INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

DISSERTATION

CLASSIFICATION AND QUANTIFICATION OF HYDROMETEORS BASED ON POLARIMETRIC RADAR MEASUREMENTS: DEVELOPMENT OF FUZZY LOGIC AND NEURO-FUZZY SYSTEMS AND IN-SITU VERIFICATION

Submitted by

Hongping Liu

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

for the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

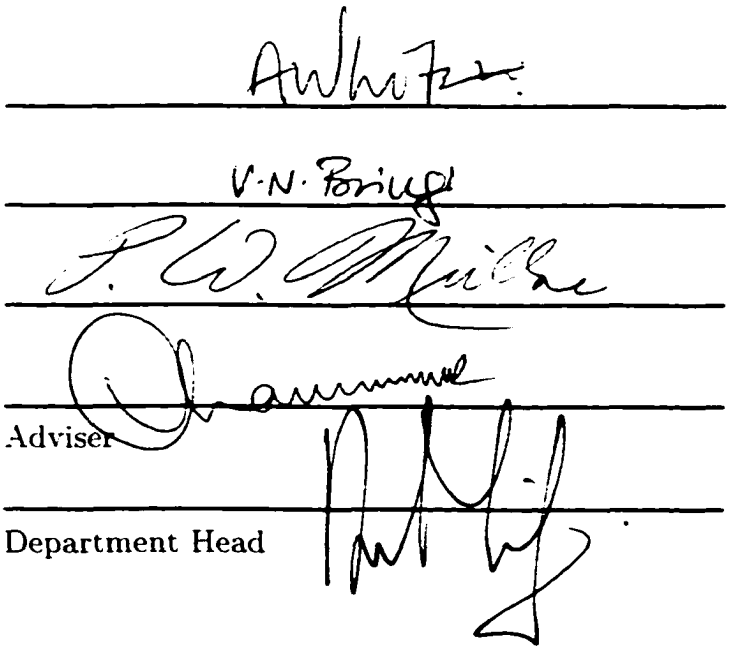
Summer, 2000

COLORADO STATE UNIVERSITY

November 4, 1999

WE HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER OUR SUPERVISION BY HONGPING LIU ENTITLED CLASSIFICATION AND QUANTIFICATION OF HYDROMETEORS BASED ON MULTI-PARAMETER RADAR MEASUREMENTS BE ACCEPTED AS FULFILLING IN PART REQUIREMENTS FOR THE DEGREE OF PH.D DEGREE.

Committee on Graduate Work



Adviser

Department Head

ABSTRACT OF DISSERTATION

CLASSIFICATION AND QUANTIFICATION OF HYDROMETEORS BASED ON POLARIMETRIC RADAR MEASUREMENTS: DEVELOPMENT OF FUZZY LOGIC AND NEURO-FUZZY SYSTEMS AND IN-SITU VERIFICATION

One of the main applications of polarimetric radar is to retrieve the characteristic information about the hydrometeors in the radar resolution volume. The information extraction from polarimetric radar data includes identification of hydrometeor type, quantitative estimation of precipitating hydrometeors, and forecast of hydrometeor evolution. Some Artificial Intelligent (AI) methods, such as fuzzy logic and Neural Network techniques, are proposed in this research to address the issues of hydrometeor classification and quantification.

Fuzzy logic and Neuro-Fuzzy systems for the classification of hydrometeor type based on polarimetric radar measurements is developed. The hydrometeor classification system is implemented by using fuzzy logic and neural network, where the fuzzy logic is used to infer hydrometeor type, and the neural network learning algorithm is used for automatic adjustment of the parameters of the fuzzy sets in the fuzzy logic system. Five radar measurements, namely, horizontal reflectivity (Z_H), differential reflectivity (Z_{DR}), differential propagation phase shift (K_{DP}), correlation coefficient ($\rho_{HV}(0)$), and linear depolarization ratio (LDR), and corresponding altitude have been used as input variables to the hydrometeor

classifier. The output is one of the many possible hydrometeor types, namely 1)drizzle, 2)rain, 3)dry and low density snow, 4)dry and high density crystals, 5)wet and melting snow, 6)dry graupel, 7)wet graupel, 8)small hail, 9)large hail, and 10)mixture of rain and hail. The Neuro-Fuzzy classifier is more advantageous than a simple Neural Network or a fuzzy logic classifier because it is transparent rather than a "black box" (unlike a neural network), and can learn the parameters of the system from the past data (unlike a fuzzy logic system). The Neuro-Fuzzy hydrometeor classifier has been applied to several case studies and the results are compared against in-situ observations.

An adaptive neural network scheme for quantitative precipitation estimation is developed in this research. The neural network is a non-parameteric method for representing the relationship between radar measurements and rainfall rate. The relationship is derived directly from a data set consisting of radar measurements and raingage measurements. The effectiveness of rainfall estimation by using a neural network can be influenced by many factors, such as the representativeness and sufficiency of the training data set, the generalization capability of the network to new data, season change or location change, etc. To achieve the best performance, the neural network may have to be refined to accommodate changes such as change of seasons. In this study, 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 implement the non-stationary relationship between radar measurements and precipitation estimation with change of season and other environment conditions, and also can incorporate new information, without re-training the complete network from the beginning. This precipitation estimation scheme is a good compromise to the dilemma of accuracy and generalization. Data collected by a WSR-88D radar and network of raingages were used to evaluate the performance of the adaptive neural network for

rainfall estimation. It was shown that the adaptive neural network can reach the same estimation accuracy compared with the neural network which is trained with all the available data, but the implementation of the network is much faster, more efficient and convenient for real time rainfall estimation to be used with WSR-88D.

Another important issue for the application of radar rainfall algorithm addressed in this dissertation is the detection of rain/no-rain condition on the ground. A Radial Basis Function (RBF) neural network based scheme for rain/no-rain determination on the ground using radar data is described in this research. Vertical reflectivity profiles of radar observations are used as input variables to the rain/no-rain determination. Radar data and ground raingage measurements are used to train the neural network. This rain/no-rain classifier is evaluated using the radar data collected by the WSR-88D radar over central Florida for two different years. Results indicate that rain/no-rain condition on the ground can be inferred from the procedure developed in this paper fairly accurately. It is shown that by using rain/no-rain classification scheme (prior to radar rainfall estimation) the accuracy of rainfall accumulation estimates can be improved greatly.

Hongping Liu
Department of Electrical and Computer Engineering
Colorado State University
Fort Collins, Colorado 80523
Summer, 2000

ACKNOWLEDGEMENTS

I would like to take this opportunity to express deep gratitude to my adviser Dr. V. Chandrasekar for his inspiring direction, unceasing encouragement and support during the course of my graduate studies. I am also so grateful to Dr. Bringi for his excellent guidance and numerous discussions, he has always been tremendous source of knowledge, wisdom, keen insight to any complicated issues.

I was extremely fortunate to have Dr. Andy White in my committee, who devoted his precious time not only in reviewing the manuscript, but also providing valuable suggestions and corrections, which helped enhance the dissertation greatly. Dr. Andy White deserves my heartfelt gratitude.

I would like to acknowledge Dr. D. Lile and Dr. P. Mielke for serving on my committee.

Additionally, I sincerely acknowledge Dr. J. Hubbert, Dr. J. Beaver, sharing their diverse knowledge and experience; and those fellows in the Electrical Engineering Radar Group at CSU: R. Xiao, GwoJong Huang, L. Zhao, H. Tong, S. Bollen, Y. Cho, G. Xu, Dr. Abou-El-Magd, Boyack and all others, for their direct or indirect involvement to help complete this dissertation.

Finally and foremost, a special thanks to my family, who gave me continuous support and encouragement that kept me constantly motivated towards the completion of my study. My husband, Michael Porter, who patiently withstood my noturnal and weekend working on my research and dissertation and spent enormous amount of time to proofread the manuscript and correct my "Chinese English". I could not have accomplised this without his help and understanding.

This research was supported by the National Science Foundation (ATM-9413453). This research was also supported by CSU-CHILL (supported by NSF/ATM9500108) and the NASA TRMM program.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 General Description of Problem

Polarimetric radar is a useful remote sensing tool for the study of storm microphysics. The polarimetric radar measurements are sensitive to the types, shapes and size distributions, as well as fall behaviors of the hydrometeors in the radar resolution volume. As a result, extensive information about the microphysics of the hydrometeors is contained in the multiparameter radar measurements. Retrieving the microphysical information from polarimetric radar observation has been a topic of active research. Eventually, these microphysical inferences can be utilized to initialize cloud/mesoscale numerical weather prediction models.

Information retrieval from polarimetric radar data can be summarized as 1) identification of hydrometeors (or hydrometeors classification), 2) quantitative estimation of precipitating hydrometeors, and 3) forecast of the hydrometeor evolution. The goal of this study is to develop techniques to classify hydrometeor types and quantify precipitating hydrometeors based on polarimetric radar measurements. Some Artificial Intelligent (AI) methods are proposed in the study to address the issues of hydrometeor type classification and precipitation quantification. Fuzzy logic and neural network techniques are the two main tools used in the research.

In the following sections, some of the relevant background is reviewed first, then the motivation of employing the fuzzy logic technique to classify hydrometeor

types is provided. Subsequently, a review of the present state of quantitative estimation of precipitating hydrometeors is presented. Finally, the research objectives and the outline of the dissertation is presented.

1.2 Hydrometeor Type Classification

Hydrometeor type classification has a wide variety of applications such as interpretation of polarimetric radar data, study of precipitation formation and life cycle, and selection of the right algorithm for precipitation estimation.

1.2.1 Background

The process of hydrometeor type classification from remotely sensed radar data cannot be obtained as a simple look-up-table because the radar signature for different hydrometeors is not mutually exclusive and unique. Current state of the art involves use of several of the polarimetric radar data fields and some prior knowledge to arrive at a decision on the hydrometeor type. This procedure is reliable, if it is done by experts, but is very inefficient, and cannot process large volumes of radar data in reasonable time. Therefore it is important to develop an automatic classification system. Some potential techniques that can be used for automatic hydrometeor classification are a) decision tree method, b) classical statistical decision theory, c) neural network techniques, and d) fuzzy logic. Among these, the fuzzy logic method is perhaps the best suited for the problem at hand as explained in the next section.

Over the last decade, when polarimetric radar observations became more prevalent, many advances were made in the context of hydrometeor identification. Some of these advances are as follows:

(i) Classification using differential reflectivity (Z_{dr})

Since the introduction of Z_{dr} measurement for rainfall estimation applications, it has been used to distinguish between rain and ice, as well as to identify other

targets. For example, Bringi et al. (1984) utilized the differential reflectivity measurements to detect hail, and showed that reliable differentiation between regions of hail and rainfall is possible in convective storms. Hall et al. (1984) used differential reflectivity and reflectivity measurements to identify various hydrometeor types and ground targets.

(ii) Classification using linear depolarization ratio

Bringi et al. (1986) studied the profiles of Z_{dr}, linear depolarization ratio (LDR) and reflectivity at horizontal polarization (Z_h) through the core of convective storms and found that these three measurements are useful in identifying graupel region. At the same time, they found that the vertical structure of Z_{dr} (below the melting level), LDR (above the melting level), and dual-frequency ratio can provide information on the structure and vertical extent of hail sharft.. A fairly detailed study about the vertical profiles of Z_h, Z_{dr} and dual wavelength ratio in terms of the size, shape and fall behavior of the hail-stones were presented by Aydin et al. (1990).

