Journal of Mobile Communication 1 (3): 101-108, 2007
ISSN: 1990-794X

Intelligent Location Tracking Strategy for Managing Users' Mobility in UMTS Networks

[1]J. Amar Pratap Singh, [2]V.K. Govindan and [2]M.P. Sebastin
[1]N.I. College of Engineering, Thuckalay, TamilNadu, India
[2]National Institute of Technology,Calicut, Kerala, India

Abstract: Present generation mobile systems provide access to a wide range of services and enable mobile users to communicate regardless of their geographical location and their roaming characteristics. Due to the growing number of mobile users, global connectivity and the small size of cells, one of the most critical issues regarding these networks is location management. In recent years, several strategies have been proposed to improve the performance of the location management procedure in UMTS networks. In this study, we present a User Pattern Learning strategy (UPL) using cascaded correlation neural networks to reduce the location update signaling cost by increasing the intelligence of the location procedure in UMTS. This strategy associates to each user a list of cells where the user is likely to be with a given probability in each time interval. The implementation of this strategy has been subject to extensive tests. The results obtained confirm the efficiency of UPL in significantly reducing the costs of both location updates and call delivery procedures when compared to the UMTS standard and with other strategies well-known in the study.

Key words: Location update, mobility management, user pattern learning, UMTS, neural networks

INTRODUCTION

Owing to the increasing population of mobile subscribers, smaller sized cells have been used to accommodate the large number of Mobile Terminals (MT's). Sending paging signals to all cells within a Location Area (LA) to locate an MT may result in an excessive amount of network bandwidth. Therefore, more sophisticated schemes (Akyildiz *et al*., 1996; Jeong and Jeong, 1998) were proposed to make the location update and terminal paging operations more efficient. These schemes include the time-based, movement-based and distance-based schemes which locate an MT by paging the LA's ring by ring from its last updated location. In this paper, we propose a location update scheme based of cascaded correlation neural networks where an MT updates its location only when it's moving direction changes. To locate an MT, paging can be carried out along its moving direction and hence the paging cost is reduced. Moreover. The MT's moving direction can be determined by simple numerical calculations.

Location management methods are classified into two major groups: Memory-based and non-memory-based methods. The first group includes methods based on learning processes, which require knowledge of mobile user behavior, while the second group includes methods based on specific algorithms and network architectures. The strategy proposed in this paper belongs to the first group. In North America, the IS-41 standard is used for both the location update and call delivery procedures. This standard deploys a two-level database architecture consisting of a single Home Location Register (HLR) and several Visitor Location Registers (VLR). The HLR for a given network contains the network's subscriber profiles, while a VLR stores the profiles of the users that are currently roaming within LAs associated with that specific VLR. Third-generation mobile networks are characterized by high user density and high mobility (like current 2G systems) and small cell sizes, which will increase the number of location updates and handoff messages, thus limiting the switching capacity and available bandwidth. Reducing the signaling and database access costs of location management introduces significant technical challenges which have to be dealt with and constitutes an important research area. Several alternative strategies have recently been proposed to improve the performance of the location management scheme (Cayirci and Akyildiz, 2002; Cken and Gu, 2003; Ho and Akyildiz, 1997, 1995; Hu and Tan, 2004).

There are 2 basic operations in location management: Location update and paging. Location update is the process through which system tracks the locat ion of mobile terminals that are not in conversations. The mobile terminal reports its up-to-date location

Corresponding Author: J. Amar Pratap Singh, N.I. College of Engineering, Thuckalay, TamilNadu, India

information dynamically. A PA may include one or more cells. When an incoming call arrives, the system searches for the mobile terminal by sending polling signals to cells in the PA. This searching process is referred to as paging. To perform location update or paging will incur a significant amount of cost (e.g., wireless bandwidth and processing power at the mobile terminals, the base stations and databases), which should be minimized in the systems.

