

Environment-Adaptation Mobile Radio Propagation Prediction Using Radial Basis Function Neural Networks

Po-Rong Chang, *Member, IEEE*, and Wen-Hao Yang

Abstract—This paper investigates the application of a radial basis function (RBF) neural network to the prediction of field strength based on topographical and morphographical data. The RBF neural network is a two-layer localized receptive field network whose output nodes from a combination of radial activation functions computed by the hidden layer nodes. Appropriate centers and connection weights in the RBF network lead to a network that is capable of forming the best approximation to any continuous nonlinear mapping up to an arbitrary resolution. Such an approximation introduces best nonlinear approximation capability into the prediction model in order to accurately predict propagation loss over an arbitrary environment based on adaptive learning from measurement data. The adaptive learning employs hybrid competitive and recursive least squares algorithms. The unsupervised competitive algorithm adjusts the centers while the recursive least squares (RLS) algorithm estimates the connection weights. Because these two learning rules are both linear, rapid convergence is guaranteed. This hybrid algorithm significantly enhances the real-time or adaptive capability of the RBF-based prediction model. The applications to Okumura's data are included to demonstrate the effectiveness of the RBF neural network approach.

Index Terms—Propagation prediction, RBF neural networks.

I. INTRODUCTION

THE PROBLEM of finding an exact or approximation model for propagation loss occurs frequently in planning mobile communication systems. Nowadays, two strategies for propagation loss prediction are in use [1]–[3]. One is to derive an empirical formula for propagation loss from measurement data. The other strategy is a deterministic method that is based on the theory of diffraction. However, the deterministic methods suffer from excessive computation time and the need for very detailed databases. An empirical formula based on Okumura's results [1] has been developed by Hata [2] in order to make the propagation loss prediction easy to apply. Since the Okumura's results are based on field measurements taken in the Tokyo area, Hata's formulation is limited to certain types of environments that are similar to the Tokyo area. Meanwhile, propagation prediction models should be adapted to different types of environments. Hawthorne [4] has proposed

an environmental propagation loss model that is based on a plane earth propagation model. This method involved the use of a Kalman filter that utilizes propagation measurements to derive a best fit of the model to the measurement data. An alternative approach to a propagation loss prediction model with the flexibility to adapt to different environments is based on the multilayer perceptron (MLP) [5], [6]. Stocker *et al.* [6] demonstrated that a neural-based propagation loss model trained by back-propagation algorithms showed the superior accuracy of its prediction. However, the MLP has problems of slow convergence and unpredictable solutions during training.

To tackle this difficulty, this paper presents a means to approximate the propagation loss based on radial basis function (RBF) neural networks [7]–[9]. RBF neural networks offer a viable alternative to the two-layer perceptron in system identification and have recently attracted attention in the neural network community [7]–[9]. Similar to MLP neural networks, RBF's possess the properties of approximating nonlinear functions of several variables. Unlike the former, radial basis function expansions have a “linear in the parameters” representation. As a result, rapid convergence of the parameters can be guaranteed. Furthermore, Girosi and Poggio [10] showed that the RBF neural networks have the best approximation property, while MLP neural networks do not have this property. An approximation scheme is said to have this property if in the set of approximating functions there is one that has the minimum distance (norm) from the given function.

One of the major advantages of the RBF network is that learning tends to be much faster than in the MLP since the learning process of an RBF network can be broken into two stages, and the algorithms used in both stages can be made relatively efficient. In the first stage, a number of unsupervised clustering algorithms [11], [12] are used to adjust the RBF centers. Moody and Darken's m -means clustering algorithm [11] is the well-known method used to adjust centers in real-time. However, a key problem with the m -means algorithm is that m (the number of clusters) should be appropriately pre-selected, otherwise the algorithm will perform badly. For tackling this problem, a unsupervised competitive learning called rival penalized competitive learning (RPCL) is proposed to automatically and efficiently allocate appropriate locations of RBF centers [13]. The RPCL algorithm has been developed from the basic idea that for each input is not only the winner unit modified to adapt to the input, but also its rival

Manuscript received January 25, 1995; revised October 22, 1995. This work was supported in part by the National Science Council, Taiwan, R.O.C., under Contract NSC-84-2212-E-009-016.

P.-R. Chang is with the Department of Communication Engineering, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

W.-H. Yang is with Telecommunication Laboratories, DGT, MOTC, Taiwan, R.O.C.