(iii) Classification using specific differential propagation phase (K_{dp}) and copolar correlation coefficient (ρ_{hv})

Zrnic et al. (1993) examined the specific differential phase, the backscatter differential phase (δ), ρ_{hv}, and Z_{dr} observed over a severe hailstorm and demonstrated that these three measurements can be used to detect hail, as well as to identify and quantify mixed-phase hydrometeors.

(iv) Classification using difference reflectivity (Z_{dp})

Tong et al. (1998) used the difference reflectivity to estimate the fraction of rain and ice in mixed phase precipitation. Z_{dp} is the difference reflectivity which can be estimated from Z_h and Z_{dr}.

(v) Empirical algorithm for classifying hydrometeor types

Building on the results of hydrometeor classification in the existing literature, Straka and Zrnic(1993), and Holler(1994) described classification schemes to discriminate the different hydrometeors. They both used the decision tree method, in which predefined boundaries were used to define the decision region.

Thus the polarimetric radar signatures of precipitation have shown the potential for hydrometeor classification and have set the foundation for the development of automatic classification procedures, using advanced techniques such as neural network and fuzzy logic.

1.2.2 Motivation for using fuzzy logic method

There are several methods that potentially can be used for hydrometeor identification, such as decision trees, classical statistical decision theory, neural networks and fuzzy logic. Fuzzy logic is used in this study for classification because it has many inherent advantages over other methods. Many polarimetric radar measurements lie in a limited measurement space for each hydrometeor type. A decision tree method essentially is based on thresholds and boolean logic. The decision tree method is inadequate for the hydrometeor classification problem because the measurement sets for different hydrometeors are not mutually exclusive. It also does not provide allowance for measurement errors. The fuzzy logic system possesses the ability to reach distinct decisions based on overlapping and "noise contaminated" data.

Statistical decision theory is another potential technique that can be considered for the hydrometeor classification problem. However, statistical models are difficult to construct. For example, the statistical model for the rain identification can be expressed as follows,

$$P(C = Rain|Z_h, Z_{dr}, K_{dp}, LDR, \rho_{hv}) = \frac{f(Z_h, Z_{dr}, K_{dp}), LDR, \rho_{hv}|C = Rain) \times p(C = Rain)}{f(Z_h, Z_{dr}, K_{dp}, LDR, \rho_{hv})} \quad (1.1)$$

where,

C is the hydrometeor type;

$P(C = Rain)$ is the prior probability of class rain;

$f(Z_h, Z_{dr}, K_{dp}, LDR, \rho_{hv})$ is the joint probability density of the five polarimetric radar parameters;

$f(Z_h, Z_{dr}, K_{dp}, LDR, \rho_{hv}|C = Rain)$ is the joint probability density function of the five polarimetric radar parameters under the condition of Rain.

It is very difficult to obtain the prior probability and the probability density functions. However, fuzzy logic uses simple rules to describe the system of interest rather than analytical equations, thus it is easy to implement for hydrometeor classification. Based on the above reasons as well as other advantages, such as robustness and speed, fuzzy logic method is the best choice for hydrometeor classification.

1.2.3 Advantages of neuro-fuzzy classification system

In this study, a neuro-fuzzy system instead of a pure fuzzy logic system is also implemented for hydrometeor classification problem. This is because the combined usage of neural network and fuzzy logic enables the system to learn and improve its performance based on past data. Therefore, the neuro-fuzzy system with the learning capability of a neural network and with the advantage of the rule-based fuzzy system can improve the classifier's performance immensely and provides a mechanism to incorporate past observations into the classification process. Unlike the neural network scheme where the training essentially builds the system, in a neuro-fuzzy scheme, the system is built by fuzzy logic definitions and then it is refined using "neural network Type Training." Therefore, extensive training from the beginning is inappropriate for this system. Future in-situ data collection will have a role to play in only refining the system.

1.3 Quantitative Estimation of Precipitation

One of the most important applications of weather radar is the quantitative estimation of precipitation. The main advantage of using radar measurements for precipitation estimation is that radars can obtain measurements over large areas (about 10,000 km^2) with fairly high temporal and spatial resolution. Development of algorithms for remote estimation of precipitation based on radar measurements has been an active research topic for many years. Generally, there are two categories of precipitation: frozen precipitation (also called solid precipitation) and liquid precipitation. The precipitation also may exist as a mixed phase of these two. Basically, precipitation estimation includes liquid water content, ice water content, rainfall rate, snowfall rate and hail-fall rate, etc. The scope of quantitative estimation of precipitation in this research is to investigate a practical and accurate precipitation estimation algorithm.

1.3.1 Background

The problem of rainfall estimation on the ground based on radar measurements is complicated due to the space-time variability of the rainfall field. Marshall and Palmer (1948) found the relationship between rain rate and Drop Size Distribution (DSD) and derived a simple rainfall esimate from radar reflectivity Z (Marshall, 1948). After that, Seliga and Bringi (1976) suggested using differential reflectivity to improve rainfall estimates. Combination of specific differential phase and differential reflectivity could be used to obtain rainfall estimates independent of absolute calibration (Seliga and Bringi, 1976). Humphries (1974) showed that differential phase K_{dp} is almost linearly related to rain rate R and that it is not sensitive to the DSD. Sachidanada and Zrnic (1986) obtained a R(K_{dp}) which is much less sensitive to the DSD. All the proposed algorithms are based on the three radar measurements Z_h, Z_{dr} and K_{dp} which are correlated to the characteristics of the rain

medium. All these algorithms can be categorized as parametric algorithms. The details about these algorithms can be found in Bringi and Chandrasekar (2000).

Apart from polarimetric algorithms, statistical/engineering solutions have also been used for rainfall estimates. Zawadzki (1984) proposed to derive rainfall algorithm using probablity matching techniques. Probability matching method is a statistical procedure to derive the average relation between radar measurement and ground rainfall rate. This method introduced by Calheiros and Zawadzki (1987) and essentially evaluated by Rosenfeld et al () matches the cumulative distribution function of R and Z yielding a $Z - R$ relationship.

Many factors cause bias in radar rainfall estimation such as variation in drop size distribution, radar calibration error, propagation attenuation, and some other measurement error caused by ground clutter, anomalous propagation, beam blockage and presence of hail, etc.. A robust and accurate algorithm is desired for practical application of radar rainfall estimation. Neural network based rainfall algorithm offers immense potential to progress towards the goal.

1.3.2 neural network approach for Precipitation Estimation

Rainfall rates obtained on the ground can be potentially dependent on the four dimensional structure of precipitation aloft (3 spatial dimensions and time). In principle, one can obtain a functional approximation between the rainfall on the ground and the 4-D radar observations aloft. This function will be much more complicated than a simple Z-R algorithm or a polarimetric radar rainfall algorithm. Therefore the ground rainfall estimation can be viewed as a complex function approximation problem. Neural networks are well suited for complex function approximation, and the theoretical basis is provided by the Universal Function Approximation theorem (Fumahashi, 1989). Recent research has shown that neural network techniques can be used successfully for ground rainfall estimation from radars (Xiao and Chandrasekar, 1997). This technique includes two

stages, a training stage and an application stage. In the training stage, the neural network learns the potential relationship between rainfall rate and the radar measurements from a training data set. When a radar measurement vector (several radar measurements as components) is applied to the neural network, the network yields a rainfall rate estimate as output. This output is compared against the raingage 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 finished, a relationship between rainfall rate and radar measurements is established and the network is ready for operation. Subsequently, a radar measurement vector is applied to the network, and 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 data set, and therefore it is not influenced by the radar system errors. The neural network can be tuned very well for one specific storm or several storms. Once the neural network is trained, it represents a relationship between radar measurements and rainfall rate. If the training data set is large enough and representative enough, the neural network can perform very well.

1.3.3 Adaptive neural network Scheme for Precipitation Estimation

There is a common limitation with respect to the trade-off between generalization and accuracy of neural network based rainfall estimation. For example, a neural network based rainfall estimate, constructed such that it works for summer storms as well as winter rainfall may not be very accurate for one single season. Therefore, developing a flexible network instead of a fixed network for rainfall estimation is better. A neural network can learn its structure and parameters automatically from training data. One way to solve the problem is to collect new data

and re-train the neural network all over again from the beginning. However, the training process is very tedious and time-consuming, and to re-start the training all over is not a practical solution. Our goal of the study is to develop an adaptive neural network, which can continuously update the structure by incorporating the latest information into an existing neural network without having to re-train from the beginning. Therefore, the network will have the "dynamic" characteristic, and it can fine-tune the functional mapping over time.

In this dissertation, an adaptive neural network scheme has been developed that can be fine-tuned continuously. For this purpose, Radial Basis Function (RBF) neural network is chosen because its architecture is well suited for adaptive modification.

1.4 Objectives of the Research

In this research, a fuzzy logic scheme is proposed and developed to address the problem of "hydrometeor type classification." Once the fuzzy logic classifier has incorporated the existing expert knowledge about the polarimetric signatures of the different hydrometeors, it can make fast and fairly reasonable decision on the hydrometeor types according to the inputs of several polarimetric data fields. In addition, a more advanced system called a neuro-fuzzy classifier is proposed to improve the performance of the fuzzy logic classifier. The neuro-fuzzy classifier combines neural network techniques and fuzzy logic functions. It is a fuzzy logic system in the sense of structure and inference, where the neural network learning algorithm is embedded in the automatic adjustment of the parameters of the fuzzy logic structure. This architecture enables the system to have the capability of learning from the data and to adjust its parameters in the fuzzification blocks.

Subsequently, "quantitative estimation of precipitating hydrometeors," is addressed by using neural network techniques. Based on the framework developed

by Xiao and Chandrasekar (Xiao, 1997), the issues involved with neural network radar rainfall estimation are investigated further. The focus of the research is to develop an adaptive neural network scheme for radar rainfall estimation.

1.5 Organization of the Dissertation

Following the introduction in Chapter 1, the organization of this dissertation is as follows:

- Chapter 2 introduces the two main radar systems used in this research, namely CSU-CHILL and WSR-88D (also called NEXRAD). Chapter 2 also reviews the fundamentals of neural networks and the application of neural networks in function approximation. The last section of this chapter reviews the background of fuzzy sets and fuzzy logic relevant to this research.

- Chapter 3 describes the development of a fuzzy logic system for hydrometeor classification. Subsequently, a neuro-fuzzy system for hydrometeor classification and its learning algorithm are presented. The main purpose of the neuro-fuzzy system is to adjust the parameters of the fuzzy logic classifier by, incorporating the neural network learning capability, so that it can adjust its membership function according to in-situ verification information.

- Chapter 4 presents the in-situ verification of the hydrometeor classification technique developed in this dissertation.

- Chapter 5 introduces a classification scheme for rain/no-rain condition on the ground based on the vertical reflectivity profiles of the radar observations. This rain/no-rain classifier is evaluated using the radar data collected by WSR-88D radar over central Florida for two different years. The impact of using this classification scheme prior to applying any rainfall estimation algorithm is also evaluated in this chapter.

- Chapter 6 discusses several important issues for successful application of neural networks for rainfall estimation. An adaptive neural network scheme for radar rainfall estimation is also described.

- Chapter 7 presents the applications of the adaptive RRN (Radar Rainfall neural network) to several case studies.

- Chapter 8 summarizes the important results of this dissertation and provides suggestions for future research.

