Node localization in Wireless Sensor Networks Using Artificial Neural Networks and Optimization Based on Simulated Annealing Algorithm

[1,2]Stephan H. Chagas, [1]Leonardo L. de Oliveira, [1,2]João Baptista S. Martins
{stephan.chagas,leonardo}@mail.ufsm.br, batista@inf.ufsm.br

[1]GMICRO – Universidade Federal de Santa Maria
[2]PPGI – Universidade Federal de Santa Maria

Abstract

Localization of nodes in wireless sensor networks (WSNs) is one of the most popular problems that came out to be solved by the scientific community in the last years. This issue is well-known because most of wireless sensor networks applications need location awareness as one of their sensor's main features. In order to localize network nodes, many approaches have been proposed across the years as can be seen in literature. However, few of them use artificial neural networks (ANNs) as the operating principle of their methods. Therefore, this paper presents a meta-heuristic way to select the best suited structure of an ANN aiming node localization in WSNs. The optimization procedure was made by using Simulated Annealing (SA) algorithm and artificial neural networks toolbox under MATLAB environment. The MATLAB-based Probabilistic Wireless Network Simulator (Prowler) was used to generate the WSN topology, as well the training and testing data to feed de ANNs under evaluation. Results using the best artificial neural network structure found after the optimization process had a root mean square error of 0.61 meter when analyzing the accuracy of the localization estimation.

1. Introduction

Applications regarding wireless sensor networks (WSNs) have been explored in several areas, such as domotics, robotics, industrial monitoring, environmental monitoring, military surveillance, and many others. The location awareness is one of the most desired features when talking about WSNs, and it is not hard to understand why. Frequently, the wireless sensors nodes are supposed to be monitoring some processes or systems variables (temperature, humidity, etc.) at some specific point of a geographic area. Therefore, making these readings without addressing the positioning information as well could become a meaningless action.

The easiest way to keep the nodes up to date with their own positions would be through the use of one GPS (Global Positioning System) receiver for each node. However, it is not possible due to the small form factor of the devices, as well their low-power and low-cost profiles.

In order to solve this issue, several localization algorithms have been proposed by the scientific community. Most of the related work found by the authors use analytical methods like trilateration and triangulation when estimating the nodes coordinates in the network. However, some of the well-known approaches use the connectivity information extracted from the messages exchanged among the nodes. The Centroid algorithm [1] is a great example of this class of algorithms.

This work proposes an artificial neural network (ANN) approach to the localization problem. An optimization procedure is performed through the use of Simulated Annealing algorithm (SA) to tune the feedforward ANN to the best operation point when localizing nodes in a certain WSN. This allows the solution to be more accurate while simultaneously avoid the trial and error procedure that can be seen in [2] - [4] when selecting the best suited ANN structure.

The remainder of this paper is organized as follows. In the next section, the basic theory regarding artificial neural networks will be addressed, as well the basics about Simulated Annealing algorithm. In section 3, the simulation design will be described and the results will be discussed. This article is concluded in section 4.

2. Background

2.1. Artificial Neural Networks

The first article related to neurocomputing was made in 1943 by McCulloch and Pitts [5]. They created the first mathematical model for an artificial neuron. The first training method for artificial neural networks was developed in 1949. It aimed to describe a neurophysiologic hypothesis proposed by Donald Hebb.

Artificial Neural Networks are interconnection structures among artificial neurons (also called nodes). The artificial neurons are modeled in order to mimic biological neurons through the use of activation functions. Each neuron consists of multiple inputs, weights and a single output. Also, its transfer function is responsible for mapping its inputs to output.

Basically, an ANN is an adaptive system that receives a set of inputs, processes the data and provides an output. It may have a single or multiple layers in their structure having the following designations: input layer, hidden layer or output layer.

However, before using the ANN, there is still need to train it. This process is performed by giving the correct answers for a set of inputs and adjusting the weights based on the response given by the network. When the outputs provided by the ANN are under a specific error limit, the training is done. This is called supervised training.

2.2. Simulated Annealing

The Simulated Annealing optimization is based on the physical process of annealing in metallurgy. It has the same principle as the Hill Climbing, except that it is also able to accept worst solutions given a certain probability.

It starts with a user given solution, then evaluates it and performs a small modification on the solution. This new candidate solution is evaluated and if it is better than the previous, SA accepts it and assumes it as the current solution. If it is not better than the previous, there will be a probability of this new worst solution to be accepted based on the cost of each solution and the present temperature in the system. This formula can be seen in (1), where *A* is the probability of accepting the worst solution, *c(N)* is the cost of the new solution, *c(P)* is the cost of the present solution and *t* is the temperature.

$$A = e^{-\frac{c(N)-c(P)}{t}} \quad \textbf{(1)}$$

This procedure is performed in a loop and after an arbitrary number of iterations occur the temperature is multiplied by a reduction factor. This will make the temperature decrease and consequently more difficult to accept worst solutions. This technique uses only one solution and tries to guide it to best place in the design space.