A User Pattern Learning strategy (UPL) associates with each user a list of Location Areas (LA) where she is most likely to be located within a given time interval. When a call arrives for an MT, each location within the list is paged sequentially until the MT is found. When a user moves between locations within the list, no location update is required. The list is stored at an Intermediate Location Database (ILD) associated with a Mobile Switching Centers (MSC) as well as within the user's MT. The cost reduction depends on the behavior of each class of user. It can be assumed that, when the user follows its expected behavior, the cost of a location update is reduced.

In the UPL, if the position of a user is always known in advance, then no explicit registration is necessary. Thus, the optimal location area is given by a single cell, which, in turn, minimizes paging costs. Stationary users on fixed schedules exhibit this type of behavior. If mobile users are classified into categories, as was done earlier, the system could treat each category differently to minimize system costs. Furthermore, user mobility information can be used to assist mobility management (traffic routing), to manage network resources (resource allocation, call admission control, congestion and flow control) and to analyze handoff algorithms in integrated wired/wireless networks. User mobility patterns can also be useful for system recovery (Gil *et al*., 2001). By means of a fuzzy logic algorithm, a users' location is forecast by the system, which eliminates the need of a backup. This way, users will not experience long service delays.

Our strategy differs from the others user profile strategies in the following aspects:

We use a Cascaded Correlation Artificial Neural Network (CCANN) (Fahlman and Lebiere, 1990) to learn about the users' regular routines. Pattern recognition is one of the fields where neural networks (ANNs) have been strongly applied from many years. Pattern learning and classification can be stated as the problem of labeling test patterns derived by a particular application domain. A classification system may be trained by a set of data features, adequately prepared or not. We have divided our strategy into two steps: Training and application. In general, ANN systems are capable of learning trends in a given data set and establishing input-output relationships based strictly on a test set of data. It is desirable for the test data that the system learns from to be as representative of the complete data set as possible; trends not seen in the test data set will not be "learned" by the neural network system. After training, the network is ready for application. For satisfactory application, it is essential that the training data contain input sets (and the associated output values) that represent the entire range of possible future inputs; the system will only perform as well as it has been trained. The training examples may contain errors. ANN learning methods are quite robust to noise in the training data. The ability to learn is a fundamental trait of intelligence. Although a precise definition of learning is difficult to formulate, a learning process in the ANN context can be viewed as the problem of updating network architecture and connection weights (an elementary structure and functional unit between two neurons) so that a network can efficiently perform a specific task. The network usually must learn the connection weights from available training patterns. Performance is improved over time by iteratively updating the weights in the network. ANN's ability to automatically learn from examples makes them attractive. Instead of following a set of rules specified by human experts, ANNs appear to learn underlying rules (like input-output relationships) from the given collection of representative examples. This is one of the major advantages of neural networks over traditional expert systems. Neural networks derive their computing power through their ability to learn and then generalize; generalization refers to the ability of the neural network to produce reasonable outputs for inputs not encountered during training. It is this quality that we utilize to predict the movement of mobile users so that we can predict the position of a user in advance and reduce the paging cost based on the predicted destination cell.

Finally, the impact on the performance of location management with CCANN is reduced. The cost of the UPL is decomposed into four components: training procedure, maintenance and update of the user's profile, location update and call delivery. Although CCANN learning times are relatively long, evaluating the learned network in order to apply it to a subsequent instance (maintenance and update of the user's profile, location update and call delivery) is very fast. In CCANN, performance is improved over time by iteratively updating the weights in the network as compared to conventional ANN algorithms. This study proposes a user pattern learning strategy that reduces the signaling cost of a location update by increasing the intelligence in the location procedure.

UMTS NETWORK ARCHITECTURE

Universal Mobile Telecommunications Service (UMTS) can be viewed as an evolution of GSM that supports 3G services. Generally, a UMTS network is divided in an access network and a core network (Fig. 1). The former is dependent on the access technology, while the latter can theoretically handle different access networks. The access network is known as the UMTS Terrestrial Radio Access Network (UTRAN). The UTRAN is comprised of two types of nodes, the Radio Network Controller (RNC) and the Node B, which is a Base Station (BS). It controls the radio resources within the network and can interface with one or more stations (Node Bs).