Chapter 2

THEORY AND BACKGROUND

2.1 CSU-CHILL and WSR-88D radars

The two main radars which collected the data for the research described in this dissertation are the CSU-CHILL and the WSR-88D radar systems. The purpose of this chapter is not to describe the two radars in detail. It just focuses on the introduction of the signal flow, so as to understand system error and data quality. The quality of data plays an important role in the determination of the accuracy of rainfall estimation and in the explanation of the hydrometeor classification results.

2.1.1 General Description of CSU-CHILL radar

The CSU-CHILL radar is an S-band dual-polarization pulsed-Doppler radar, which is currently located at Greeley, Colorado, and is operated by Colorado State University. In this pulsed-Doppler radar, a pulse modulator is used to alternately switch on and off a high power amplifier to generate a train of pulses at microwave frequencies. The pulses of eletromagnetic energy are then transmitted via the antenna in a narrow beam in space. As the RF pulses impinge on objects in space, the incident energy contained in the pulse is scattered in many directions. That portion of the scattered field which is incident on the radar antenna during the receive time (governed by the transmit-receive(T/R) switch) constitutes the received signal and is applied to the receiver for frequency down-conversion and subsequently to the signal and data processors for display.

The CSU-CHILL radar is a dual-polarization radar having two transmitters and two receivers. This configuration permits the measurement of the polarimetric covariance matrix in the horizontal/vertical polarization basis. Its antenna has very good matched beams between horizontal (H) and vertical (V) polarization states, and it also has very good isolation between H and V channels. The two identical transmitters drive the antenna's H and V ports, and nearly identical two receivers handle the co-polarized and cross-polarized signal returns. The received signals are processed in real time, and are presented on an interactively controlled color display.

2.1.1.1 The signal/data flow in CSU-CHILL radar

(1) Transmit path

The transmitters generate, amplify, and modulate an RF signal and feed it to the antenna through H/V polarization channels. One of the component of the transmitter is a stable local oscillator (STALO), which is a crystal controlled device for generating a sinusoidal signal with frequency of 2.785 GHz. This signal is then mixed with an intermediate frequency (60 MHz) signal (IF) from a coherent oscillator (COHO). After that, the signal is sent to two stage amplifiers, the first one is a 30 dB intermediate amplifier (IPA) and the second stage is a 55 dB high power Klystron transmitter amplifier (VA87B). This amplified RF signal will be fed to the antenna through a waveguide. The antenna used in this radar system is a parabolic dish with 8.5 m diameter. The antenna can concentrate the transmitting power to a certain direction, so that the transmitted eletromagnetic wave can pinpoint to a specific direction. The signal flow in the forward path is shown in figure 2.1 as form of block diagram.

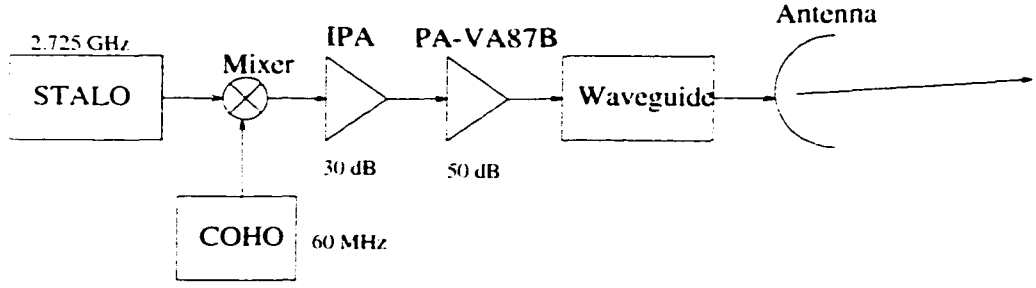


Figure 2.1: The transmitting signal flow path of CSU-CHILL Radar

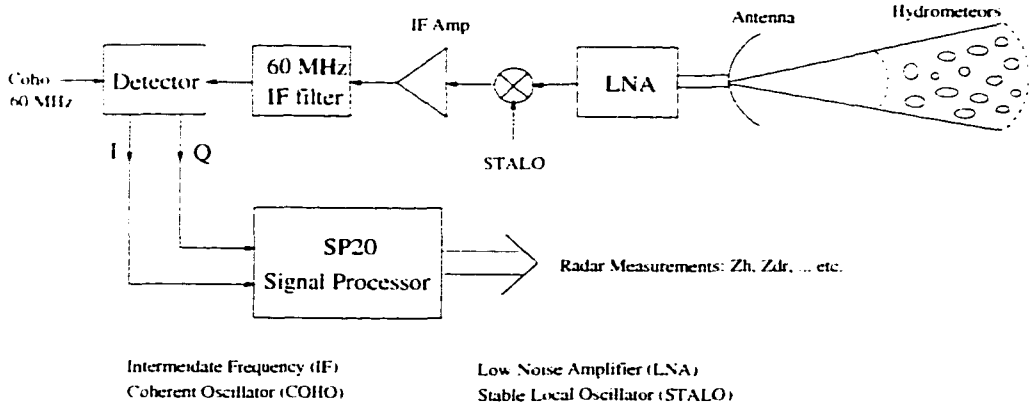


Figure 2.2: The receiving signal flow path of CSU-CHILL Radar

(2) Signal flow in the receive path

Once the transmitted electromagnetic wave hits hydrometers, a small portion of it gets backscattered and travels back to radar. The receiver can detect, amplify, filter, demodulate the signal, and then send the signal to a signal processor to calculate the polarimetric radar parameters. Figure 2.2 shows the diagram of the signal flow during this process. The return signal from a single resolution volume consists of the contributions from each of the scatterers within the resolution volume. The number of scatters within the resolution cell is dependent on the concentration as well as resolution cell volume.

The received signals are first amplified by a 30 dB low noise amplifier(LNA). The output of the LNA is filtered by a band-pass filter. This signal then is mixed

with STALO and the result an intermediate frequency signal is at 60 MHz. From this point, the signals are separated into two paths. The first one is sent into the log receiver and digitized by the signal processor to set the gain of the linear channel on a gate-by-gate. The other one is delayed by approximately 350 ns to allow the step attenuators to be set for each gate based on the sample from the log receiver. The linear signal passes through a 700 kHz bandwidth matched filter. These procedures prevent saturation of the next IF stages and allow for around 90 dB dynamic range in the linear receivers. After the signals are amplified by around 55 dB, it is sent into quadrature detectors to provide in-phase(I) and quadrature(Q) base-band video signal for each channel. These I and Q signals along with the log receiver signal are applied to a signal processor for computing the polarimetric radar parameters.

2.1.1.2 System Characteristics of the CSU-CHILL radar

The block diagram of the CSU-CHILL radar is shown in Figure 2.1.1.2. The important characteristics of the CSU-CHILL radar relevant to this study are given in table 2.1.

2.1.1.3 Polarimetric Measurements of the CSU-CHILL radar

The CSU-CHILL radar can provide a complete set of polarization measurements. The available radar parameters include received power (dBm), reflectivity (dBZ), differential reflectivity (Z_{DR}), normalized coherent power (NCP), differential propagation phased (ϕ_{DP}), radial velocity (pulse pair method), zero lag HV correlation ($\rho_{HV}(0)$), spectral width (second lag method), linear depolarization

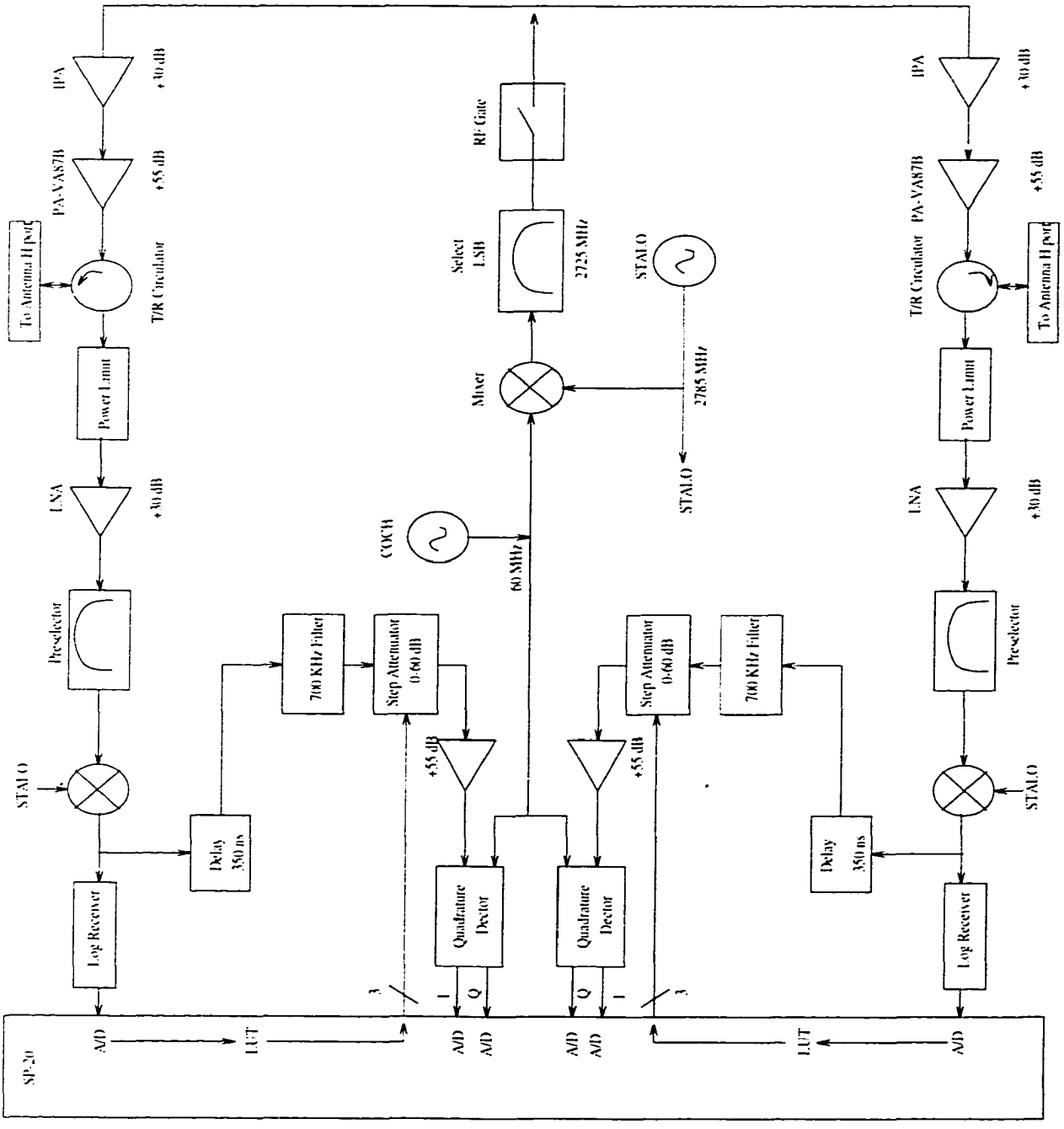


Figure 2.3: The Block Diagram of CSU-CHILL Radar

Table 2.1: CSU-CHILL Radar Characteristics

Antenna	
Type	fully steerable , prime focus parabolic reflector
Size	8.5 m
Feed	scalar horn
Half power beam width	1.0^o
Gain (directivity)	45 dB
Side lobe level	$\geq$ -27 dB
Cross-pol. level	$\leq$ -35 dB
Polarization radiated	Horizontal or Vertical
Scan type	PPI , RHI , sector scan
Transmitter	
Type	klystron , modernized FPS-18
Wavelength	10.7 cm
Peak Power	700 $\sim$ 1000 KW
Pulse Width	0.1 $\sim$ 1.0 μs in steps of 0.1 μs
PRT	800 $\sim$ 2500 μs
Max. unambigu.range	375 Km
Max. unambigu. vel.	$\pm$ 34.4 ms^{-1}
Receiver	
Noise figure	$\sim$ 4.3 dB
Transfer function	linear
Dynamic range	> 85 dB
Min. detectable signal	-114 dBm (SNR = 0 dB)
Data Acquisition	
Signal processor	SP20 made by Lassen Research
Number of range gates (bins)	variable 64 - 2048
Range gate (bin) spacing	0.2 μs or 1.0 μs
sampling rate/avg. opt.	under micro-code control
Video digitizer	12-bit , in the SP20 input card for I, Q and log P
Time series capability	up to 150 gates with continuous recording .
Variables Available	
Reflectivity at H and V polarizations .	
Mean Doppler Velocity (v) and Spectral Width (σ_v)	
Differential Phase between H and V states (ϕ_{dp})	
Copolar Correlation Coefficient ($\rho_{hv}(0)$)	
Doppler Spectra from FFT processing	
I , Q and log P for every pulse in time series mode	

ratio (LDR), unprocessed complex voltage (I, Q), complex correlations (RHV1, etc.).