3. Simulation Design and Results

All simulations made in this work were performed using MATLAB from MathWorks and the WSN toolbox Prowler [6], developed by Vanderbilt University. The original target device used in Prowler was Berkeley's MICA mote. However, in this work, Prowler was customized to fit the Crossbow's MICAz model. Furthermore, the propagation model used to simulate this particular WSN was also customized to the fit the well-known log-normal shadowing path loss model. This model can be seen in (2), where L is the attenuation for distance d, L_0 is the loss on reference distance d_0, γ is the path loss exponent and X_σ is a zero-mean Gaussian random variable with standard deviation σ. The values used in this simulation were extracted from empirical measurements made in [7]. The reference distance was 0.1 meter with P_{out} = -25dBm, L_0 is 30dBm, γ is 2.5 and σ = 4.

$$L(d) = L_0 + 10\gamma log_{10}\frac{d}{d_0} + X_\sigma \quad \textbf{(2)}$$

The simulated scenario was an indoor squared area of 26x26 meters. This particular simulated area size was due to the MICAz output power, which can reach approximately 40 meters when applying the log-normal shadowing path loss propagation model with these parameters arrangement. A total of 8 anchor nodes were distributed on the edges of the area. A set of 81 grid sensors with known positions were used in order to collect training data for the artificial neural networks. For testing purposes, a set of 64 nodes inside the area covered by the training grid were used. Fig. 1 shows the location of these nodes.

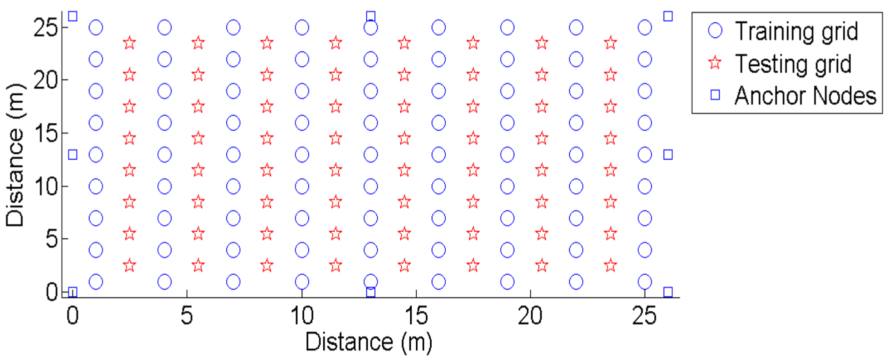

Fig. 1 – Training and testing grid.

Each of the training grid sensors collected a set of samples. An input sample contains the received signal strength indicator (RSSI) measurement, x and y position, and ID from all of the anchor nodes. In the simulation environment, the anchor nodes sent 30 beacons each, from which the first 10 collected were saved by every sensor node in the training grid. This resulted in a data set with 810 samples (81 training nodes collecting 10 input samples each). Due to the nondeterministic nature of propagation channel used in simulation and packet collisions, some beacons could not be delivered. Hence, the anchor nodes sent 30 beacons although aiming the reception of only 10. All nodes were within one hop distance to each other.

With the input data gathered, the optimization procedure could be made. The ANN's parameters that were allowed to be modified by the Simulated Annealing algorithm were: number of hidden layers, number of nodes on each hidden layer, and transfer function on each hidden layer. Tab. 1 shows the boundaries for each of these variables.

Tab.1 – Boundaries of artificial neural networks parameters

Parameter	**Value**
Number of hidden layers	[0-3]
Number of nodes per layer	[1-16]
Transfer function for each layer	[tansig, logsig, purelin, radbas]

These parameters boundaries could be customized, but in this simulation they were selected aiming the computational and energy constraints of the MICAz network nodes. Therefore, only a few hidden layers were allowed, as well a small number of artificial neurons per layer. The Simulated Annealing optimization requires an initial ANN structure (initial solution) to begin the procedure. Every feedforward ANN structure suggested by the SA algorithm was trained and tested with the data acquired by Prowler. The evaluation (cost function) of these structures was made through the use of root mean square error (RMSE) between the real positions and estimated positions of the unknown nodes. The calculation of the RMSE can be seen in (3), where *n* is the number of testing nodes, x_i and y_i are the real node coordinates, $\hat{x}_i$ and $\hat{y}_i$ are the estimated node coordinates, and i is the node index.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2]} \quad \textbf{(3)}$$

All simulations and optimizations were done using an Intel Core i3 3.1 GHz and 4 Gb of RAM memory. Total processing time for the Simulated Annealing optimization was about 30 minutes. The best ANN structure found can be seen in tab. 2.