Publisher Item Identifier S 0018-9545(97)01311-X.

is de-learned by a smaller learning rate. Once the centers of RBF neural networks are fixed, learning in the RBF connection weights is supervised. It is intrinsically easier since the network output is linear in the weights. The weights are typically trained using orthogonal least squares [7] and Givens least squares [8] algorithms. In this paper, we propose a new formulation for applying the fast-convergent recursive least squares (RLS) algorithm [14] to the weight-adjustment in the RBF networks in order to improve convergence performance of the above two least squares algorithms. The derivation of the hybrid RPCL and RLS algorithm is given in Section III. Section IV describes the procedure for the approximation of propagation loss model using the trained RBF neural network. Finally, Section V demonstrates the application of the proposed RBF neural network to the propagation loss prediction of selected field measurements.

II. DESCRIPTION OF MOBILE RADIO PROPAGATION LOSS MODEL IN BUILT-UP AREA

The propagation of radio waves in built-up areas has been found to be strongly influenced by the nature of the environment, in particular the size and density of buildings. Generally, a qualitative description of the environment is often employed using terms such as rural, urban, and suburban. Rural areas define open farmland with sparse buildings, woodland, and forests. Urban areas are generally defined as being dominated by tall buildings, office blocks, and other commercial premises while suburban areas comprise residential houses, parks, and gardens. Usually the propagation loss model is used to express the measured propagation loss as a function of the variables associated with the environment and terrain of the mobile, base station antenna height h_b , frequency f_c , mobile antenna height h_m , and the distance between transmitting and receiving antennas d . The degree of terrain undulation is given by a parameter known as intercede range Δh [15]. The value of Δh depends on the terrain topography. For example, the value of Δh falls within an interval (20 m, 40 m) for quasismooth terrain or (40 m, 80 m) for rolling terrain. Thus, a simple mathematical expression for the propagation loss, L_p , in a specific type of built-up environment and size of city is represented by

$$L_p = f(h_b, h_m, f_c, d, \Delta h). \quad (1)$$

Okumura *et al.* [1] published an empirical path loss prediction model based on field measurements taken in the Tokyo area. It provides an initial path loss estimate for the urban area with a quasismooth terrain ($\Delta h \simeq 20$ m). In addition, some correction factors must be used to adapt to the results in some other conditions, for example, the type of propagation environment and the size of city. However, Okumura's method cannot be easily automated, because it involves various curves. In an attempt to make Okumura's method easy to apply, Hata [2] established empirical mathematical relationships to describe the graphical information given by Okumura. Hata's formulation is limited to certain ranges of input parameters and is applicable only over quasismooth terrain. The mathe-

tical expressions for path loss L_p in dB and their range of applicability in urban areas are

$$L_p = 69.55 + 2.16 \log f_c - 13.82 \log h_b - a(h_m) + (44.9 - 6.55 \log h_b) \log d \text{ dB} \quad (2)$$

where

$$\begin{aligned} 150 &\leq f_c \leq 1500 \text{ MHz} \\ 30 &\leq h_b \leq 200 \text{ m} \\ 1 &\leq d \leq 20 \text{ km.} \end{aligned}$$

$a(h_m)$ is the correction factor for mobile antenna height and is computed as follows:

1) *Small- or Medium-Sized City:*

$$a(h_m) = (1.1 \log f_c - 0.7)h_m - (1.56 \log f_c - 0.8) \quad (3)$$

where $1 \leq h_m \leq 10m$.

2) *Large City:*

$$a(h_m) = \begin{cases} 8.29(\log 1.54h_m)^2 - 1.1 : & f_c \leq 200 \text{ MHz} \\ 3.2(\log 11.75h_m)^2 - 4.97 : & f_c \geq 400 \text{ MHz.} \end{cases} \quad (4)$$

3) *Suburban Areas:*

$$L_p = L_p(\text{urban}) - 2 \left[\log \left(\frac{f_c}{28} \right) \right]^2 - 5.4 \text{ dB.} \quad (5)$$

4) *Open Areas:*

$$L_p = L_p(\text{urban}) - 4.87(\log f_c)^2 - 18.33 \log f_c - 40.94 \text{ dB.} \quad (6)$$

These expressions have considerably enhanced the practical value of the Okumura's method, although Hata's formulations do not include any of the path-specific corrections available in the original method. However, Hata did not provide a systematic procedure to determine his empirical formula based on measurement data. In order to adapt to a different environment database, Hawthorne [4] applied Kalman filtering techniques to obtain the path loss prediction model for Florida area. More recently, Stocker *et al.* [6] proposed a novel adaptive learning algorithm based on a multilayer neural network to train their prediction model with measurement data from different environment databases. Their neural-based prediction model yields acceptable conformity with measurements. Although the multilayer neural network is a useful method for approximating the propagation loss, many researchers [5], [6] have shown, however, that it suffers from drawbacks of slow convergence and unpredictable solutions during learning. To overcome this difficulty, radial basis function neural networks that have a "linear in the parameters" representation are proposed to enhance the real-time learning capability and achieve the rapid convergence. More details about the RBFNN will be discussed in the next section.