The air interface used between the User Equipment (UE) and the UTRAN is WCDMA. This interface is called Uu in UMTS. The UTRAN communicates with the core network over the Iu interface. The Iu interface has two components: the Iu-CS interface, supporting Circuit-Switched (CS) services and the Iu-PS interface, for Packet-Switched (PS) services.

The RNC that controls a given Node B is known as the Controlling RNC (CRNC). For a given connection between a UE and the core network, only one RNC can be in control, the Serving RNC (SRNC). The CRNC controls the management of radio resources for the Node B that it supports. The SRNC controls the radio resources that the UE is using. It is possible for a CRNC and a SRNC to coincide. UTRAN supports soft handovers (where an UE is communicating with a Node B whose CRNC is not the SRNC). The Iur interface's purpose is to support this type of handover, that is, inter-RNC mobility.

Mobility management issues in multitier PCS systems are presented in (Cayirci and Akyildiz, 2002). Hwang *et al.* (2000) propose a direction-based location update method that uses line paging for reducing paging costs. This approach is based on a conventional two-dimensional random walk model, where the directions of the MTs are assumed to be independent and identically distributed.

Aljadhai and Znati (2001) deal with the question of how to support Quality of Service (QoS) through an exact knowledge of the trajectory followed by the MT. For this solution to work, it is crucial to have an efficient prediction mechanism for the user's trajectory. With an estimation of trajectories, arrival and departure times, the system is able to support QoS requirements. This approach does not necessitate the memorization of users' profile within the network.

Wu *et al.* (2002) present a new analytic framework for dynamic location management of PCS networks.

Mobility Management in UMTS: In standard UMTS, Mobile Switching Centers (MSCs) are responsible for the circuit switched location management, while Serving GPRS Support Nodes (SGSNs) assume the packet switched location management. Both domains are linked over some interfaces, but the information is kept in separate network entities: The CS location information is in the MSC, while the PS location information is in the SGSN. The HLR is a common location information database for both domains. Several area types have been defined in UMTS to handle the location information:

- Location Areas (LA) are the same as in GSM.

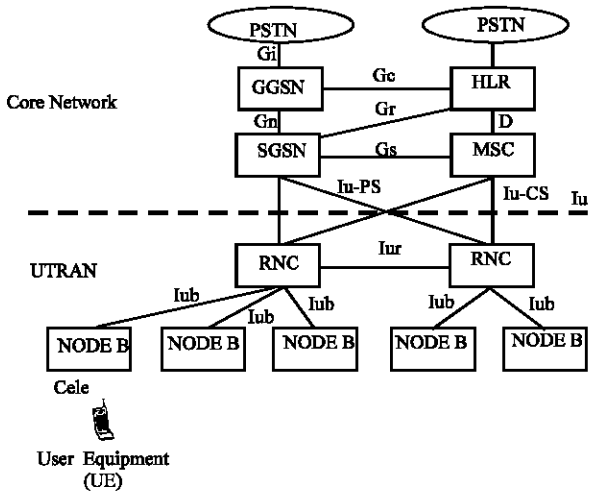

Fig. 1: UMTS network architecture

- A Routing Area (RA) is composed of a group of cells that belong to only one RA. Several RAs can be included in the same LA, but an RA cannot span more than one LA. The PS domain to track the MT's location when in idle mode uses the RAs.
- UMTS Registration Areas (URAS) are an intermediate level between cells and RAs (or LAs). They are similar to RAs and LAs, but are used by the UTRAN to set trade-offs between the MT's location accuracy and signaling load. Furthermore, they are used to track the MT's location while it is in connected mode without using a logical channel. This concept is optional in UTRAN.
- Cells are related to the provision of radio coverage. The idea of having this diversity is to allow a trade-off between location accuracy and paging (Cayirci and Akylidiz, 2002).

