WSN to promote new application developments. This kind of application is a promising research field computational intelligence focusing on synergistic combinations of multiple approaches to develop intelligent systems [11].

This paper examines the possibility of embedding NN and WSN into a Smart Home application, called Smart Table. This application is an interesting case study, because it requires at least distributed representation and processing, pattern recognition and pattern classification. Moreover, this work will explore some aspects related to the topological organization, data fusion and programming support. These aspects are fundamentals to put into operation artificial neurons building insider sensor nodes. Besides, this application can be useful contribution in other application areas such as Soft Sensor and other applications in industrial environments. Preliminary prototypal results have shown that NN models are good candidates particularly for using it deployed into low-cost System-on-a-Chip (Soc), such as PIC microcontrollers.

The next sections give an overview of the current building distributed Soft Sensors. Section 2 presents the hybridization with Smart Home and WSN. Section 3 shows how to construct distributed NN. Section 4 displays concepts of NN model. Section 5 presents the design of the Smart Table project. Section 6 shows the principal experimental results. Finally, Section 7 discusses the results and some future perspectives for works.

II. Smart Home and WSN

Advances in smart devices, mobile wireless communication, sensor networks, pervasive computing, machine learning, middleware and agent technologies, and human-computer are allowing computing systems with various capabilities and interfaces to become part of daily life. Sensors, actuators, and processing units can now be purchased at very affordable prices. These technologies can be networked and used with coordination of highly intelligent software to understand the events and relevant contexts of a specific environment and to take sensible decisions in real-time or *a posteriori*. Computing devices will have to be coordinated by intelligent systems that integrate the resources available to provide intelligent environments [13].

The term “smart home” is used for a residence equipped with technology that allows monitoring of its inhabitants and/or encourages independence and the maintenance of good health [14]. Smart homes are equipped with sensors, actuators, and/or biomedical monitors. Usually, devices operate in a network connected to a remote centre for data collection and processing.

A general concept is “smart environment”. In this small world all kinds of smart devices are continuously working to make inhabitants’ lives more comfortable. These devices provide varied and impressive capabilities. When networked together and tied to intelligent sensor and the outside world, the impact of these devices becomes even more powerful. In [15] these intelligent devices are called appliances. There are many kinds of appliances, such as smart table [11], adaptive furniture [16], intelligent bed for monitoring the quality of sleep [17], coffee makers, microwaves, refrigerators, etc.

Wireless sensor nodes can be used to create a smart environment. And the nodes can be embedded into furniture and appliances, and they can communicate each other, such as by wireless. The implementation of an appliance through a WSN has some benefits, such as, processing capability, distributed processing capacity, sensing capability, etc.

Another fundamental aspect in projects of appliance using WSN refers to the capacity of data fusion, inherent in a WSN. Data fusion is a very ample term, and used in several studies as a formal structure in which are expressed methods and tools to data junction origination of different source. Your goal is higher obtaining of higher quality information, the exact definition of “higher quality” will depend of the application [18].

According [19], data fusion may be segmented at various levels of abstraction. In the lower level are basic features, such as filtering data. In the intermediary level, data are transformed into information in order to may be used in decisions making the higher level. For example, pattern recognition is a feature of intermediary level.

In a Smart Home, a variety of different sensors, such as, measuring light, sound, video cameras, temperature sensors, contact and motion can provide simultaneously multi-dimensional information about the inhabitant to support the inference of activity determination. In the home security application, information of the sensors can be used to extract a human voice, open/closed door, an animal, a personal, broken window, etc. Redundant information can come from different sensors, for example, the information “personal” can be extracted from sensors video cameras and sound. These processes have in common that the data that come from the sensor are processed, redundant data can be fused via low level data fusion, and an algorithm is used to extract the variable for each type of data and fuse the redundant variables in the high level fusion [19].

In the high level, some models may be used in the data fusion implementation to transform data. Data fusion can be classified into three different groups. First, fusion based on probabilistic models, second, fusion based on least-squares techniques and, third, intelligent fusion. The probabilistic model methods are Bayesian reasoning, evidence theory, robust statistics and recursive operators. The least-squares techniques are Kalman filtering, optimal theory, regularization and uncertainly ellipsoids. The intelligent fusion methods are fuzzy logic, NNs and genetic algorithms [20].

