Development of a Neural Network Based Algorithm for Rainfall Estimation from Radar Observations

Rongrui Xiao, *Member, IEEE,* and V. Chandrasekar, *Member, IEEE*

***Abstract*— Rainfall estimation based on radar measurements has been an important topic in radar meteorology for more than four decades. This research problem has been addressed using two approaches, namely a) parametric estimates using reflectivity-rainfall relation (Z-R relation) or equations using multiparameter radar measurements such as reflectivity, differential reflectivity, and specific propagation phase, and b) relations obtained by matching probability distribution functions of radar based estimates and ground observations of rainfall. In this paper we introduce a neural network based approach to address this problem by taking into account the three-dimensional (3-D) structure of precipitation. A three-layer perceptron neural network is developed for rainfall estimation from radar measurements. The neural network is trained using the radar measurements as the input and the ground raingage measurements as the target output. The neural network based estimates are evaluated using data collected during the Convection and Precipitation Electrification (CaPE) experiment conducted over central Florida in 1991. The results of the evaluation show that the neural network can be successfully applied to obtain rainfall estimates on the ground based on radar observations. The rainfall estimates obtained from neural network are shown to be better than those obtained from several existing techniques. The neural network based rainfall estimate offers an alternate approach to the rainfall estimation problem, and it can be implemented easily in operational weather radar systems.**

***Index Terms*—Multiparameter radar, neural networks, radar rainfall estimation.**

I. Introduction

ACCURATE estimation of ground rainfall from radar measurements is an important topic of current interest. Traditionally, radar rainfall estimates were computed from a parametric reflectivity–rainfall relation that was allowed to vary from place to place and season to season. Such an approach has not been very successful because of the extensive variability observed with Z-R relations with rainfall types and climatic regions [1]. Multiparameter radar estimates of rainfall were introduced by Seliga and Bringi [2], [3] using differential reflectivity $(Z_{\rm DR})$ and specific differential propagation phase $(K_{\rm DP})$ to address the problem of rainfall estimation by obtaining better characterization of the drop size distribution. Sachidananda and Zrnic [4] used $K_{\rm DP}$ measurement to estimate rainfall. In spite of advances made by multiparameter radar estimates of rainfall, only few experiments done in a controlled manner were able to demonstrate significant improvements using multiparameter radar based rainfall estimates [5], [6]. Zawadzki [7] suggested that significant problems exist in experiments conducted to compare rainfall estimates from radar and raingage due to the observation process and other physical mechanisms. Radar observation at one elevation angle is nearly an instantaneous snap shot of the horizontal reflectivity structure, whereas the raingage has to accumulate or count sufficient number of raindrops to obtain an accurate estimate of rainfall accumulation. The various mechanisms involved in the rainfall observation process by radar and ground based instrumentation are complex and exhibit wide variability. Both radar and surface observations may depend on the three dimensional (3-D) structure of the drop size distribution (DSD).

In this paper a totally different approach, namely, a neural network based technique, is introduced to address the rainfall estimation problem using radar data. Without assuming any parametric relation, this technique maps the three dimensional radar measurements such as $(Z_H, Z_{\rm DR})$ to the ground raingage measurements directly. Using raingage measurements as the target set of observations on the ground, the neural network approximates the relation between the radar measurements and ground rainfall observations based on a training data set. When the neural network is trained appropriately, it generalizes the relations so that it can be applied to other new data sets. The theoretical basis of this technique is the universal approximation theorem which states that a multilayer feed-forward neural network (MFNN) such as a multilayer perceptron (MLP) is capable of performing any nonlinear input-output mapping [8], [9]. Due to the dense interconnections and adaptive nature, the neural network algorithm is also robust or error tolerant [10] which implies that error at a few input nodes or links will not damage the overall performance very much.

We have developed three-layer perceptron neural networks to provide rainfall estimates on the ground from radar observations. Two such networks have been developed, one using reflectivity Z_H alone as the input, the other using both Z_H and differential reflectivity $Z_{\rm DR}$ as the input. The neural networks were trained by using a subset of the data collected by the CP-2 multiparameter radar and a few raingages from the Kennedy Space Center (KSC) raingage network during the CaPE experiment [11]. The neural networks were subsequently applied to estimate rainfall for four days during the months of July and August 1991.

The organization of this paper is as follows: Section II describes the development of neural network using multilayer

Manuscript received December 5, 1995; revised July 1, 1996. This work was supported by the National Science Foundation (ATM-9200761) and NASA.

R. Xiao was with Colorado State University, Fort Collins, CO 80523 USA. He is now with Vexcel Corporation, Boulder, CO 80301 USA.

V. Chandrasekar is with Colorado State University, Fort Collins, CO 80523 USA.

Publisher Item Identifier S 0196-2892(97)00369-0.

perceptron for the rainfall estimation problem. Section III describes the the training and testing data sets used in the development of neural network based rainfall estimation. Section IV presents the results of multiparameter radar rainfall estimation using neural network. Section V summarizes the important results of this paper.

II. The Neural Network for Radar Rainfall Estimation

A. Three-Layer Perceptron Network for Function Approximation

Rainfall rate obtained on the ground can be potentially dependent on the 3-D structure of precipitation aloft. In principle one can try to obtain a functional approximation between rainfall on the ground and the 3-D radar observation above the observation point. Therefore, the rainfall estimation problem can be viewed as a complex function approximation problem. The universal approximation theorem for neural network states that a two-layer feed forward perceptron network with nonconstant, bounded, and monotone-increasing continuous activation function can perform arbitrary nonlinear input-output relationship mapping [12]. Therefore, a two-layer perceptron network can be used for the rainfall estimation problem. The above universal approximation theorem gives the theoretical justification for the approximation of an arbitrary continuous function by a two-layer (one hidden-layer) perceptron network. In practice, however, a three-layer (two hidden-layer) perceptron network works better than a two-layer perceptron for the function approximation problem. This is because the interaction between neurons in a single hidden-layer network makes it difficult to obtain a globally good approximation, while a two-hidden layer network isolates and thus reduces the interaction effects by solving the problem in two steps, i.e., the first hidden-layer extracts the local features of the input data whereas the second hidden-layer extracts the global feature, to make the approximations in different regions of the input space individually adjusted [12]. Due to above reasons, three-layer perceptron networks are chosen in this paper for the rainfall estimation problem. The structure of a three-layer perceptron is shown in Fig. 1.

B. The Recursive Least Square Learning Algorithm

A neural network learns the input–output relationship through the training process. The learning process in a neural network is an interactive procedure in which its connection weights are adapted through the presentation of a set of input–output training example pairs. Learning algorithms dictate the changes of the weights. The gradient descent based back-propagation algorithm is the most popular learning algorithm for multilayer perceptrons [13]. However, the convergence rate of the training error in these algorithms is very slow. Several fast learning methods using recursive least square (RLS) method to speed up the learning precess of MLP have been proposed [14], [15]. The RLS learning algorithm based on Azimi and Liou [14] is implemented in this paper for training three-layer rainfall estimation perceptron networks. The details of the algorithm are described in the Appendix.

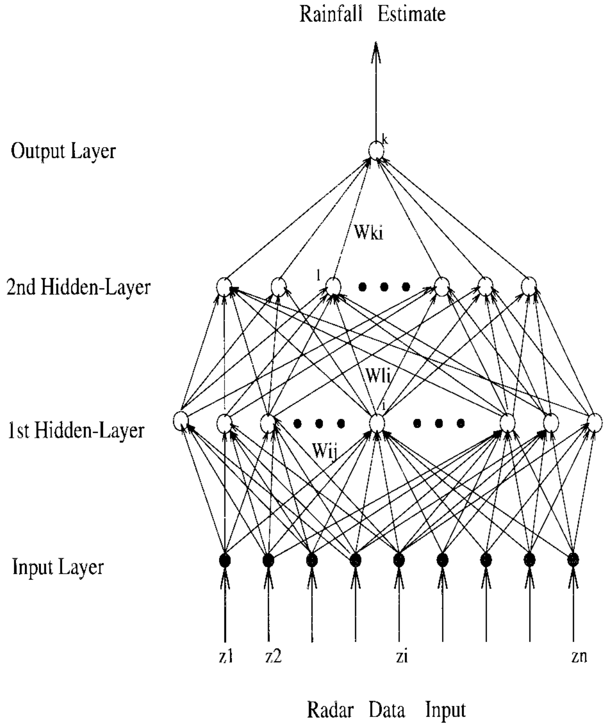


Fig. 1. The structureof the three-layer perceptron neural network.