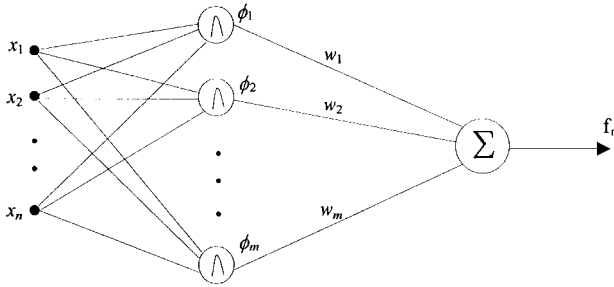


Fig. 1. Schematic diagram of RBF neural network.

III. MODELING NONLINEAR SYSTEMS USING RBF NEURAL NETWORKS

The RBF network depicted in Fig. 1 is a two-layer network whose output nodes from a linear combination of the activation (or basis) functions computed by hidden layer nodes [7]–[9]. Each node in the hidden layer contains a parameter vector called a center, and the node calculates the Euclidean distance between the center and the network input vector. The result is then passed through the activation function. These activation functions are radially symmetric and produce a localized response to input stimulus. That is, they produce a significant nonzero response only when the input falls within a small localized region of the input space. The second layer (output layer) is essentially a linear combiner with a set of connection weights. This output layer nodes form a weighted linear combination of the outputs from the hidden layer. Thus, the overall network performs a nonlinear transform from R^n to R by forming a linear combination of the nonlinear basis functions. The overall response of such a network is a mapping from $R^n \rightarrow R$, that is

$$f_r(\mathbf{x}; \mathbf{w}, \mathbf{C}) = \sum_{i=1}^m w_i \phi(\|\mathbf{x} - \mathbf{c}_i\|) \quad (7)$$

where $\mathbf{x} \in R^n$ is the network input vector, $\phi(\cdot)$ is a continuous activation function from R^+ to R^+ ; $\|\cdot\|$ denotes the Euclidean norm, $\mathbf{c}_i \in R^n$ and $1 \leq i \leq m$ are the RBF centers, w_i and $1 \leq i \leq m$ are the connection weights, m is the number of computing nodes in the hidden layer, $\mathbf{w} = [w_1, w_2, \dots, w_m]^T$ is a $m \times 1$ connection weight vector, and $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_m]$ is an $n \times m$ matrix of centers. Note that the RBF network can be extended to perform a nonlinear transformation from R^n to R^N , where $N > 1$ [17].

The aim in the present study is to use the RBF network response $g(\cdot)$ to approximate some arbitrary nonlinear real-valued continuous mapping $y = f(\cdot): R^n \rightarrow R$ by selecting appropriate centers and weights. In other words, $\hat{y}(t) = f_r[\mathbf{x}(t); \mathbf{w}, \mathbf{C}]$ acts as a one-step-ahead predictor for $y(t)$. For good modeling capability, the radial basis function, $\phi(\cdot)$, is typically chosen as the so-called thin-plate spline function [18]

$$\phi(\nu) = \nu^2 \log \nu. \quad (8)$$

Other choices of $\phi(\cdot)$ can also be employed. The well-known one is the Gaussian function. Theoretical investigations and practical results, however, seem to show that the type of nonlinearity $\phi(\cdot)$ is not crucial to the performance of RBF networks [18].

A. Functional Approximation Capability of the RBF Network

The RBF network can be used for functional approximation, just like the MLP. In theory, the RBF network is capable of forming an arbitrarily close approximation to any continuous nonlinear mapping [9], [10]. The primary difference between these two is in the nature of their basis functions. The hidden layer nodes in an MLP form sigmoidal basis functions, which are nonzero over an infinitely large region of the input space, while the basis functions in the RBF network cover only small, localized regions. The local tuning property of the RBF networks is considered to be particularly suited for fast learning [11].

