Intelligent Wireless Sensor Networks Using FuzzyART Neural-Networks

Andrea Kulakov, Danco Davcev

Faculty of Electrical Engineering and Information Technologies
Ss Cyril and Methodius University, Skopje, Macedonia
E-mail: kulak@feit.ukim.edu.mk

Abstract

An adaptation of one popular model of neural-networks algorithm (ART model) in the field of wireless sensor networks is demonstrated in this paper. The important advantages of the ART class algorithms such as simple parallel distributed computation, distributed storage, data robustness and auto-classification of sensor readings are confirmed within the proposed architecture consisting of one clusterhead which collects only classified input data from the other units.

This architecture provides a high dimensionality reduction and additional communication savings, since only identification numbers of the classified input data are passed to the clusterhead instead of the whole input samples.

We have adapted and implemented the FuzzyART neural-network algorithm and used it for initial clustering of the sensor data as a sort of pattern recognition. This adaptation was made specifically for MicaZ sensor motes by solving mainly problems concerning the small memory capacity of the motes. At the final clusterhead - server, the data are stored in a database and the results of the data processing are continuously presented in a classification graph.

1. Introduction

Although neural networks have been extensively studied and developed in the last thirty years, their beneficial properties have not been used so far in wireless sensor networks, unlike other mathematical methods like nearest neighbor search, principal component analysis and multidimensional scaling (e.g [1] and [2]).

Sensor networks place several requirements on a distributed storage infrastructure. These systems are highly data-driven and are deployed to observe and analyze the physical world. A fully centralized data collection strategy is impractical, knowing the energy constraints on sensor node communication. It is also inefficient, given that sensor data have significant redundancy both in time and in space.

Unsupervised classification of the sensor readings is important in sensor networks since the data obtained with them is with high dimensionality and in huge amounts, which could easily overwhelm the processing and storage capacity of a centralized database system. On the other hand, the data obtained by the sensor networks are often self-correlated over time, due to the nature of the sensed physical phenomena, which are usually slowly changing. The data are redundant over space as well, due to the redundant sensor nodes dispersed near each other. Finally the data are also redundant over different sensor inputs due to the fact that often the sensor readings are correlated over different modalities sensed at one node (e.g. sound and light from cars in a traffic control application).

When the application demands compressed summaries of large spatio-temporal sensor data and similarity queries, such as detecting correlations and finding similar patterns, the use of a neural-network algorithm is a reasonable choice.

Still, up until recently, the only application of neural-networks algorithms for data processing in the field of sensor networks was [3], where they have slightly modified the Kohonen Self Organizing Maps model, and [4], where they have used another model of neural network for simulating real-time forest fire detection with wireless sensor networks.

In our previous work ([5] and [6]) we have shown through simulations how can some more sophisticated models of neural-networks, namely the ART models ([7], [8], [9], [10], and [11]), be used as data management algorithms in wireless sensor networks and how it can be a practical solution which reduces the amount of data being communicated.

We have chosen the ART neural networks mostly because of their documented advantages over other types of neural networks ([12]), one of them being the

1-4244-1521-7/07/$25.00 ©2007 IEEE

possibility for classifying the input data and learning new categories simultaneously and another one being the option not to constrain the number of different categories in which the input data will be clustered.

Recently, we have fully adopted and implemented the FuzzyART neural network algorithm [10] in MicaZ sensor motes in the same time solving problems regarding the small memory capacity of the motes.

2. Applying the FuzzyART Neural Network

2.1. The Original FuzzyART Algorithm

Adaptive Resonance Theory (ART) has been developed by Grossberg and Carpenter for pattern recognition primarily. Models of unsupervised learning include ART1 [9] for binary input patterns and FuzzyART [10] for analog input patterns.

ART networks possess several features such as robustness to variations in intensity, detection of signals mixed with noise, and model both short- and long-term memory to accommodate variable rates of change in the environment.

