UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

LOCATION-AWARE COMPUTING: A NEURAL NETWORK MODEL FOR DETERMINING LOCATION IN WIRELESS LANS

Roberto Battiti, Thang Le Nhat and Alessandro Villani

February 2002

Technical Report # DIT-02-0083

Location-aware computing: a neural network model for determining location in wireless LANs

Roberto Battiti, Thang Le Nhat, and Alessandro Villani

Università di Trento, Dipartimento di Informatica e Telecomunicazioni
via Sommarive 14, I-38050 Pantè di Povo (TN), Italy
battiti|thang|avillani@science.unitn.it

Abstract. The strengths of the RF signals arriving from more access points in a wireless LANs are related to the position of the mobile terminal and can be used to derive the location of the user.
In a heterogeneous environment, e.g. inside a building or in a variegated urban geometry, the received power is a very complex function of the distance, the geometry, the materials. The complexity of the inverse problem (to derive the position from the signals) and the lack of complete information, motivate to consider flexible models based on a network of functions (*neural networks*).
Specifying the value of the free parameters of the model requires a supervised learning strategy that starts from a set of labeled examples to construct a model that will then generalize in an appropriate manner when confronted with new data, not present in the training set.
The advantage of the method is that it does not require ad-hoc infrastructure in addition to the wireless LAN, while the flexible modeling and learning capabilities of neural networks achieve lower errors in determining the position, are amenable to incremental improvements, and do not require the detailed knowledge of the access point locations and of the building characteristics. A user needs only a map of the working space and a small number of identified locations to train a system, as evidenced by the experimental results presented.

Keywords: location- and context-aware computing, wireless LANs, IEEE802.11b, neural networks, machine learning.

1 Introduction

Sentient computers, that take the current context (e.g. location, time, activity, previous history) into account when interacting with the user, hold significant promises for a seamless use of tomorrow's wireless networks in which mobile computing and Internet connectivity will be provided for professional and recreational activities through PDAs, smart phones, laptops, and other mobile appliances.

Knowledge of the location and suitable models are important in order to reduce the cognitive burden on the users in context- and location-aware systems [8, 1, 15]. Location awareness is considered for example in the infostation-based hoarding work of [14], and in the websign system of [17]. Some techniques for determining the location in indoor and urban context (where GPS assisted localization encounters problems) are based on pattern recognition: from the signature of the signal received by multiple antennas one derives the position of the mobile device [3]. A complication is caused by the fact that signal propagation is influenced by environmental factors like, for example, the number of people located in the working area [2], the position and material of walls and, in general, the infrastructure contained in a building.

The research in this paper proposes a method based on neural networks for reducing the errors in the determination of the current location of the user. One executes measurements of the strength of signals coming from the different antennas at a series of points distributed in the environment. These data are a training set that can be used by a learning algorithm (e.g. a neural net) to develop an association between signal strengths and location. We propose to use neural networks and a training algorithm based on second-order information in order to develop flexible models of the relationship between the raw signal measurements and the location data.

The following part of this paper is organized as follows. Section 2 summarizes previous approaches to the problem, Section 3 describes the methodology for modeling the input-output relationship through multi-layer perceptron neural networks, Section 4 describes the system and the collection of data points for the experiments, Section 5 discusses the experiments dealing with the selection the neural architecture and the length of the training phase. Finally, Section 6 describes the obtained measurement error test results as a function of the number of training examples.

2 Previous approaches

Advances in indoor, short-range wireless communication technology and the increasing trend toward portable, hand-held personal computers equipped with high-speed radio access have made wireless LANs popular. Currently, there are several alternative wireless LAN technologies such as IEEE 802.11 a, b, HIPERLAN and Bluetooth. Among them, the IEEE 802.11 standard is gaining a growing support as a solution for transmitting/receiving data with high-speed rate in indoor networks with a bandwidth of up to 54Mbps [16].

Many different systems and technologies to determine the location of users for mobile computing applications have been proposed. Global Positioning System (GPS)is a satellite-based navigation aid originally developed by the US military. GPS systems receive signals from multiple satellites and use a triangulation process to determine physical locations with approximately 10 meters accuracy. GPS is very successful in open areas but ineffective for indoor use or in urban areas with tall buildings that shield the satellite signals.

The Active Badge system [23], [9] is one of the earliest indoor systems for determining the location, based on diffused infrared technology. A badge emits a unique IR signal periodically or on demand. Infrared sensors placed in the building pick up these periodic signals and transfer them to a master station for processing. Although the Active Badge system provides accurate location information, it also subject to some restrictions such as line-of-sight limitations, poor performance with fluorescent lighting or direct sunlight.