A. Definition of the radar parameters

The following five radar parameters play an important role in the research of hydrometer classification and rainfall estimation.

(1) Reflectivity factor at horizontal polarization is given by

$$Z_h = (4\lambda^4/\pi^4|K_w|^2) < |s_{hh}|^2 > \tag{2.1}$$

where λ is the wavelength of the radar: $< |s_{hh}|^2 >$ is a term in the backscattering covariance matrix, which is proportional to the power received at H-port when transmitting wave is horizontally polarized: $K_w = \frac{\epsilon_r - 1}{\epsilon_r + 2}$, ϵ_r is dielectric constant. From this definition, we can conclude that Z_h is proportional to the received power at h-port, which is proportional to the mean cross-section of the hydrometeors in the radar scan volume.

(2) Differential Reflectivity Z_{dr} can be defined as,

$$Z_{dr} = 10log(< |s_{hh}|^2 > / < |s_{vv}|^2 >) \tag{2.2}$$

where $< |s_{vv}|^2 >$ is a term in the backscattering covariance matrix, which is proportional to the power received at V port when transmitting wave is vertically polarized:

Z_{dr} is related to the difference between Z_h and Z_v, where Z_v is the reflectivity factor at vertical polarization. Z_{dr} is a very reliable polarimetric parameter, which is a good indicator of shape and orientation of precipitation particle.

(3) Differential Propagation Phase Shift K_{dp} can be defined as,

$$K_{dp} = K_{eff}^h - K_{eff}^v \tag{2.3}$$

where K_{eff}^{h} is the effective propagation constant for horizontally polarized waves. K_{eff}^{v} is the effective propagation constant for vertically polarized waves. then K_{dp} is the difference between these two propagation constants. K_{dp} is zero for isotropic hydrometeors and K_{dp} is non-zero for anisotropic hydrometeors.

(4) Correlation coefficient at zero lag ρ_{hv} is defined as:

$$\rho_{hv}(0) = \frac{< S_{vv} S_{hh}^{*} >}{< |S_{vv}|^2 >^{1/2} < |S_{hh}|^2 >^{1/2}} \tag{2.4}$$

The correlation coefficient between horizontally and vertically polarized echoes ($\rho_{hv}(0)$) is affected mainly by the variability in the ratio of the vertical-to-horizontal size of individual hydrometeors and depends on the orientation and shape (e.g., irregular shape) of hydrometeors.

(5) Linear depolarization ratio is defined as:

$$LDR_{hv} = 10log(< |s_{hv}|^2 > / < |s_{vv}|^2 >) \tag{2.5}$$

or

$$LDR_{vh} = 10log(< |s_{vh}|^2 > / < |s_{hh}|^2 >) \tag{2.6}$$

If a horizontally polarized wave is transmitted. The backscatter from hydrometeors predominently horizontal polarized. But a small portion of vertically polarized waves also will return to the radar. This can occur because factors such as hydrometeor shape, oblateness, radar view angle and canting with respect to axis of polarization (Herzegh and Jameson 1992) depolarize some of the transmitted engery (Herzegh and Jameson 1992). The ratio of the received cross-polar power to the transmitted co-polar power defines the Linear Depolarization Ratio (LDR_{hv}) and this depends on the dielectric constant. Detailed discussions of these

parameters can be found in Bringi and Chandra (2000) and Doviak and Zrnic (1993).

B. The errors in the radar parameters

The quality of the radar parameters influences the accuracy of the inference of hydrometeor types and rainfall estimation directly. Many factors contribute errors to the measured radar parameters, such as radar system error, clutter from the environment, anomalous propagation, 3-body effect, etc..

The separate waveguides to the antenna ports, i.e., one for H and one for V polarized channel cause a differential gain between the two receivers due to the different waveguide loss factors. This will result in the deviation of the measured radar parameters from the intrinsic radar parameters. The differential gain of the two receivers can be calibrated by using Sun as a randomly polarized exciting source. This is based on the fact that Sun emits radiation with random polarization which equally excite the two parts. Therefore, LDR and Z_{DR} can be calibrated by this method for details refer to Huang (1997).

The environment includes ground clutter, atmospheric clutter, and other sources of interference. Clutter can contaminate the radar data. Presence of clutter influences the range fluctuations in ϕ_{DP}.

Z_{DR} can be further calibrated by using a Z_{DR_bias} which is determined by pointing the antenna vertically (elevation angle = 90^o) and changing the azimuth angle from 0^o to 360^o. Ground-clutter can at times cause a dependency of Z_{DR} on azimuth angle but this should be averaged out. This calibration is based on the assumption that the ice particles are oriented randomly (Hubbert, 1998).

When the back-scattered signal received by the radar is not scattered back directly from a hydrometeor, it is scattered back from the radar likely to the ground, then from ground to hydrometeors, finally from hydrometeors to the radar.

This is called the three-body effect. Three-body scatter typically causes differential reflectivity to be very high at high elevations and to be negative at lower elevations at the rear of the storm core (Hubbert, 1997).

The path of radar signals in the atmosphere depends on the variation of the atmosphere's refractive index with height. The refractive index is a function of temperature, pressure, and water vapor content. Under the conditions of temperature inversion in the lower atmosphere, it is possible that the radar beam hits the ground and returns a false echo. This situation is called anomalous propagation (Doviak and Zrnic, 1993).

2.1.2 General Description of WSR-88D radar

The WSR-88D (Weather Service Radar, 1988-Doppler) radar is a single polarization radar. It measures reflectivity and Doppler velocity. The operational frequency is 2.7 to 3.0 Gigahertz. The WSR-88D transmitter generates and radiates a peak power output of 750 kilowatts. The transmitter is of Klystron type, so it is easy to control the waveform of its transmitted pulses. The antenna is a 28 foot parabolic dish that directs the radar signal. The angular width of the antenna pattern (beam-width) is about 0.96 degrees. The receiver receives and amplifies the returned radio frequency signal and sends the signal to the signal processor. The signal processor transforms the analog signal, that was sent from the receiver, into digital base data. It performs clutter suppression and other signal conditioning. The WSR-88D radar can work in one of the two operational modes: precipitation mode and clear air mode. In precipitation mode, it can scans either 14 elevations in 5 minutes or 9 elevations in 6 minutes. In the clear air mode, it scans 5 elevations in 10 minutes.

The system characteristics of the WSR-88D radar are listed in table 2.2.

Table 2.2: General WSR-88D Radar Characteristics

Antenna	
Antenna size	8.53 m
Beam width	$0.88 - 0.96^{o}$
Gain	39.1 dB
Azimuth speed	0-3 RPM
Elevation speed	0-15 deg/sec
Waveguide loss	Variable
Transmitter	
Wave length	10.0 - 11.1 cm
Peak power	750 KW (peak) 1.5 kw (Avg)
Pulse width	1.5 - 4.5 ms
Polarization	Single, Horizontal
Doppler capability	Yes
Velocity unfolding	+/- 50 m/s
Unambiguous range	460 km
PRF	320 - 1300 Hz
RF frequency	2.7 - 3.0 GHz
Sensitivity	10 dBZe for a 0 SNR at 50 km
Clutter cancellation	56 dB

2.2 Background of neural network

Neural networks are information processing systems which consists of a large number of highly interconnected processing units called neurons. Neural networks are designed to mimic the way of human brain functions, and they are able to learn from their environment and store the learned information in the form of interconnecting weights. Since the 1980's, neural networks have become popular and have been applied successfully in almost all fields of scientific research and industry. The main applications are function approximation, pattern classification/recognition, and time series prediction. In this dissertation, neural network techniques will be used to implement radar rainfall estimation, which is a function approximation problem. Neural network learning algorithms also are used in the fuzzy logic classification system to fine-tune the parameters of the system.

Of the various neural network architectures, two are very suitable for function approximation, namely, multilayer perceptron neural network (MLPNN) and radial basis function neural networks (RBF). These two neural network structures and learning algorithms will be introduced in the following section. Both MLPNN and RBF also are good at applications of classification. There are two more other neural networks which fit for classification application: One is the probability neural network; the other is Kohonen neural network. Generally, the combination of Kohonen neural network and Learning Vector Quantization (LVQ) can perform very well in the applications of pattern classification.

2.2.1 Multilayer Perceptron Neural Network (MLPNN)

2.2.1.1 MLPNN architecture

The basic element in a neural network is the **neuron**. The modeling of a neuron in the MLPNN is shown in the Figure 2.4. In this figure, $[x_1 \quad x_2 \quad \ldots$

$x_n]$ stands for the n inputs of the neuron; in the neuron, the weighted inputs combined with bias b are summed up to obtain net input h first, then go through a activation function block to produce output y. The input/output relation can be expressed as,

$$h = \mathbf{w}^T\mathbf{x} + b \tag{2.7}$$

$$Y = \psi(h) \tag{2.8}$$

where the superscript T denotes the transpose operation, $\mathbf{x} = [x_1 \ x_2 \ \ldots \ x_n]$is a n by 1 input vector, $\mathbf{w} = [w_1 \ w_2 \ \ldots \ w_n]$ is a n by 1 weight vector, b is a bias, and ψ is an activation function. The common forms of the activation function are hard limit, sigmoid, tangent and linear function (Haykin, 1997).

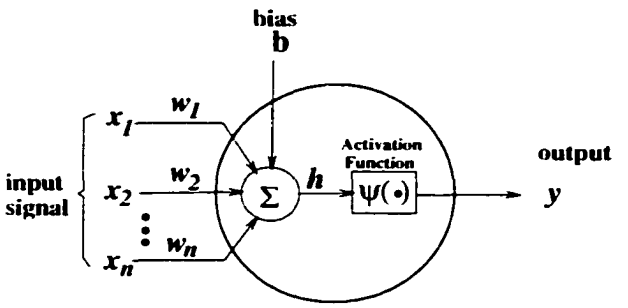


Figure 2.4: Modeling of a neuron: w_1, w_2, w_n are the synaptic weights, b_k is the bias, $\psi(.)$ is the activation function.