In Fig. 1, a typical representation of an ART Artificial Neural Network is given. Winning L2 category nodes are selected by the attentional subsystem. Category search is controlled by the orienting subsystem. If the degree of category match at the L1 layer is lower than the sensitivity threshold Θ, originally called vigilance level, a reset signal will be triggered, which will deactivate the current winning L2 node for the period of presentation of the current input.

The learning process of the network can be described as follows: At each presentation of a non-zero binary input pattern, the network attempts to classify it into one of its existing categories based on its similarity to the stored prototype of each category node. More precisely, for each node in the category layer, a weighted bottom up activation is calculated, which gives a degree of match between the current input and each category node. Then the category node that has the highest bottom-up activation is selected (realizing the so called winner-takes-all competition). The weight vector of the winning node will then be compared to the current input at the comparison layer. If they are similar enough, i.e. if they satisfy the matching condition compared to a sensitivity threshold Θ, then this category node will capture the current input and the network learns by modifying the weight vector.

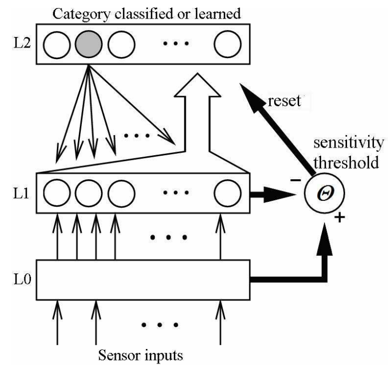

Fig. 1. Architecture of the ART network.

If, however, the stored prototype does not match the input sufficiently, the winning category node will be reset for the period of presentation of the current input. Then another category node is selected with the highest bottom-up activation, whose prototype will be matched against the input, and so on. This "hypothesis-testing" cycle is repeated until the network either finds a stored category whose prototype matches the input well enough, or allocates a new category node for the current input. Details about the FuzzyART algorithm, together with the corresponding equations can be found in [10].

Due to their so called stability-plasticity property, the ART neural networks are capable of learning "on-line", i.e. refining their learned categories in response to a stream of new input patterns, as opposed to being trained "off-line" on a finite pre-chosen training set. We have modified the ART cycle of testing and learning in a way that it is not necessary to load a certain set of input patterns on which the learning will take place, but rather it is done after each new signal pattern has been given to the inputs.

The number of developed categories can be controlled by setting the sensitivity threshold Θ: the higher the threshold, the larger number of more specific categories will be created. At its extreme, the network will create a new category for every unique input pattern. The sensor data can be classified with different sensitivity threshold Θ, thus providing a general overall view on the data, (smaller Θ) or more and more detailed views of the sensed data (greater Θ).

2.2. Two-tier Data Aggregation Architecture

In the proposed architecture for data aggregation in wireless sensor networks, each MicaZ unit has FuzzyART implementations classifying only its sensor readings. One of the MicaZ units can be chosen to be a clusterhead collecting and classifying only the classifications obtained at other units. Since the clusters at each unit can be represented with integer values, the neural-network implementation at the clusterhead can be ART1 with binary inputs (see Fig. 2). Depending on the requirements, it can be even a supervised neural network (e.g. FuzzyARTMAP [13]) where the user or another system can apply the teacher input to the neural network.

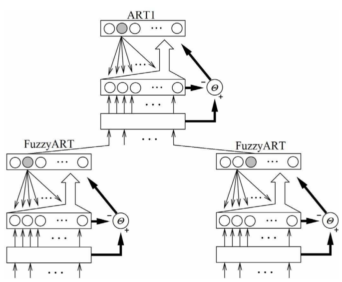

Fig. 2. One clusterhead collecting and classifying the data after they are once classified at the lower level

With this architecture a great dimensionality reduction can be achieved depending on the number of sensor inputs in each unit. In our case it's a 7-to-1 ratio when used with MTS310 sensor boards and 3-to-1 ration when used with MTS300 sensor boards (see section III). In the same time communication savings benefit from the fact that the classification identification numbers are usually small binary numbers unlike raw sensory readings which can be several bytes long real numbers converted from the analog inputs.

