

## An Adaptive Neural Network Scheme for Radar Rainfall Estimation from WSR-88D Observations

HONGPING LIU, V. CHANDRASEKAR, AND GANG XU

*Colorado State University, Fort Collins, Colorado*

(Manuscript received 22 July 1999, in final form 28 February 2001)

### ABSTRACT

Recent research has shown that neural network techniques can be used successfully for ground rainfall estimation from radar measurements. The neural network is a nonparametric method for representing the relationship between radar measurements and rainfall rate. The relationship is derived directly from a dataset consisting of radar measurements and rain gauge measurements. The effectiveness of the rainfall estimation by using neural networks can be influenced by many factors such as the representativeness and sufficiency of the training dataset, the generalization capability of the network to new data, season change, location change, and so on. In this paper, a novel scheme of adaptively updating the structure and parameters of the neural network for rainfall estimation is presented. This adaptive neural network scheme enables the network to account for any variability in the relationship between radar measurements and precipitation estimation and also to incorporate new information to the network without retraining the complete network from the beginning. This precipitation estimation scheme is a good compromise between the competing demands of accuracy and generalization. Data collected by a Weather Surveillance Radar—1988 Doppler (WSR-88D) and a rain gauge network were used to evaluate the performance of the adaptive network for rainfall estimation. It is shown that the adaptive network can estimate rainfall fairly accurately. The implementation of the adaptive network is very efficient and convenient for real-time rainfall estimation to be used with WSR-88D.

### 1. Introduction

Radar is a useful remote sensing tool for precipitation estimation on the ground. The development of algorithms for the remote estimation of precipitation based on radar measurements has been an active research topic for many years. The problem of rainfall estimation on the ground based on radar measurements is complicated because of the space–time variability of the rainfall field. The rainfall rate  $R$  obtained on the ground can be potentially dependent on the four-dimensional structure of precipitation aloft (three spatial dimensions and time). In principle, one can obtain a functional approximation between the rainfall on the ground and the 4D radar reflectivity observations  $Z$  aloft. This function will be more complicated than a simple  $Z$ – $R$  algorithm or a multiparameter radar rainfall algorithm. Therefore the ground rainfall estimation can be viewed as a complex function approximation problem.

Neural networks are well suited for this problem, and the theoretical basis is provided by the universal function approximation theorem (Funahashi 1989). Recent research has shown that neural network techniques can be used successfully for ground rainfall estimation from

radars (Xiao and Chandrasekar 1995, 1997) and other such applications (Krasnopolsky et al. 1995). This technique includes two stages, namely, 1) the training and validation stage and 2) the application stage. In the training stage, the neural network learns the potential relationship between the rainfall rate and the radar measurements from a training dataset. When a radar measurement set is applied to the neural network, the network yields a rainfall-rate estimate as output. This output is compared with the rain gauge measurement, and their difference or the error is propagated back to adjust the parameters of the network. This learning process is continued until the network converges. Once the training process is complete, a relationship between the rainfall rate and the radar measurements is established and the network is ready for operation. When a radar measurement vector subsequently is applied to the network, it yields a rainfall-rate estimate.

Neural networks have many advantages in the context of rainfall estimation from radar measurements. The relationship between radar measurements and rainfall rate on the ground is derived directly from a training dataset, and therefore it is not influenced by systematic variations in the radar system characteristics. The neural network can be tuned very well for one specific kind of storm or for several storms. Once the neural network is trained, it represents a relation between radar measure-

---

*Corresponding author address:* V. Chandrasekar, Colorado State University, Fort Collins, CO 80523-1373.  
E-mail: chandra@engr.colostate.edu

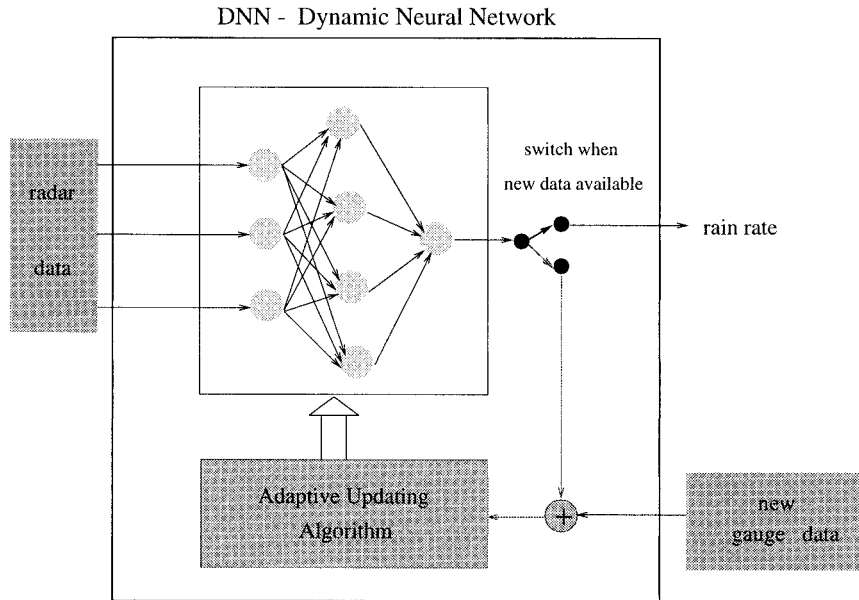


FIG. 1. Structure of the adaptive neural network for rainfall estimation. The radar data block indicates input, and gauge data are used as the target output for the neural network. Once trained, the neural network estimates rainfall based on radar data. When new data are available, the network switches to an updating mode.

ments and rainfall rate. If the training dataset is large enough and representative enough, the neural network can perform very well.

There is a common limitation with respect to the trade-off between generalization and accuracy of neural network-based rainfall estimation. Therefore, developing a flexible network instead of a fixed network for rainfall estimation may be better. A neural network can learn its structure and parameters automatically from the training dataset. One way to solve the problem is to collect new data and retrain the neural network all over again from the beginning. However, this training process is very tedious and time consuming, and to restart the training every time new data are available is not a practical solution. The goal of this paper is to develop an adaptive neural network that is easy to train and can continuously update the structure by incorporating the latest information into an existing neural network without having to retrain from the beginning. Therefore, the network has the "dynamic" characteristic, and it can fine-tune the functional mapping over time.

The main feature of the adaptive neural network is that the network can adjust itself whenever new rain gauge data are available (as shown in Fig. 1). To start with, the network can be built by initial training using all the available data. The network is in the application mode after the initial training. Once new rain gauge data are collected, the network switches into an updating mode. By using an adaptive updating algorithm, the network adjusts some of its parameters, adding or removing some neurons so as to fine-tune its structure

with the new information. The scheme not only provides a fast and efficient way to build a new neural network rainfall estimation model but also can provide a way to maintain an existing neural network rainfall estimation model and make it evolve gradually.

In this paper, we have developed an adaptive neural network scheme that can be modified continuously. For this purpose, a radial basis function (RBF) neural network is chosen because its architecture is suited well for adaptive modification. In the next section, the development of the adaptive neural network for rainfall estimation is presented. The performance of the adaptive neural network is evaluated in section 3. The important conclusions are summarized in section 4.