C. Training and Testing Data Generation

A representative training data set consisting of the radar data and corresponding ground raingage data are needed to develop a multilayer perceptron for the rainfall estimation problem. Radar data and other related information are applied to the network as the input and the corresponding raingage data are used as the target or desired output [Fig. 2(a)]. The training procedure for a multilayer perceptron includes two steps, namely forward propagation and backward propagation. The connectional weights are updated during the backward error propagation according to the learning algorithm. This process is repeated until the error between the network output and desired output (raingage measurement) meets the prescribed requirement. When the training process is complete, the network is ready for application. Rainfall estimates can be obtained if radar data are applied to the network at this stage [Fig. 2(b)].

The neural network developed here accepts radar data as input to provide an estimate of rainfall on the ground. Radar data over a storm are typically collected in a sequence of plan position indicators (PPI) or range height indicators (RHI). The sequence is usually extended in space to obtain full coverage over the storm cell. In order to provide a consistent format for the input data to the neural network, we generate constant altitude PPI's (CAPPI) at different altitudes. These CAPPI's can be generated from both PPI and RHI volume scans. Fig. 3 shows the schematic of the input data generation for the neural

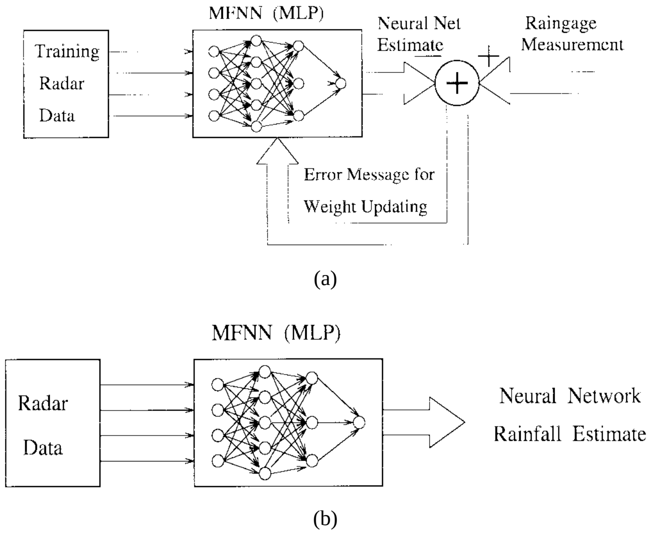


Fig. 2. Development of the multilayer perceptron (MLP) for rainfall estimation: (a) Training of a MLP for rainfall estimation. (b) Application of MLP for rainfall estimation.

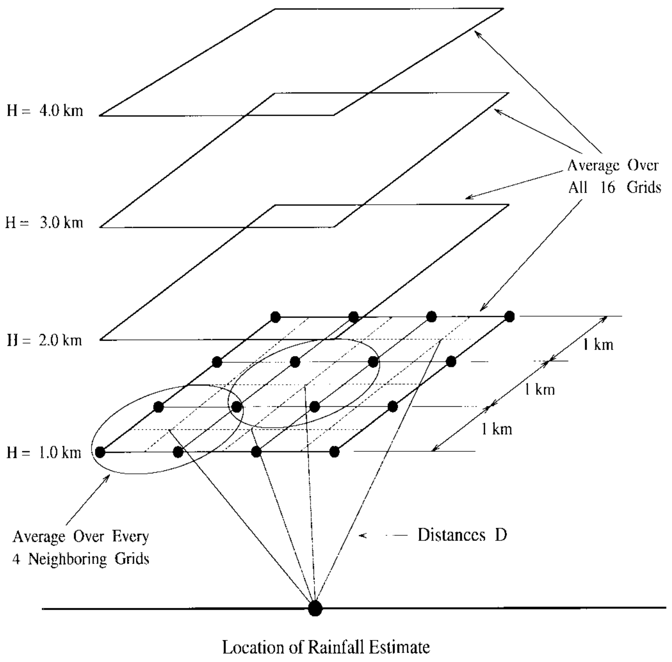


Fig. 3. Data structure that forms the input to the neural network.

network. Let the point marked "Location of Rainfall Estimate" on Fig. 3 be the target point where rainfall estimate is desired. This estimate may be influenced by the precipitation above, over a much wider area extending to several kilometers in all directions. However, it is also reasonable to assume that the storm cells far away from the observation point are less likely to influence precipitation observed at a point. The input data set to the neural network is as follows (see Fig. 3):

1. Radar measurements at every kilometer over a 3 × 3 km square grid at the lowest elevation CAPPI (Fig. 3 shows the lowest elevation CAPPI for the data set used in this paper as 1.0 km).