It is essential to establish the approximation capabilities of the RBF networks to a nonlinear continuous function $y = f(\mathbf{x}): D \subseteq R^n \rightarrow R$ from input/output data pairs $\{\mathbf{x}, y\}$, where D is a compact set on R^n . Park and Sandberg [9] showed that for any $f \in C^2(D)$ with respect to \mathbf{x} , an $f_r(\mathbf{x}; \mathbf{w}, \mathbf{C})$ with a sufficient number of hidden nodes, appropriate weights, and centers can be found such that $\|f_r(\mathbf{x}; \mathbf{w}, \mathbf{C}) - f(\mathbf{x})\|_\infty = \sup_{\mathbf{x} \in D} \|f_r(\mathbf{x}; \mathbf{w}, \mathbf{C}) - f(\mathbf{x})\| < \epsilon$ for an arbitrary $\epsilon > 0$. Empirical results obtained in [16] have indicated that an RBF network with centers on a regular mesh can achieve ϵ when the number of hidden nodes grows exponentially with the dimension (n) of the input space for large n . Moreover, RBF networks also have the best approximation property [9], [10], i.e., there exists a set of weights, $\{w_i\}$, and a set of centers, $\{\mathbf{c}_i\}$, of the network such that the network output, $f_r(\mathbf{x}; \mathbf{w}, \mathbf{C})$, has the minimum distance from a given function $f(\mathbf{x})$.

B. Learning Algorithms for RBF Neural Networks

One of the major advantages of the RBF networks is that learning tends to be much faster than in the MLP [11]. The main reason for this is that the learning process is broken into two stages, and the algorithms used in both stages can be made relatively efficient. In the first stage, the positions of the m centers, initially chosen at random from the training set, are gradually adjusted using the unsupervised adaptive m -means clustering algorithm [11], [12]. In the second stage, the hidden-layer representation becomes the input to the second layer. The second layer is trained in supervised mode. Typically, its connection weights, w_{ij} , are derived by minimizing a sum of weighted error squares at the output of the network (over all training patterns) using the orthogonal least squares [7] or Given least squares [8] algorithm. However, here, we would like to give a new formulation for applying the fast-convergent RLS algorithm to the weight-adjustment in the RBF network.

Moody and Darken [11] first introduced the m -means clustering procedure as a good updating rule for the RBF centers. However, this clustering algorithm has a problem of selecting an appropriate m , the number of clusters. It has been shown that the m -means clustering algorithm significantly deteriorates its performance when the number of clusters is inappropriately chosen [13]. More recently, Xu *et al.* [13] proposed a new algorithm called rival penalized competitive learning (RPCL) to determine the RBF centers. The experimental results show that significant improvements have been

obtained by their RPCL. The basic idea of RPCL is that for each input, not only the winner center is modified to adapt to the input, but also its rival (the second winner) is de-learned by a smaller learning rate. RPCL can automatically allocate an appropriate number of clusters for an input data set. RPCL can be regarded as an unsupervised extension of Kohonen's LVQ_2 , which is a supervised vector quantization algorithm, and is closer in effect to Bayes decision theory.

For each input, the center vector \mathbf{c}_s of the RBF network that wins the competition is modified to adapt to the input with a learning rate α_s , but also the center vector \mathbf{c}_r of its rival is de-learned by a learning rate α_r , much smaller than that used by \mathbf{c}_s , i.e., $\alpha_s(k) \gg \alpha_r(k)$ for $k \geq 0$. The RPCL algorithm is summarized in the following.

Given initial centers $\mathbf{c}_i(0)$ and $1 \leq i \leq m$ and two initial learning rates $\alpha_s(0)$ and $\alpha_r(0)$, where $\alpha_s(0) \gg \alpha_r(0)$, at each time point k , the RPCL algorithm consists of the following computational steps:

1) Let an indicator, u_i be

$$u_i = \begin{cases} 1 & \text{if } i = s \text{ such that } \gamma_s \|\mathbf{x}(k) - \mathbf{c}_s(k-1)\|^2 \\ & = \min_{1 \leq j \leq m} \gamma_j \|\mathbf{x}(k) - \mathbf{c}_j(k-1)\|^2 \\ -1 & \text{if } i = r \text{ such that } \gamma_r \|\mathbf{x}(k) - \mathbf{c}_r(k-1)\|^2 \\ & = \min_{j \neq s} \gamma_j \|\mathbf{x}(k) - \mathbf{c}_j(k-1)\|^2 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where $\gamma_j = n_j / \sum_{i=1}^m n_i$ and n_i is the cumulative number of occurrences of $u_i = 1$.