III. Distributed Neural Networks

Distributed learning is a particular problem both WSN and NN. WSNs are a fortiori designed for the purpose of

making inferences about the environments that they are sensing, and they are typically characterized by limited communication capabilities due to tight energy and bandwidth limitations. As a result, WSNs have inspired a resurgence in research on decentralized inference or learning [10]. On the other hand, some studies of NN have been suggested that intelligence emerges through a large number of interconnected processing elements in which any individual processing element performs a simple computational task, beyond more flexibility in design and implementation [12].

NN and WSN present similar characteristics. NN is a biologically inspired computational model that consists of processing elements (artificial neurons), connections (weights) between them and an output. The sum of all weights and a threshold determine the output of neuron. Analogically to the neuron, a node sensor converts physical quantities to an electrical output signal which is filtered similarly to weighting. Hence, a sensor node can be extended in a WSN topology. Then, a WSN can be seen as a NN, which a sensor node inside the WSN could run an NN model to decide on the output action.

The division of a NN in units or modules is also known as Modular Neural Network (MNN). NN is said to be a modular if the computation performed by the network can be decomposed into two or more modules (subsystems) that operate on distinct inputs without communicating with each other. The benefits of creating a MNN are model complexity reduction, robustness, scalability, computational efficiency, learning capacity, economy of learning and knowledge integration [21]. The MNN characteristics enable the integration with a WSN whose advantages are mainly low-cost, small physical dimensions and incremental processing capacity.

When a MNN is embedded into a WSN, it can assume several configurations. A MNN can receive different information source (different sensors) available; use heterogeneous or homogenous NN models; different combination mechanisms for NN modules; cooperative or competitive modules; hierarchical architecture; etc. Figure 1 shows three possibilities of MNN architectures into a WSN. The first layer is composed by wireless sensor nodes and their inputs (sensors). The second layer combines the outputs of first layer in a central sensor node. This node can

contain algorithms for data fusion, or another NN. In (a), all NN models share inputs (or sensors), and the sensor node f is able to combine NN outputs of the sensor nodes n . In (b), the NNs in the first layer have different inputs, S and T . The inputs S can be different types of sensors, such as luminosity or temperature. In (b), the NN also not are shared inputs. At second layer, the NN contained in the *sensor node f* can combine NN outputs in order to generate relevant information to the system.

Embedding a complex neural structure into a node sensor gives rise some challenges, for example, energy consumption that is directly proportional to computation complexity. In [9] is proposed a NN called Laguerre Neural Network. The goal of this technique is to reduce the complexity and energy consumption in a sensor node. Moreover, embedding NN into a sensor node imposes some limitation on the architecture. However, more optimized NN implementations with this purpose are required. In [22], a NN is implemented on a low and inexpensive microcontroller.

IV. Neural Networks

Artificial Neural Networks perform the *subsymbolic* paradigm of representing and processing of information. The area of science that deals with methods and systems for information processing using NNs is called *neurocomputation* [23]. The learning ability of NN is achieved through applying a learning (training) algorithm. Training algorithms are mainly classified into three groups: supervised, unsupervised and reinforcement learning. In supervised learning, the target outputs are known and given to the NN during training, so that the NN can adjust its weights to try matching its outputs to the target value. Multilayer Perceptron (MLP) and Radial Basis Function are examples of supervised learning. In unsupervised learning, the NN is not provided with the correct results during training. The principal unsupervised NNs are Self-Organizing Map Kohonen and Adaptive Resonance Theory (ART). In reinforcement learning, no desired category of target value is given, instead, only a “teacher” is available, and the learner receives a feedback about the appropriateness of its response.