The Active Bat location system [24, 10] developed by AT&T researchers uses a combination of RF and ultrasound time-of-flight to estimate the distance. When a controller connected to the PC sends a radio request message, an Active Bat tag attached to the object reacts by emitting an ultrasonic pulse directed to a matrix of receiving elements mounted on the room ceiling. At the same time, the controlling PC sends a reset signal to the receivers over the serial network, so that they can measure the time interval and calculate the distances from the tag to the receivers. The use of ultrasound time-of-flight requires a large fixed-sensor infrastructure throughout the ceiling and the accuracy, that can reach about 9 cm, is rather sensitive to the precise placement of the sensors.

PinPoint Corp. develops a product named 3D-iD local positioning system [25] for determining the 3D locations of items inside buildings. In this architecture, 3D-iD readers emit codes that are received by the tags attached to mobile devices. Then the tags simply change the signal's frequency and transmit back to the reader with tag ID information phase-modulated onto it. The reader extracts the tag ID from this returned signal and also determines the tag's distance from the antenna by measuring the round trip time of flight. The PinPoint system is composed of cells within a building and uses spread-spectrum radio signals and multiple antennas (up to 16) at the cell controller to process the signal from a tag. It can detect reliably items from about a 30-meter distance with 1 to 3-meter accuracy. The disadvantages are that each antenna has a narrow cone of influence, so that ubiquitous deployment becomes prohibitively expensive. Additional difficulties arise when interoperating with the IEEE 802.11 wireless networking infrastructure because of radio spectrum collision [11].

Microsoft Research RADAR location system used the IEEE 802.11b wireless LAN technology [3]. In the RADAR system, the RF signal strength is used as a measure of distance between Access Point (AP) and mobile terminal, and then this information is used to compute the 2D position by triangulation, with both an empirical method and a signal propagation modeling method. The results show that the empirical method is superior in terms of accuracy with median resolution in the range of about 3 meters, while the signal propagation modeling method has 4.3 meters accuracy (median), but it makes deployment easier.

Similar to Active Bat system, Cricket, a location-support system for in-building, mobile, location-dependent applications, uses a combination of RF and ultrasound hardware to enable a small device attached to mobile user (the listener) to estimate the distance to the nearest beacon[18]. The listener performs the timing and computation functions. On each transmission, a beacon, a small device attached to some locations within the geographic space, sends both space information and an ultrasonic pulse. When the listener hears the RF signal, it uses the first few bits as training information and turns on it ultrasonic receiver to listen to the ultrasonic pulse, which arrive in short time later. Based on the time interval between the first bit of RF information and the ultrasonic signal, the listener can determine the distance to the beacon. Cricket's main features are user privacy, decentralized administration, network heterogeneity, low cost and a portion-of-a-room granularity of 4x4 feet.

SpotON, a new tagging technology for three-dimensional location sensing based on radio signal strength analysis was introduced in [12]. The system is built by using RFIDeas badge and AIRID base station - the product of Illinois Company and Hydra microwebserver that has both an Ethernet and serial port for the AIRID internetworking task. In general, the SpotON system is similar to Microsoft Research wireless LAN and the PinPoint system in developing a fine grained tagging technology based on RF signal strength. However, following the authors' laboratory experiments, the SpotON can archive better resolution and accuracy than the Microsoft Research system with a much lower cost than the product from PinPoint.

3 Methodology: Models based on multi-layer perceptron neural nets

We introduce a new method to determine the locations of mobile terminal in high-speed wireless LAN environment using the IEEE 802.11b standard that is based on neural network models and automated learning techniques. As it is the case for the RADAR system, no special-purpose equipment is needed in addition to the wireless LAN, while the flexible modeling and learning capabilities of neural networks achieve lower errors in determining the position, are amenable to incremental improvements, and do not require the detailed knowledge of the access point locations and of the building characteristics in addition to a map of the working space.

In our system we use the signal strengths received at a mobile terminal from different access points (at least three) to determine the position of the terminal inside a working area. The starting point of the method is the relationship between distance and signal strength from a given access point. In a free space environment the power received by a receiver antenna which is separated from a radiating transmitter antenna by a distance d is given by the following Friis free space equation[19].

$$P_r(d) = \frac{P_t G_t G_r l^2}{(4\pi)^2 d^2 L} \tag{1}$$

where P_t is the transmitted power, $P_r(d)$ is the received power, G_t, G_r are transmitter and receiver antenna gain respectively, d is the $T-R$ separation distance in meters, L is the system loss factor not related to the propagation $(L > 1)$ and l is the wavelength in meters. More detailed radio propagation models for indoor environments are considered for example in [3]. If one knows distances d_i from the mobile terminal to at least three different APs, one can calculate the position of the mobile terminal in the system.