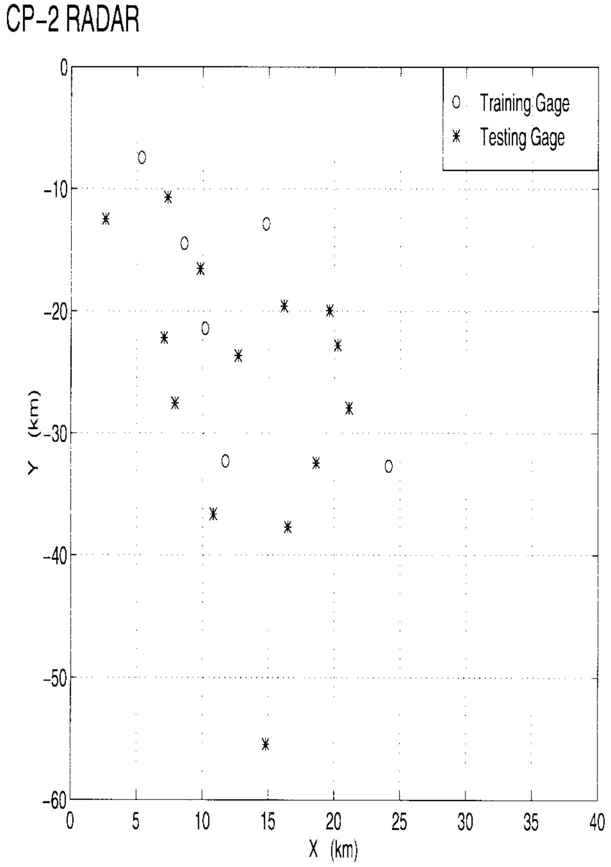
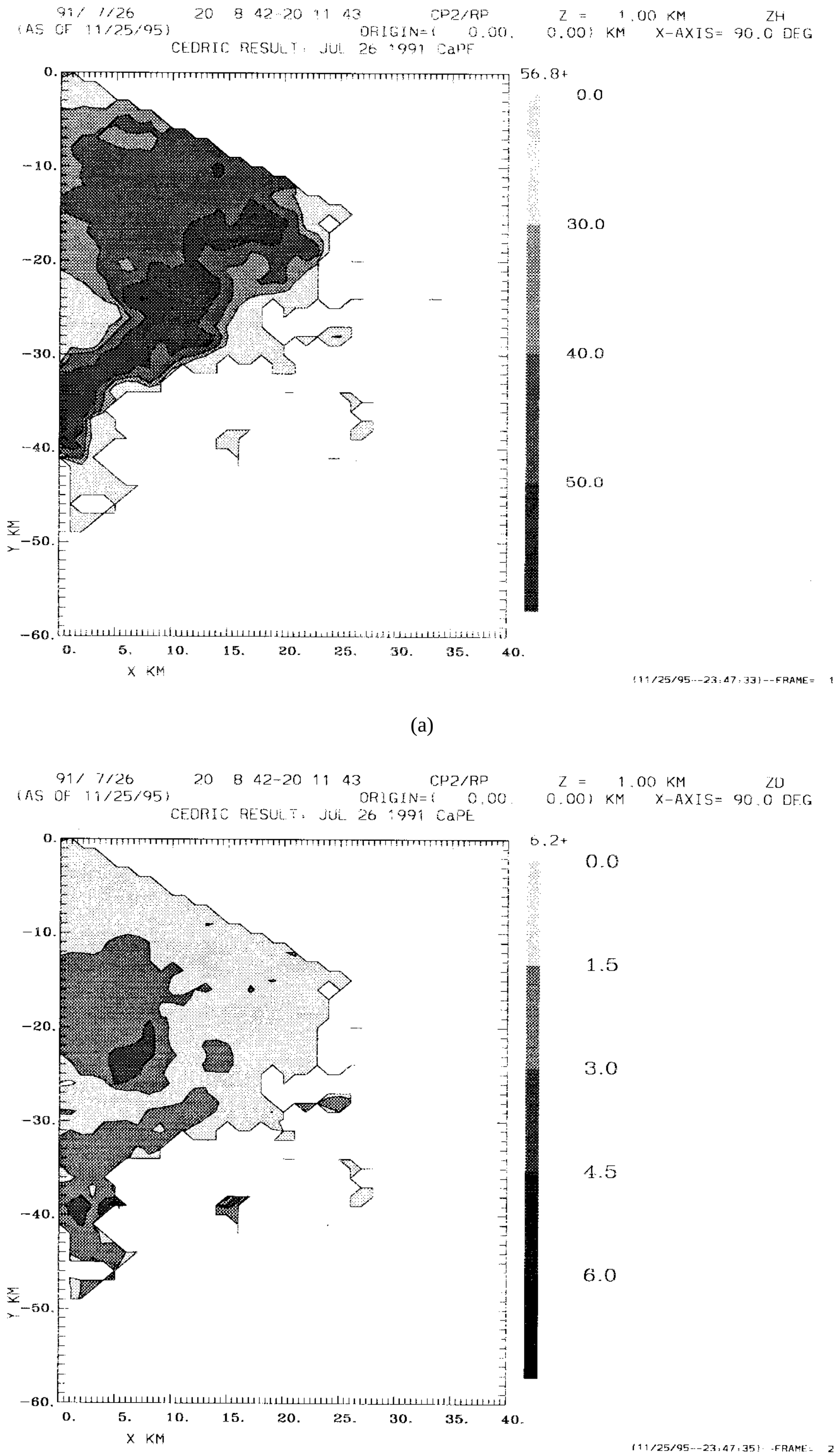


Fig. 4. KSC raingage locations relative to the CP-2 radar (located at the origin). Twenty raingages were used in the experiment. Data from six raingages are used to form the training data set for the neural network.

2. Mean radar observations averaged over 3 × 3 km grids for four levels such as 1.0 km, 2.0 km, 3.0 km, and 4.0 km (1 km apart). By utilizing mean vertical profile above the lowest level we reduce the size of the input and therefore the size of the network. Note that the radar measurements referred in item 1 and item 2 above can be either reflectivity (Z_H), differential reflectivity (Z_{DR}), or any other measurement from a multiparameter radar. In this paper, we have restricted our analysis to Z_H and Z_{DR} measurements.
3. Spatial information is used to form the other part of the input data as an attempt to provide the spatial information of the data points with respect to the target point where rainfall estimate is made. The distances from the center of each four neighboring grids at the lowest elevation are provided as part of the input data. The heights of the four levels corresponding to the four averaged Z_H values representing the vertical profile are also provided as part of the input data set. In this paper these distances are represented by D.

Therefore, each input vector includes Z_H (and Z_{DR}) values, and distance information D. The target output of the neural network corresponding to each input vector is the average rainfall rate during the period of a radar volume scan. The resolution of raingage data used in this paper is one minute. It



(a)

(b)

Fig. 5. CAPPI scans of CP-2 radar during a rainstorm on July 26, 1991. (a) Horizontal reflectivity Z_H, and (b) differential reflectivity $Z_{\rm DR}$.

typically takes several minutes to complete a volume scan over a storm cell with current weather radar systems. Therefore the accumulating time period should be of the same order of radar volume scan times. During the CaPE experiment, CP-2 radar took approximately 5 min to finish a PPI volume scan. We have used mean rainfall rate based on 5 min gage accumulation as the target output.

III. Data Sources and Processing

A. Radar and Raingage Data Sources

Radar and raingage observations collected during CaPE experiment are used to demonstrate the neural network based rainfall estimation procedure. CaPE experiment was conducted in the central Florida region during the summer of 1991 [11].

TABLE I
THE CHARACTERISTICS OF THE CP-2 MULTIPARAMETER RADAR

Characteristic	CP-2 Radar
Polarization type	Linear
Wavelength (cm)	10.7
Peak power (kW)	1200
Pulse length (μs)	1.0
PRF (s^{-1})	960
Antenna type	Center fed paraboloid
Antenna size (m)	8.5
Beamwidth	0.93°
Polarizations transmitted	Linear V or H
Polarization received	Copolar to transmit
Maximum sidelobe level (dB)	-21
Polarization control method	Ferrite switch
Polarization control period	Pulse by pulse

One of the objectives of CaPE program was remote estimation of rainfall. The radar data were collected by the NCAR CP-2 radar and the raingage data were collected by 20 tipping bucket raingages with a recording resolution of 1 min. The gage data were maintained by the NASA Tropical Rainfall Measurement Mission (TRMM) program. The gages were located in the vicinity of the Merrit Island area at the KSC. Fig. 4 shows the locations of CP-2 radar and the raingages. The gages will be referred hereforth as KSC raingages. The radar parameters of interest to this paper are the reflectivity factor at horizontal polarization (Z_H) and the differential reflectivity (Z_{DR}) both measured at S-band frequency by the CP-2 multiparameter radar. Table I lists the main features of the CP-2 radar that are relevant to this paper. Data used in this paper were collected by integrating 64 sample pairs with 1 ms PRT (pulse repetition time). CP-2 radar performed sequences of PPI volume scans over the KSC raingage area during four storm events on days July 26, July 30, August 9, and August 19, 1991. Constant Altitude PPI (CAPPI) data containing Z_H and Z_{DR} values at four levels (1 km, 2 km, 3 km, and 4 km) were generated from the radar data. Fig. 5(a) and (b) show examples of radar Z_H and Z_{DR} CAPPI scans at 1 km level.

Fig. 6 shows the raingage measurements over all 20 KSC gage sites on July 26, 1991. Five-minute integration is performed on the raingage data to obtain the corresponding mean rainfall rate measurement in the period of a radar PPI volume scan. This mean rainfall rate obtained over 5 min over raingage measurements is used as the target output for the training data set and as the ground truth for the network performance evaluation.