## 2. Adaptive neural network algorithm

Multilayer feedforward neural networks (MLFNN) can be used successfully for radar rainfall estimation from remotely sensed data (Xiao and Chandrasekar 1995; Tsintikidis et al. 1996). Xiao and Chandrasekar (1997) showed that a back-propagation neural network (BPN), which is a class of the MLFNN, can be used for radar rainfall estimation. Some of the disadvantages of BPN are that the training process is computationally demanding and the learning process is very tedious. However, once trained, BPN can be used successfully for radar rainfall estimation. The structure and learning algorithm of a BPN make it difficult for implementing adaptive rainfall estimation algorithms. One of the alternatives suited for rainfall estimation is an RBF neural network. The RBF network has a unique structure that

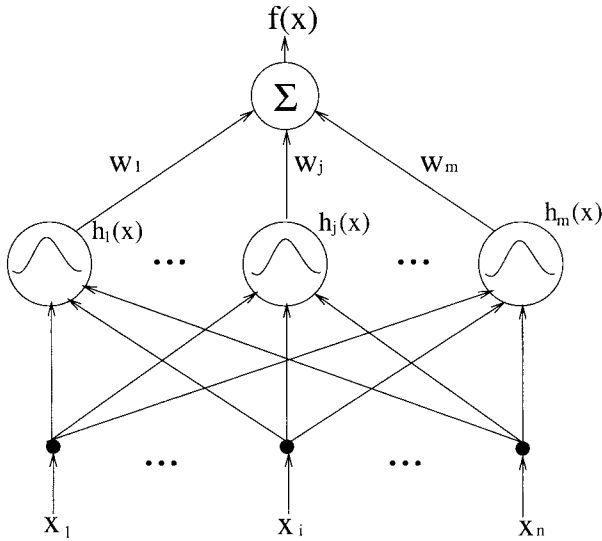


FIG. 2. Architecture of a typical RBF neural network.

will make it conducive for adaptive radar rainfall estimation. The following section describes the structure of an RBF network for radar rainfall estimation.

*a. RBF network architecture and parameters*

An RBF network has three layers: 1) an input layer consisting of input variables  $x_1, x_2, \dots, x_n$ ; 2) a hidden layer, consisting of neurons with RBF as transfer func-

tion  $h_j(\mathbf{x})$ ; and 3) an output layer, which consists of linear combinations of the hidden-layer output. The block diagram of the RBF neural network is shown in Fig. 2.

The Gaussian RBF is used in this study, which can be expressed as

$$h_j(\mathbf{x}) = \exp\left[-\sum_{i=1}^n \frac{(x_i - c_{ij})^2}{r_{ij}^2}\right], \quad (1)$$

and the output  $f(\mathbf{x})$  for an input vector  $\mathbf{x}$  is given by

$$f(\mathbf{x}) = \sum_{j=1}^m w_j h_j(\mathbf{x}), \quad (2)$$

where  $\mathbf{x} = [x_1, x_2, \dots, x_n]$  is the input vector,  $\mathbf{c}_j = [c_{1j}, c_{2j}, \dots, c_{nj}]$  is the center vector of neuron  $j$ ,  $\mathbf{r}_j = [r_{1j}, r_{2j}, \dots, r_{nj}]$  is the size vector of neuron  $j$ ,  $m$  is the number of neurons in the hidden layer, and  $w_j$  is the weight from neuron  $j$  to the output.

For the problem of rainfall estimation at a point on the ground, the input to the network can be chosen from available radar measurements over the three-dimensional space aloft. For Weather Surveillance Radar—1988 Doppler (WSR-88D) data, reflectivity factor  $Z_H$  is used for developing rainfall products. The reflectivity at 1-km height ( $Z_{H1}, \dots, Z_{H9}$ ) is used as inputs, where the spatial locations of these reflectivities are shown in Fig. 3. The input vector size depends on the input data used (nine for the structure shown in Fig. 3). The rainfall rate is chosen as the output of the network. The hidden

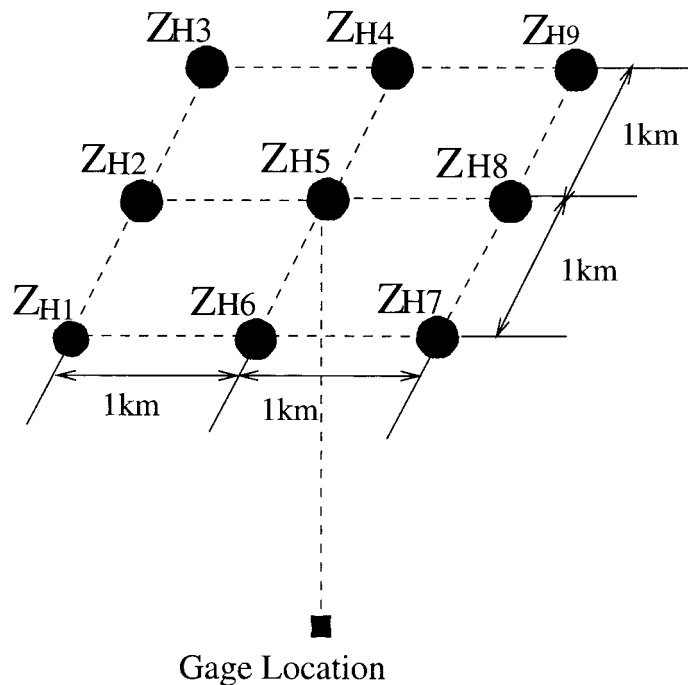


FIG. 3. The location of inputs to the RBF network.  $Z_{H1}, \dots, Z_{H9}$  are the nine reflectivity inputs to the neural network. The gage location is at the center of the grid.

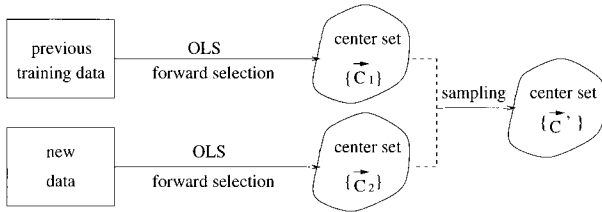


FIG. 4. A new center subset  $\{C_2\}$  is obtained by using ordinary least squares forward selection from the new available data. From  $C_2$  and center set  $\{C_1\}$  from the existing network, a new center vector set  $\{C\}$  is selected.

layer is an important part of the network, which determines the accuracy of the network. From Eq. (2) it can be seen that the RBF network performs a linear superposition of the localized basis function  $[h_1(x), \dots, h_m(x)]$ , where the accuracy of the output depends on the number of the basis functions and the centers and the widths of the basis functions.

For this RBF neural network, the following three parameters need to be determined (Mark 1996): 1) center vector of all the neurons in the hidden layer  $c_j$ ; 2) size vector of all the neurons in the hidden layer  $r_j$ , and 3) weights from the hidden layer to the output ( $w_1, \dots, w_m$ ). Once all these parameters are determined, the network can be used for applications. If an input vector  $x$  is applied to the RBF network, the distance of the vector to every center vector of the neurons in the hidden layer is calculated. The output of the neuron is a function of

the distance [as shown in Eq. (1)]: it is 1 (maximum) when the input vector  $x$  is equal to the center vector and then, as the distance increases, the output decreases. A linear combination of the outputs from all the hidden units is the final output.