User pattern learning strategy: In this study, we introduce our UPL location management strategy. We also present protocols and algorithms supporting this scheme. Furthermore, several application scenarios are proposed. In the following discussion, we define the anchor LA of an MT as the LA whose location is updated at the HLR. When an MT changes LA, it updates the pointer at the anchor LA. This way, no update is made at the HLR.

Overview of the user pattern learning: The main motivation behind the use of Artificial Neural Networks (ANN) is their ability to learn relationships in complex data sets that may not be easily perceived by humans. Learning and generalization are perhaps the most important topics in neural network research. Learning is the ability to approximate the underlying behavior adaptively from the training data, while generalization is the ability to predict well beyond the training data. The basic element of an Artificial Neural Network (ANN) system is called a neuron. The neuron accepts one input x, which may actually be a sum of multiple inputs and produces an output value y based on a nonlinear function. In general, ANN systems are capable of "learning" trends in a given data set and establishing input-output relationships based strictly on a "test" set of data. It is desirable for the "test" data that the system "learns" from to be as representative of the complete data set as possible; trends not seen in the test data set will not be "learned" by the neural network system (Halpin and Burch, 1997).

Typically, ANNs are organized in several layers of elementary units called neurons, each computing a non-linear function of a weighted sum of its inputs (Fig. 2). The learning phase of an ANN is based on algorithms

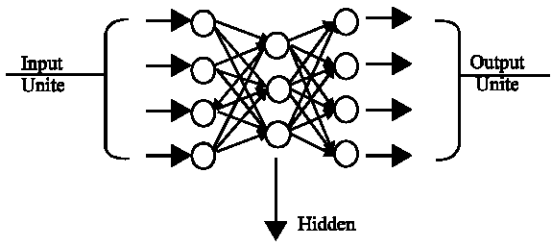

Fig. 2: ANN Architecture

(e.g., backpropagation) able to set the weight connections by training the net with a known data set until a certain (small) error is achieved. Unfortunately, the design of networks of practical interest is not an easy task as the whole topology of the network, i.e., number of hidden layers, number of neurons inside each hidden layer, the connections between the neurons, etc., should be specified in advance. The hidden layers are called "hidden" units because their output is available only within the network and is not available as part of the global network output. Typically, trial-and-error and heuristic procedures are used to determine the network topology according to some criteria (e.g., minimum classification error). In practice, for a given network topology and (real) training set, the best possible set of weights cannot be guaranteed by the back propagation algorithm (Halpin and Burch, 1997).

The number of nodes used in the input layer usually depends on the type and amount of input data; several hundred input nodes may be used in large applications. The number of nodes in the hidden layer determines, in general, the ability of the network to learn complex relationships. There may be multiple hidden layers to increase the network's ability to learn. In our model, with the BP algorithm, there are three layers in the Neural Networks, input layer, hidden layer and output layer. The role of the hidden layer is to remap the inputs and results of previous layers to achieve a more separable or classifiable representation of the data and allow attachment of semantics to certain combinations of layer inputs.

ANNs perform their calculations using nonlinear functions and simple multiplying factors, called weights, which are associated with a pathway between any two nodes. While the functions remain constant for any given application, the weights are updated in such a manner that the complete network "learns" to produce a specific output for a specific input. The process of adjusting the weights to achieve a specified accuracy level is referred to as "training." The backpropagation (BP) training algorithm is a method for iteratively adjusting the

weighting factors until the desired accuracy level is achieved. This algorithm is based on a gradient-search optimization method applied to an error function (i.e., the sum of squared error). In our approach, we use the learning process to derive a list from which we can find, with high accuracy, the exact cell in which the MT resides at any time of every day. The learning process is able to derive such a list after observing (learning) the behavior of a mobile user for a certain period of time.