B. Preprocessing of the Training and Testing Data

The input vector is scaled such that elements in the input vector are of similar magnitude, in order to ensure that no parameter is dominant over other parameters and the importance of each parameter is equally presented to the network. For example, the value of reflectivity Z_H is normalized with the possible maximum reflectivity value such as 60 (this maximum value can be easily changed), i.e.,

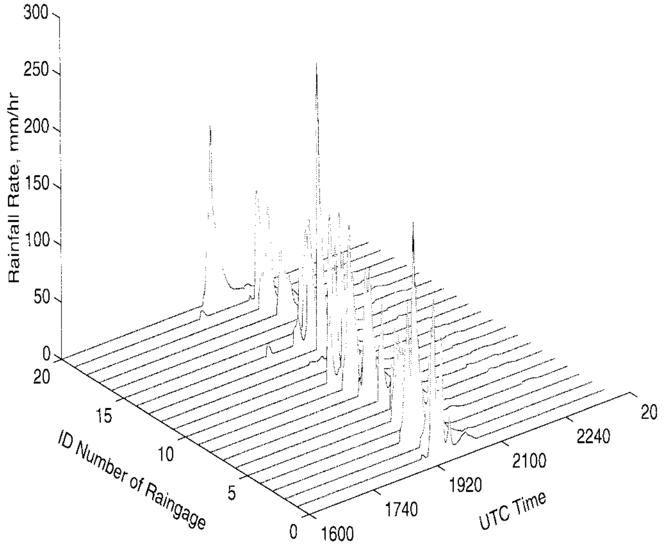


Fig. 6. Raingage measurements over 20 the KSC gage sites during a storm on July 26, 1991, Florida.

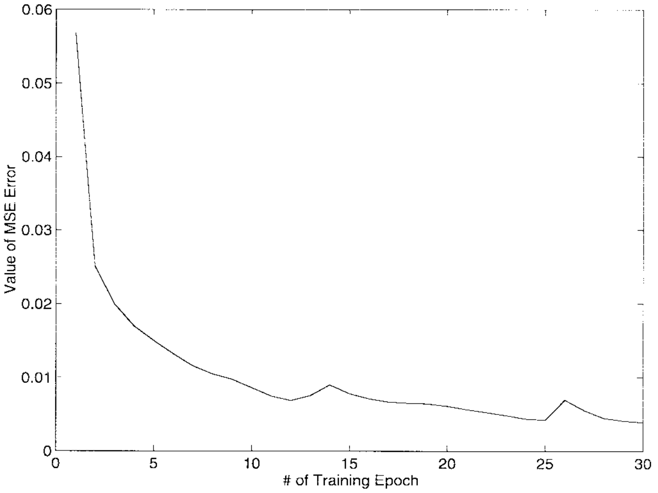


Fig. 7. Training error of the 3-layer perceptron neural network with Z_H as the input.

$$z_h = Z_H/60.0. \tag{1}$$

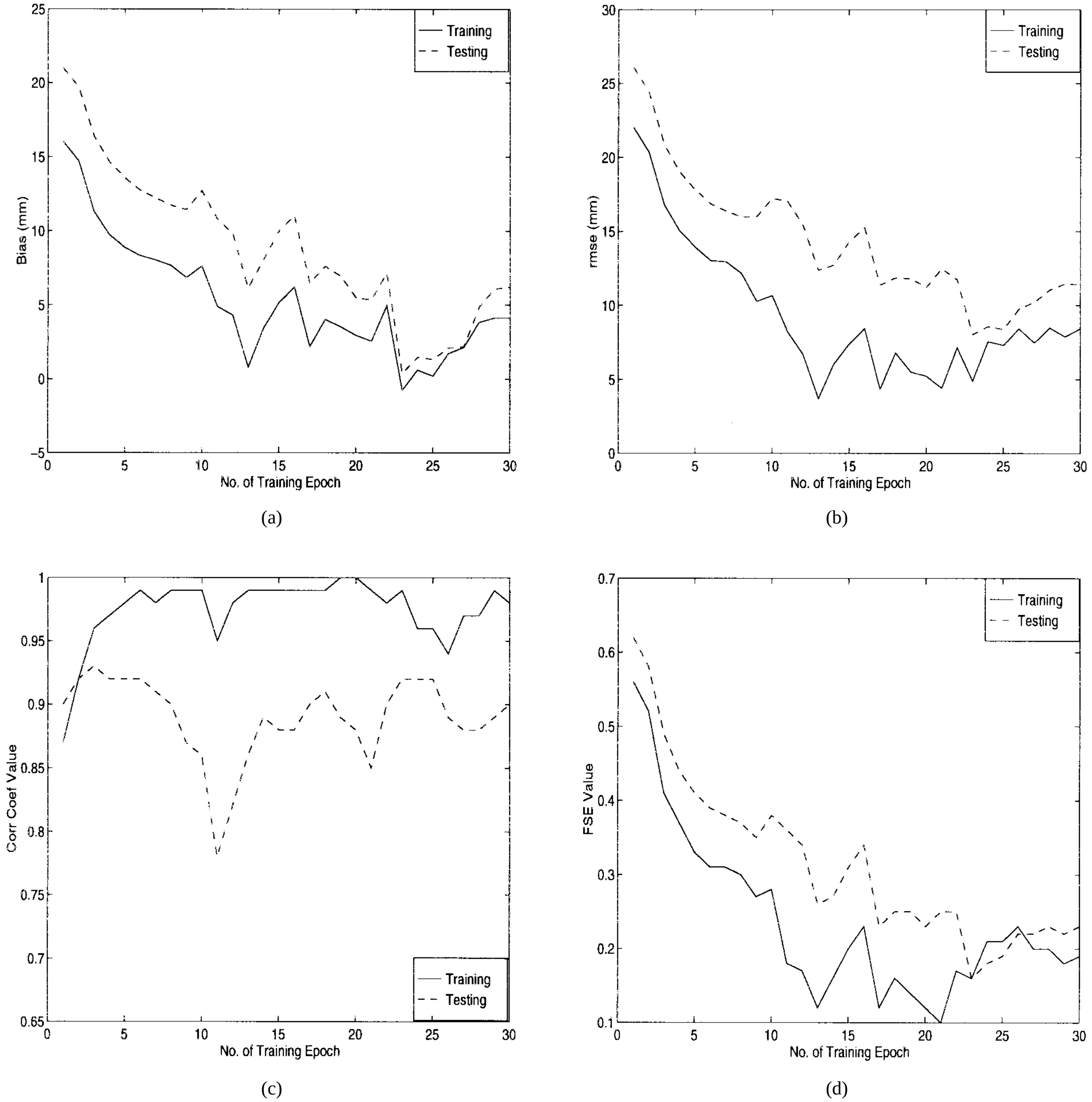
No scaling is applied to Z_{DR} values. For distance D as mentioned in Section II, a Gaussian weighting function is applied as follows

$$d = \exp[-(D/4.0)^2]. \tag{2}$$

A log function transformation is applied to the 5-minute mean rainfall rate values to provide a target output between 0 to 1, to make the output layer of the perceptron network work in a linear region. Let $R_{\max}$ be the maximum possible rainfall rate to be encountered in the observation. Then the transformation applied to rainfall rate is

$$r = \log_{10}(R+1)/C. \tag{3}$$

Where $C = \log_{10}(R_{\max} + 1)$.



(a)

(b)

(c)

(d)

Fig. 8. Performance of a three-layer perceptron network using Z_H as input during training and testing, shown as a function of the training epoch. The evaluation parameters in (a)–(d) are the bias, root mean square error (rmse), correlation coefficient (corr coef), and fractional standard error (FSE), respectively.