However, in a variegated and heterogeneous environment, e.g. inside a building or in a complex urban geometry, the received power is a very complex function of the distance, the geometry of walls, the infrastructures contained in the building. Even if a detailed model of the building is available, solving the direct problem of deriving the signal strength given the location requires a lengthy simulation. The inverse problem, of deriving the location from the signal strengths is more complicated and very difficult to solve in realistic situations. Furthermore, in order to facilitate the deployment of the system, it is unrealistic to require a detailed exhaustive specification of the building geometry, materials, infrastructures. The two reasons, complexity of the problem and lack of complete information, motivate to consider flexible models based on networks of functions. These models are termed ”non-parametric models” in statistics, and neural networks in other contexts.

The non-linear transformation of each unit and a sufficiently large number of free parameters guarantee that a neural network is capable of representing the relationship between inputs (signal strengths) and outputs (position). Let us note that the distance from the access points, and therefore the detailed knowledge of their position, is not required by the system: a user may train and use the system without requiring this information.

Specifying the value of the free parameters of the model (also called ”weights” of the network) requires a learning strategy that starts from a set of labeled examples to construct a model that will then generalize in an appropriate manner when confronted with new data, not present in the training set.

3.1 The One-Step Secant method for training neural networks

Efficient optimization algorithms are crucial in the learning phase of models like neural networks and have been studied for example in [5], [6]. Let us briefly define the notation. We consider the ”standard” multi-layer perceptron (MLP) architecture, with weights connecting only nearby layers and the sum-of-squared-differences *energy* function defined as:

$$E(w) \;=\; \frac{1}{2}\sum_{p=1}^{P} E_p \;=\; \frac{1}{2}\sum_{p=1}^{P}(t_p \;-\; o_p(w))^2 \tag{2}$$

where t_p and o_p are the target and the current output values for pattern p, respectively, as a function of the parameters of the networks (“weights” w). The architecture of the multi-layer perceptron is organized as follows: the signals flow sequentially through the different layers from the input to the output layer. For each layer, each unit (”neuron”) first calculates a scalar product between a vector of parameters (weights) and the vector given by the outputs of the previous layer. A transfer function is then applied to the result to produce the input for the next layer. The transfer function for the hidden layers is the sigmoidal function: $f(x) = 1/(1+e^{-x})$, while for the output layer it is the identity function, so that the output signal is not bounded.

It has been demonstrated that a network with a single hidden layer is sufficient to approximate any continuous function to a desired accuracy, provided that the number of hidden neurons is sufficiently large [13]. In this work we consider a single-hidden-layer MLP and a training technique that uses second-derivatives information: the one-step-secant method with fast line searches OSS, see [5], [4].

The standard back propagation technique uses only first-order information (the gradient). In particular, the stochastic on-line back-propagation update is given by:

$$w_{k+1} = w_k \;-\; \epsilon\, \nabla E_p(w_k) \tag{3}$$

where the pattern p is chosen randomly from the training set at each iteration, ∇E_p is the gradient, and ϵ is a fixed learning rate.

Faster training can be obtained by using also second derivatives, but computing all second derivatives (the Hessian) requires order $O(N^2)$ operations [7] and order $O(N^2)$

memory to store the Hessian components. In addition, the solution of equation to find the step (or search direction) in Newton's method requires $O(N^3)$ operations, at least when using traditional linear algebra routines. Fortunately, some second-order information can be calculated by starting from the last gradients, and therefore reducing the computation and memory requirements to find the search direction to $O(N)$.

Historically the one-step-secant method *OSS* is a variation of what is called *one-step (memory-less) Broyden-Fletcher-Goldfarb-Shanno* method, see [20]. The OSS method is described in detail and is used for multilayer perceptrons in [4] and [5].

Note that BFGS stores the whole approximated Hessian, while the *one-step* method requires only vectors computed from gradients. In fact, the new search direction p_{k+1} is obtained as:

$$p_{k+1} = -g_{k+1} + A_k s_k + B_k y_k \tag{4}$$

where the two scalars A_k and B_k are the following combination of scalar products of the previously defined vectors s_k, g_{k+1} and y_k (last step, gradient and difference of gradients):

$$A_k = -\left(1 + \frac{y_k^T y_k}{s_k^T y_k}\right) \frac{s_k^T g_{k+1}}{s_k^T y_k} + \frac{y_k^T g_{k+1}}{s_k^T y_k} \;\; ; \;\; B_k = \frac{s_k^T g_{k+1}}{s_k^T y_k}$$

The search direction is the negative gradient at the beginning of learning and it is restarted to $-g_{k+1}$ every N steps (N being the number of weights in the network).