*b. Development of an RBF network*

It was shown in the previous section that three sets of parameters need to be determined when constructing an RBF neural network. The orthogonal least squares method is used to determine the center vectors for the hidden neurons. Once the parameters in the hidden layer are determined, the weight vector from the hidden layer to the output layer can be obtained by the linear least squares method. This combined learning algorithm is fast, because no back propagation is involved in the process.

The size vectors  $r_j$  must be determined in conjunction with the center vectors  $c_j$ . The generalization capability of the RBF neural network is sensitive to the size vector. If the size vector is small, the network will function very well with the training set but will have poor generalization capability. On the other hand, if it is too large, then the network will be overgeneralized. Therefore, an appropriate size vector should be determined by several trials.

A subset ( $i = 1, p$ ) of training input ( $x$ ) to the network is chosen as the center of the RBFs. Then the training starts with an empty subset and adds one basis function

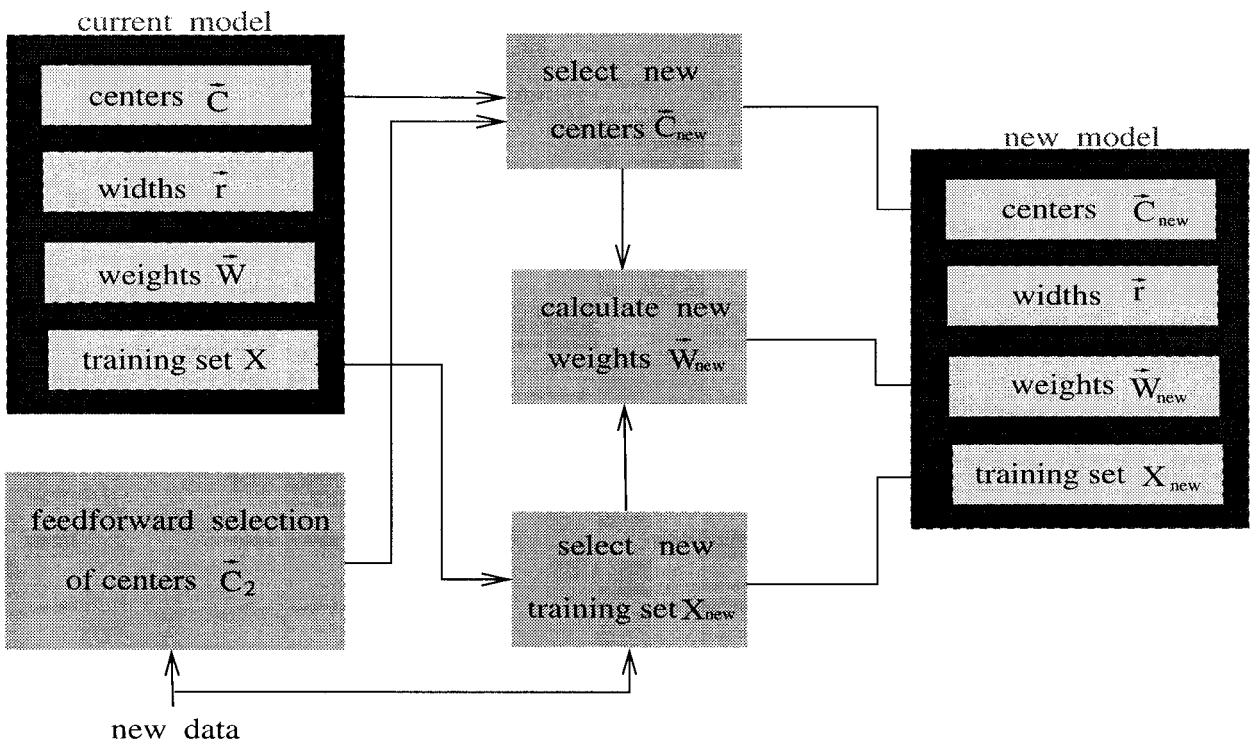


FIG. 5. Scheme of adaptive RBF neural network for rainfall estimation.

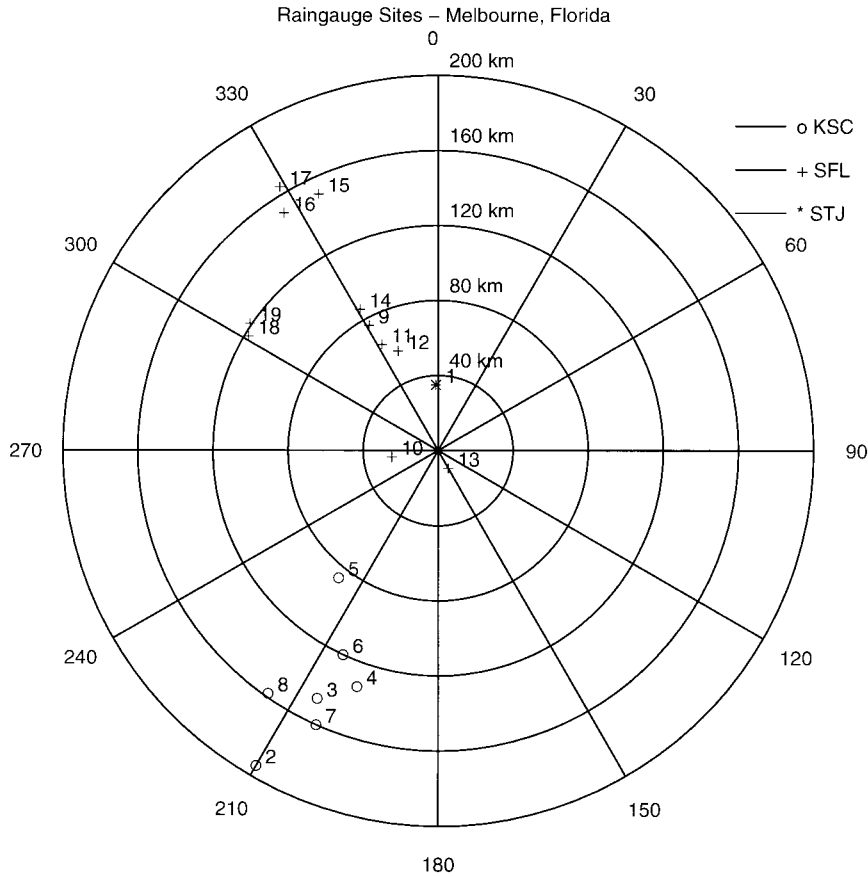


FIG. 6. The locations of the rain gauges on a radar plan position indicator (PPI) overlay. The circles are range rings of 40-km intervals, and the radials indicate azimuth angle in degrees. The symbols  $\circ$ ,  $+$ , and  $*$  indicate Kennedy Space Center (KSC), South Florida Water Management District (SFL), and St. Johns Water Management District (STJ) gauges, respectively.

at a time. The sum of squared error  $S$  is used to determine convergence according to the least squares algorithm,

$$S = \sum_{i=1}^p [\hat{y}_i - f(\mathbf{x}_i)]^2, \tag{3}$$

where  $\hat{y}$  is the target output or rainfall estimate. Lowest prediction error is the convergence criterion that is used to determine if any additional RBFs are needed. The network has the lowest prediction error when the optimum subset of RBFs is chosen. Standard measures, such as final prediction error, can be used to compute prediction error. When one of these measures stops decreasing, then no more RBFs should be added to the hidden layer.