In the case when only reflectivity factor Z_H is used to obtain rainfall estimates, a vector of normalized reflectivities z_h and corresponding scaled distances d are used as the input to the multilayer perceptron, and the length of input vector is 26 $(13,\ z_h$ plus $13,\ d)$. In the case when both Z_H and $Z_{\rm DR}$ values are used to obtain rainfall estimates, z_h, z_{dr} and d are combined to form the input vector and the size of input vector is 39 $(13,\ z_h$ plus $13,\ z_{dr}$ plus $13,\ d)$. The value of normalized mean rainfall rate r is the desired output for both cases. An inverse function is applied to the network output to retrieve the rainfall estimate given by

$$\hat{R} = 10^{C\cdot r} - 1.0 \tag{4}$$

We choose data from six out of 20 gages to train the neural network. The location of the six training gages are shown in Fig. 4 with the symbol of "o." The trained network was subsequently utilized to estimate the rainfall at the other 14 gage locations to evaluate the performance. The neural network sizes are determined after several trial and error experiments. The size of the input layer is decided by the size of the input vector, which is 26 or 39 depending on the data used. The size of the output layer is decided by the size of the target output vector, which is 1 for this application. The size of the two hidden layers are decided by observing the training convergence error and generalization performance of the trial network. The size of the first hidden layer should be larger than the input layer to ensure the network handle the complexity of the input properly, whereas the size of the second hidden layer should not be too large to avoid poor generalization performance. For the case when only Z_H is used for rainfall

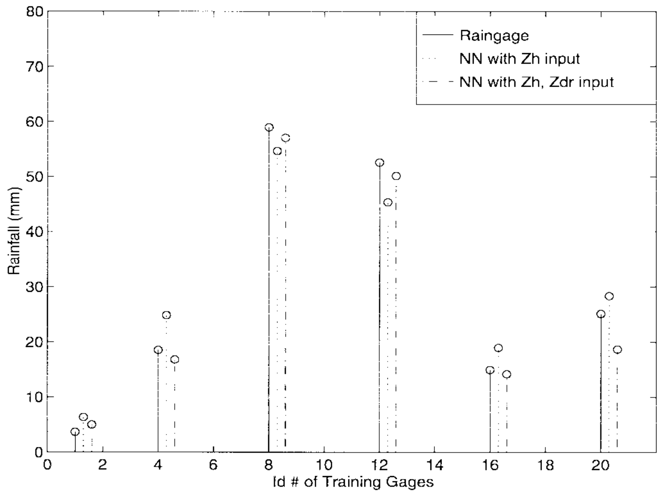


Fig. 9. Comparison of accumulated rainfall at six training gage sites.

estimation, the size of the three-layer perceptron is chosen as 26 : 27 : 17 : 1 representing the number of nodes (neurons) in the input layer, first hidden layer, second hidden layer and the output layer, respectively. When both Z_H and $Z_{\rm DR}$ are used, the size of the network is chosen as 39 : 47 : 21 : 1. Note that these network sizes are not necessary the optimal for the rainfall estimation problem. Rigorous evaluation of the optimum network size is beyond the scope of this paper.

IV. Neural Network Rainfall Estimation Results

A. Performance Evaluation Criteria

Several statistical measures, namely, the bias, the root mean squared error, the correlation coefficient and the fractional standard error between the raingage measurements and the neural networks estimates were computed to evaluate the performance of the network. Let N_g be the total number of raingages involved, and $RF_g(i)$, $RF_n(i)$ be the raingage accumulations and the corresponding neural network rainfall estimates at the ith gage location, then the definitions of above statistical quantities are as follows:

1) *Bias*

$$\text{bias} = \frac{1}{N_g}\sum_{i=1}^{N_g}[RF_g(i) - RF_n(i)]. \tag{5}$$

2) *Root mean squared error (rmse)*

$$\text{rmse} = \left\{\frac{1}{N_g}\sum_{i=1}^{N_g}[RF_g(i) - RF_n(i)]^2\right\}^{\frac{1}{2}}. \tag{6}$$

3) *Correlation coefficient (coef)*

$$\text{coef} = \frac{\frac{1}{N_g}\sum_{i=1}^{N_g}\left[RF_g(i) - \overline{RF}_g\right]\left[RF_n(i) - \overline{RF}_n\right]}{\sigma(RF_g)\cdot\sigma(RF_n)}; \tag{7}$$

where $\overline{RF}_g$ is the mean of $RF_g(i)$ and $\overline{RF}_n$ is the mean of $RF_n(i)$. $\sigma(RF_g)$ and $\sigma(RF_n)$ are standard deviations. They are expressed as follows

$$\overline{RF}_g = \frac{1}{N_g}\sum_{i=1}^{N_g} RF_g(i) \tag{8}$$

$$\overline{RF}_n = \frac{1}{N_g}\sum_{i=1}^{N_g} RF_n(i) \tag{9}$$

$$\sigma(RF_g) = \left\{\frac{1}{N_g}\sum_{i=1}^{N_g}[RF_g(i) - \overline{RF}_g]^2\right\}^{\frac{1}{2}} \tag{10}$$

$$\sigma(RF_n) = \left\{\frac{1}{N_g}\sum_{i=1}^{N_g}[RF_n(i) - \overline{RF}_n]^2\right\}^{\frac{1}{2}}. \tag{11}$$

4) *Fractional standard error (FSE)*

$$\text{FSE} = \frac{\frac{1}{N_g}\sum_{i=1}^{N_g}|RF_g(i) - RF_n(i)|}{\overline{RF}_g}. \tag{12}$$

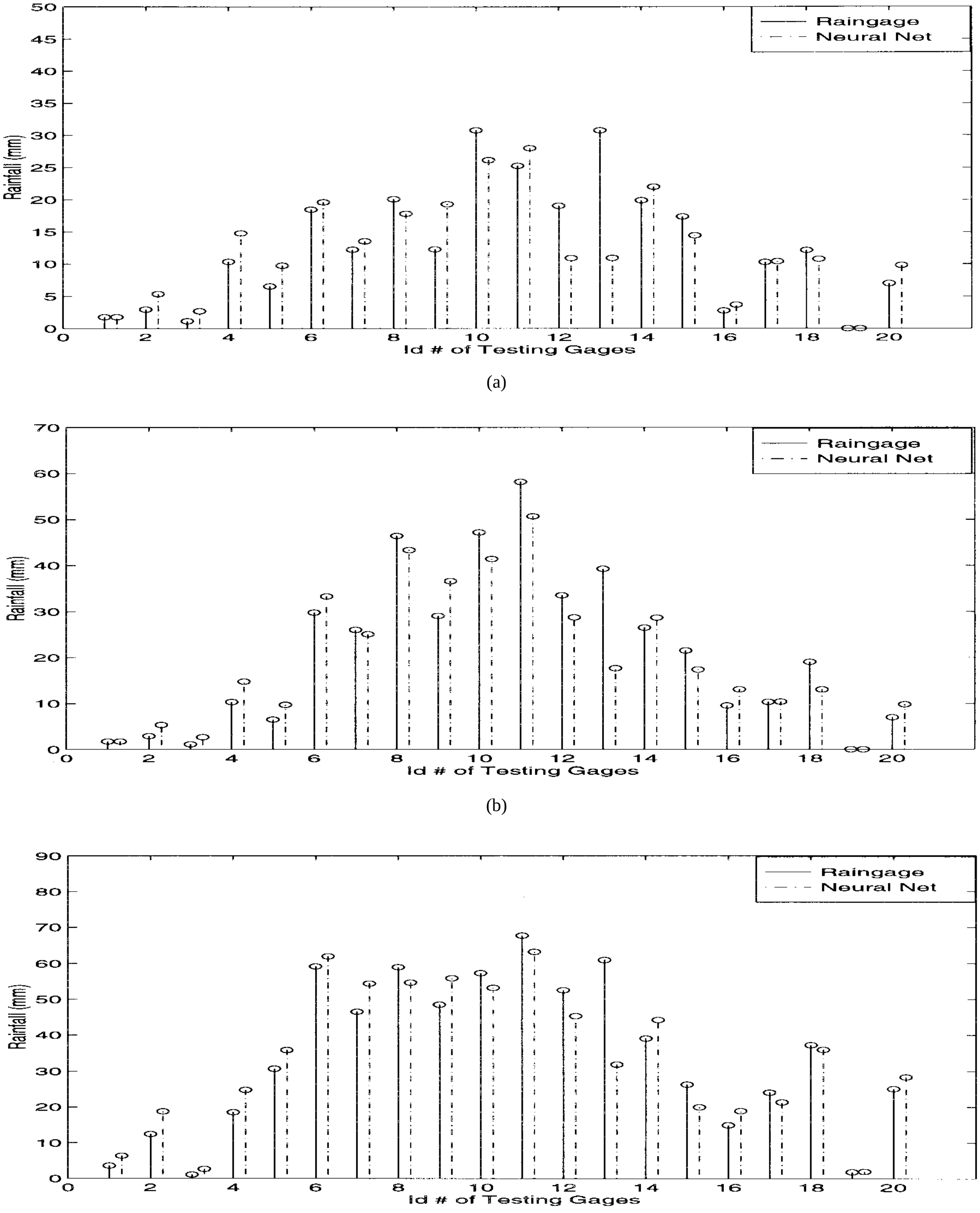
B. The Cross-Validation Scheme

The performance of the network can be evaluated at every stage of the training process. This evaluation is done by comparing the network outputs with actual raingage measurements at the gage locations that are used in the training process as well as at the gage locations that are not part of the training data set. This procedure is called cross-validation, which is designed to check the generalization performance of the neural network. Fig. 7 shows the training error (the mean-squared error between the desired output and the actual network output) versus training epoch for the neural network that uses reflectivity only. The term epoch represents the training cycle during which the whole training data set is applied to the neural network once.