The fast one-dimensional minimization along the direction p_{k+1} is crucial to obtain an efficient algorithm. The one-dimensional search is based on the "backtracking" strategy. The last successful learning rate λ is increased ($\lambda \leftarrow \lambda \times 1.1$) and the first tentative step is executed. If the new value E is not below the "upper-limiting" curve, then a new tentative step is tried by using successive quadratic interpolations until the requirement is met. Note that the learning rate is decreased by L_{decr} after each unsuccessful trial.

Quadratic interpolation is not wasting computation, in fact, after the first trial one has exactly the information that is needed to fit a parabola: the value of E_0 and E_0' at the initial point and the value of E_λ at the trial point. The parabola $P(x)$ is:

$$P(x) = E_0 + E_0' x + \left[\frac{E_\lambda - E_0 - \lambda E_0'}{\lambda^2}\right] x^2 \tag{5}$$

and the minimizer λ_{min} is:

$$\lambda_{min} = \frac{-E_0'}{2\left[\frac{E_\lambda - E_0 - \lambda E_0'}{\lambda^2}\right]} \leq \frac{1}{2(1 - G_{decr})}\lambda \tag{6}$$

If the "gradient-multiplier" G_{decr} is 0.5, the λ_{min} that minimizes the parabola is less than λ, see [5] for the complete details.

4 System description and experimental setup

Our system consists of a wireless Local Area Network using the IEEE 802.11b standard. It is located on the first floor of a 3-storeyed building.The layout of the floor and the positions of the three Access Points (APs) are shown in the Fig. 1. The floor has dimensions of 25.5 m x 24.5 m , for a total area of 624.75 m^2 and includes more than eleven rooms (offices and classrooms).

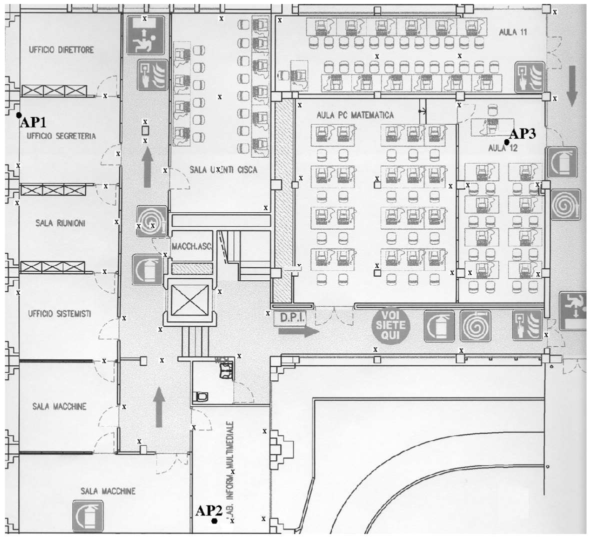

Fig. 1. The floor layout of the experiment, with access points locations

The origin of the coordinate system (0,0) is placed at the left bottom corner of the map. The (x, y) coordinates of the access points are as follows: AP1 $= (0.66m, 19.36m)$, AP2 $= (10m, 0.5m)$ AP3$= (23.41m, 17.90m)$.

The Access Points are AVAYA WP-II E model by Lucent Technologies Netherland B.V., two with external antennas. The wireless stations are Pentium-based Laptop computers running Linux version 7.2. Each Laptop is equipped with the ORiNOCO PC card - a wireless network interface card by Lucent Technologies.

The network operates in the 2.4 GHz license-free ISM band and supports data rates of 1, 2, 5.5, and 11Mbps. The 2.4 GHz ISM band is divided into 13 channels (IEEE & ETSI Wireless LAN Standard). In our system we use three channels: channel 1– at 2412 MHz; channel 7 – at 2442 MHz and channel 13 – at 2472 MHz (European Channel Selection–non–overlapping). Additional details of the system specifications are collected in Table 1.

4.1 Collection of example patterns

In order to facilitate the collection of labeled example patterns, the map of the area is stored on a laptop and a user interface has been designed based on a single click on the displayed map.

Frequency	2.4 GHz ISM Bandwidth
Modulation Method	Direct Sequence Spread Spectrum CCK at 11 Mbps and 5.5 Mbps, DQPSK at 2 Mbps, DBPSK at 1 Mbps
Media Access Protocol	CSMA/CA (Collision Avoidance) with Acknowledgment (ACK)
Bit Error Rate (BER)	Better than 10^{-5}
Nominal Output Power	15 dBm
External Antenna Gain	2.5dBi
Transmission Speed	Auto select 1, 2, 5.5, and 11Mbps
Spreading	11–chip Barker Sequence
Encryption	128 bit - (RC4)- Gold, also supports 64-bit Wired Equivalent Privacy WEP (RC4)- Silver
Receiver Sensitivity	-83,-87,-91,-94 dBm at 11, 5.5, 2, 1Mbps
Number of APs	3
Number of Measurement Points	56
Floor Dimensions	25.5 m x 24.5 m
OS platform	Linux 7.2

Table 1. The system specifications

When the user is at an identifiable position in the experimental area (e.g., at the entrance of a room, close to a corner, close to a column, etc.) he clicks on the displayed map in a point corresponding to the current position. Immediately after the click, the three received radio signal strengths from the APs are automatically measured and they are saved together with the point's coordinates in a file, to prepare the examples for training and testing the neural network. A total of 56 measurement points are identified on the map and collected, during different periods of the day.