A typical configuration of a multilayer perceptron neural network is shown in the Figure 2.5. It consists of one input layer, one output layer, and one or more hidden layers in between. Each layer contains a number of neurons. The input layer consists of n inputs, $[x_1 \quad x_2 \quad \ldots \quad x_n]$; the output layer has only one output Y(it may have several outputs, generally); there are three hidden layers in the figure (assume that the ith hidden layer has n_i neurons). The inputs are fully connected to the first hidden layer. Each hidden layer is fully connected to the next, and the last hidden layer is fully connected to the outputs. The net input

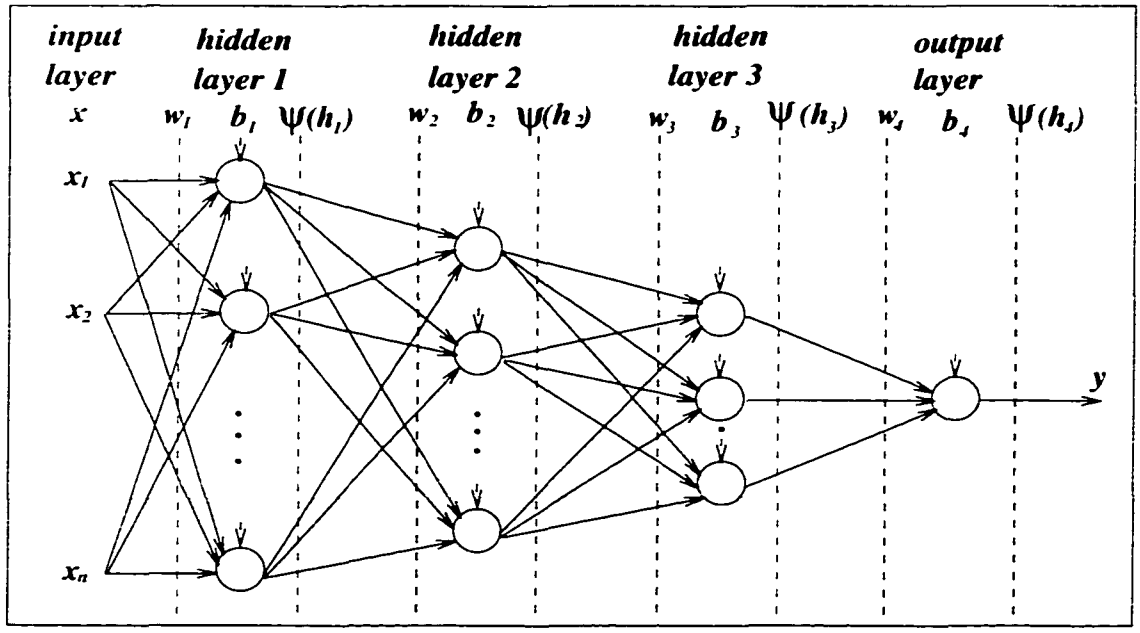


Figure 2.5: Typical configuration of a multilayer perceptron neural network

to neurons in the first hidden layer can be expressed as shown in the following equation.

$$h_1 = \mathbf{w_1 x} + b_1 \tag{2.9}$$

where $b_1 = [b_{11}\ b_{12}\ \ldots\ b_{1n_1}]$ is a n_1 by 1 bias vector, $h_1 = [h_{11}\ h_{12}\ \ldots\ h_{1n_1}]$ is a n_1 by 1 activation vector, and $\mathbf{w_1} = [w_{11}\ w_{12}\ \ldots\ w_{1n_1}]$ is a n_1 by n weight matrix, while the 1 by n weight vector at layer *1* is defined as $w_{1j} = [w_{1j1}\ w_{1j2}\ \ldots\ w_{1jn}]$ for j = 1, 2, ... n_1.

Therefore, the output of the neurons in the first layer is $\psi(h_1)$. Similarly, the net inputs to all the neurons in the following layers can be obtained by the following equation,

$$h_i = \mathbf{w_i}\psi(\mathbf{h_{i-1}}) + b_i \tag{2.10}$$

where $b_i = [b_{i1}\ b_{i2}\ \ldots\ b_{in_i}]$ is a n_i by 1 bias vector, $h_i = [h_{i1}\ h_{i2}\ \ldots\ h_{in_1}]$ is a n_i by 1 activation vector, and $\mathbf{w_i} = [w_{i1}\ w_{i2}\ \ldots\ w_{in_i}]$ is a n_i by n_{i-1} weight matrix.

while the 1 by n_{i-1} weight vector at layer n_i is defined as $w_{ij} = [w_{ij1}\ w_{ij2}\ \ldots\ w_{ijn_{i-1}}]$ for j = 1, 2, ... n_i.

Some notations used in above two equations are defined as

b_{ij} : bias of the jth node at layer i

h_{ij}: net input of the jth node at layer i

$o(h_{ij})$: output of the jth node at layer i

w_{ijk}: weight connected between the jth node at layer i and the kth node at layer i-1

2.2.1.2 Back-Propagation Learning Algorithm

The back-propagation learning algorithm is a gradient descent learning algorithm is designed to minimize the error function iteratively. In the MLP neural network, the error function can be defined as,

$$e = \sum_{j=0}^{N} e^j \tag{2.11}$$

where N is the total number of training patterns, all training patterns are applied to the network sequentially. e^j is the mean square error between the desired output t^j and the actual neural network output Y^j with respect to the jth training pattern. e^j can be expressed as

$$e^j = \frac{1}{2}(t^j - Y^j)^T(t^j - Y^j). \tag{2.12}$$

After the jth training pattern is applied to the neural network, weights and bias at each layer are updated according to the following rules,

$$\mathbf{w}_i^{j+1} = \mathbf{w}_i^j + \mu^j \triangle \mathbf{w}_i \tag{2.13}$$

$$b_i^{j+1} = b_i^j + \mu^j \triangle b_i \tag{2.14}$$

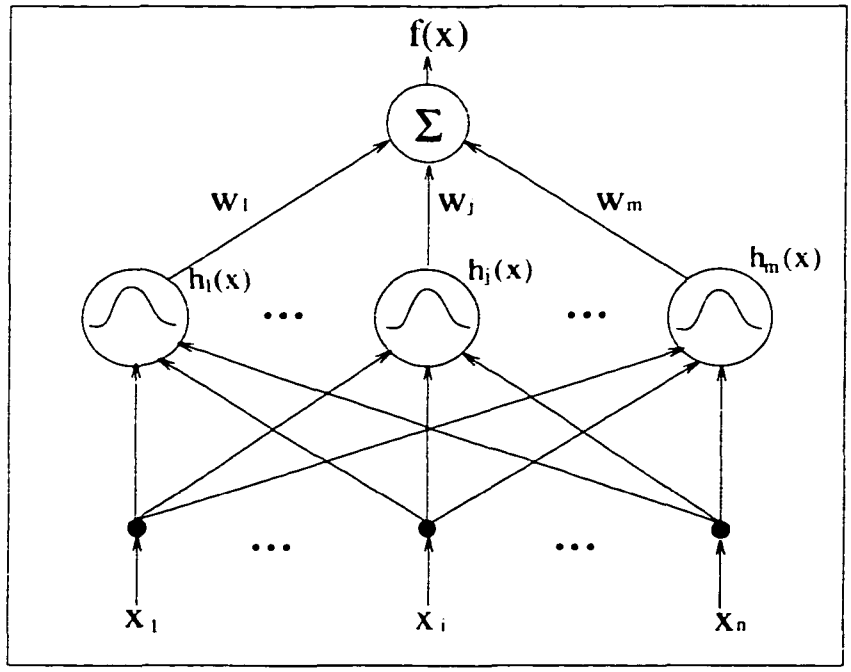


Figure 2.6: Architecture of a typical RBF neural network

where μ^l is the learning rate, $\triangle \mathbf{w}_i$ is the derivative of error function with respect to the weight vector $\mathbf{w}_i$, and $\triangle b_i$ is the derivative of error function with respect to the bias vector b_i.

The parameters of the MLPNN are updated iteratively. When the network converges, it reaches the global minimum on the error surface. Then the learning process is finished, and it can be used for generalization for those input patterns which are not in the training set.

2.2.2 Radial Basis Function Neural Network

2.2.2.1 RBF Network Architecture and Parameters

A radial basis function network has three layers (Figure 2.6), namely, (i) input layer, consisting of input variables $x_1, x_2, \ldots, x_n$, (ii) hidden layer, consisting neurons with radial basis function as transfer function $h_j(\vec{x})$, (iii) output layer, consisting output variables, which are linear combinations of hidden layer output.

The Gaussian radial basis function is used in this study, which can be expressed as

$$h_j(\vec{x}) = exp(-\sum_{i=1}^{n} \frac{(x_i - c_{ij})^2}{r_{ij}^2}) \tag{2.15}$$

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

$$f(\vec{x}) = \sum_{j=1}^{m} w_j h_j(\vec{x}). \tag{2.16}$$

where $\vec{x} = [x_1\ x_2\ ...\ x_n]$ is the input vector, $\vec{c}_j = [c_{1j}\ c_{2j}\ ...\ c_{nj}]$ is the center vector of neuron j, $\vec{r}_j = [r_{1j}\ r_{2j}\ ...\ r_{nj}]$ 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.

The hidden layer is an important part of the network, since it directly determines the accuracy of the network. The radial basis function network is a modified form of exact interpolation. From equation 2.16 it can be seen that the RBF network performs a linear superposition of the localized basis function ($h_1(x)$, ... , $h_m(x)$), where the accuracy of the network function depends on the number of the basis functions, the centers and the widths of the basis functions.

For this RBF neural network, the following three parameters need to be determined. (i) *center vector* of all the neurons in the hidden layer $\vec{c}_j$ (ii) *width vector* of all the neurons in the hidden layer $\vec{r}_j$ (iii) *weights* from the hidden layer to the output (w_1 ... w_m) Once all these parameters are determined, the network is complete, and it can be used for application. If an input vector $\vec{x}$ 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 2.15), and takes the value 1 (maximum) when the input vector $\vec{x}$ is equal to the center vector. As the distance increases, the output decreases. Linear combination of the outputs from all the hidden units is the final output.

2.2.2.2 Learning Algorithm for RBF Neural Network

The training process for a RBF neural network involves selecting a set of radial basis functions for the hidden layer neurons and finding the optimum weight vector. Generally, there exist three categories of learning algorithm for determining all the free parameters in the RBF neural network, namely, *fixed centers selected at random*, *recursive hybrid learning procedure*, and *stochastic gradient approach*.

(i) fixed centers selected at random

The simplest approach for choosing radial basis functions for the hidden layer neurons is to pick up a fixed number of *centers* $\vec{c}_j$ from the training data set and set the *widths* r_{ij} a constant r. The constant r is fixed at the common value

$$r = \frac{d_{max}}{\sqrt{K}}, \tag{2.17}$$

where K is the number of centers, and d_{max} is the maximum distance between the chosen centers.