Fig. 8 shows the cross-validation results based on four-day rainfall accumulations obtained from the raingages and from the three-layer perceptron neural network that uses Z_H as input for both the training and the testing data sets. Fig. 8(a)–(d) show the performance evaluation parameters, namely, bias, root mean-squared error (rmse), correlation coefficient (coef), and fractional standard error (FSE) as a function of training epoch, respectively. The solid lines in Fig. 8 are results for training data whereas dotted lines are results for testing data. We can see that the bias, rmse and FSE all have a decreasing trend but the correlation coefficient has an increasing trend, indicating that the neural network learns and generalizes the radar measurement-rainfall relations as the training proceeds. We can also see from Fig. 8 that the bias and the errors increase after epoch 23, indicating that “overtraining” is occurring after this point and training process should be stopped here. The cross-validation results for the case when both Z_H and $Z_{\rm DR}$ are used as input are similar to the results shown in Fig. 8.

C. Rainfall Estimation Results and Performance Evaluation

1) Comparison of Daily Rainfall Estimation: Rainfall accumulations obtained from raingage measurements and neural



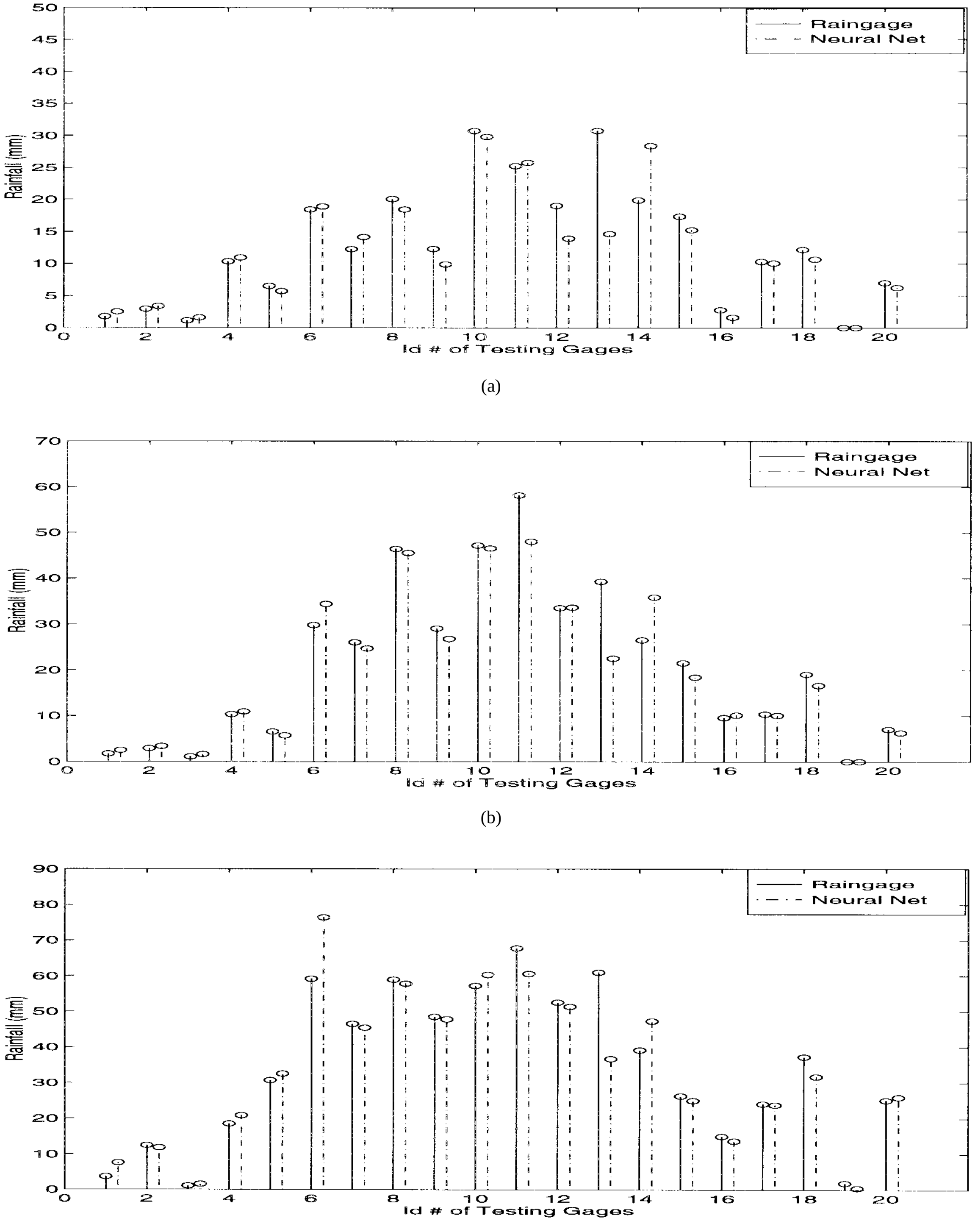
(a)

(b)

(c)

Fig. 10. Comparison of one-day, two-day, and four-day rainfall accumulations obtained from raingages measurements and neural network based estimates using Z_H: (a) one-day rainfall accumulations at 19 gage sites on 07/26, (b) two-day rainfall accumulations at 17 gage sites, 07/26 and 08/19, (c) four-day rainfall accumulations at 20 gage sites, 07/26, 07/30, 08/09, 08/19.

networks are compared in this section. Fig. 9 shows the comparisons of four-day rainfall accumulations at the six training gage sites. Fig. 10(a) shows the comparison of one-day (July 26) accumulations observed by raingages and inferred by neural network estimates obtained using Z_H. Fig. 10(b) and (c) show similar results except two-day accumulations and four-day accumulations are compared. Note again that data from six of the twenty gages were used in training the network. The results of Fig. 10 show that the gage and neural network based estimates of rainfall accumulations agree well at all gage sites including those that were not part of the training data set. Fig. 11(a)–(c) show comparison of one-day, two-day and four-day rainfall accumulations obtained by the neural network using Z_H and Z_{DR} as well as by raingages, respectively. We can see from Fig. 11 that radar rainfall accumulations and gage measurements compare very



(a)

(b)

(c)

Fig. 11. Comparison of one-day, two-day, and four-day rainfall accumulations obtained from raingages measurements and neural network based estimates using Z_H and Z_{DR}: (a) one-day rainfall accumulations at 19 gage sites on 07/26, (b) two-day rainfall accumulations at 17 gage sites, 07/26 and 08/19, (c) four-day rainfall accumulations at 20 gage sites, 07/26, 07/30, 08/09, and 08/19.

well. The statistics of the comparison shown by Figs. 10 and 11 are summarized later in Table II to Table IV.