By observing the mobile user's daily behavior, we use the BP algorithm to learn the behavior. With some useful data from observation of the mobile user as the input nodes, we can obtain the output as the result we want, which is the cell information of the mobile user on observation, that is to say, the cell list for a mobile user.
For every mobile user there is a user pattern learning process associated to it. We may classify the users into three different categories depending the predictability of their daily routine: Users who have a very high probability of being where the system expects them to be (deterministic users), users who have a certain likelihood of being where the system expects them to be (quasi-deterministic users) and users whose position at a given moment is unpredictable (random users), similar to the classification proposed in Pollini and Chih-Lin (1997). The predictability of deterministic and quasi-deterministic users can be used by the system to reduce the number of location update operations. So, after the learning process completes, we get the mobile user's behavior associated with known location areas. Then, a profile is built for the mobile user (Fig. 3). When a call arrives for a mobile, it is paged sequentially in each location within the list. When a user moves between location areas in this list, no location updates are required. The list is stored at the HLR in the Information Database (ID) as well as in the user's mobile terminal. The cost reduction depends on the behavior of each class of user. It can be assumed that, when the user follows its expected behavior, the location update cost is reduced, even if accesses to HLR are minimized when calls are received from relatively close areas.

Our strategy increases the intelligence of the location update procedure and utilizes replication and locality to reduce the cost incurred from the paging procedure.

An Intermediate Location Database (ILD) is added to the UPL scheme. This database is located on the same architectural level as the MSC and contains the profile of each user. Furthermore, the ILD contains a flag for each registered MT and indicates whether or not the MT is roaming under that particular MSC. The flag helps us exploit the "locality" property of calls (i.e., incoming calls from the same MSC). Moreover, our strategy takes advantage of calls placed regularly to a particular user

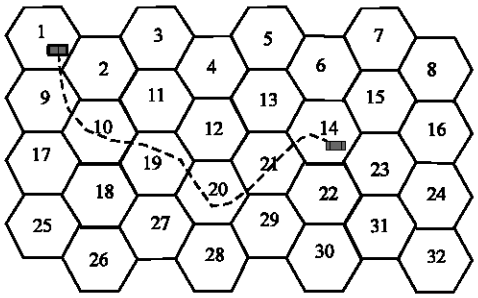

Fig. 3: User movement path

(i.e., a great amount of calls are placed by the MT's top five callers). Thus, some ILDs will store "location data tables" that contain pointers tracking the called MT's location. This may somewhat increase the location update cost since the MT must use these pointers each time it changes location. On the other hand, using pointers significantly reduces paging costs.

An user behaviour is as shown in Fig. 3. A user's profile is made up of several fields. The first field is the profile number. Each user might have several profiles, each containing a different behavioral pattern. The next two fields contain the IDs of the MSC and the LA under which an MT might be roaming. The Expected Entry Time (EET) field indicates the time interval within which the system expects to locate the MT in a particular LA. The Timestamp field indicates the date and time an MT has entered a new LA. Finally, the Number of Visits field saves the number of times that an MT has been roaming under a particular LA. When a MT enters a new LSTP that he has never seen before, the MSC creates a basic profile containing a single record with MT's current LA.

The UMTS network consists of three interacting domains: Core Network, UMTS Terrestrial Radio Access Network and Mobile Terminal (Fig.1). The main function of the core network is to provide switching, routing and transit for user traffic. The Core network also contains databases (like HLR) and network management functions. The UPL process is in the Mobile Switching Center (MSC) in the core network, The UPL associates with each user a list of Location Areas (LA) where she is most likely to be located within a given time interval. This list is stored at an Intermediate Location Database (ILD) associated with an MSC as well as within the user's MT.

Backpropagation algorithm: We use the BP algorithm to implement the learning process. In our problem, the learning process aims to derive a list with which we can find the cell in which the MT locates at any time of every day with high accuracy after observing the behavior of the mobile user for a period, for example, a month.

There are three layers in the Neural Networks: Input layer, hidden layer and output layer. The role of the hidden layer is to remap the inputs and results of previous layers to achieve a more separable or classifiable representation of the data and allow attachment of semantics to certain combinations of layer inputs.