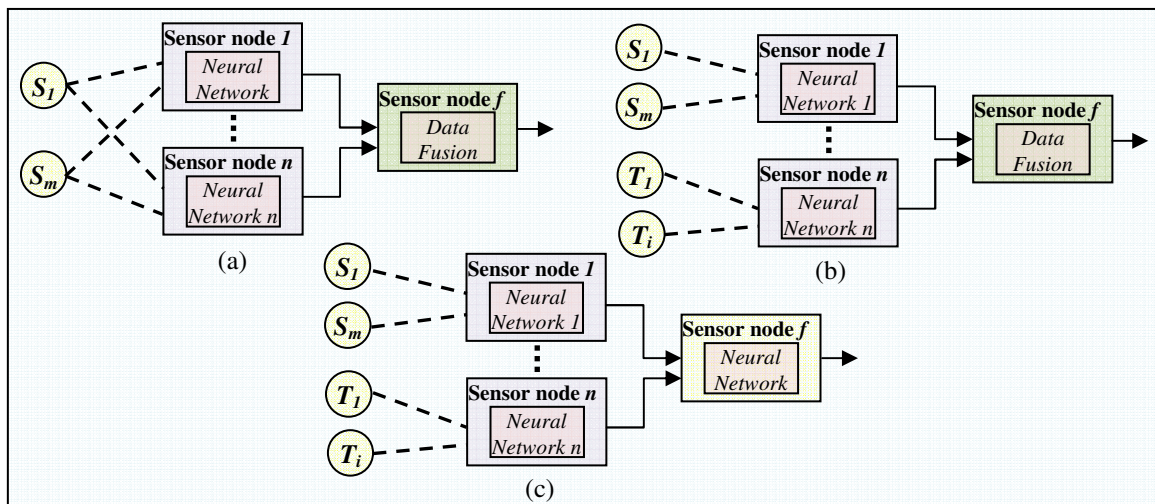


Figure 1. Examples of Distributed Neural Networks

A. Multilayer Perceptron (MLP)

The Multilayer Perceptron (MLP) consists of multiple layers of simple, two-state, processing elements (nodes) or neurons that interact using weighted connections. After a lowermost input layers, there are usually any numbers of intermediate, or hidden, layers followed by an output layer at the top. There exist no interconnections within a layer while all neurons in a layer are fully connected to neurons in adjacent layers. Weights measure the degree of correlation between the activity levels of neurons that they connect. An external input vector is supplied to the network by clamping it at the nodes in the input layer [24]. The MLPs were put into practice only when learning algorithms were developed for them, one of them being the so-called *backpropagation algorithm* (BP) [23].

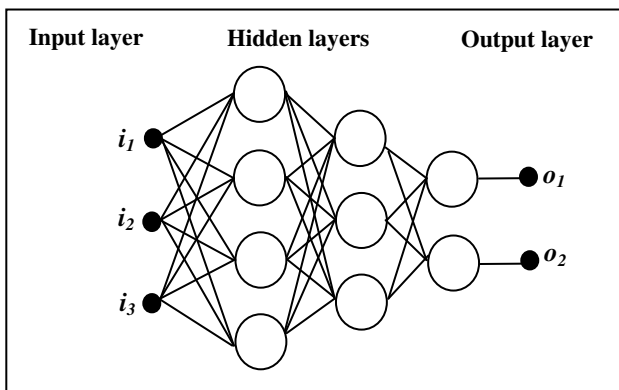


Figure 2. Multilayer Perceptron with Two Hidden Layers

The BP training algorithm uses gradient descent procedure to attempt to locate the absolute (or global) minimum of the error surface. BP has been shown to perform adequately in many applications. The weights in the network are initially set to small random values. The BP algorithm then calculates the local gradient of the error surface and changes the weights in the direction of steepest local gradient. The learning cycle of training algorithm consists of two principal steps: a *forward pass*, when inputs are supplied and propagated through the intermediate layers to the output layer; and, a *backward pass*, when an error is calculated at the outputs and propagated backward for calculating the weights' changes. The BP algorithm can be divided into:

1. initialize network weights;
2. apply an input vector x and its corresponding output vector y ;
3. propagate forward the vector x through all neurons in all the layers and obtain the output signals;
4. calculate an error signal for every output neuron, by comparing actual output to the desired output;
5. adjust the weights between the intermediate neurons and output neurons;
6. calculate the error for neurons in the intermediate layer;
7. propagate the error back to neurons of lower level;
8. repeat steps 2-7 with next input vector, until overall error is satisfactorily small.

This BP implementation is known as on-line training whereby the network weights are adapted after each pattern

has been presented. The alternative is known as batch training, where the summed error for all patterns is used to update the weights. The BP algorithm is very time-consuming and various modifications are suggested trying to improve its performance [23].

V. Embedding Neural Networks into WSN Smart Table

This paper discusses a Smart Table as a case study. This appliance enables interacting with objects on its surface through localization and identification of shape of these objects. The purpose of this furniture is the interaction of objects in a Smart Home environment. However, the smart table can be adapted for several areas. For example, in education, kids can match different objects on the table surface, and then, teachers can observe the learning progress of a particular child.

The principal objectives of the smart table are localization and classification of multiples objects on its surface. As shown in Figure 3, the appliance can identify how many objects are on the surface, beyond to classify their shapes in *small* and *large*.