If the *centers* and *widths* of the radial basis functions are determined, then the determination of weights w_j is very straightforward. w_j are determined by minimizing the sum of squared error ϵ given by

$$\epsilon = \sum_{i=1}^{p} (\hat{y}_i - \sum_{j=1}^{m} w_j h_j(\vec{x}))^2 \tag{2.18}$$

The optimum w_j is given by the generalized inverse equation

$$\hat{w} = [\hat{w}_1 \ \hat{w}_2 \ ... \ \hat{w}_m]^T = (H^T H)^{-1} H^T \hat{Y} \tag{2.19}$$

where H is the matrix of basis functions given by (Mark, 1996)

$$H = \begin{bmatrix} h_1(\vec{x}_1) & h_2(\vec{x}_1) & ... & h_m(\vec{x}_1) \\ h_1(\vec{x}_2) & h_2(\vec{x}_2) & ... & h_m(\vec{x}_2) \\ \vdots & \vdots & ... & \vdots \\ h_1(\vec{x}_p) & h_2(\vec{x}_p) & ... & h_m(\vec{x}_p) \end{bmatrix} \tag{2.20}$$

Alternatives to the method by equation 2.19 used to compute the weight vector are the Least-Mean-Square (LMS) algorithm (Haykin, 1996) and the Recursive-Least-Square (RLS) algorithm (Haykin, 1996).

(ii) recursive hybrid learning procedure

One of the limitations of the fixed centers method is that it requires a large dataset for good performance. Another learning method is called *recursive hybrid learning procedure*, in which the *centers* of the radial basis functions are selected from training data set by using self-organized learning algorithm. The commonly used one is called *k-means clustering algorithm* (Duda and Hart, 1973), in which the input data are partitioned into k clusters, the *centers* of the radial basis functions are placed in the centers of the k clusters. The *widths* r_{ij} are obtained by equation 2.17. The *weights* are calculated by equation 2.19, or, they can also be obtained by using LMS and RLS algorithms.

To avoid getting stuck in local optimum, *enhanced k-means clustering algorithm* was proposed by Chen (1995).

(iii) stochastic gradient approach

Unlike the fixed centers method and recursive hybrid learning procedure, the stochastic gradient approach aims to find the free parameters ($\vec{c}_j$, r_{ij}, and w_j) by supervised learning process based on the error defined as

$$\epsilon = \sum_{i=1}^{p} (\hat{y}_i - f(\vec{x}_i))^2. \tag{2.21}$$

This learning algorithm is very similar to the back-propagation learning algorithm introduced in section 2.2.1.2. The only difference is that there is no back-propagation of errors involved in the learning process because of the simple structure of the RBF neural network (one hidden layer).

2.3 Background of Fuzzy Logic

2.3.1 Introduction

Since the introduction of the concept of Fuzzy set in 1965 by Lofti A. Zadeh, the fuzzy logic had been mostly viewed as a controversial technology for some time. Since the late 1980s, fuzzy logic has been used in many successful applications from consumer products, to industrial process control.

2.3.2 Fuzzy Set and Fuzzy Logic

A fuzzy set is a set with a smooth boundary. Fuzzy logic refers to a logical system that *generalizes* classical two-valued Boolean logic for reasoning under uncertainty.

2.3.2.1 Fuzzy Set vs. Classical Set

The difference between a fuzzy set and a classical set is that a fuzzy set is a set with a smooth boundary while a classical set is a set with sharp boundary. For instance, the expression of "warm room temperature" may be expressed as an interval (e.g., [70°F, 78°F]) in classical set theory. However, the concept does not have a well-defined natural boundary. A representation of the concept closer to human interpretation is to allow a gradual transition from "not warm" to "warm." In order to achieve this, the notion of membership function in a set needs to become a matter of degree. This is the essence of fuzzy sets. An example of a classical set and a fuzzy set is shown in Figure 2.7, where the vertical axis represents the degree of membership in a set.

2.3.2.2 Membership Function

The definition of membership function is as follows: $\mu_A(x)$ *is called membership function of fuzzy set* A *(for a fuzzy variable* x*), whose value is the degree to which* x *is a member of fuzzy set* A.

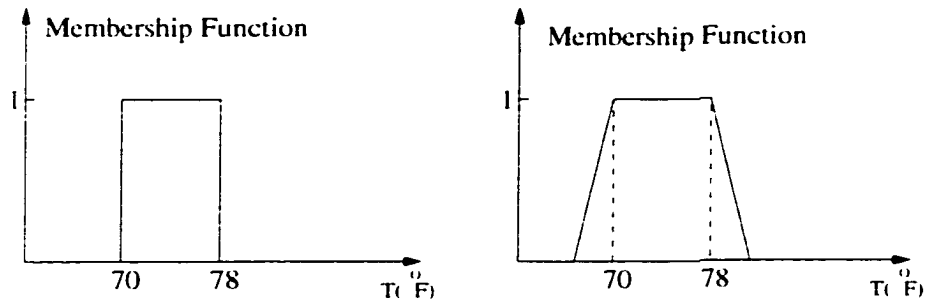


Figure 2.7: A classical set (left) and a fuzzy set (right) representation of "Warm Room Temperature."

There are several general forms for membership functions, as shown in Figure 2.8, i.e., triangular shape, trapezoidal shape, Gaussian shape, Beta functions, sigmoidal function, π function, and S and Z curve. The mathematical expressions for these forms are given by,

1. Triangular Shape

$$Tri(x, e, f, g) = \begin{cases} 0: & x \leq e \\ 1 - \frac{f-x}{f-e}: & e < x < f \\ \frac{g-x}{g-f}: & f < x < g \\ 0: & x > g \end{cases} \tag{2.22}$$

2. Trapezoidal Shape

$$Tra(x, a, b, c, d) = \begin{cases} 0: & x \leq a \\ 1 - \frac{b-x}{b-a}: & a < x < b \\ 1; & b < x < c \\ \frac{d-x}{d-c}: & c < x < d \\ 0: & x > d \end{cases} \tag{2.23}$$

3. Gaussian membership function

$$G(x, m, a) = e^{-\frac{(x-m)^2}{a^2}}; \tag{2.24}$$

4. Beta membership function

$$Beta(x, m, a, b) = \frac{1}{1 + ((\frac{x-m}{a})^2)^b} \tag{2.25}$$

5. Sigmoidal membership function

$$Sigm(x, a, c) = \frac{1}{1 + e^{-a(x-c)}} \tag{2.26}$$

6. π membership function

$$\pi_1(x, a, b) = \frac{1}{1 + \frac{x-a}{b}^2} \tag{2.27}$$

$$\pi_2(x, lw, lp, rp, rw) = \begin{cases} \frac{lw}{lp+lw-x}: & x \leq lp \\ 1: & lp \leq x \leq rp \\ \frac{rw}{x-rp+rw}: & x \geq rp \end{cases} \tag{2.28}$$

7. S-Curve Shape

$$S(x, a, b) = \begin{cases} 0: & x \leq a \\ 2(\frac{x-a}{b-a})^2: & a \leq x \leq \frac{a+b}{2} \\ 1 - 2(\frac{x-b}{b-a})^2: & \frac{a+b}{2} \leq x \leq b \\ 1: & x \geq b \end{cases} \tag{2.29}$$

8. Z-Curve Shape

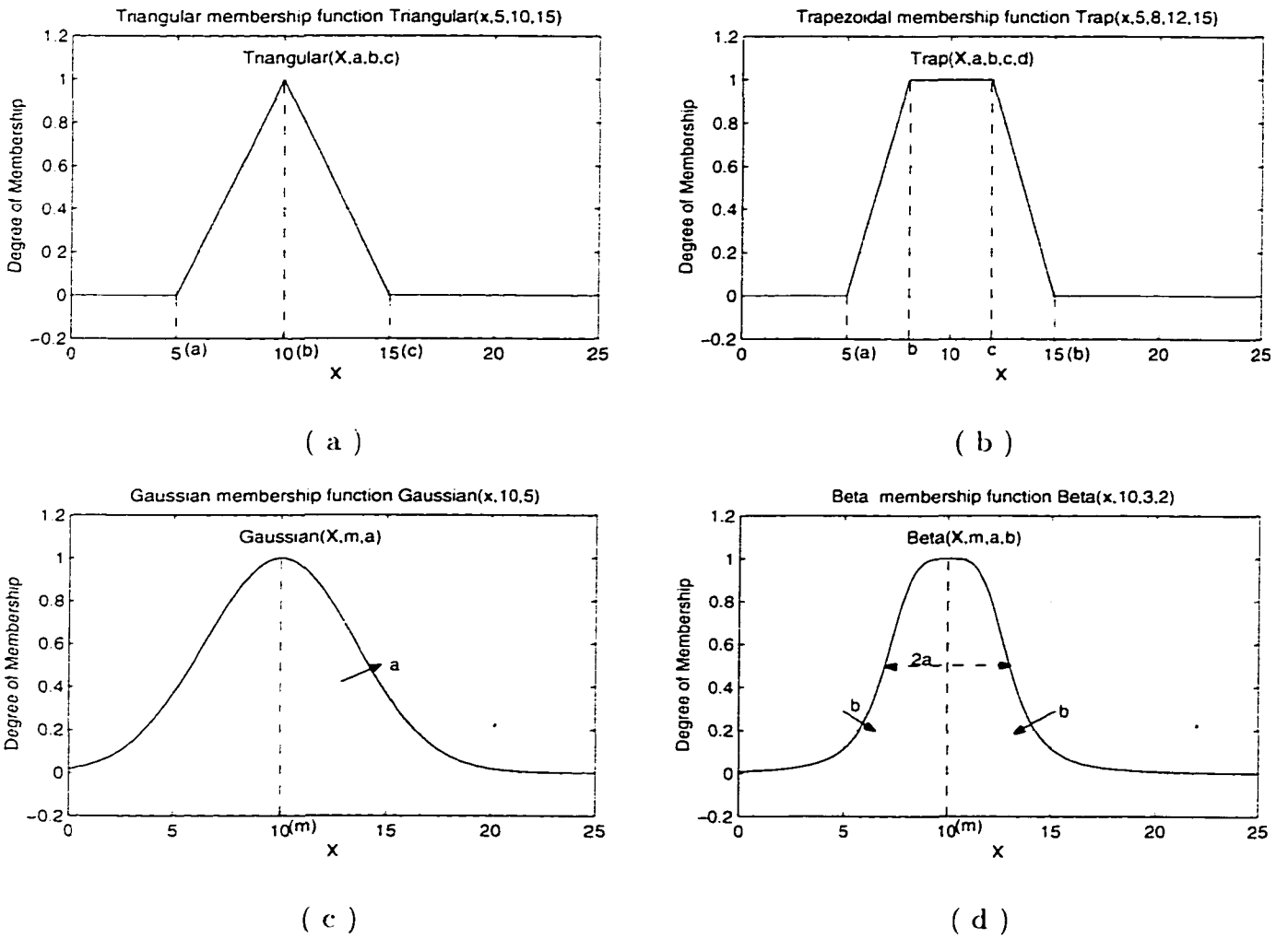
$$Z(x, a, b, c) = \begin{cases} 1: & x < a - b \\ 1 - \frac{(x-(a-b))^2}{2b^2}: & a - b \leq x \leq a \\ \frac{((a+b)-x)^2}{2b^2}: & a < x \leq a + b \\ 0: & x > a + b \end{cases} \tag{2.30}$$

2.3.2.3 Basic Operations in A Fuzzy Set

Generally, there are three major operations on sets, namely, *intersection*, *union*, and *complements*. The three operations on fuzzy sets have been extended accordingly because the notion of set membership has been generalized into a matter of degree.