In Fig. 2, we present a simplified ANN use in our approach. The circles represent units or neurons that are extremely simple analog computing devices. The numbers within the circles represent the activation values of the units. The main nodes are arranged in layers. In this study, there are three layers, the input layer that contains the values for Day and Time, a hidden layer that contains three nodes, h1, h2, h3 and two output units that gives the value of the output values, it's the probability that reside each cell in the User Mobile History (UMH). The hidden layer is so named because the network can be regarded as a black box with inputs and outputs that can be seen, but the hidden units cannot be seen. The lines connecting the circles represent weights and the number beside a weight is the value of the weight.

To compute the value of the output units, residence probability, we place values for Day and Time on the input layer units. Let these values be Monday, 10:15, as shown in Table 1. First, we compute the value of the hidden layer unit, hs. The first step of this computation is to look at each lower level unit. For each of these connections, find the value of the unit and multiply by the weight and sum all the results.

The steps of BP: The error back-propagation algorithm can be outlined as:

Step 1: Initialize all weights to small random values.

Step 2: Choose an input-output training pair.

Step 3: Calculate the actual output from each neuron in a layer by propagating the signal forward through the network layer by layer (forward propagation).

Step 4: Compute the error value and error signals for output layer.

Step 5: Propagate the errors back ward to update the weights and compute the error signals for the preceding layers.

Table 1: An example training set

Examples	Di(Day)	Ti(Time)	Ci (Cell Id)	Probability(%)
E1	Monday	02.15	5,1,6	90,50,10
E2	Monday	11.45	5,2,1	95,50,10
E3	Sunday	12.00	5,6,3	40,90,15
E4	Thursday	14.15	5,2,1	95,50,10
E5	Thursday	05.30	5,1,6	90,50,10

Step 6: Check whether the whole set of training data has been cycled once, yes-go to step 7; otherwise go to step 2.

Step 7: Check whether the current total error is acceptable; yes- terminate the training process and output the field weights, otherwise initiate a new training epoch by going to step 2.

The cascade correlation algorithm

Initial configuration: The algorithm begins with a simple perceptron with N input units (Fahlman and Lebiere, 1990) and M output units. N and M are chosen on the basis of the problem that the network is to learn as shown in Fig.4. Initial training:

The perceptron is trained on the entire training set {(Vp,Tp) | p = 1,..., P }, until the performance of the network is as good as possible. If the desired performance is obtained, the algorithm stops. Otherwise: Start adding hidden units to the network, one by one.

Training of candidates: A pool of candidates for a new hidden unit is generated. This pool emulates a stochastic search in the weight space, which will decrease the risk of inserting a candidate stranded in a local minimum with high error. Each node in the pool of candidates is connected to all input nodes and all previously inserted hidden units. Each of the candidates is trained with the purpose of maximizing some measure of "goodness" of the candidate.

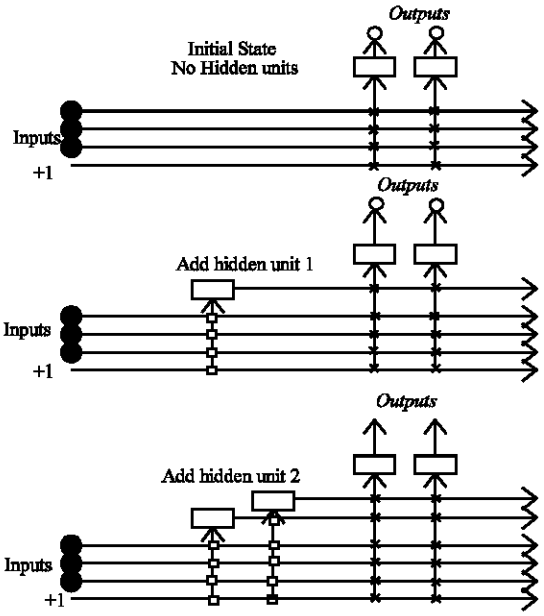

Fig. 4: The Cascade architecture, initial state and after adding two hidden units

Inserting a new hidden unit: The candidate with the highest score is inserted "for real" in the network as a new hidden unit. The incoming weights to the new hidden unit are then frozen, i.e. they are not to be changed anymore. The new hidden unit is connected to all output nodes with random weights.