If the centers and sizes of the RBFs are fixed, then the determination of weights  $w_j$  is straightforward. The  $w_j$ s are determined by minimizing the sum of squared error  $S$  given by

$$S = \sum_{i=1}^p \left[ \hat{y}_i - \sum_{j=1}^m w_j h_j(\mathbf{x}) \right]^2. \tag{4}$$

The optimum  $w_j$  is given by the generalized inverse equation,

$$\hat{\mathbf{w}} = [\hat{w}_1 \hat{w}_2 \cdots \hat{w}_m]^T = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{Y}, \tag{5}$$

where  $\mathbf{H}$  is the matrix of basis functions given by

$$\mathbf{H} = \begin{bmatrix} h_1(\mathbf{x}_1) & h_2(\mathbf{x}_1) & \cdots & h_m(\mathbf{x}_1) \\ h_1(\mathbf{x}_2) & h_2(\mathbf{x}_2) & \cdots & h_m(\mathbf{x}_2) \\ \vdots & \vdots & \cdots & \vdots \\ h_1(\mathbf{x}_p) & h_2(\mathbf{x}_p) & \cdots & h_m(\mathbf{x}_p) \end{bmatrix}, \tag{6}$$

and  $\mathbf{Y}$  is the output vector (of rainfall observations).

*c. Dynamic updating scheme for the RBF network*

We can use the procedure introduced above to develop an RBF neural network for rainfall estimation. As days go by, more data become available. Some of the data are completely new to the network, and some are similar to what the network has seen before but with slightly different output. To incorporate the information from the new data, it is necessary to refine the network

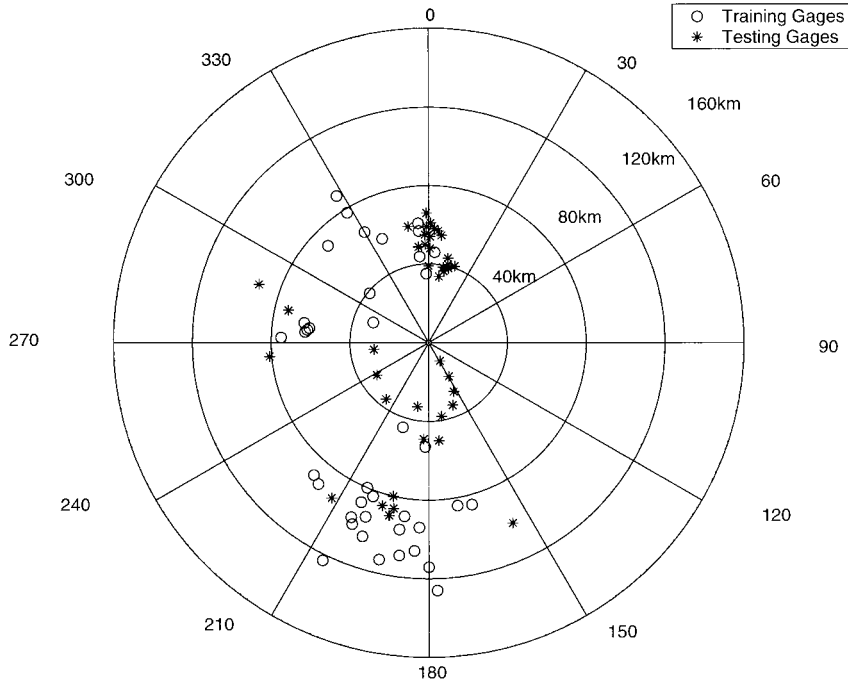


FIG. 7. The locations of the rain gauges on a radar PPI overlay. The circles are range rings of 40-km intervals, and the radials indicate azimuth angle in degrees. The symbols  $\circ$  and  $*$  indicate training gauges and testing gauges, respectively.

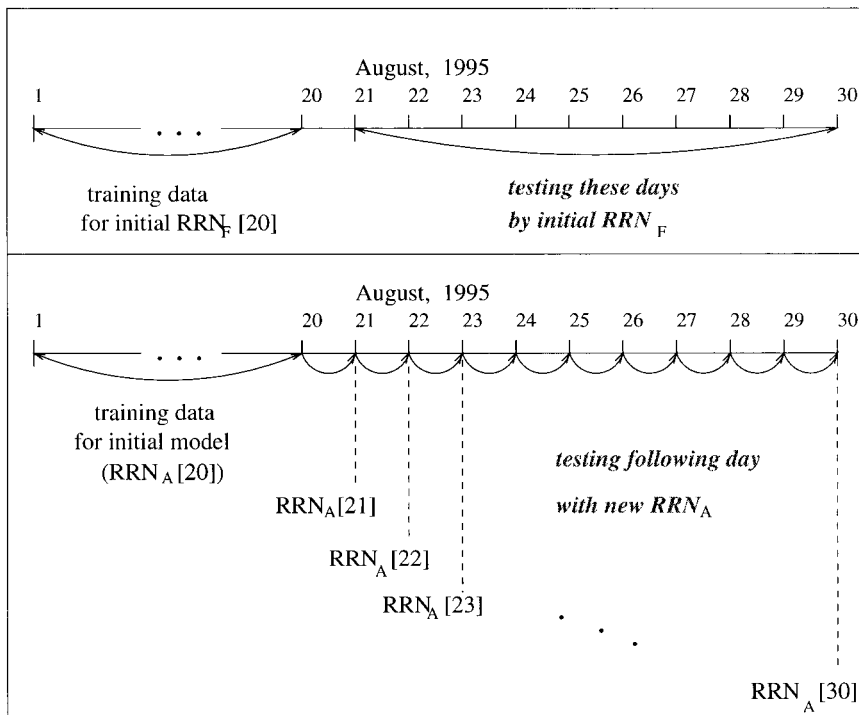


FIG. 8. The comparison scheme between a fixed network and an adaptive neural network for rainfall estimation for a 10-day period (21–30 Aug 1995).