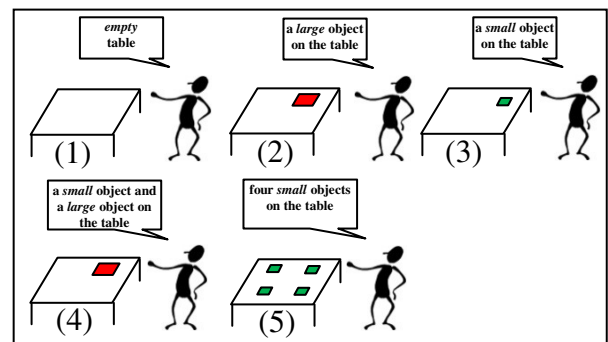


Figure 3. Main Interactions with Smart Table

Our goal is to implement a WSN which supports the requirement of the smart table. This includes the construction of NN models into each sensor node in the network. The mapping of NN into nodes brings some advantages in system architecture. Such as, it increases the performance, because hybridization reduces the abstraction in software, besides it reduces the time of prototyping, since the process is composed of a smaller number of activities.

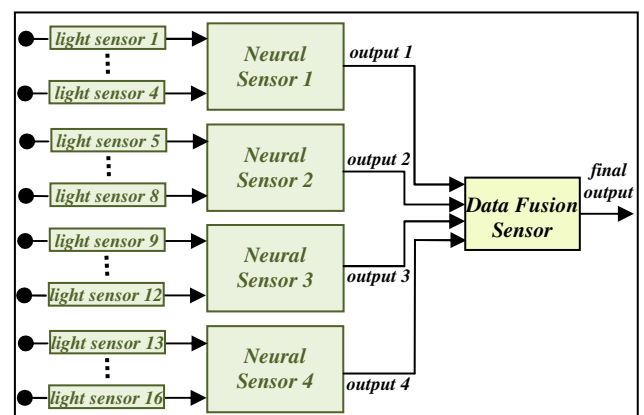


Figure 4. System Architecture

A. System Architecture

Figure 4 displays the system architecture which uses two layers. The first layer contains neural sensors, and the second layer contains a data fusion sensor node. In this prototype, each neural sensor contains two neural models, a Perceptron (a single layer NN) and a MLP. Each neural sensor is connected to four light sensors. And the outputs of neural sensors are connected to data fusion node. This block is able to unite the information of each input block and generate the application states.

On the other hand, the Figure 5 shows another view of the proposed system. Smart Table is mapped into four sensor nodes. Each sensor node works like an artificial sensor that captures luminosity and identifies an object on its surface through NN models. Sensor nodes are interconnected by ring through physical standard RS-232. And data of each node cross the network in circular shape until the node 4. State information of the table is requested through the user's application contained in the PC. The requisition is received by the node 4 which sends to the PC. Communication between the PC and the Smart Table is performed through the wireless standard *ZigBee*.

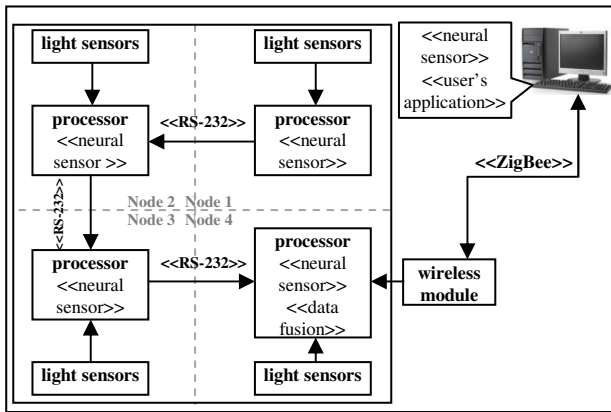


Figure 5. Overview of the Developed Prototype

B. Hardware Design Description

Figure 6 shows hardware architecture of a neural sensor. A node consists of a PIC18F4550 microcontroller and four light sensors. The PIC18F4550 is a specific purpose chip. This circuit has features such as: 40 pines, 32Kbytes flash memory, 2Kbytes SRAM memory, 256bytes EEPROM memory, 35 i/o pines, 13 i/o conversion channels, RS-232 communication and USB.