5 Selection of the multi-layer perceptron architecture

A labeled training set (given by inputs signals and corresponding output locations) is used by the OSS learning algorithm to determine the free parameters of the flexible MLP architecture. The measure of the error on the training set given by eq. 2 is minimized by OSS during the learning procedure.

It is essential to note that the objective of the training algorithm is to build a model with good *generalization* capabilities when confronted with new input values, values not present in the training set. The generalization is related both to the number of parameters and to the length of the training phase. In general, an excessive number of free parameters and an excessively long training phase (*over-training*) reduce the training error of eq. 2 to small values but prejudicate the generalization: the system *memorizes* the training patterns and does not extract the regularities in the task that make generalization possible.

The theoretical basis for appropriate generalization is described by the theory of the Vapnik - Chervonenkis (VC) dimension [22]. Unfortunately, the VC dimension is not easily calculated for a specific problem and experimentation is often the only way to derive an appropriate architecture and length of the training phase for a given task.

The purpose of the experiments in this section is to determine the architecture, in our case the number of hidden units, and the length of the training phase leading to the best generalization results. Fig. 2 describes a significant summary of the results obtained in the experiments.

The three architectures considered are given by 4, 8, and 16 hidden units. A set of labeled examples (signal strengths and correct location) has been collected as described in Sec. 4.1. Among all examples collected, 200 are extracted randomly and are used for the training phase, the remaining ones are used to test the generalization, at different steps during the training process. The plotted value is the average absolute distance error DE over all patterns:

$$DE = \frac{1}{P}\sum_{p=1}^{P}\sqrt{(tx_p \; - \; ox_p(w))^2 + (ty_p \; - \; oy_p(w))^2} \tag{7}$$

where $ox_p(w)$ and $oy_p(w)$ are the x and y coordinates obtained by the network and tx_p and ty_p the correct ”target” values. P is the number of test or training patterns, depending on the specific plot.

Both the training error and the generalization error are shown in each figure. As expected, the training error decreases during training, while the generalization error first decreases, then reaches a plateau value and finally tends to increase (*over-training* effect). The *over-training* effect is particularly strong for the architecture with 16 hidden units. The best generalization values (of about 1.52 meters) are reached after about 4,000 iterations for both the 8 and 16 hidden units architectures.

The robustness of the MLP model for different architectures and for different lengths of the training phase is to be noted. When the architecture changes from 4 to 8 to 16 hidden units, the optimal generalization value changes only by less than 5% (from 1.6 meters to about 1.52 meters). When the number of iterations increases from 2,000 to 20,000 the generalization error worsens only by a few percent points, in particular for the more compact architectures (4 and 8 hidden units).

After this series of tests, the architecture $3 \rightarrow 8 \rightarrow 2$ apparently achieves close to optimal generalization values (of about 1.53 meters) and is less subject to overtraining that the more redundant $3 \rightarrow 16 \rightarrow 2$ architecture. The structure of the neural network used in the subsequent tests consists of three layers as shown in Fig. 3: 3 input units, 8 hidden layer units and 2 outputs. The network structure is feed-forward and fully connected.

The CPU time for a single training session (2,000 iterations with 300 examples) on the architecture $3 \rightarrow 8 \rightarrow 2$ is of about 13.2 seconds on a 1400 MHz Pentium IV.

6 Improvement of measurement test error with number of examples

While the tests in Section 5 have been dedicated to evaluating the impact of the architecture and the length of the training period on the location accuracy, the experiments in this Section analyse in more detail the accuracy that can be obtained as a function of the number of training examples.

The first experiment considers 56 examples collected at a specific period during the day. Fig. 4 shows the average distance error DE as a function of the number of examples present in the training set, the remaining examples being used to test the trained neural net. The examples used for the training are selected randomly for each trial. We made 100 repetitions for the selection of the training sets. For convenience, the average over all trials is also plotted.