Comparison With Parametric Rainfall Algorithms: Rainfall algorithms in the past have been obtained as parametric equations based on Z_H or Z_H and Z_{DR}. Numerous Z-R algorithms are available in the literature. We have chosen the Marshall–Palmer [16] Z-R relation for comparison here. The Marshall–Palmer Z-R algorithm is given by

$$R(Z_H) = 0.0365 Z_H^{0.625}. \tag{13}$$

This choice is fairly arbitary and it should not be seen as evaluating the performance of Marshall–Palmer Z-R relation. The (Z_H, Z_{DR}) based parametric algorithm given by Gorgucci

TABLE II
PERFORMANCE EVALUATION OF Z-R RELATION AND NEURAL NETWORK BASED RAINFALL ALGORITHM USING Z_H. VALUES OF $\overline{RF}_g$, $\overline{RF}_{zr}$, $\overline{RF}_n$, BIAS, AND RMSE ARE IN MILLIMETERS

Experiment	Raingage	Z_H, Z_{DR}-R Relation					Neural Network, Z_H, Z_{DR} Input				
Day	$\overline{RF}_g$	$\overline{RF}_{zr}$	bias	rmse	coef	FSE	$\overline{RF}_n$	bias	rmse	coef	FSE
07/26/91	13.74	6.86	6.88	9.34	0.75	0.51	13.25	0.50	5.66	0.78	0.26
07/30/91	8.83	2.83	5.99	7.56	0.96	0.68	7.10	1.73	3.32	0.94	0.29
08/09/91	9.67	5.10	4.57	6.07	0.89	0.48	11.92	-2.25	3.64	0.92	0.26
08/19/91	13.75	6.85	6.90	8.88	0.77	0.52	12.67	1.08	3.66	0.91	0.19

TABLE III
PERFORMANCE EVALUATION OF Z_H, $Z_{\rm DR}$-R RELATION AND NEURAL NETWORK BASED RAINFALL ALGORITHM USING Z_H and $Z_{\rm DR}$. VALUES OF $\overline{RF}_g$, $\overline{RF}_{\rm dr}$, $\overline{RF}_n$, BIAS, AND RMSE ARE IN MILLIMETERS

Experiment	Raingage	Z_H, Z_{DR}-R Relation					Neural Network, Z_H, Z_{DR} Input				
Day	$\overline{RF}_g$	$\overline{RF}_{dr}$	bias	rmse	coef	FSE	$\overline{RF}_n$	bias	rmse	coef	FSE
07/26/91	13.74	14.01	-0.26	6.28	0.79	0.29	12.74	1.01	4.49	0.88	0.18
07/30/91	8.83	6.24	2.58	3.30	0.98	0.29	7.56	1.27	2.79	0.94	0.22
08/09/91	9.67	9.22	0.45	2.83	0.91	0.23	11.52	-1.85	4.58	0.88	0.29
08/19/91	13.75	11.49	2.26	5.53	0.79	0.34	13.47	0.28	3.83	0.89	0.18

TABLE IV
STATISTICAL COMPARISON OF FOUR RAINFALL ALGORITHMS BASED ON TWO-DAY AND FOUR-DAY ACCUMULATIONS. VALUES OF $\overline{RF}_{2d}$, $\overline{RF}_{4d}$, BIAS AND RMSE ARE IN MILLIMETERS

Rainfall	2-Day Accumulation					4-Day Accumulation				
Algorithm	$\overline{RF}_{2d}$	bias	rmse	coef	FSE	$\overline{RF}_{4d}$	bias	rmse	coef	FSE
Z-R Relation	8.93	9.66	12.93	0.86	0.53	16.41	17.93	22.05	0.90	0.52
NN, Z_H Input	18.51	0.08	5.67	0.92	0.20	33.99	0.35	8.04	0.92	0.16
Z_H, Z_{DR}-R Relation	16.92	1.67	6.73	0.87	0.26	31.32	3.02	8.41	0.93	0.17
NN, Z_H, Z_{DR} Input	18.49	0.10	5.85	0.91	0.17	33.93	0.41	7.34	0.94	0.12

[6] as

$$R_{\rm DR} = 0.01 Z_H^{0.91} 10^{-0.385 Z_{\rm DR}} \qquad (14)$$

is used for making comparison with neural network rainfall estimates based on Z_H and $Z_{\rm DR}$.

The four algorithms, namely a) Z-R relation given by (13), b) $(Z_H, Z_{\rm DR})$ based rainfall algorithm given by (14), c) neural network based algorithm using Z_H, and d) neural network based algorithm using Z_H and $Z_{\rm DR}$, are evaluated on common data set in this section. The comparison is done by evaluating the performance for rainfall estimation on four different days. In addition, two-day and four-day accumulation results obtained by the various algorithms are computed. The comparison is made in terms of bias, RMS error, correlation coefficient and FSE.

Table II shows the statistical performance comparison of one-day rainfall accumulation obtained from the Marshall–Palmer Z-R relation and neural network based algorithm using Z_H as input, whereas in Table III the $(Z_H, Z_{\rm DR})$ based estimates are compared with neural network estimates. Note that $\overline{RF}_g$ in Table II and Table III is defined as the daily average raingage measurement over all gage sites, $\overline{RF}_n$ is defined as the daily average rainfall accumulation estimate over all gage sites, and $\overline{RF}_{zr}$ and $\overline{RF}_{dr}$ are similar to $\overline{RF}_n$ except they are computed using Z-R and Z_H, $Z_{\rm DR}$-R relations, respectively. The results of Table II show that the neural network based rainfall estimates using Z_H performs in a fairly consistent manner over the four different days, providing rainfall estimation to an average FSE (over four days) of about 25%. Similarly, the results of Table III show that the neural network based algorithm using Z_H, $Z_{\rm DR}$ produced rainfall estimates to an average FSE (over four days) of about 22%. In addition, neural network based rainfall estimates have smaller biases than those obtained from Z-R or Z_H, $Z_{\rm DR}$-R relation. From Table II the average bias over four days for the Z-R relation estimates is 6.09 mm, whereas the average bias for the neural network based estimates is 0.27 mm. Similarly, in Table III the average bias for the Z_H, $Z_{\rm DR}$-R estimates is 1.26 mm, however the corresponding bias for neural network based estimates is only 0.18 mm.

Table IV summarizes the statistical comparison of the four different algorithms based on two-day and four-day rainfall accumulations. Note that $\overline{RF}_{2d}$ and $\overline{RF}_{4d}$ are defined as the average two-day and four-day rainfall accumulation estimates over all gage sites, respectively. The neural network based algorithm using Z_H estimates two-day accumulation of rainfall to an accuracy of 20% and four-day accumulation to an accuracy of 16%. The corresponding values for the Marshall–Palmer Z-R relation are 53% and 52%. Similarly, the neural network based rainfall algorithm using Z_H and Z_{DR}, estimates two-day accumulations to an accuracy of 17% and four-day accumulations to an accuracy of 12%, whereas the corresponding accuracies for the Z_H, Z_{DR}-R relation are 26% and 17%. The bias values for neural network based estimates are significantly lower than those for Z-R or Z_H, Z_{DR}-R estimates.

It can be observed from Table II to Table IV that the neural network based rainfall algorithms produce more accurate rainfall estimates than the corresponding parametric rainfall algorithms. In summary, neural network based rainfall algorithms perform very well, providing accurate and consistent rainfall estimates. In addition, the estimates have low bias, and are highly correlated with raingage measurements of rainfall. One more observation that can be made from above studies is that utilizing Z_{DR} as an input of neural network improves the accuracy of rainfall estimates. However, a rigorous evaluation of the utility of Z_{DR} in neural network formulation for rainfall estimates is beyond the scope of this paper.

V. Summary and Conclusion

A neural network technique to estimate rainfall based on multiparameter radar measurements is introduced in this paper. The neural network directly maps the radar observations to rainfall on the ground. The training and testing of multilayer perceptrons based on radar and raingage data from four different days during the CaPE experiment demonstrate that the neural network has the ability to generate potentially more accurate and robust rainfall estimates than the existing Z-R or Z_H, Z_{DR}-R relations. In addition, the neural network estimates of rainfall using both Z_H and Z_{DR} are better than the neural network estimates based on Z_H alone, thereby indicating the contribution of Z_{DR} measurement for rainfall estimation in a neural network context.