The communication standard used for interconnected nodes, RS-232, performs asynchronous serial communication and uses two routes for data: transmission (TX) and reception (RX). In the RS-232 data transmission, a RX line of a node is connected to the TX line from another node. On the other hand, *ZigBee* wireless standard is used to connect the appliance a computer. *ZigBee* is a standard with focus on application for monitoring and sensing, and, may control 65535 nodes in a network with low power consumption. Choose *ZigBee* possibilities more convenience to the system due to absence of wires and, also, enables the construction of a home network with several interconnected appliances.

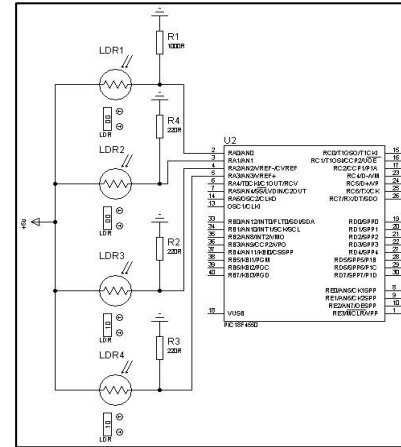


Figure 6. Hardware of the Neural Sensors

The communication standard used for interconnected nodes, RS-232, performs asynchronous serial communication and uses two routes for data: transmission (TX) and reception (RX). In the RS-232 data transmission, a RX line of a node is connected to the TX line from another node. On the other hand, *ZigBee* wireless standard is used to connect the appliance a computer. *ZigBee* is a standard with focus on application for monitoring and sensing, and, may control 65535 nodes in a network with low power consumption. Choose *ZigBee* possibilities more convenience to the system due to absence of wires and, also, enables the construction of a home network with several interconnected appliances.

Light sensor is a low-cost sensor which reduces electrical resistance when it captures a light energy. So that the light captured may be converted into voltage, each sensor is configured as a voltage divisor. In Equation (1) shows behavior of the LDR sensor wherein V_{SAIDA} is output voltage divisor, R_{LDR} is LDR resistance and R is a resistance connected in series to the sensor.

$$V_{SAIDA} = V_{CC} * R / (R + R_{LDR}) \tag{1}$$

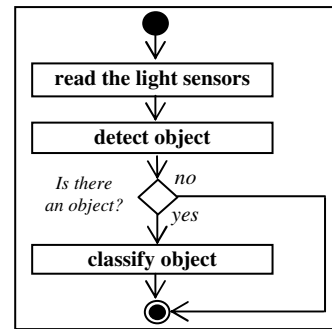


Figure 7. Activity Diagram of the Neural Sensors

C. Software Design Description

Each neural sensor node is composed by a Simple Perceptron and a MLP. The Perceptron is able to detect the presence of objects on each node. And the MLP routine classifies objects on the table. Figure 7 shows the activity diagram of the software developed. First, a node sensor reads the light sensors. Later, a Perceptron detects objects.

And finally, if an object is found on the table, a MLP is called to classify objects. Otherwise, the node ends its activity. The division of the software into two neural models reduces the time of processing, because the MLP is executed just if an object is detected.

1) Simple Perceptron

Simple Perceptron is an NN model used to detect objects. The development process of the Perceptron is divided in: NN configuration, training, test and integration. The NN configuration of the Perceptron is a neuron in a single layer. The light sensors work as inputs for the neuron. In the training, the NN is able to adjust the weights in order to later classify light patterns. In this work, the training is done on the computer. This technique not affects the performance of the system, because NN is a robust model, and even if one of the weights is damaging on microcontroller, the system would work well. In the Perceptron learning is used AND logic function and considers “true” state as light presence. Thus, Perceptron produces “true” output when all sensors are “true”. On the other hand, if at least one of sensors is “false”, the Perceptron produces output “false”. At the end of training, weights will be returned by the program. The integration in the microcontroller requires: sensor reading, conversion of reading, normalization, activation function calculation and threshold function calculation.

2) Multilayer Perceptron

The MLP model is able to classify patterns of objects according to *large* and *small*. The MLP contains three-layer configuration. In the reference model, the amount of inputs is equal to neurons in the hidden layer, i.e. four neurons; and the output layer has one neuron. The training was performed with 16 (2^4) samples. The logic used in the training is: if three or four sensors are without light, then object is classified as *large*; and zero to two sensors are without light, then object is *small*. At the end of training, the weights were integrated in the microcontroller PIC.

VI. Experimental Results

The goal of following tests is to evaluate the implementation of a neural model in a node sensor.