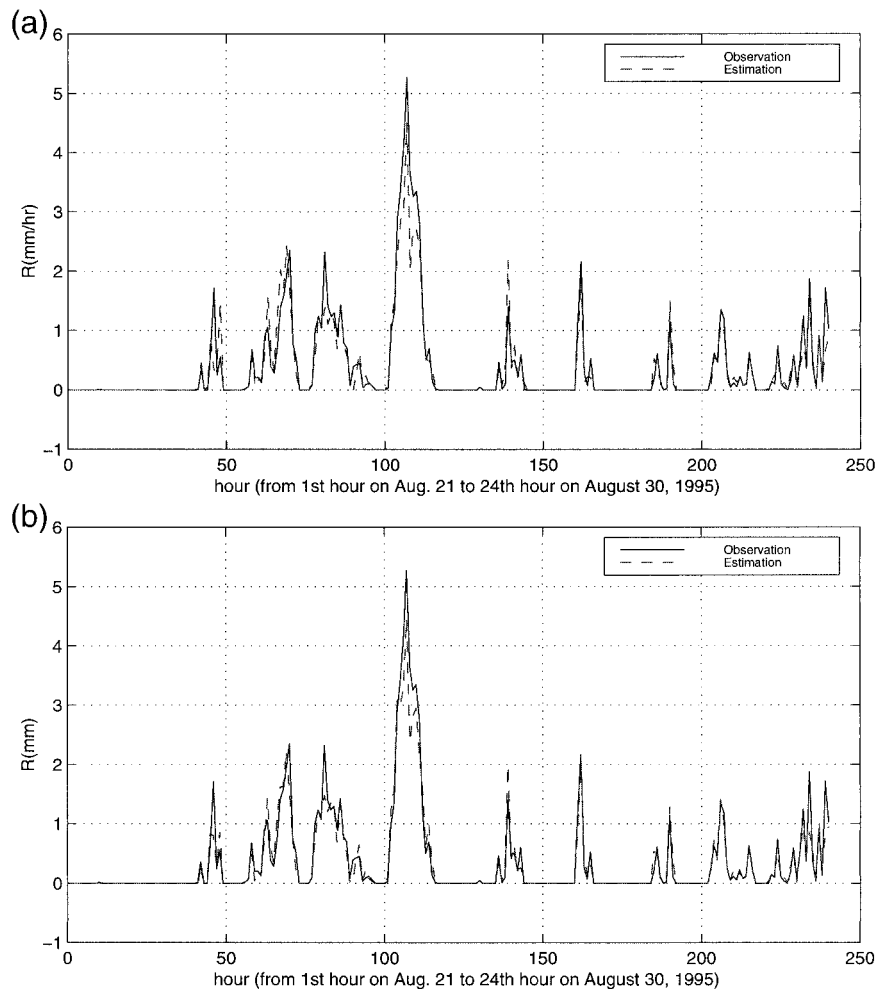


FIG. 9. Mean hourly rainfall accumulation results from (a) a fixed neural network and (b) an adaptive neural network (for 21–30 Aug 1995).

by adding or removing some neurons in the hidden layer. For data similar to what the network has seen but with different desired output as compared with what the network saw before, it is necessary to adjust the weights from the hidden layer to the output layer with the latest input–output pairs, maintaining the structure of the network.

One way to incorporate the new information from the new data into the network is simply by combining the new data with the old training dataset to form a new larger training dataset and to retrain from the beginning. The most important part in the retraining process is searching for the optimum center set from the new training dataset, and this process is tedious. This solution is neither convenient nor practical. Another disadvantage is that a simple retraining process may not give preference to the latest data in the training process, which is required if the most current relation between reflectivity structure and rainfall is to be maintained by the neural network. Based on these reasons, it may be better

to use an adaptive RBF neural network for rainfall estimation.

In the adaptive learning scheme illustrated in Fig. 4, the new network is based on the existing network; however, it is modified according to the new data. One of the simplest ways to modify the network for new data is to add or replace neurons as well as change the center vector. The schematic of altering the center vectors is shown in Fig. 4. The procedure used to modify the RBF network is as follows. First, the standard orthogonal least squares method that was used to build the network from the beginning can be applied to the new dataset to come up with new center vector  $C_2$ . The existing model has a center vector set (called center vector set 1,  $C_1$ ). A new center vector set can be constructed from these two center vector sets  $C_1$  and  $C_2$ .

For  $C_i \in C_1$  and  $C_j \in C_2$ ,

$$\text{if } \|C_i - C_j\| \leq T, \text{ then remove } C_i, \quad (7)$$

where  $\| \cdot \|$  indicates the Euclidean norm and  $T$  is thresh-

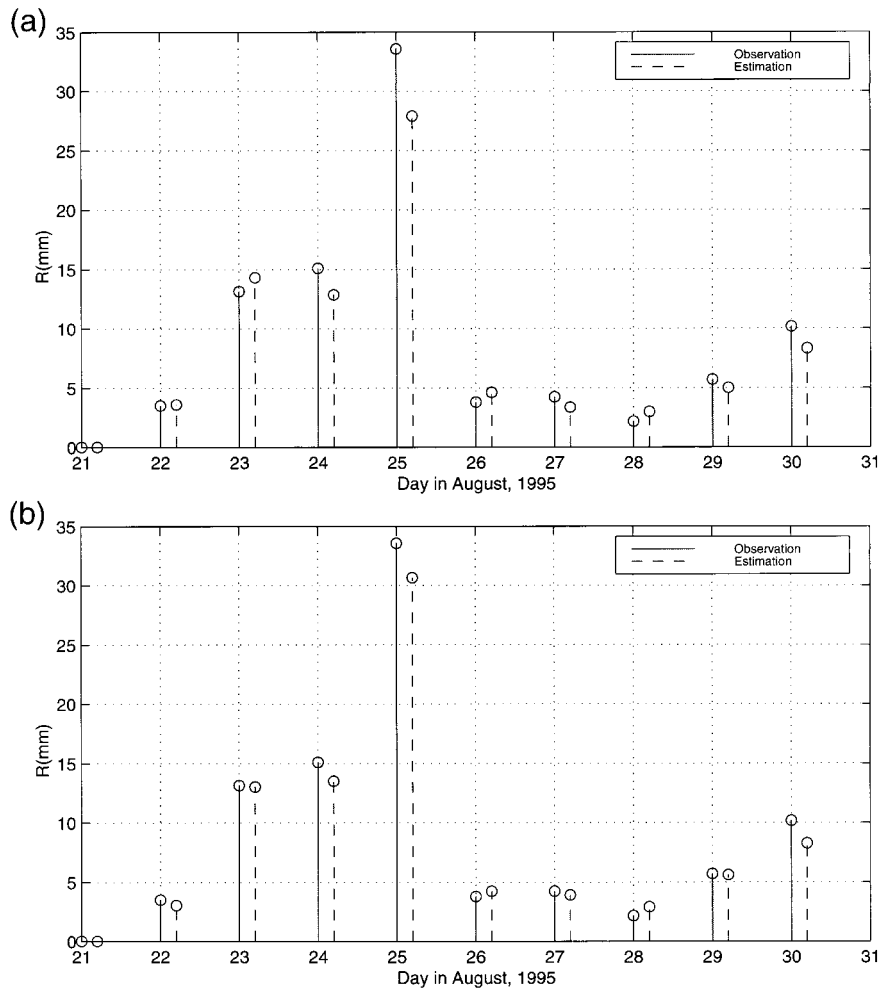


FIG. 10. Mean daily rainfall accumulation estimation results from (a) a fixed neural network algorithm and (b) the adaptive neural network algorithm (for 21–30 Aug 1995).

old. If the distance from a center vector in  $C_2$  to any of the center vectors in the  $C_1$  is greater than a threshold, then this center vector is added to the RBF, and one more neuron is added to the hidden layer. The next parameter to be determined is the set of weights  $W$ , which is determined from Eq. (5). In this process, it is important to determine how different the new data are

when compared with the old training data. If the new data are very different, then those data are included in the set to determine the weights. This procedure ensures that the new data have higher priority in the determination of weights of the modified network. Figure 5 shows the schematic diagram of the adaptive updating scheme for an RBF neural network for rainfall estimation.

TABLE 1. Mean rainfall estimation comparisons between two algorithms during a 10-day period (21–30 Aug 1995). CORR is the correlation coefficient; NE is the normalized error.