In this paper we have presented a framework to develop neural network estimates of rainfall. The input data set to our development involved surface distribution of radar observations as well as limited vertical profile. This format is not necessarily the only possibility of inputs to the neural network for rainfall estimation. Potentially several other input data/structures can be used as input for the network. For example, the range to the radar has significant effect on the rainfall estimation process and can be used as an input parameter. Advection of the storm cell can also be accounted for by using radar scans at earlier time intervals. Thus, in principle, the neural network based rainfall estimation provides an easy basis to include a large number of physical factors that affect remote estimation of precipitation using radar, and these issues are currently being investigated by the authors.

Appendix
Recursive Least Square Learning Algorithm [14]

Forward Propagation: For jth neuron in layer i, at iteration n, the input is

$$y_j^{(i)}(n) = \sum_{k=0}^{N_{i-1}-1} w_{k,j}^{(i)}(n) z_k^{(i-1)}(n) \tag{15}$$

where N_{i-1} is the number of neurons in layer $i-1$, and $z_k^{(i-1)}$ is the output of the kth neuron in layer $i-1$, $w_{k,j}^{(i)}$ is the connection weight between the kth neuron in layer $i-1$ and the jth neuron in layer i. Note $i = 1, 2, 3$ represents the first hidden layer, second hidden layer and the output layer, respectively.

Using the threshold-logic (raising ramp) as the activation function, the output of this neuron is

$$z_j^{(i)}(n) = \begin{cases} 0, & y_j^{(i)}(n) \le 0 \\ \frac{1}{a_i} y_j^{(i)}(n), & 0 < y_j^{(i)}(n) < a_i \\ 1, & y_j^{(i)}(n) \ge a_i \end{cases} \tag{16}$$

where $a_i, i = 1, 2, 3$, are the slopes of activation function of each layer.

Backward Error Propagation and Weight Updating:

$$e_k^{(i)}(n) = \frac{1}{a_{i+1}} \sum_{l=0}^{N_{i+1}-1} w_{k,l}^{(i+1)}(n) e_l^{(i+1)}(n) \tag{17}$$

$$d_k^{(i)}(n) = z_k^{(i)}(n) + e_k^{(i)}(n) \tag{18}$$

$$K^{(i)}(n) = \frac{P^{(i)}(n-1) \cdot Z^{(i-1)}(n)}{\lambda + (Z^{(i-1)}(n))^T \cdot P^{(i)}(n-1) \cdot Z^{(i-1)}(n)} \tag{19}$$

$$P^{(i)}(n) = \frac{1}{\lambda}[I - K^{(i)}(n) \cdot (Z^{(i-1)}(n))^T] \cdot P^{(i)}(n-1) \tag{20}$$

$$\begin{aligned} W_k^{(i)}(n) = W_k^{(i)}(n-1) + K^{(i)}(n) \\ \times \left[d_k^{(i)}(n) - \frac{1}{a_i}(Z^{(i-1)}(n))^T \cdot W_k^{(i)}(n-1) \right] \end{aligned} \tag{21}$$

where $e_k^{(i)}$ is the backward propagated error at the kth neuron of layer i from layer $i+1$, $d_k^{(i)}$ is the desired output of this neuron. $Z^{(i-1)} = (z_1^{(i-1)}, z_2^{(i-1)}, \ldots, z_{N_{i-1}}^{(i-1)})^T$, is the output vector of layer $i-1$. $P^{(i)}$ is a matrix of size $N_{i-1} \times N_{i-1}$. $K^{(i)}$ is the gain vector of size $N_{i-1} \times 1$. $W_k^{(i)} = (w_{1,k}^{(i)}, w_{2,k}^{(i)}, \ldots, w_{N_{i-1},k}^{(i)})^T$, $k = 1, 2, \ldots, N_i$, are the weight vectors between layer $i-1$ and layer i.

Initialization:

$$\begin{aligned} w_{k,l}^{(i)}(0) &\in [-0.5, 0.5] \\ P^{(i)} &= 0.5I \\ a_i &= 1.0. \end{aligned}$$

Choose λ, the "forgetting factor," close to but less than 1.0.

Acknowledgment

The authors would like to thank Prof. V. N. Bringi, Dr. Gorgucci, and Prof. W. Krajewski for many helpful discussions.

REFERENCES

[1] L. J. Battan, *Radar Observation of the Atmosphere*. Chicago, IL: The University of Chicago Press, 1973.

[2] T. A. Seliga and V. N. Bringi, "Potential use of radar differential reflectivity measurements at orthogonal polarizations for measuring precipitation," *J. Appl. Meteor.*, vol. 15, pp. 69–76, 1976.

[3] ——, "Differential reflectivity and differential phase shift: Application in radar meteorology," *Radio Sci.*, vol. 13, pp. 271–275, 1978.

[4] M. Sachidananda and D. S. Zrnic, "Rain rate estimates from differential polarization measurements," *J. Atmos. Oceanic Technol.*, vol. 4, pp. 588–598, 1987.

[5] K. Aydin, Y. M. Lure, and T. A. Seliga, "Polarimetric radar measurements of rainfall compared with ground-based rain gauges during MAYPOLE'84," *IEEE Trans. Geosci. Remote Sensing*, vol. 28, pp. 443–449, 1990.

[6] E. Gorgucci, V. Chandrasekar, and G. Scarchilli, "Radar and measurement of rainfall during CaPE: 26 July 1991 case study," *J. Appl. Meteor.*, vol. 34, pp. 1570–1577, 1995.

[7] I. I. Zawadzki, "Factors affecting the precision of radar measurements of rain," in *Preprint 22nd Radar Meteorol. Conf.*, Zurich, Switzerland, Amer. Meteor. Soc., 1984, pp. 251–256.

[8] K. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Networks*, vol. 2, pp. 183–192, 1989.

[9] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward network are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[10] R. A. Lippman, "An introduction to computing with neural nets," *IEEE ASSP Mag.*, vol. 4, pp. 4–22, 1987.

[11] G. B. Foote, "Scientific Overview and Operations Plan," National Center for Atmospheric Research, Boulder, CO, p. 145, 1991.

[12] S. Haykin, *Neural Networks*. New York: Macmillan, 1994.

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representation by error propagation," *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: MIT Press, 1986, vol. 1, pp. 318–362.

[14] M. R. Azimi-Sadjadi and R. Liou, "Fast learning process of multilayer neural networks using recursive least squares method," *IEEE Trans. Signal Proc.*, vol. 40, pp. 446–450, Feb. 1992.

[15] S. Shah and Palmieri, "Training the feed-forward networks with the extended Kalman filter," in *Proc. Int. Joint Conf. Neural Network*, San Diego, CA, 1990, vol. 3, pp. 41–46.

[16] J. S. Marshall and W. M. Palmer, "The distribution of raindrops with size," *J. Meteor.*, vol. 10, pp. 25–29, 1948.

Rongrui Xiao (S'94–M'96) was born in Jiangxi, China, in 1964. He received the B.S. and M.S. degrees from the Shanghai Jiao Tong University, Shanghai, China, in 1985 and 1988, respectively, and the Ph.D degree from Colorado State University, Fort Collins, in May 1996, all in electrical engineering.

He was with the Shanghai Broadcasting and TV Technology Research Institute between 1988 and 1991. From 1992 to 1996 he was a graduate Research Assistant in the Department of Electrical Engineering, Colorado State University. He worked as a post-doctoral research fellow in the Radar Lab of the Department of Electrical Engineering at Colorado State University after he graduated. He joined Vexcel Corporation, Boulder, CO, in December 1996. He is currently involved in research and development of advanced SAR processing algorithms and satellite ground station for remote sensing. His research interests include signal processing, image processing, neural networks, pattern recognition, and radar remote sensing.

Dr. Xiao is a member of Sigma Xi.

V. Chandrasekar (S'83–M'87) photograph and biography not available at the time of publication.