Retraining the network: All the incoming weights to the output units are retrained in order to adjust the weights from the newly inserted hidden unit. If the performance of the network is satisfying after retraining, the algorithm stops. Otherwise: Go to 3.

Cascade correlation neural network architecture: A cascade correlation network consists of input units, hidden units and output units. Input units are connected directly to output units with adjustable weighted connections. Connections from inputs to a hidden unit are trained when the hidden unit is added to the net and are then frozen. Connections from the hidden units to the output units are adjustable consequently.

Cascade correlation network starts with a minimal topology, consisting only of the required input and output units (and a bias input that is always equals to 1). This net is trained until no further improvement is obtained. The error for each output until is then computed (summed over all training patterns). Next, one hidden unit is added to the net in a two-step process. During the first step, a candidate unit is connected to each of the input units, but is not connected to the output units. The weights on the connections from the input units to the candidate unit are adjusted to maximize the correlation between the candidate's output and the residual error at the output units. The residual error is the difference between the target and the computed output, multiplied by the derivative of the output unit's activation function, i.e., the quantity that would be propagated back from the output units in the back propagation algorithm. When this training is completed, the weights are frozen and the candidate unit becomes a hidden unit in the net. The second step in which the new unit is added to the net now begins. The new hidden unit is then connected to the output units and the weights on the connections being adjustable. Now all connections to the output units are trained. (Here the connections from the input units are trained again and the new connections from the hidden unit are trained for the first time.) A second hidden unit is then added using the same process. However, this unit receives an input signal from the both input units and the previous hidden unit. All weights on these connections are adjusted and then frozen. The connections to the output units are then established and trained. The process of adding a new unit, training its weights from the input units and the previously added hidden units and then freezing the weights, followed by training all connections to the output units, is continued until the error reaches an acceptable level or the maximum number of epochs (or hidden units) is reached.

The purpose of inserting a new unit is to reduce the total error of the network. The way the CCA does this is to train the candidate unit so the correlation between the residual error and the output from the candidate is maximized. Let X and Y be two stochastic variables. Then the correlation between X and Y is defined as:

$$\mathrm{Corr}[X,Y] = \frac{\mathrm{Cov}[X,Y]}{\sqrt{\mathrm{Var}[X]\mathrm{Var}[Y]}}$$

IMPLEMENTATION AND ANALYSIS

We analyze the sensitivity of the CCA algorithm to the learning rate in the UPL approach. We generated 2400 samples for each set, assuming normal distributions using the statistics estimated from the real data. We used 1500 samples of each set as training data and the rest as test data. Then, we chose to use the generated data to avoid the problem of incorrect estimation of parameters. If training data is not a good representative of the sets, then the classification accuracies of training and test data might be very different, resulting in difficulty in comparing the performance of the learning algorithms. The training performance and the classification accuracies are shown in Fig. 5 and Table 2, respectively.