Mean hourly rainfall accumulation			
Algorithms	Bias (%)	CORR	NE
Fixed RRN	9.5	0.95	0.29
Adaptive RRN	6.7	0.95	0.27
Mean daily rainfall accumulation			
Algorithms	Bias (%)	CORR	NE
Fixed RRN	9.2	0.89	0.16
Adaptive RRN	6.8	0.90	0.10

### 3. Performance evaluation of the adaptive neural network

Radar data used in this study were collected by the Melbourne WSR-88D in the summers of 1995 and 1998 over central Florida. Two consecutive months (August and September) of radar data and the corresponding rain gauge measurement records were used for this study. The WSR-88D volume scans were done every 6 min. Data from rain gauges within a 200-km radius of the radar were used. We construct radar data constant-altitude plan position indicators, and, based on the time



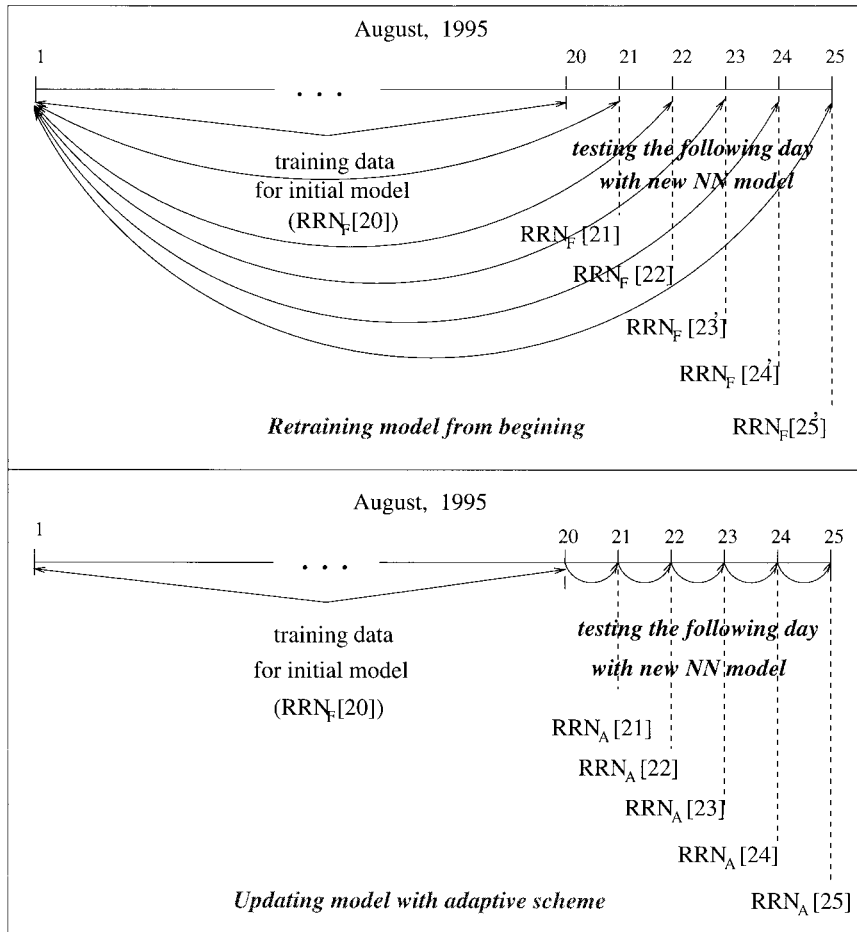


FIG. 11. The comparison scheme between a simply retrained neural network and a dynamic neural network for rainfall estimation (for 21–25 Aug 1995).

of the radar data, 5 min. of gauge data are averaged about that time. The locations of the rain gauges are shown in Figs. 6 and 7. Figure 6 shows the location of the 19 rain gauges used in the 1995 dataset, and Fig. 7 shows the 81 gauges used in the 1998 dataset.

*a. Comparison of a fixed network with an adaptive network for rainfall estimation*

The analysis performed in this section is done using data collected during 1995. The rainfall estimation calculated using an adaptive neural network is compared with a fixed neural network for the period between 21 and 30 August 1995. The fixed neural network is set up based on the radar data and rain gauge measurements during the period of 1–20 August. The RBF network is constructed using the algorithm described in section 2. This fixed neural network is denoted “RRN<sub>F</sub>[20]”, where RRN stands for radar rainfall neural network, the subscript *F* indicates fixed network, and the number 20 indicates that it is based on the first 20 days of data. The adaptive network is based on the initial model

(RRN<sub>F</sub>[20]), and is adaptively updated everyday when new rain gauge data become available. The adaptive networks are denoted as RRN<sub>A</sub>[21], RRN<sub>A</sub>[22], RRN<sub>A</sub>[23], . . . , which are used to estimate rainfall for the following day. The subscript *A* with RRN indicates the adaptive network. Figure 8 shows a schematic that describes the test scheme.

The mean hourly rain-rate estimation from the fixed network and the adaptive network over the gauge are shown in Fig. 9 during the period of 21–30 August (240 h). Fig. 9a shows the hourly rain accumulation estimated using the fixed network. Fig. 9b compares the hourly rain accumulation derived by using the adaptive network with the corresponding ground observations. For the convenience of evaluation, the gauge observation is shown by a solid line in all figures. The mean daily rainfall accumulation results based on the two rainfall algorithms are shown in Fig. 10. The statistical analysis of the rainfall estimation results is listed in table 1. The statistical parameters used to evaluate the rainfall estimates are 1) bias, 2) normalized error (NE, mean absolute deviation normalized with respect to the mean),

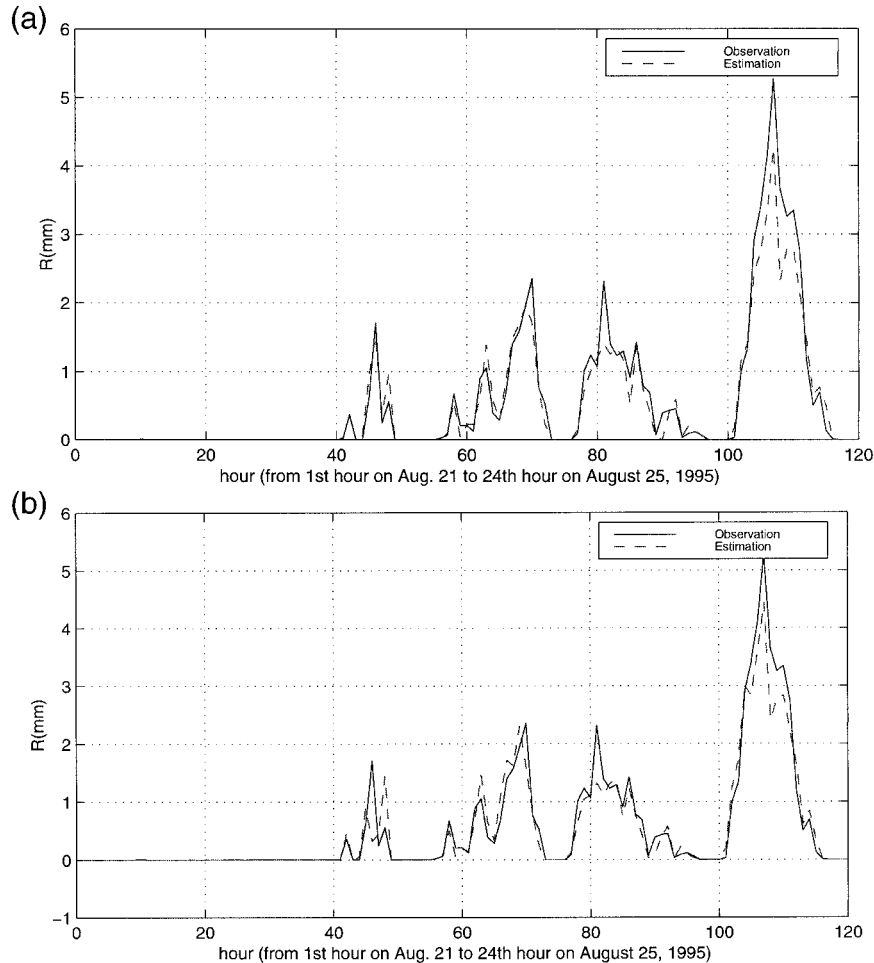


FIG. 12. Mean hourly rain rate estimation compared with gauge data from (a) a simply retrained neural network and (b) the adaptive neural network.