(i) Intersection and Union of Fuzzy Sets

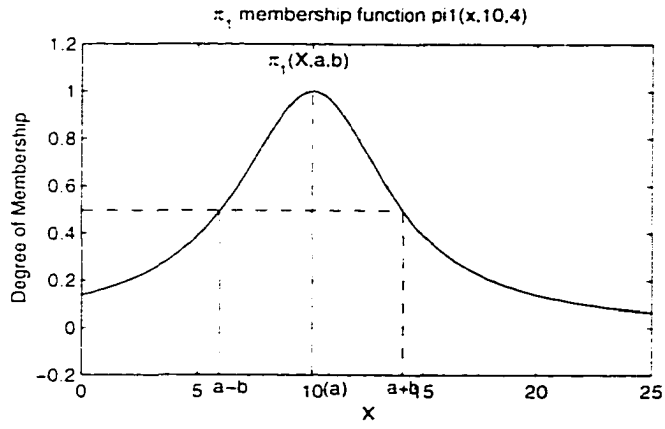
Zadeh (1973) suggested the minimum operator for the intersection and the maximum operator for the union of two fuzzy sets. For an example, there is a



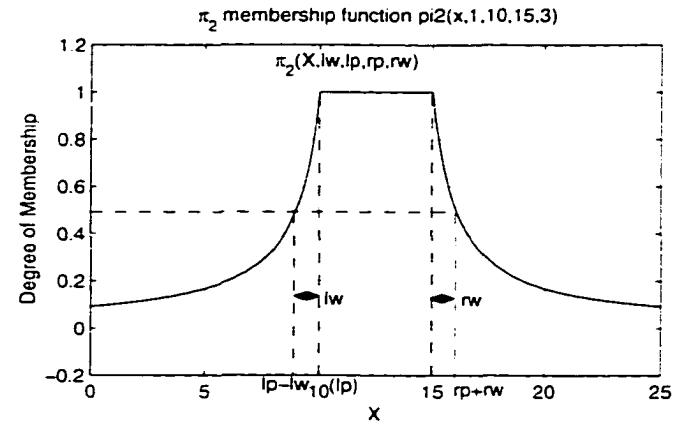
(a) (b)

(c) (d)

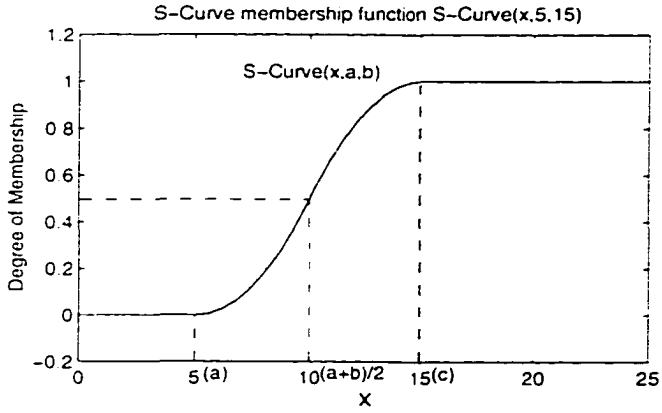
Figure 2.8: Shapes of membership function (a) Triangular (b) Trapezoidal (c) Gaussian (d) Beta function



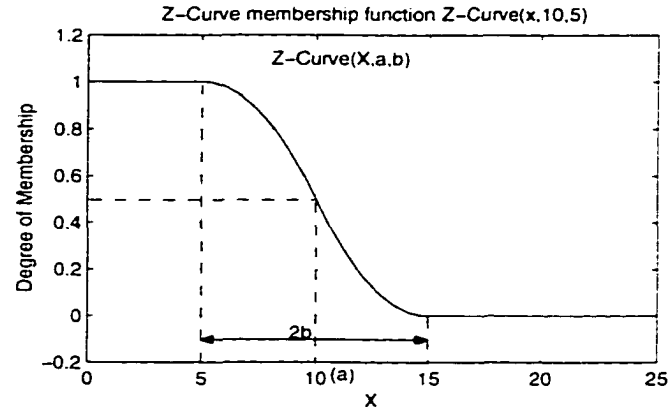
(e)



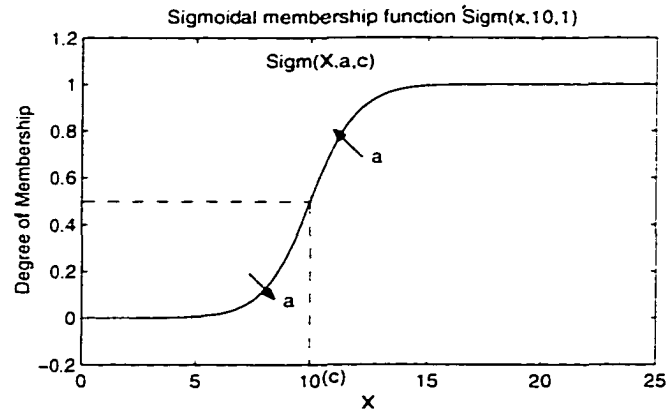
(f)



(g)



(h)



(i)

Figure 2.8: Shapes of membership function (e) π_1 (f) π_2 (g) S-Curve (h) Z-Curve (i) Sigmoidal

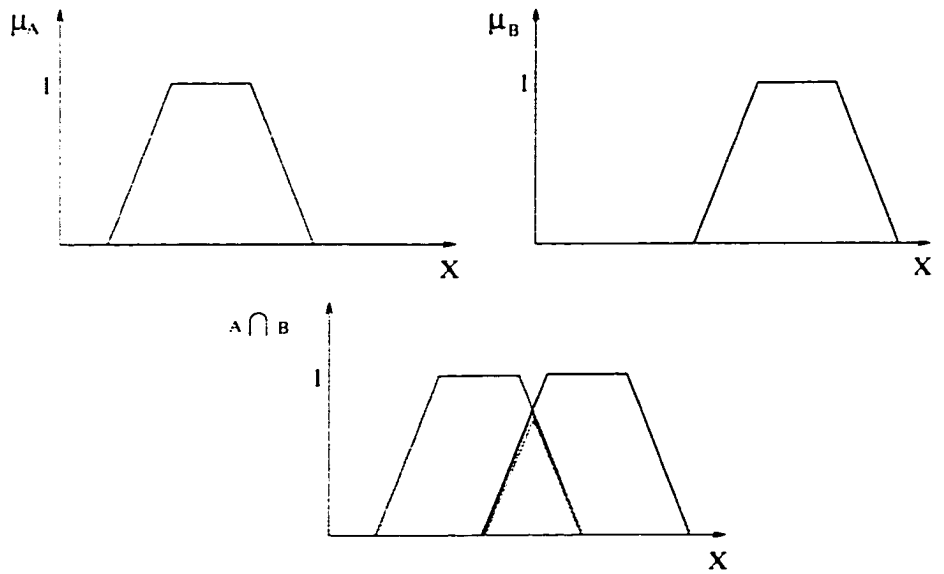


Figure 2.9: Intersection Operation: (a)fuzzy set **A** with a membership function μ_A (b) fuzzy set **B** described by the membership function μ_B (c) intersection operation of the two fuzzy sets (**A** AND **B**) (in green)

fuzzy set **A** with a membership function μ_A (shown in figure 2.9 (a) and a fuzzy set **B** described by the membership function μ_B (shown in figure 2.9 (b), the intersection operation of the two fuzzy sets (**A** AND **B**) can be expressed as (shown in figure 2.9 (c))

$$\mu_{A \cap B} = min(\mu_A, \mu_B) \tag{2.31}$$

if the minimum operator is used for the intersection. The union operation of the fuzzy set A and B is shown in figure 2.10 where the maximum operator is used for union, which can mathematical represented as:

$$\mu_{A \cup B}(x) = max(\mu_A(x), \mu_B(x)) \tag{2.32}$$

In addition, there are multiple choices for fuzzy intersection (also called fuzzy conjunction) and fuzzy union (also called fuzzy disjunction) operation. The following is the list of typical intersection and union operator pairs.

(a) Algebraic Product and Algebraic Sum:

$$\mu_{A \cap B} = \mu_A \times \mu_B \tag{2.33}$$

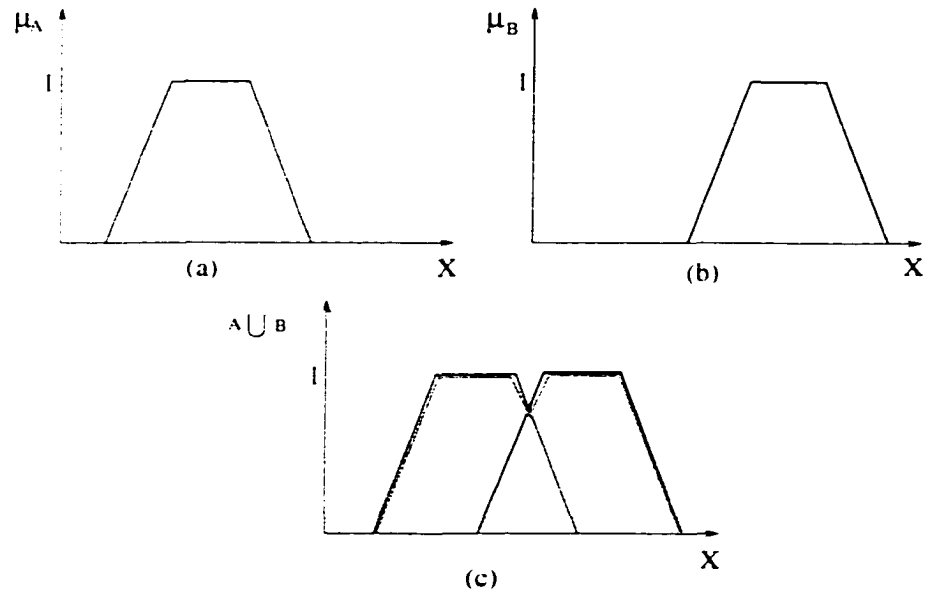


Figure 2.10: Union Operation: (a)fuzzy set **A** with a membership function μ_A (b) fuzzy set **B** described by the membership function μ_B (c) union operation of the two fuzzy sets (**A** AND **B**) (in green)

$$\mu_{A \cup B} = \mu_A + \mu_B - \mu_A \times \mu_B \tag{2.34}$$

(b) Bounded Difference and Bounded Sum:

$$\mu_{A \cap B} = max\{0, \mu_A + \mu_B - 1\} \tag{2.35}$$

$$\mu_{A \cup B} = min\{1, \mu_A + \mu_B\} \tag{2.36}$$

(c) Einstein Product and Einstein Sum:

$$\mu_{A \cap B} = \frac{\mu_A \mu_B}{2 - [\mu_A + \mu_B - (\mu_A \mu_B)]} \tag{2.37}$$

$$\mu_{A \cup B} = \frac{\mu_A + \mu_B}{1 + \mu_A \mu_B} \tag{2.38}$$

(d) Hamacher Product and Hamacher Sum:

$$\mu_{A \cap B} = \frac{\mu_A \mu_B}{\mu_A + \mu_B - \mu_A \mu_B} \tag{2.39}$$

$$\mu_{A \cup B} = \frac{\mu_A + \mu_B - \mu_A \mu_B}{1 - (\mu_A \mu_B)} \tag{2.40}$$

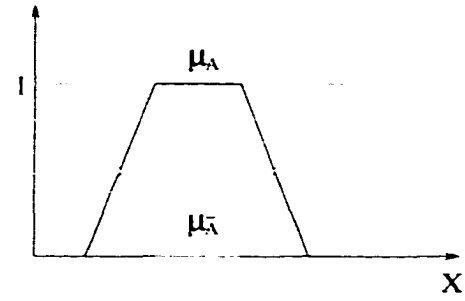


Figure 2.11: Complement Operation: (a)fuzzy set **A** with a membership function μ_A (in red) and the complement of **A** is in green.