It can be observed that about five random examples are sufficient to produce a test distance error of less than 3 meters, already sufficient to localize a mobile terminal within a single room, in most cases. This is an indication that, once a map of the environment is available (without knowing the position of the three APs), a user may quickly train the system to recognize the position by visiting about five different places and determining their positions on the given map. When the number for examples increases, the accuracy improves, to reach a value of about 1.9 meters for a number of examples equal to 45. After a careful examination of the data we discovered that in a fraction of the test points, only two of the three signals are present (when this event occurs one of the signals is set to the lowest possible value). This is also caused by the fact that only two APs are equipped with an external antenna. In this case, the distance error tends to be larger.

A second experiment considers also the variability of the signal strengths during the day, a variability caused for example by the different number of people in the rooms affecting the signal propagation characteristics. A total of 8 collections of signal strengths has been executed at different times of the day, ranging from 8:30am to 6:30pm, for a total of 448 examples.

Fig. 5 shows the (test) distance error obtained as a function of the number of training examples. For each trial, the specified number of examples is extracted randomly from the complete series, while the remaining examples are used for testing. For this experiment we repeat 100 times the random selection of the training sets.

It can be observed that the distance error decreases rapidly as a function of the number for training examples, to reach a limiting value of about 1.5 meters for approximately 300 examples. Let us note that the second experiment is more difficult because now the environmental characteristics must be also taken into account by the neural network model.

The detailed histogram of the test error is shown in Fig. 6. The 50^{th} percentile (median) is 1.69 meters.

For a comparison, the results presented in [3] are of 8.16 meters (50^{th} percentile) for the “strongest base station” method (location of terminal guessed to be the same as the base station with the strongest signal), 2.94 meters for the empirical method proposed in the cited paper, and 2.75 meters by averaging over 5 nearest neighbors.

7 Conclusions

We considered a neural network (the multi-layer perceptron) for building a flexible mapping between the raw signal measurements and the position of the mobile terminal.

The average accuracy reached when the environmental changes during the day are also taken into account is of approximately 2.3 meters, therefore improving the previous state of the art results [3]. The positive features of the system are its reliance on a standard wireless LAN infrastructure, its respect for privacy (the position computation is executed at the mobile station, the system is informed only if the user desires), its simplicity and speed. The training phase does not require the knowledge of the positions of the base stations and training can be done incrementally, by identifying points on a map and running the OSS algorithm. The collection of about five known points is sufficient to determine the position within about 3 meters of accuracy.

We plan to extend the present work to consider different neural network and machine learning methods, in particular by using the structured risk minimization principle presented in [21].

Acknowledgments

We would like to thank Pietro De Vigili, Dr. Kiss Kallo Csaba, and Dr. Davide Aprea for their help in collecting the examples and conducting the experiments, Dr. P. Bahl and Prof. A. Wolisz for their help in obtaining the software for getting the signal strengths data in the IEEE802.11b system.

References

1. J. Anhalt, A. Smailagic, D. P. Siewiorek, F. Gemperle, D. Salber, S. Weber, J. Beck, and J. Jennings. Toward context-aware computing: Experiences and lessons. *IEE Intelligent System*, 3(16):38–46, May-June 2001.
2. P. Bahl, V. N. Padmanabhan, and A. Balachandran. A software system for locating mobile users: Design, evaluation, and lessons. Technical report, Microsoft Research, MSR-TR-2000-12, April 2000.
3. Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *IEEE INFOCOM 2000*, pages 775–784, March 2000.
4. R. Battiti. Accelerated back-propagation learning: Two optimization methods. *Complex Systems*, 3(4):331–342, 1989.
5. R. Battiti. First-and second-order methods for learning: Between steepest descent and newton's method. *Neural Computation*, 4:141–166, 1992.
6. R. Battiti and G. Tecchiolli. Training neural nets with the reactive tabu search. *IEEE Transactions on Neural Networks*, 6(5):1185–1200, 1995.
7. C. Bishop. Exact calculation of the hessian matrix for the multilayer perceptron. *Neural Computation*, 4:949–501, 1992.
8. G.M. Djuknic and R.E. Richton. Geolocation and assisted GPS. *IEEE Computer*, pages 123–125, February 2001.
9. Andy Harter and Andy Hopper. A distributed location system for the active office. *IEEE Network*, 6(1):62–70, jan-feb 1994.
10. Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster. The anatomy of a context-aware application. In *MOBICOM 1999*, pages 59–68, August 1999.
11. Jeffrey Hightower and Gaetano Borriello. Location system for ubiquitous computing. *IEEE Computer*, 34(8):57–66, August 2001.
12. Jeffrey Hightower, Gaetano Borriello, and Roy Want. *SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength*. The University of Washington, Technical Report: UW-CSE 2000-02-02, February 2000.

13. Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4:251–257, 1991.
14. U. Kubach and K. Rothermel. Exploiting location information for infostation-based hoarding. In *Seventh Annual International Conference on Mobile Computing and Networking, Rome*, pages 15–27, 2001.
15. A. Misra, S. Das, A. McAuley, and S. K. Das. Autoconfiguration, registration and mobility management for pervasive computing. *IEEE Personal Communications (Special Issue on Pervasive Computing)*, 8(4):24–31, Aug 2001.
16. Luis Munoz, Marta Garcia, Johnny Choque, Ramon Aguero, and Petri Mahonen. Optimizing internet flows over IEEE 802.11b wireless local area networks: A performance-enhancing proxy based on forward error correction. *IEEE Communications Magazine*, 39(12):60–67, December 2001.
17. S. Pradhan, C. Brignone, J.-H. Cui, A. McReynolds, and M.T. Smith. Websigns: hyperlinking physical locations to the web. *IEEE Computer*, 34(8):42–48, August 2001.
18. Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *MOBICOM 2000*, pages 32–43, August 2000.
19. T. S. Rapport. *Wireless Communications - Principles and Practice*. IEEE Press, 1996.
20. D. F. Shanno. Conjugate gradient methods with inexact searches. *Mathematics of Operations Research*, 3(3):244–256, 1978.
21. V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.
22. V.N. Vapnik and A. Ja. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probab. Apl.*, 16:264–280, 1971.
23. Roy Want, Andy Hopper, Veronica Falcao, and Jonathan Gibbons. The active badge location system. *ACM Transaction on Information Systems*, 10(1):91–102, January 1992.
24. Andy Ward, Alan Jones, and Andy Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5):42–47, October 1997.
25. Jay Werb and Colin Lanzl. Designing a positioning system for finding things and people indoors. *IEEE Spectrum*, 35(9):71–78, September 1998.

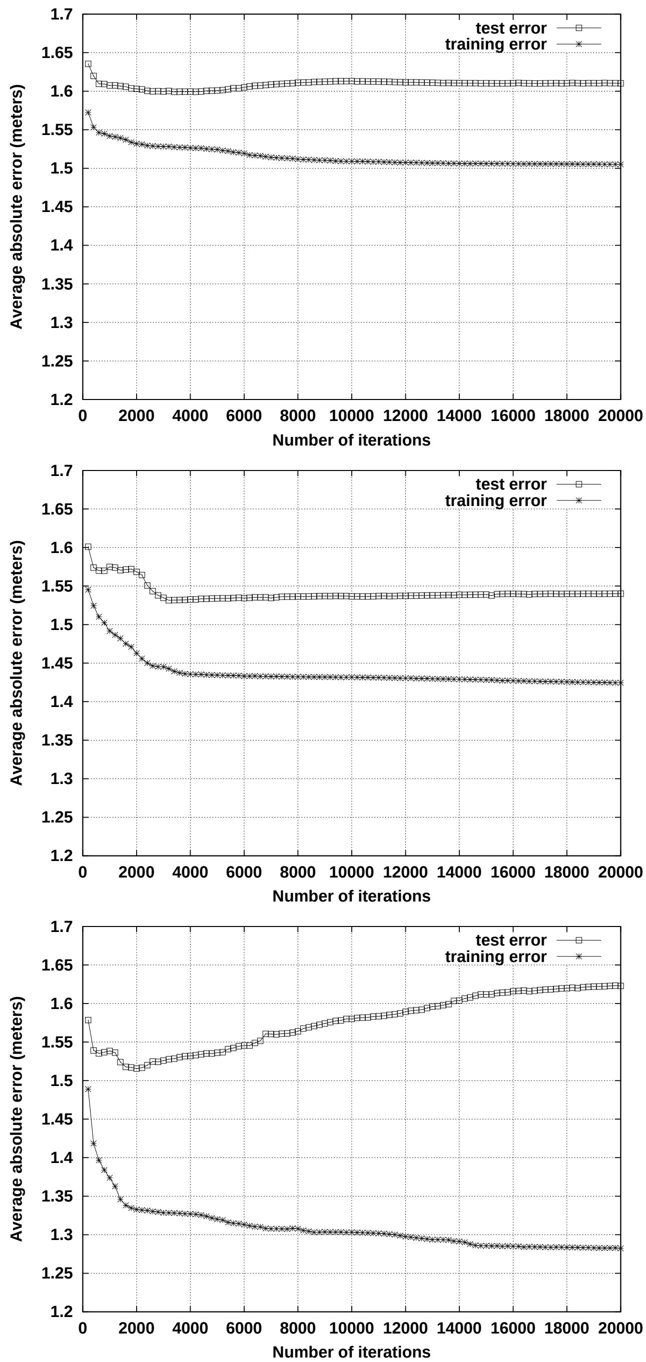

Fig. 2. Training and test error for architecture $3 \rightarrow 4 \rightarrow 2$ (top), $3 \rightarrow 8 \rightarrow 2$ (middle), $3 \rightarrow 16 \rightarrow 2$ (bottom).