The first test aims to investigate the number of hidden neurons that the microcontroller can support. Figure 8 shows the percentage of ROM and RAM memory consumption based on the number of hidden neurons. As can be seen, there is a linear increase in memory consumption when the number of hidden neurons is expanded. On the other hand, Figure 9 displays the time of response in a microcontroller based on the number of hidden neurons. Time of response includes reading of the sensors and neural classification of the node.

The second test examines the number of inputs (or light sensors) that the microcontroller can contain. Figure 10 shows the amount of inputs according to memory

consumption. And Figure 11 displays the number of inputs based on the time of response. As can be seen, the time of response grows as the number of sensors increases.

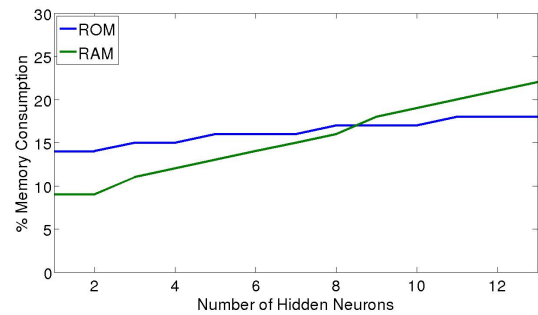


Figure 8. No. of Hidden Neurons versus Memory

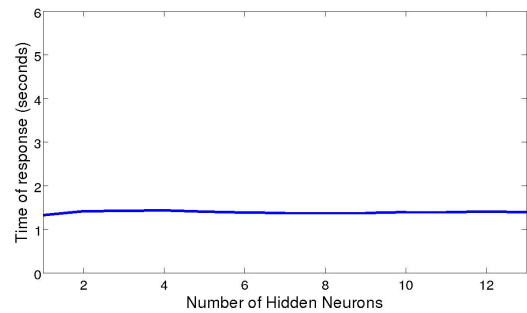


Figure 9. No. of Hidden Neurons versus Time

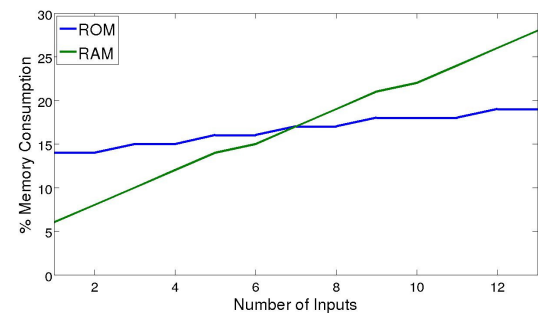


Figure 10. No. of Inputs versus Memory

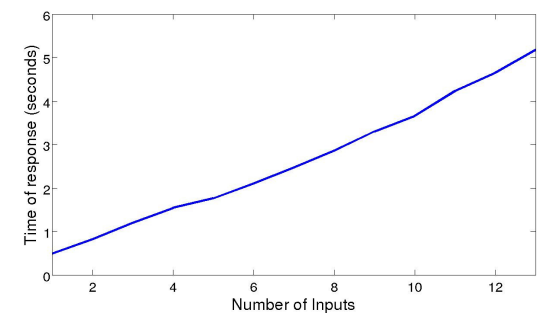


Figure 11. No. of Inputs versus Time

Third test analysis the final output of the neural node. At the end, the microcontroller is able to classify objects in *none*, *small* and *large*, as shown in the Table 1. The shape of the *small object* is 15x15cm, and the *large object* is 30x30cm. In the Table 1, column “d” represents the distance between sensors and objects. The “size of object” shows the shape of the object used in the test. The “output of NN” displays the output of the neural sensor. “Result” analysis if the sensor classified correctly the objects. This test displays that only 7 and 9 have incorrect results. It is attributed by the low accuracy of the sensor light.