and 3) the correlation coefficient. From this table, it can be seen that the mean bias in hourly rain-rate estimates from the fixed RRN and the adaptive RRN are 9.5% and 6.5%, respectively. The corresponding NE are 29% and 27%, respectively. For the daily rainfall accumulation, the mean biases in the two algorithms are 9.2% and 6.8%, respectively; the NE are 16% and 10%, respectively. It is obvious that the  $RRN_A$  (adaptive neural network) performs slightly better than the  $RRN_F$  (fixed neural network) for rainfall estimation, demonstrating the validity of  $RRN_A$ .

#### b. Comparison of the adaptive RRN network with the completely retrained network

To show that the adaptive updated neural network can reach nearly the same level of accuracy as the network that is completely retrained with all the available data, their performances are compared for the 5 days toward the end of August (from 21 to 25 August). This exercise also reveals one of the practical advantages of the adap-

tive scheme over the simple retraining scheme. The schematic of the experiment is shown in Fig. 11. In addition,  $RRN_F$  [21] indicates the fixed RRN obtained by combining data from the first 21 days to form a training dataset and then simply training without using the adaptive updating scheme introduced in section 2. During the process, the advantage of the adaptive scheme became obvious, because the complete training process got computationally very demanding as new data became available. This complexity is a very important practical problem. However, by using the adaptive updating scheme introduced in this paper, this process is simplified. This experiment was done only for 5 days because of the computational complexity of complete retraining. We can compare the performance of the two schemes. The mean hourly rain-rate estimation comparison over the gauge locations is shown in Fig. 12. Table 2 shows the statistical evaluation of this analysis. From the results of Table 2, it can be seen that the performances of the simply retrained RRN and the adaptive updated RRN are very similar. Therefore, we can

TABLE 2. Mean rainfall estimation comparisons between simply retrained network and the adaptive neural network during a 5-day period (21–25 Aug 1995). CORR is the correlation coefficient; NE is the normalized error.

Algorithms	Mean hourly rainfall accumulation		
	Bias (%)	CORR	NE
Simply retrained RRN	10.43	0.97	0.22
Adaptive RRN	7.33	0.95	0.25
Algorithms	Mean daily rainfall accumulation		
	Bias (%)	CORR	NE
Simply retrained RRN	10.42	0.80	0.13
Adaptive RRN	7.66	0.80	0.09

conclude from this analysis that the adaptive RRN can perform as well as the completely retrained RRN in all aspects of estimation accuracy, such as bias and percentage error. However, the adaptive RRN is much simpler, faster, and easy to train, and it never computationally grows out of control.

### c. Further evaluation of the adaptive neural network

The last two sections demonstrated that the simplified adaptive neural network is as accurate as a fully retrained neural network. In this section data from the Tropical Rainfall Measuring Mission (TRMM) Texas and Florida Underflights (TEFLUN) program is used for independent validation of the adaptive neural network rainfall products. This dataset has a large number of gauges in the vicinity of the Melbourne, Florida, radar. Approximately one-half the gauges were chosen randomly for use in training, and the other set of gauges was used for testing. Figure 7 shows the location of the training and testing gauges with respect to the radar. The adaptive neural network for rainfall estimation was

applied to two months of data, August and September of 1998, from the Melbourne radar. As described in the previous sections, the adaptive neural network trains using the training data up to a certain day.  $RRN_A$  is ready for testing the next day, except the testing is done on gauges that are not used in the training at all. In addition, the WSR-88D algorithm and a best  $Z-R$  algorithm computed from the training data for each day (adaptively changed for each day) were also used to evaluate rainfall over the testing gauges. In practice, the best  $Z-R$  algorithm is available only after the day is over; nevertheless, it serves as one of the best-case scenarios as a reference to compare with neural network rainfall estimates. Figure 13 shows the mean daily accumulation time series over the testing gauges for the months of August and September. The three time series correspond to the three algorithms, namely, the adaptive neural network, the WSR-88D algorithm, and the current day-based best  $Z-R$  algorithm. Note that this is the mean daily accumulation over many gauges distributed over a 100-km radius. Individual gauges have higher daily accumulations for some days. Figure 14 shows the mean hourly and daily accumulations in the form of scatterplots. The statistics of the results shown in Figs. 13 and 14 are summarized in Tables 3 and 4. It can be seen that the daily accumulation can be estimated with greater accuracy than the best  $Z-R$  algorithm (adaptively changed) for each day. In addition, the adaptive neural network is also much better than the fixed WSR-88D algorithm.

## 4. Summary and conclusions

An adaptive neural network scheme for rainfall estimation is developed in this paper. The motivation for this method is to develop a scheme in which a

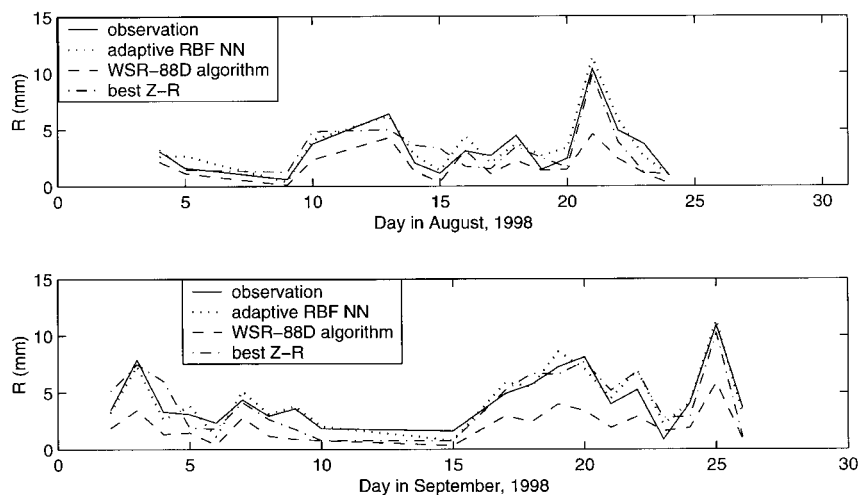


FIG. 13. Mean daily rainfall accumulation time series from the gauges and from the adaptive RBF neural network, the WSR-88D algorithm, and the current day-based best  $Z-R$  algorithm: the results for (top) Aug 1998 and (bottom) Sep 1998.