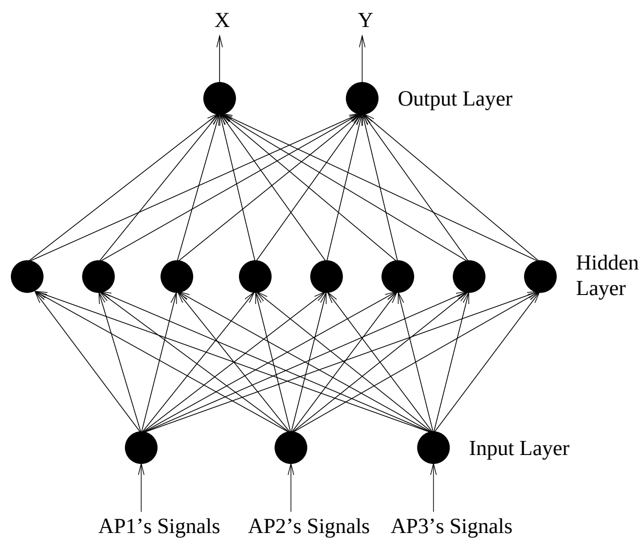

Fig. 3. The multi-layer perceptron configuration

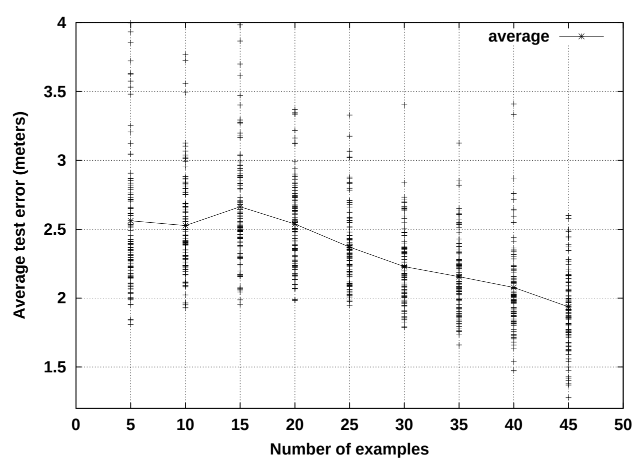

Fig. 4. Reduction of average distance error (test) as a function of the number of training examples (patterns at a single time of the day)

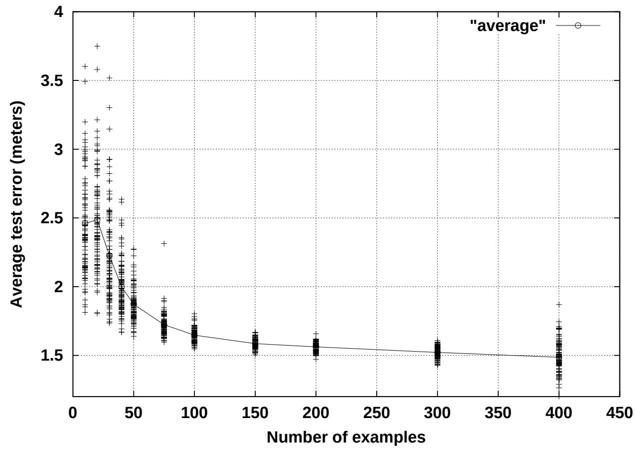

Fig. 5. Reduction of average distance error (test) as a function of number of training examples (patterns at different times of the day).

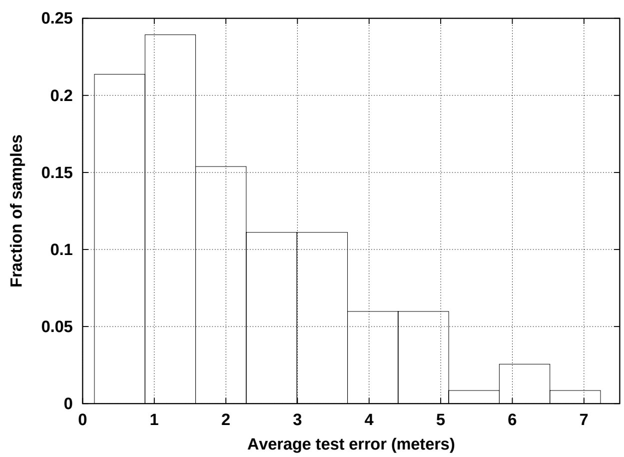

Fig. 6. The histogram of the test error.