2.3. Adaptation of the FuzzyART Algorithm

One of the limitations of the ART neural networks that had to be overcome during the adaptation is the fixed number of inputs to the neural-network. It is not a problem in each of the nodes in the wireless sensor network, since the physical sensors are supposed to function continuously and it is rare that some of the individual sensors malfunction, which requires separate special algorithm for detection. On the other hand, in the neural-network implemented into the clusterhead, some of the inputs otherwise obtained from certain sensor nodes, may become unavailable due to the power failure of the nodes' batteries.

We have augmented the ART neural-network architecture in the clusterhead with the possibility to define some of the inputs as unused. In that way only the maximal number of inputs is predefined and thus limited, while the actual number of inputs to the ART neural-network depends on the actual aggregated data.

This is necessary in order to further reduce the communication usage, in a way that the sensed inputs are first classified and only in the case when the new sensor input sample is classified as different from the previous one, the new classification result is sent over the communication channel. Still, another parameter is needed for the whole network to function correctly. Specifically, it is the time needed to declare one network node as power exhausted. If certain amount of time has passed, even in the case that the neural-network is classifying the sensors' data as the same with the previous ones continuously, a data packet that is announcing it is sent, in order to inform the clusterhead that the node is still functioning and that its reported data should be taken into account when classifying it again in the ART neural-network implemented into the clusterhead.

On the other hand, what had to be limited in each MicaZ node is the number of different categories into which the sensory inputs can be classified. Due to the limited memory, their number has to be limited by the maximum allowed memory. We have modified the ART algorithm in way that after the maximum number of different categories is reached, the sensitivity threshold is lowered a little and no further additions of new categories is allowed. Only modifications of the existing weights among the neural network nodes are performed, in that allowing the adaptation of the existing categories to the future input data. This modification does not apply to the clusterhead unit that has practically no memory limits since it is running on a PC.

Before applying the sensed input data into the ART neural-network, each sensor value has to be normalized and transformed into the real numbers' interval from 0 to 1.

3. Hardware and Software Platform

MicaZ motes (see Fig. 3) are 8-bit microcontrollers running at 16 MHz and have only 4 Kbytes of RAM. We have used 7 such motes, 4 of them equipped with MTS310 sensor boards having a light sensor, 2 accelerometers, 2 magnetic sensors, a thermometer and a microphone, while 3 of them were equipped with MTS300 sensor boards having only a light sensor, a thermometer and a microphone. The operating system that runs on the motes is TinyOS [15].

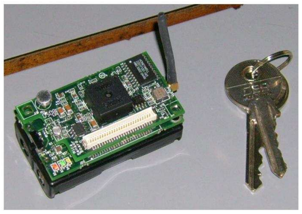

Fig. 3. A single MicaZ mote

The program for the MicaZ units is written in NesC [14] where the platform-independent adapted code for the FuzzyART neural network is about 300 lines long. After compilation it requires 17942 bytes of program ROM and 3086 bytes of data RAM, having the limit of 48 different categories into which the input data can be categorized. (The source code is made available at http://odl-skopje.etf.ukim.edu.mk/WSN)

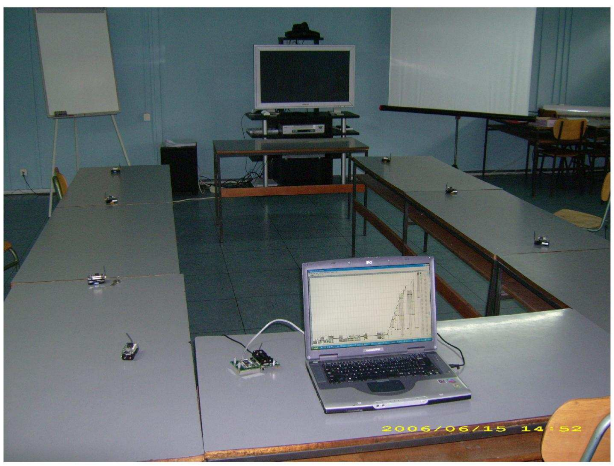

Fig. 4. Experimental setup with seven MicaZ motes and one clusterhead mote connected to a laptop.