2) Update the center vector by

$$\Delta \mathbf{c}_i(k) = \begin{cases} \alpha_s(k) [\mathbf{x}(k) - \mathbf{c}_i(k-1)] & \text{if } u_i = 1 \\ -\alpha_r(k) [\mathbf{x}(k) - \mathbf{c}_i(k-1)] & \text{if } u_i = -1 \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

The initial centers are often chosen randomly. Two learning rates α_s and α_r with $0 \leq \alpha_s, \alpha_r \leq 1$ should slowly decrease to zero. In the present application, both $\alpha_s(k)$ and $\alpha_r(k)$ are computed according to the same rule

$$\alpha_s(k) = \frac{\alpha_s(k-1)}{1 + \text{int} \left[\frac{k}{m} \right]^{1/2}} \quad (11)$$

$$\alpha_r(k) = \frac{\alpha_r(k-1)}{1 + \text{int} \left[\frac{k}{m} \right]^{1/2}}$$

where $\text{int}[\cdot]$ denotes the integer part of the argument. In (10), $\alpha_s(k) \gg \alpha_r(k)$, for $k \geq 1$ is always true when $\alpha_s(0) \gg \alpha_r(0)$.

The RPCL clustering is based on a linear learning rule, thus guaranteeing rapid convergence. It is also an unsupervised procedure using only the network input data. No desired response is required and the procedure will not be affected by the learning rule used for weights.

In this paper, we use the well-known RLS algorithm for adjusting the connection weights of the second layer. The learning is based on a sequence of random input/output pairs $\{\mathbf{x}(k), y(k) = f[\mathbf{x}(k)] + v(k)\}$, where $f(\mathbf{x})$ is the function to be approximated and $\mathbf{x}(k)$, $y(k)$, and $v(k)$ are the samples of

\mathbf{x} , y , and observation noise at time instant k , respectively. So, at each time instant $k = 1, 2, \dots$, determine the appropriate RBF network, $g_k(\mathbf{x}) = f_r(\mathbf{x}; \mathbf{w}, \mathbf{C}) = \mathbf{w}^T(k) \mathbf{u}[\mathbf{x}(k)]$ such that

$$J(k) = \sum_{i=0}^k \lambda^{k-i} \{y(i) - g_k[\mathbf{x}(i)]\}^2 \quad (12)$$

is minimized, where $\mathbf{u}(\mathbf{x}) = [\phi(\|\mathbf{x} - \mathbf{c}_1\|), \phi(\|\mathbf{x} - \mathbf{c}_2\|), \dots, \phi(\|\mathbf{x} - \mathbf{c}_m\|)]^T$, the appropriate centers $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_m$ are determined by the above-mentioned clustering method, $\mathbf{w}(k) = [w_1(k), w_2(k), \dots, w_m(k)]^T$, $w_i(k)$, $1 \leq i \leq m$ are the connection weights estimated from k input/output pairs, and $\lambda \in (0, 1]$ is a forgetting factor.

Since $g_k(\mathbf{x}) = \mathbf{u}^T \mathbf{w}$ is linear in the parameter vector \mathbf{w} , we can therefore use the following fast-convergent RLS algorithm to update \mathbf{w} . At the k th time instant, let us define the input vector $\mathbf{u}(k) = \mathbf{u}[\mathbf{x}(k)] = [\phi(\|\mathbf{x}(k) - \mathbf{c}_1\|), \phi(\|\mathbf{x}(k) - \mathbf{c}_2\|), \dots, \phi(\|\mathbf{x}(k) - \mathbf{c}_m\|)]^T$, the desired output is $y(k)$, and the estimated weight vector $\hat{\mathbf{w}}(k) = [\hat{w}_1(k), \hat{w}_2(k), \dots, \hat{w}_m(k)]^T$. The on-line RLS estimation of an optimal weight vector \mathbf{w}^* is then recursively expressed by

$$\Psi(k) = \frac{\lambda^{-1} \mathbf{P}(k-1) \mathbf{u}(k)}{1 + \lambda^{-1} \mathbf{u}^T(k) \mathbf{P}(k-1) \mathbf{u}(k)} \quad (13)$$

$$\hat{\mathbf{w}}(k) = \hat{\mathbf{w}}(k-1) + \Psi(k) [y(k) - \hat{\mathbf{w}}^T(k-1) \mathbf{u}(k)] \quad (14)$$

$$\mathbf{P}(k) = \lambda^{-1} \mathbf{P}(k-1) - \lambda^{-1} \Psi(k) \mathbf{u}^T(k) \mathbf{P}(k-1) \quad (15)$$

where $\mathbf{P}(k)$ denotes the inverse of the input correlation matrix and $\Psi(k)$ represents the Kalman gain vector. For the initial condition at $k = 0$, we have $\hat{\mathbf{w}}(0) = \mathbf{0}$ and $\mathbf{P}(0) = \delta^{-1} \mathbf{I}$, where δ is a small positive constant and \mathbf{I} represents an identity matrix. While reaching a optimal weight vector \mathbf{w}^* , this would lead to a RLS-based RBF network prediction of an unknown function $f(\mathbf{x})$.