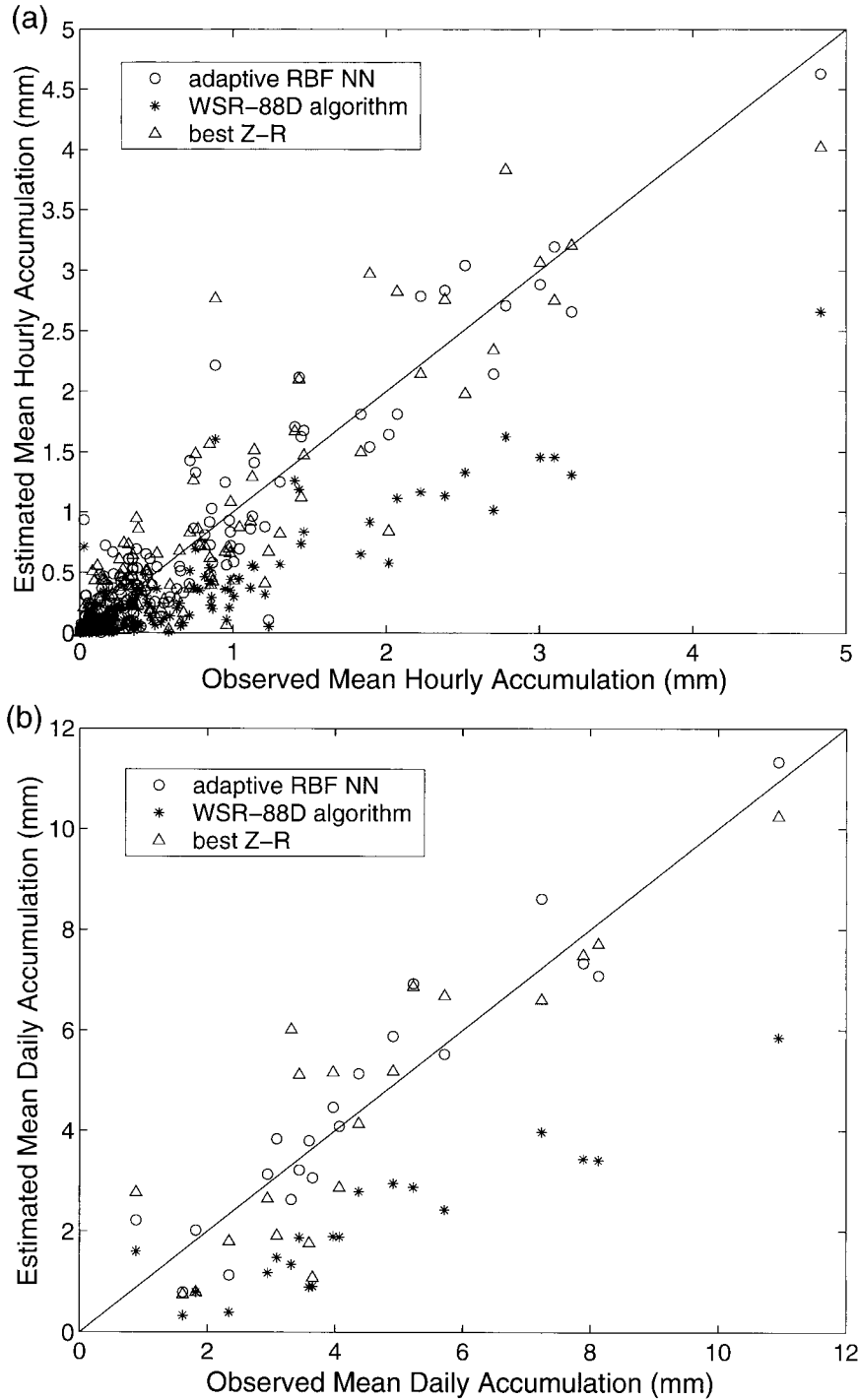


FIG. 14. Scatterplots of the mean (a) hourly and (b) daily rainfall accumulations for the dataset from Sep 1998.

neural network built for radar rainfall estimation can be gradually modified over time without retraining the network from the beginning. Such a network is very practical for real-time implementation on WSR-88D systems. This goal was achieved by using a radial basis function neural network in which the neural net-

work was adjusted adaptively. The algorithm also ensures priority for new data in the training process. The performance of the adaptive neural network is evaluated by using 2 months of WSR-88D data collected for the TRMM TEFLUN field program. The analysis indicates that the neural network-based tech-

TABLE 3. Mean rainfall estimation comparisons among three algorithms for the period of 1–30 Aug 1998. CORR is the correlation coefficient; NE is the normalized error.

Hourly rainfall accumulation			
Algorithms	Bias (%)	CORR	NE
WSR-88D Z-R	44.68	0.90	0.47
Best Z-R	6.38	0.82	0.46
Adaptive RRN	-8.51	0.92	0.38
Daily rainfall accumulation			
Algorithms	Bias (%)	CORR	NE
WSR-88D Z-R	44.24	0.84	0.44
Best Z-R	6.06	0.81	0.32
Adaptive RRN	-8.48	0.89	0.21

TABLE 4. Same as Table 3 but for the period of 1–26 Sep 1998.

Hourly rainfall accumulation			
Algorithms	Bias (%)	CORR	NE
WSR-88D Z-R	51.85	0.90	0.57
Best Z-R	1.85	0.80	0.40
Adaptive RRN	-3.70	0.93	0.32
Daily rainfall accumulation			
Algorithms	Bias (%)	CORR	NE
WSR-88D Z-R	52.47	0.87	0.54
Best Z-R	2.02	0.82	0.25
Adaptive RRN	-3.59	0.90	0.15

nique estimates rainfall better than the simple WSR-88D Z-R algorithm and the best Z-R algorithm estimated for each day. In addition, the adaptive network also reaches nearly the same estimation accuracy as a completely retrained fixed RRN with all the available data. When compared with a completely retrained neural network, the adaptive neural network is easier and faster to set up and is very suitable for real-time implementation on WSR-88D radars.

*Acknowledgments.* This research was supported by the NASA TRMM program.

#### REFERENCES

- Funahashi, K., 1989: On the approximate realization of continuous mappings by neural networks. *Neural Networks*, **2**, 183–192.
- Krasnopolsky, V. M., L. C. Breaker, and W. H. Gemmill, 1995: A neural network as a nonlinear transfer function model for retrieving surface wind speeds from the Special Sensor Microwave Imager. *J. Geophys. Res.*, **100**, 11 033–11 045.
- Mark, J. L., cited 1996: Radial basis function networks. [Available online at <http://www.anc.ed.ac.uk/mjo/intro/intro.html>.]
- Tsintikidis, D., J. Haferman, E. Anagnostou, and W. Krajewski, 1996: A neural network approach to estimating rainfall from spaceborne microwave data. *IEEE Trans. Geosci. Remote Sens.*, **35**, 1079–1093.
- Xiao, R., and V. Chandrasekar, 1995: Multiparameter radar rainfall estimation using neural network techniques. Preprints, *27th Conf. on Radar Meteorology*, Vail, CO, Amer. Meteor. Soc., 199–204.
- , and —, 1997: Development of a neural network based algorithm for rainfall estimation from radar measurements. *IEEE Trans. Geosci. Remote Sens.*, **36**, 160–171.