For the clusterhead we have used NesC to program the mote which is connected by Ethernet link to a PC. The collected data are first saved into a PostgreSQL database and are further classified using the ART1 neural network classifier. For the PC we have implemented a graphical server application which classifies the data and displays them continuously on a graph. This program written in C++ for Microsoft Visual Studio is about few thousand lines of code.

The whole experimental setup can be seen in Fig. 4. where 7 motes are spread in our lab, together with the clusterhead mote which is connected by Ethernet link to the laptop computer. We have influenced the environment by switching lights on/off, by shouting and by knocking and shaking the desks. The results of the classification are displayed in real-time in a specially developed application, running at the laptop computer, and can be seen enlarged in Fig. 8.

The debugging of the programs written in NesC is very difficult since the memory model simulated for the debugging process in a PC, does not reflect correctly the real memory organization in MicaZ motes. In that way, the original functions of the ART neural-networks written in C, had to be rewritten, because they were made for a memory organization of a PC, not for embedded systems like MicaZ motes.

At the end, one of the most useful debugging tools for tracking the execution of our programs in real Micaz motes, turned to be the three LED lights and the sounder integrated into motes. The most time-consuming part of the debugging process was the practice of taking out the batteries and plugging the motes into the programmer card, which had to be repeated many times.

We have also tested the prototype in a more realistic environment – in an application of a traffic control at a parking. The experimental setup can be seen in Fig. 5, where 7 motes are distributed along the road leading to the parking in front of our lab. The clusterhead had to be placed near the entrance of our lab, since the communication range of the motes is very small.

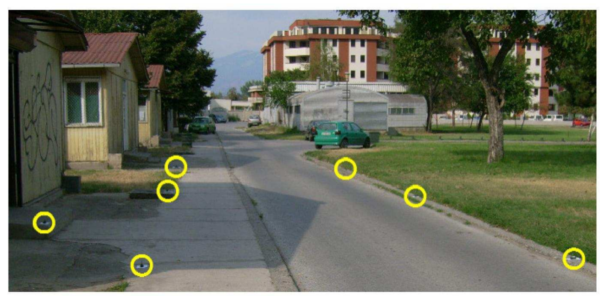

Fig. 5. The experimental setup in a traffic control application. With yellow circles are highlighted the motes distributed along the road leading to the parking.

4. Experimental Results

The data obtained from this experiment can be seen in Fig. 6. Although the photo of the experimental lab is taken by daylight, we have carried the experiments by twilight, when the illumination was slowly extinguishing and the temperature was slowly decreasing. Each time some car has passed, the lights sensor, the microphone and the temperature sensor reacted correspondingly. The accelerometers and even the magnetic sensors reacted as well, but only at those sensors which were nearest to the road.

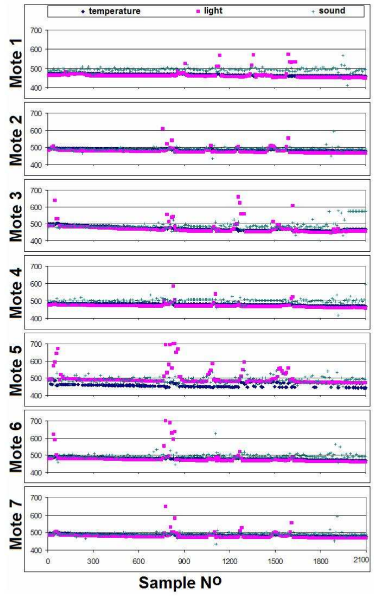

Fig. 6. Part of the data collected in the traffic control application, collected after around 2100 samples.

Results from the simulations, where we have even deliberately made some of the sensor inputs malfunctioning in order to show the data robustness of this approach can be seen in [5] and [6] and are shown in Fig. 7 for comparison reasons.