IV. PROPAGATION LOSS MODEL IDENTIFICATION BY RBF NEURAL NETWORKS

In general, model identification is usually recognized as a process to train RBF neural networks $f_r(\mathbf{x}; \mathbf{w}, \mathbf{C})$ to represent nonlinear systems $f(\mathbf{x})$. This would be distinctly helpful in achieving the accurate propagation loss prediction for a specified mobile environment.

From (1), the system plant dynamics for the propagation loss can be modeled as follows:

$$L_p = f(\mathbf{x}) = f_r(\mathbf{x}; \mathbf{w}, \mathbf{C}) + e \quad (16)$$

and the measurement equations are given by

$$y = L_p + n \quad (17)$$

where $\mathbf{x} = [h_b, h_m, f_c, d, \Delta h]^T$, n denotes the measurement noise and e represents the unmodeled system error.

The basic configuration for achieving the propagation loss prediction is shown schematically in Fig. 2. A RBF neural network with a single hidden layer is placed in parallel with the propagation loss system and receives the same input \mathbf{x} as the system. The system output provides the measured propagation loss, y , during training. The purpose of the identification is

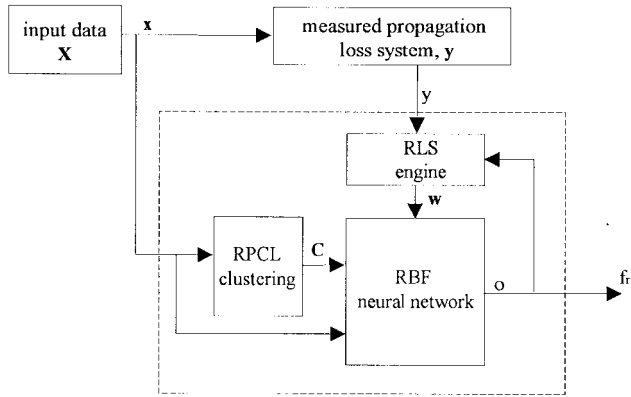


Fig. 2. Schematic configuration of RBF neural network for propagation loss prediction.

to find the appropriate centers and weights, i.e., c_i 's and w_i 's with response o that matches the response of the system for a given set of inputs x . During the identification, the norm of the error, $\|e + n\| = \|y - o\|$, is minimized through a number of center and weight adjustments by RPCL and RLS algorithms, respectively. Fig. 2 shows the case for which the network attempts to model the mapping of propagation loss system input to output with both input and output measured at the same time. As mentioned in Section III, the RBF neural network will be identical to the system in the domain of interest when the number of hidden units is sufficiently large.

V. SIMULATION RESULTS

In order to compare our method with the well-known Hata formula, the RBF-based propagation loss prediction model is trained by Okumura's field measurements taken in the Tokyo area. Since the terrain of Okumura's method is assumed to be quasismooth ($\Delta h \approx 20$ m), the propagation loss L_p of (1) becomes a function of four variables and is given by

$$L_p = f(h_b, h_m, f_c, d). \tag{18}$$

This implies that the RBF neural network used to approximate the unknown propagation loss function of (18) has four inputs; $n = 4$. For training and test purposes, a set of 292 measurement points taken in the large city was divided into 231 training patterns and 61 test patterns. The parameters for the RPCL algorithm are chosen as $\alpha_s(0) = 0.75$, $\alpha_r(0) = 0.08$, and $c_0(0) = [2, 2, 1.5]^T$. For the RLS algorithm, λ and δ are chosen as one and 10^5 , respectively. In our simulation, the number of hidden nodes is chosen as 30. Note that $m (= 30)$ initial centers are randomly chosen from the domain of interest. By inputting 231 training patterns into the RBF neural network and then performing both RPCL and RLS algorithms, the training mean-squared error can be found as 2.4 dB. Table I shows the training mean-squared errors for various numbers of hidden nodes. The training mean-squared error decreases when the number of hidden nodes increases. For Hata's formula, its mean-squared error for these 231 data is 5.84 dB.