Experiment	Covered Sensors	d (cm)	Size of Object	Output of NN	Result
1	0	*	None	None	Correct
2	1	0	Small	Small	Correct
3	1	10	Small	Small	Correct
4	2	0	Small	Small	Correct
5	2	10	Small	Small	Correct
6	3	0	Large	Large	Correct
7	3	10	Large	Small	Incorrect
8	4	0	Large	Large	Correct
9	4	10	Large	Small	Incorrect

Table 1. Main Results of the Smart Table

VII. Conclusions

In this paper, we analyzed the hybridization with NN and WSN into a Smart Home application called Smart Table. This appliance is interesting as a case study, because it requires distributed processing and representation, pattern recognition and pattern classification, which are advantages inherent of WSN and NN applications. The mapping and hybridization of these networks provide benefits, such as, reduction of the time of prototyping and composition of the system in a smaller number of activities. Preliminary tests show possible to integrate a chip with low memory capacity and a complex computational model. Moreover, the work shows that the introduction of new inputs increases time of response, but the increasing of the number of hidden neurons not affects the time of response. Another test is performed to verify whether the outputs of the network are corrects. The test shows that just two outputs are incorrect. This occurs due to the low accuracy of the light sensor. Future works include the development of a programming language which enables the direct mapping of the functional requirements that describe an application in a NN model and, consequently, in a WSN topology. Moreover, this hybridization model can be expanded in others models, such as, Hidden Markov Model and Self-Organizing Maps.

VIII. References

- [1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless Sensor Network Survey", *Computer Networks*, Vol. 52(12), 2008, pp. 2292-2330.
- [2] S. Tripathy, and S. Nandi, "Defense Against Outside Attacks in Wireless Sensor Networks", *Computer Communications*, Vol. 31 (4), 2008, pp. 818-826.
- [3] N. A. Benjamin, and S. Sankaranarayanan, "Performance of Wireless Body Sensor based Mesh Network for Health Application", *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)*, Vol. 2, 2010, pp. 020-028.
- [4] M. Nakamura, A. Sakurai, and J. Nakamura, "Autonomic Wireless Sensor/Actuator Networks for Tracking Environment Control Behaviors", *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)*, Vol. 1, 2009, pp. 125-132.
- [5] J. I. Huircan, C. Munoz, H. Young, L. V. Dossow, J. Bustos, G. Vivallo, and M. Toneatti, "ZigBee-Based Wireless Sensor Network Localization for Cattle Monitoring in Grazing Fields", *Computers and Electronics in Agriculture*, Vol. 74(2), 2010, pp. 258-264.
- [6] E. Farella, M. Falavigna, and Bruno Ricco, "Aware and Smart Environments: The Casattenta Project," *Microelectronics Journal*, Vol. 41(11), 2010, pp. 697-702.
- [7] J.-D. Son, G. Niu, B.-S. Yang, D.-H. Hwang, and D.-S. Kang, "Development of Smart Sensors System for Machine Fault Diagnosis", *Expert Systems with Applications*, Vol. 36(9), 2009, pp. 11981-11991.
- [8] G. Vijay, E. Bdira, and M. Ibnkahla, "Cognitive Approaches in Wireless Sensor Networks: A survey", *Communications (QBSC), 2010 25th Biennial Symposium on*, 2010, pp.177-180.
- [9] J. C. Patra, C. Bornand, and P. K. Meher, "Laguerre Neural Network-Based Smart Sensors for Wireless Sensor Networks", *Instrumentation and Measurement Technology Conference, 2009. I2MTC '09. IEEE*, 2009, pp.832-837.
- [10] J. B. Predd, S. B. Kulkarni, and H. V. Poor, "Distributed Learning in Wireless Sensor Networks", *Signal Processing Magazine, IEEE*, Vol. 23(4), 2006, pp. 56- 69.
- [11] S. G. Soares, A. F. da Rocha, T. M. G. A. Barbosa, and R. A. M. Araújo, "Embedding a Neural Network into WSN Furniture", *10th IEEE International Conference on Hybrid Intelligent Systems*, 2010.
- [12] R. Polikar, "Ensemble Based Systems in Decision Making", *IEEE Circuits and Systems Magazine*, Vol. 6(3), 2006, pp. 21-45.
- [13] H. Nakashima, H. Aghajan, and J. C. Augusto. *Handbook of Ambient Intelligence and Smart Environments*. Springer, New Yourk, 2010.
- [14] M. Chan, E. Campo, D. Esteve, and J.-Y. Fourniols, "Smart Homes – Current Features and Future Perspectives", *Maturitas*, Vol. 64(2), 2009, pp. 90-97.
- [15] L. Jiang, D.-Y. Liu, and B. Yang, "Smart Home Research", *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, Vol. 2, 2004, pp. 659- 663.
- [16] A. Sprowitz, S. Pouya, S. Bonardi, J. Van den Kieboom, R. Mockel, A. Billard, P. Dillenbourg, and A. J. Ijspeert, "Roombots: Reconfigurable Robots for Adaptive Furniture", *Computational Intelligence Magazine, IEEE*, Vol. 5(3), 2010, pp.20-32.
- [17] A. Gaddam, K. Kaur, G. S. Gupta, and S. C. Mukhopadhyay, "Determination of Sleep Quality of Inhabitant in a Smart Home Using an Intelligent Bed Sensing System", *Instrumentation and Measurement Technology Conference (I2MTC), 2010 IEEE*, 2010, pp.1613-1617.
- [18] L. Wald, "Some Terms of Reference in Data Fusion", *Geoscience and Remote Sensing, IEEE Transactions on*, Vol. 37(3), 1999, pp.1190-1193.
- [19] H. S. Carvalho, W. B. Heinzelman, A. L. Murphy, and C. J. N. Coelho, "A General Data Fusion Architecture", *Information Fusion, 2003. Proceedings of the Sixth International Conference of*, Vol. 2, 2003, pp. 1465- 1472.