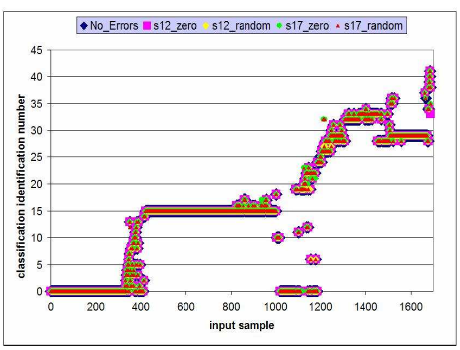

Fig.7. Results of the classifications show significant data robustness of the architecture with one clusterhead node collecting only classified input data from the other units. s12_zero and s_17_zero are signals where the 12th or the 17th sensor readings are deliberately forced to zero values, while s12_random and s17_random are signals where the 12th or the 17th sensor readings are consisted of random values in the designated interval.

We have conducted experiments with the original data obtained from another experimental setup [3] and with the synthetically made erroneous data. In Fig. 7. we give the results of the classifications at the clusterhead collecting only the prior classifications from the other units. The results show no significant difference among the classifications when all sensors are functioning correctly or when some of the sensors give only zero or random signal (in our case sensors number 12 and 17). In Fig. 7 to Fig. 9, the classification identification number is the number of the cluster where the corresponding input sample is classified and has no additional meaning since the unsupervised learning is used in this scheme.

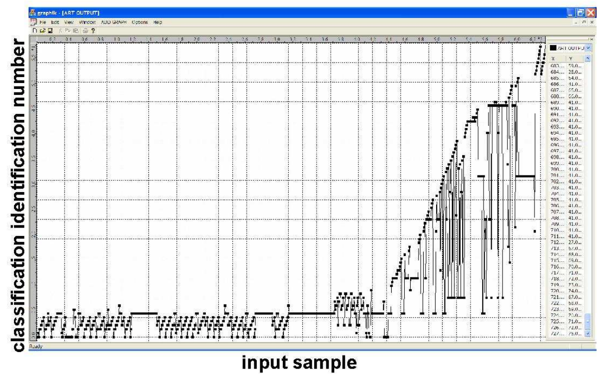

Fig.8. Results of the classification at the clusterhead node, of the already classified input samples at the previous level, displayed in real-time, in the first experimental set-up.

Here we present only the classification results at a clusterhead node after collecting the classification results of the input data sensed at the previous level. The data presented in Fig. 8. are obtained after around one thousand input samples were sensed and classified.

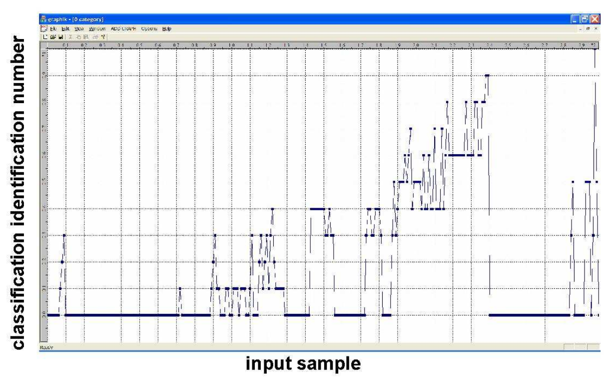

Fig. 9. Results of the classification of the data obtained at the clusterhead node in the prototype of a traffic control application.

The inputs used for the simulation and for the real experiment are different, which is reasonable since we can not repeat the same conditions. Still, it can be seen that the results from the simulation have very similar nature as the ones obtained in a real experiment. At the beginning the number of different categories into which the data is classified is small and later, when some events change the environmental conditions, the number of different categories proliferate.

In Fig. 9. are shown the results of the classification of the data obtained from the prototype of the traffic control application.

What can be further done is to add certain annotation labels to certain classification identification numbers, and in that way to obtain a supervised learning scheme with attached meanings to these classification identification numbers.

For future work we envisage a comparison of the FuzzyART with other well-known methods in data processing of wireless sensor networks, by using the comparison metrics which include: accuracy of final decision, sensor network power consumption, reduction of the communication costs, and complexity of implementation.