Next, we would like to verify the approximation capability of the trained RBF-based propagation prediction model by a set of 61 test patterns. A comparison of curves shown in

TABLE I
TRAINING MEAN-SQUARED ERROR (σ_e) VERSUS NUMBER OF HIDDEN NODES (m)

m	30	40	50	80	Hata
σ_e	2.4 dB	2.3 dB	1.72 dB	1.65 dB	5.84 dB

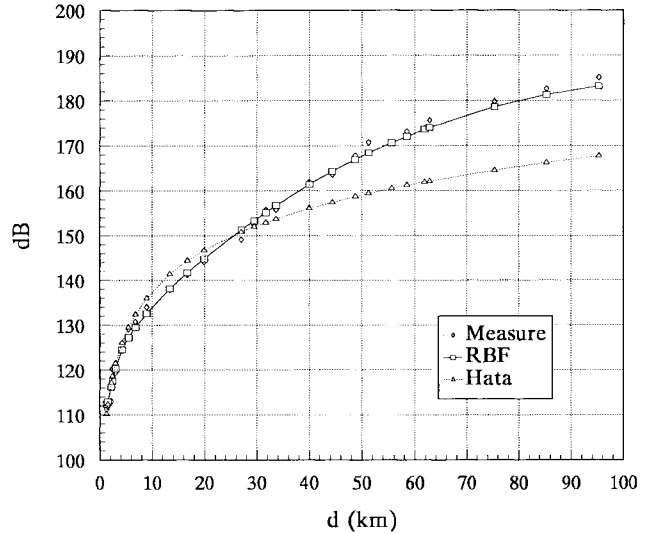


Fig. 3. Comparison of propagation losses achieved by Okumura's measurement, RBF neural network, and Hata's empirical model versus path profile when $f_c = 453$ MHz, $h_b = 140$ m, and $h_m = 3$ m.

Fig. 3 shows that the RBF-based propagation loss prediction curve along the path profile ($0 < d < 70$ km) is much closer to the measurement curve achieved by 26 test patterns than Hata's prediction curve when f_c , h_b , h_m are set to be 453 MHz, 140 m, and 3 m, respectively. The deviations between the Hata's approach and the measurement curve, in particular, become extremely large when $d \geq 40$ km. However, the RBF approach provides a uniform approximation to the propagation loss over a wider range of path profile. Figs. 4 and 5 show that this argument is also true for various values of the parameters f_c , h_b , and h_m . The mean-squared error achieved by the RBF-based prediction model over the 61 test patterns is 2.35 dB. For the same 61 test patterns, the mean-squared error for Hata's model is 7.5 dB.

VI. CONCLUSION

This paper has presented a new propagation prediction model based on a two-layer RBF neural network that is capable of predicting the field strength by using a hybrid RPCL and RLS algorithm. In particular, the advantage of this approach is that a particular propagation prediction model can be constructed to take account of various types of environments based on measurement data taken in the desired environment. This approach enhances the flexibility of the RBF-based prediction model to adapt to the terrain database of the environment. Simulation results have shown that the RBF approach provides more accurate predictions of field strength loss than that of Hata's model. This verifies the effectiveness of the best approximation capability of the RBF neural network.

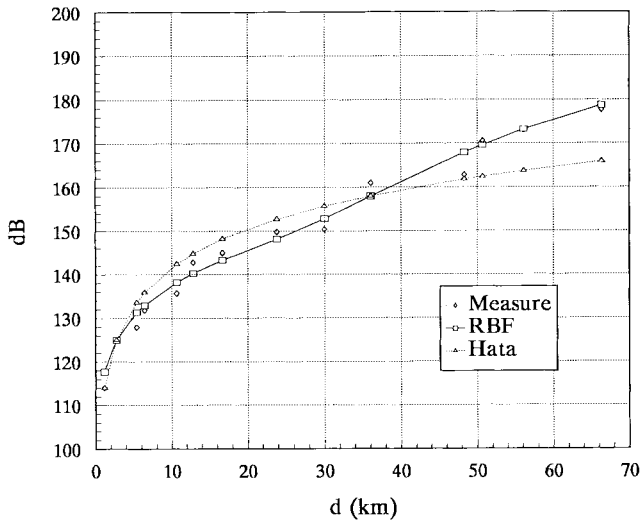


Fig. 4. Comparison of propagation losses achieved by Okumura's measurement, RBF neural network, and Hata's empirical model versus path profile when $f_c = 922$ MHz, $h_b = 220$ m, and $h_m = 3$ m.