Tab.2 – Best artificial neural network found after optimization

Parameter	**Value**
Number of hidden layers	2
Number of nodes on input layer	4
Number of nodes on hidden layer 1	15
Number of nodes on hidden layer 2	5
Number of nodes on output layer	2
Transfer function on input layer	Logsig
Transfer function on hidden layer 1	Tansig
Transfer function on hidden layer 2	Logsig
Transfer function on output layer	Purelin

The localization results using the best ANN structure had a root mean square error of 0.61 meter, a maximum error of 1.71 meter and a minimum error of 0.041 meter. The real positions and estimated positions for this test are illustrated in fig. 2. The optimization procedure was made by a PC, but the resulting ANN could be used in MICAz mote's firmware in real world application. This could be done training the ANN with data extracted by MICAz motes in the field application and using the ANN weights provided by the optimization algorithm.

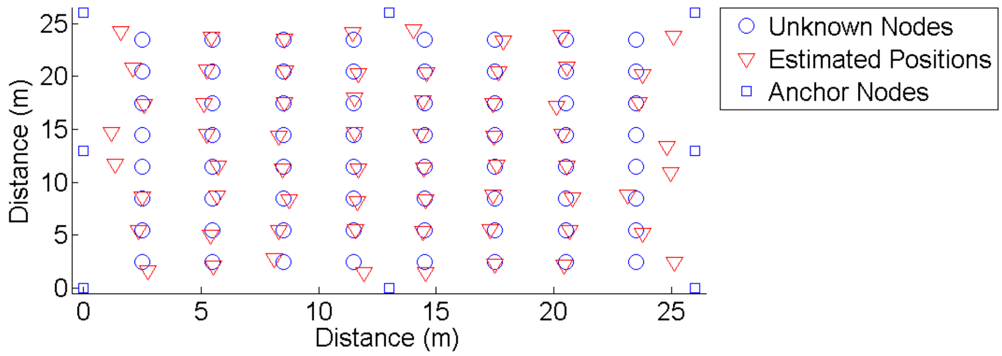

Fig. 2 – Estimated positions and real positions for the testing data set.

4. Conclusion

This paper presented an approach to localization scheme for wireless sensor networks using artificial neural networks as the machine learning algorithm and Simulated Annealing optimization to select the best ANN structure. An arbitrary initial structure was manipulated by the SA algorithm in order to tune the ANN to the best performance for the simulated WSN topology. The method was tested in an indoor simulated environment of 26x26 meters with 8 anchor nodes.

Results using the best ANN structure found after the optimization process through the Simulated Annealing algorithm had a root mean square error of 0.61 meter, a maximum error of 1.71 meter and a minimum error of 0.041 meter. The optimization procedure was performed in approximately 30 minutes.

The results indicate that this approach is effective and can be used as a way to reduce the dependency of the designer's experience when selecting the ANN parameters. Furthermore, the localization accuracy is increased and the required time to adjust ANN structure is reduced. Through the use of the weights found after optimization procedure executed by a PC, the ANN found can be used by the MICAZ motes, if the training data were collected through the use of the motes in the field application.

5. References

[1] N. Bulusu, J. Heidemann, and D. Estrin. "GPS-less low cost outdoor localization for very small devices". IEEE Personal Communications Magazine, vol 7, pp 28-34, October 2000.

[2] Ali Shareef, Yifeng Zhu, and Mohamad Musavi, "Localization Using Neural Networks in Wireless Sensor Networks," Mobilware *Proc. 1st international conferenceon MOBILe Wireless middleware, Operating systems, and Applications*, 2007.

[3] Mohammad Shaifur Rahman, Youngil Park, and Ki-Doo Kim, "Localization of Wireless sensor Network Using Artificial Neural Network", *Proc. 9th international conference on communications and information technologies*,2009.

[4] Jin-Peng TIAN, and Hui-Chang SHI, "Study of Localization Scheme base on Neural Network for Wireless Sensor Networks", *IET conference on Wireless, mobile and Sensor Networks*, Dec. 2007, Pp. 64-67.

[5] McCulloch and Pitts, and Hui-Chang SHI, "A logical calculus of the ideias immanent in nervous activity", Bull. Math. Biophysics, 1943, 5: 115-133.

[6] Simon, G et al.: "Simulation-based optimization of communication protocols for large scale wireless sensor networks", *IEEE Aerospace conference, Big Sky, MT*, March 8-15 2003.

[7] A. Barberis, L. Barboni, and M. Valle. Evaluating energy consumption in wireless sensor networks applications. In Digital System Design Architectures, Methods and Tools, 2007. DSD 2007. 10th Euromicro Conference on, pages 455–462, aug. 2007.