- [20] J.-W. Zhang, L.-P. Sun, and J. Cao, "SVM for Sensor Fusion - A Comparison with Multilayer Perceptron Networks", *Machine Learning and Cybernetics, 2006 International Conference on*, 2006, pp.2979-2984.
- [21] F. Azam, "Biologically Inspired Modular Neural Networks", *Faculty of the Virginia Polytechnic Institute, Virginia*, 2000.
- [22] N. J. Cotton, B. M. Wilamowski, and G. Dundar, "A Neural Network Implementation on an Inexpensive Eight Bit Microcontroller", *Intelligent Engineering Systems, 2008. INES 2008, International Conference on*, 2008, pp. 109-114.
- [23] N. K. Kasabov, "Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering", *MIT Press*, 1996.
- [24] S. K. Pal, and S. Mitra, "Multilayer Perceptron, Fuzzy Sets, and Classification", *Neural Networks, IEEE Transactions on*, Vol. 3(5), 1992, pp.683-697.



Talles Marcelo G. de A. Barbosa is an assistant professor in the Department of Computing at Pontifical Catholic University of Goiás (PUC Goiás) since 2001. He received a B.S. degree in Computer Science from PUC Goiás in 1999 and M.S. in Systems and Computing from Military Institute of Engineering in 2002 and Ph.D. degrees in Electrical Engineering from University of Brasília in 2008. His current research interests lie in the area of Applied Computing, including Medical Informatics, Embedded Systems and Mobile Robots.

Author Biographies



and Sensor Networks and Embedded Systems.

Symone Gomes Soares received degree in Computer Engineering from Pontifical Catholic University of Goiás in 2009. She is currently pursuing her Ph.D. degree in Electrical and Computer Engineering at the University of Coimbra. Since 2010, she is a Researcher at the Portuguese Institute for Systems and Robotics (ISR). Her research interests include Artificial Intelligence, Soft Sensors, Soft Computing



Thales Baliero Takáo is currently a Computer Engineering undergraduate student at Pontifical Catholic University of Goiás (PUC-GO). His B.Sc. Thesis focuses in hybridization between Wireless Sensor Networks, Self-Organizing Maps and Genetic Algorithms. His research interests include Artificial Intelligence, Embedded Systems, Wireless Sensor Networks and Communication Networks.



Adson Ferreira da Rocha graduated in Electrical Engineering from the University of Brasília, Brasil, received his M.S. degree from the State University of Campinas, Brasil, and the Ph.D. degree from the University of Texas at Austin. He is currently an Associate Professor at the Faculty UnB Gama. His research focuses on biomedical instrumentation and biomedical signal processing.



Rui Araújo received the B.Sc. degree ("Licenciatura") in Electrical Engineering, the M.Sc. degree in Systems and Automation, and the Ph.D. degree in Electrical Engineering from the University of Coimbra in 1991, 1994, and 2000 respectively. He joined the Department of Electrical and Computer Engineering of the University of Coimbra where he is currently an Assistant Professor. He is a founding member of the Portuguese Institute for Systems and Robotics (ISR-Coimbra), where he is now a researcher. His research interests include computational intelligence, intelligent control, learning systems, fuzzy systems, neural networks, control, embedded systems, real-time systems, soft sensors, industrial systems, sensor-based mobile robot navigation, and in general architectures and systems for controlling robot manipulators, and for controlling mobile robots.