Table 2: Classification Accuracies of CCA Neural Networks

Learning rate	Training data	Test data
0.01	86.124	85.2
0.3	85.202	84.72

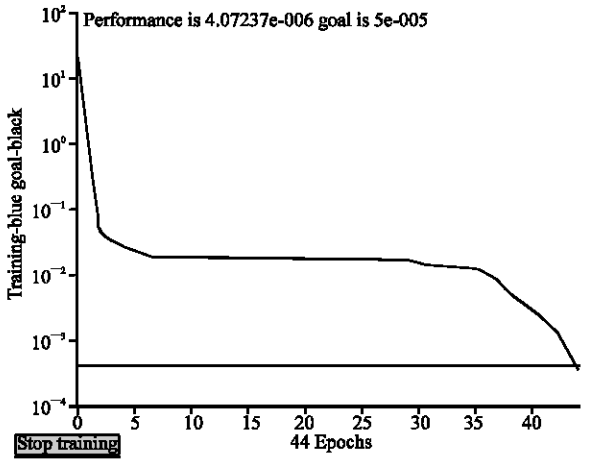

Fig. 5: Training performance curve

CONCLUSION

In this study, we present a user pattern learning strategy using cascaded neural networks to reduce location update signaling cost by increasing the intelligence of the location procedure in UMTS. This strategy associates to each user a list of cells where she is likely to be with a given probability in each time interval. The list is ranked from the most likely to the least likely place where a user may be found. When a call arrives for a mobile, it is paged sequentially in each location within the list. When a user moves between location areas in the list, no location updates are required. The results obtained from our performance evaluation confirm the efficiency and the effectiveness of UPL in comparison with the UMTS standard and other well-known strategy. This improvement represents a large reduction in location update and paging signaling costs.

REFERENCES

Akyildiz, F., J.S.M. Ho and Y.B. Lin, 1996. Movement-based location update and selective paging for PCS networks, IEEE/ACM Trans. Networking, 4: 629-638.

Aljadhai, A. and T. 2001. Znati, Predictive Mobility Support for QoS Provisioning in Mobile Wireless Environments, IEEE. J. Selected Areas in Comm., 19: 1915-1930.

Cayirci, E. and I.F. Akyildiz, 2002. User Mobility Pattern Scheme for Location Update and Paging in Wireless Systems, IEEE. Trans. Mobile Computing,1: 236-247

Chen, R. and B. Gu, 2003. Quantitative Analysis of a Hybrid Replication with Forwarding Strategy for Efficient and Uniform Location Management in Mobile Wireless Networks, IEEE. Trans. Mobile Computing, 2: 3-15.

Fahlman, S. E. and C. Lebiere,1990. The Cascade-Correlation Learning Architecture, in Neural Information Processing Systems, Editors D. Touretzky, Morgan Kaufmann Publishers, Inc., Denver, Colorado, pp: 524-532.

Gil, J., Y. Chan, C. Hwang, D. Park, J. Shon and Y. Jeong, 2001. Restoration Scheme of Mobility Databases by Mobility Learning and Prediction in PCS Networks, IEEE. J. Selected Areas in Comm., 19: 1962-1973.

Halpin, S. and R. Burch, 1997. Applicability of Neural Networks to Industrial and Commercial Power Systems: A Tutorial Overview, IEEE. Trans. Indus. Applications, 33: 1355-1361.

Ho, J. and I. Akyildiz, 1995. Mobile User Location Update and Paging under Delay Constraints, ACM. Wireless Networks, 1: 413-425.

Ho, J. and I. Akyildiz, 1997. Dynamic Hierarchical Database Architecture for Location Management in PCS Networks, IEEE/ACM Trans. Networking, 5: 646-660.

Hu, W. and T. Tan, 2004. A Hierarchical Self-Organizing Approach for Learning the Patterns of Motion Trajectories, IEEE. Trans. Neural Networks, 15: 135-144.

Hwang, H.., M. Chang and C. Tseng, A Direction-Based Location Update Scheme with a Line-Paging Strategy for PCS Networks, IEEE Comm. Letters, 4: 149-151.

Jeong, D.G. and W.S. Jeong, 1998. Probabilistic location update for advanced cellular mobile networks, IEEE. Commun. Lett., 2: 8-10.

Pollini, G. and I. Chih-Lin, 1997. A Profile-Based Location Strategy and Its Performance, IEEE. J. Selected Areas in Comm., 15: 1415-1424.

Wu, C., H. Lin and L. Lan, 2002. A New Analytic Framework for Dynamic Mobility Management of PCS Networks, IEEE. Trans. Mobile Computing, 1: 208-220.