5. Conclusion

It is not graceful to program embedded devices like MicaZ motes, mainly because there are no developed tools yet for easy debugging of the programs. Small memory capacity represents big burden which prevents more intensive pre-processing algorithms to be implemented in MicaZ motes, including time-sequence analysis of patterns either by using wavelets or other methods. In our opinion, this should be the direction for development of small embedded wireless sensor devices.

We have adapted and implemented a FuzzyART neural- networks algorithm into MicaZ motes. When used in a two-tier architecture, this method shows confirmed data robustness, leads to lower communication exchange and thus results in lower power consumption. Outcomes from the simulated deliberately erroneous sensors, where we imitate defective sensors giving only zero or random output, show that the model is robust to small variations in the input.

The proposed architecture can be expanded into a multi-tier data classification scheme, where each clusterhead would further transmit its results of the classification and at the end the main clusterhead would finally classify the event that provoked the burst of communication. Also a possibility to attach tags to the categories obtained at the main clusterhead can be added, which would later be exploited for recognition purposes.

References

[1] D. Ganesan, D. Estrin, and J. Heidemann, "DIMENSIONS: Why do we need a new data handling architecture for sensor networks?", in Proc. Information Processing in Sensor Networks, Berkeley, California, USA, 2004.

[2] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin M., and S. Madden, "Distributed Regression: an Efficient Framework for Modeling Sensor Network Data", in Proc. of Information Processing in Sensor Networks, Berkeley, CA, USA, 2004.

[3] E. Catterall, K. Van Laerhoven, and M. Strohbach, "Self-organization in ad hoc sensor networks: an empirical study", in *Proc. of Artificial Life VIII, The 8th Int. Conf. on the Simulation and Synthesis of Living Systems*, Sydney, NSW, Australia, 2002.

[4] Y. Liyang, W. Neng, M. Xiaoqiao, "Real-time forest fire detection with wireless sensor networks", in *Proc. of Int. Conf. on Wireless Communications, Networking and Mobile Computing*, Wuhan, China, 2005, Vol. 2, pp. 1214- 1217.

[5] A. Kulakov, D. Davcev, "Tracking of unusual events in wireless sensor networks based on artificial neural-networks algorithms", in *Proc. of Int. Conf. on Information Technology: Coding and Computing*, Las Vegas, NV, USA, 2005, pp. 534-539.

[6] A. Kulakov, D. Davcev, "Distributed data processing in wireless sensor networks based on artificial neural-networks algorithms", in *Proc. of The 10th IEEE Symposium on Computers and Communications*, Cartagena, Spain, 2005, pp. 353-358.

[7] S. Grossberg, "Adaptive pattern classification and universal recoding: I. parallel development and coding of neural feature detectors", *Biological Cybernetics*, vol. 23, pp. 121-134, 1976.

[8] S. Grossberg, "Adaptive Resonance Theory" in Encyclopedia of Cognitive Science, Macmillan Reference Ltd, 2000.

[9] G.A. Carpenter, and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine", *Computer Vision, Graphics, and Image Processing*, vol. 37, pp. 54-115, 1987.

[10] G.A. Carpenter, S. Grossberg, and Rosen, D.B., "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system", *Neural Networks*, vol. 4, pp. 759-771, 1991.

[11] G.A. Carpenter, and S. Grossberg, "Integrating symbolic and neural processing in a self-organizing architecture for pattern recognition and prediction". In V. Honavar and L. Uhr (Eds.), Artificial intelligence and neural networks: Steps towards principled prediction. San Diego: Academic Press, pp. 387-421, 1994.

[12] E. Sapojnikova, "ART-based Fuzzy Classifiers: ART Fuzzy Networks for Automatic Classification", PhD Thesis, Department of Computer and Cognitive Science at the University of Tübingen, Germany, 2003.

[13] G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, and D.B. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps", *IEEE Transactions on Neural Networks*, vol. 3, pp. 698-713, 1992.

[14] D. Gay et al., "The NesC Language: A holistic Approach to Networked Embedded Systems", in Proc. of ACM Conf. Programming Language Design and Implementation, ACM Press, 2003, pp. 1-11.

[15] http://webs.cs.berkeley.edu/tos/