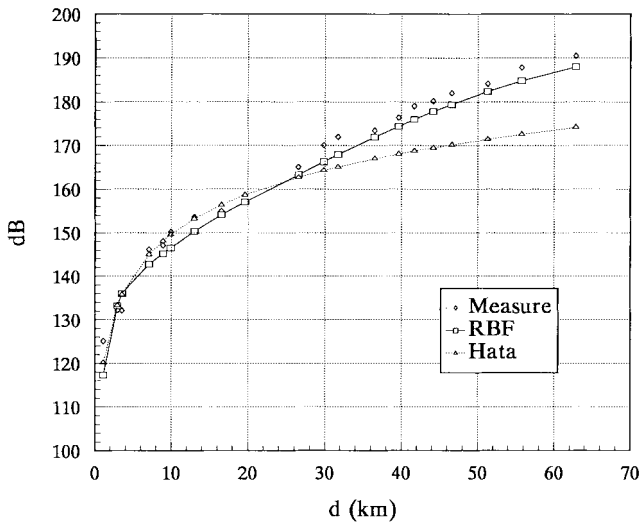


Fig. 5. Comparison of propagation losses achieved by Okumura's measurement, RBF neural network, and Hata's empirical model versus path profile when $f_c = 1.318$ GHz, $h_b = 140$ m, and $h_m = 3$ m.

REFERENCES

- [1] Y. Okumura *et al.*, "Field strength and its variability in VHF and UHF land-mobile radio service," *Rev. Elec. Common. Lab.*, vol. 16, 1968.
- [2] M. Hata, "Empirical formula for propagation loss in land mobile radio services," *IEEE Trans. Veh. Technol.*, vol. VT-29, no. 3, pp. 317-325, Aug. 1980.
- [3] J. B. Keller, "Geometrical theory of diffraction," *J. Opt. Soc. Am.*, vol. 52, pp. 116-130, 1962.

- [4] L. M. Hawthorne, "800 MHz mobile radio propagation prediction using Kalman filtering techniques," *IEEE Trans. Veh. Technol.*, vol. 38, pp. 55-68, May 1989.
- [5] D. E. Rumelhart *et al.*, "Learning internal representation by error propagation," in *Parallel Distributed Processing: Explorations in Microstructure of Cognition*. Cambridge MA: MIT Press, 1986, vol. 1, ch. 8.
- [6] K. E. Stocker *et al.*, "Neural network approach to prediction of terrestrial wave propagation for mobile radio," *IEE Proc. H*, vol. 140, no. 4, pp. 315-320, Aug. 1993.
- [7] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Networks*, vol. 2, no. 2, pp. 302-309, 1991.
- [8] S. Chen, S. A. Billings, and P. M. Grant, "Recursive hybrid algorithm for nonlinear system identification using radial basis function networks," *Int. J. Cont.*, vol. 55, no. 5, pp. 1051-1070, 1992.
- [9] J. Park and I. W. Sandberg, "Universal approximation using radial basis function networks," *Neural Computation*, vol. 3, pp. 246-257, 1991.
- [10] F. Girosi and T. Poggio, "Networks and the best approximation property," *Biological Cyber.*, vol. 63, pp. 169-176, 1990.
- [11] J. Moody and C. Darken, "Fast-learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, pp. 281-294, 1989.
- [12] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
- [13] L. Xu, A. Krzyzak, and E. Oja, "Rival penalized competitive learning for clustering analysis, RBF net, and curve detection," *IEEE Trans. Neural Networks*, vol. 4, pp. 636-649, July 1993.
- [14] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1991.
- [15] D. Parsons, *The Mobile Radio Propagation Channel*. New York: Wiley, 1992.
- [16] J. A. Leonard, M. A. Kramer, and L. M. Ungar, "Using radial basis functions to approximate a function and its error bounds," *IEEE Trans. Neural Networks*, vol. 3, pp. 624-627, July 1992.
- [17] S. V. T. Elanayar and Y. C. Shin, "Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic systems," *IEEE Trans. Neural Networks*, vol. 5, pp. 594-603, July 1994.
- [18] M. J. D. Powell, "Radial basis function approximations to polynomials," in *Proc. 12th Biennial Num. Anal. Conf.*, Dundee, 1987, pp. 223-241.

Po-Rong Chang (M'87), for a photograph and biography, see this issue, p. 95.



Wen-Hao Yang was born in Chiayi, Taiwan, R.O.C. He received, respectively, the M.S. degree in communication engineering from National Chiao-Tung University, Hsinchu, Taiwan, in 1988, and the Ph.D. degree in electronic engineering from the same university in 1994.

From 1984 to 1986, he fulfilled his military duty by serving in the Chinese Air Force. In 1988, he joined the Telecommunication Laboratories, Ministry of Transportation and Communications, R.O.C. Currently, he is an Associate Research Engineer and a Project Leader in charge of PCS system planning. His current research interests are in the area of personal communications services, radio propagation, and adaptive array.