(ii) Complement of a Fuzzy Set

The complement of a fuzzy set **A** is defined by the difference between one and the membership degree in **A**. i.e.,

$$\mu_{A^c}(x) = 1 - \mu_A(x) \tag{2.41}$$

2.3.3 Fuzzy Logic System

The block diagram of a general fuzzy logic system is shown in Figure 2.12. x_1, x_2, ... , x_n stand for n 'crisp' (or distinct) inputs, y is the crisp output. There are four types of blocks in this system, namely *fuzzification*, *rule inference*, *aggregation* and *defuzzification*(Kosko, 1997). The function of the various blocks in the fuzzy logic system are as follows:

2.3.3.1 Fuzzification

The function of the "fuzzification" block is to convert the crisp inputs (or precise measurements) to the fuzzy sets with corresponding membership degree. A specific crisp input can belong to different fuzzy sets but with different membership degrees (or, truth value). This is different from boolean logic in that one specific crisp value can belong to one and only one category. One of the examples is shown in Figure 2.13. Three fuzzy sets (short, average and tall) are involved in the

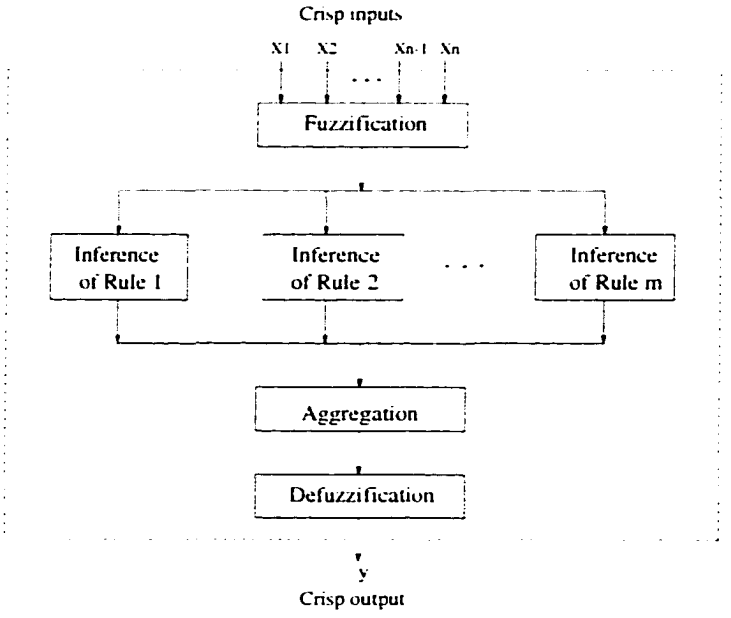


Figure 2.12: Block diagram of a general fuzzy logic system

example. Three curves ($\mu_{short}(height)$, $\mu_{average}(height)$, $\mu_{tall}(height)$) are shown in the figure, which are called membership functions for the fuzzy sets *Short, Average and Tall.* For a specific crisp height value $5'9''$, $\mu_{short}(5'9'') = 0.60$, which means that $5'9''$ height belongs to *Short* set with truth degree 0.6, or truth value for $5'9''$ height belonging to *Short* set is 0.6. Similarly $\mu_{average}(5'9'')(=1)$ and $\mu_{tall}(5'9'')(=0)$ are truth degrees to which $5'9''$ height is a member of fuzzy set *Average* and *Tall*, respectively. In this way, each crisp input height can be expressed by three truth values to the three fuzzy sets (*short, average and tall*). This process is called fuzzification. In the fuzzification process, membership functions, which are used to describe relationship of the crisp input and the fuzzy sets in the whole input domain, play a very important role.

The following describes another example of fuzzy sets in a fuzzy control system. This example will also help to explain other concepts in a fuzzy system. Figure 2.14 shows the whole fuzzy sets in a fuzzy fan-speed control system. The membership functions for the three fuzzy sets (*Cool set, Warm set, and Hot set*)

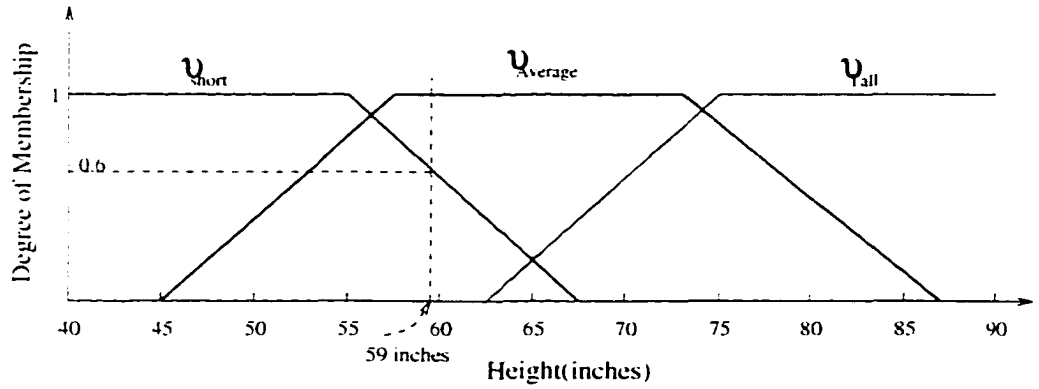


Figure 2.13: Fuzzification example

with respect to the input variable "*Temperature*" are shown in (a). *Low set, Moderate set, and high set* with respect to the other input variable "*Relative Humidity*" are shown in (b). (c) shows the membership functions of the *Low set* and *High set* for the fuzzy output variable "*fan speed.*"

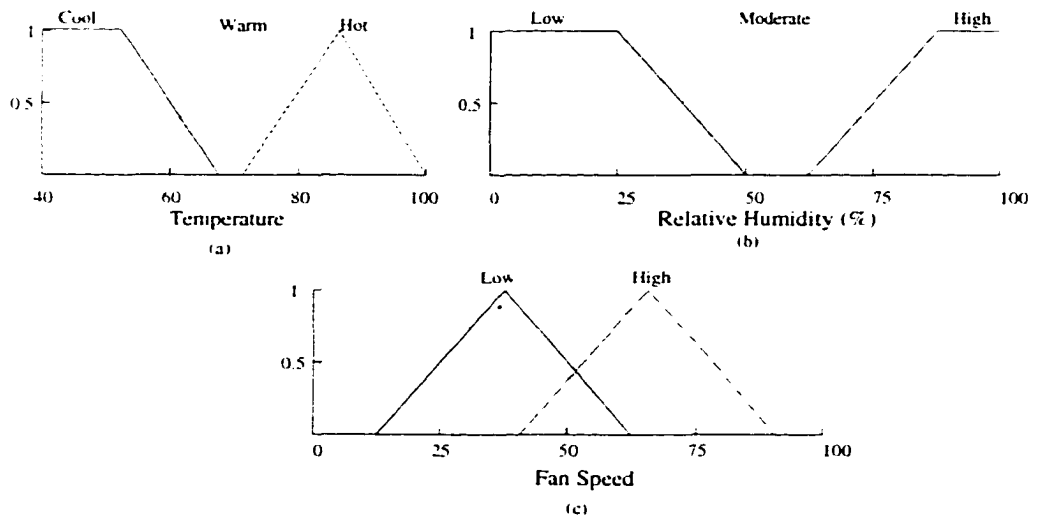


Figure 2.14: An example: fuzzy sets with respect to input fuzzy variable "*Temperature*" and "*Relative Humidity*", and output fuzzy variable "*fan speed*"

If the crisp input of temperature is 78° and the relative humidity is 40%, then the temperature with 78° is Hot with truth degree 0.42, the relative humidity with 40% is Moderate with truth degree 0.70. The following section on inference will show how these inputs are used.

2.3.3.2 Inference

In a fuzzy logic system, rules are used to describe the complex relationship between the input and output fuzzy variables in the form of IF-THEN statements linguistically. Typically, the rule is composed of several 'antecedents' in the IF statement and one or several 'consequents' in the THEN statement. The process of deducing the "strength" of these consequents from the "strength" of the antecedents is called rule *inference*. Usually a IF-THEN rule is in the form as follows.

IF *antecedent_1* AND *antecedent_2* AND ... THEN *consequent_1* AND *consequent_2* AND ...

The IF-side of the rule is composed of one or more antecedents, while the THEN-side is composed of one or more consequents. For example, the following Rule 1 is one of the rule in the fuzzy logic fan speed control system.

Rule 1: IF Temperature is Hot AND Humidity IS Moderate THEN Fan-speed is High.

This rule has two antecedent prepositions "*Temperature is Hot*" and "*Humidity is Moderate*" on the IF-side and one consequent preposition "*Fan -speed is High*" on the THEN-side. The rule inference process is explained as follows,
(1) obtain the truth degree of each preposition on the IF-side by using fuzzification results.

With 78° temperature and 40% relative humidity, the first antecedent preposition "Temperature is Hot" is true with degree 0.42 after using fuzzification as introduced in last section, the second antecedent preposition "Humidity is Moderate" is true with degree 0.70.
(2) obtain the strength of the IF-side by using one of the intersection operations

If we use *min* operator for intersection, the the strength of the IF-SIDE of Rule_1 is 0.42 for 78° temperature and 40 humity inputs.

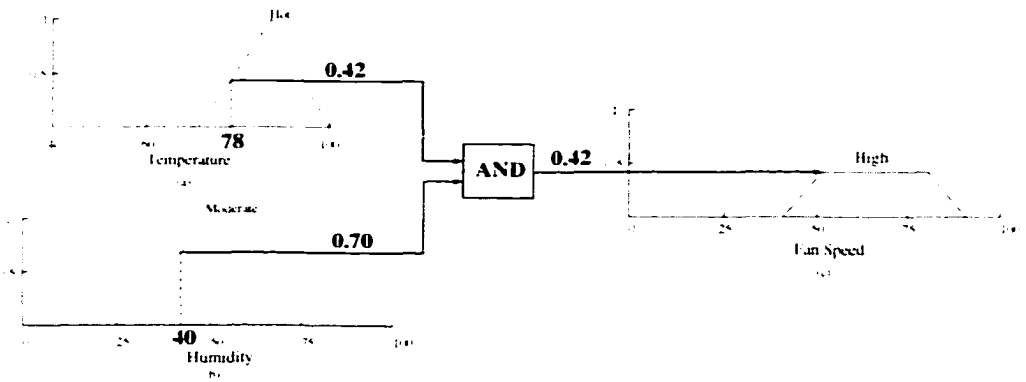


Figure 2.15: Rule 1 inference by using correlation minimum method.

(3) obtain the strength of consequences by using inference methods

From this IF-side strength, the strength of consequences can be obtained by using some inference methods. The most commonly used inference methods are *Correlation Minimum*, *Correlation Product* and *MIN-MAX* (Heske and Heske, 1996).

- Correlation Minimum

  Correlation Minimum Inference method uses the truth value of the IF-side to truncate the consequent fuzzy sets.

  For the previous example (Rule 1), the strength of the IF-side has been obtained as 0.42. By using the *correlation minimum* method, the fuzzy set **High** for fan speed has been truncated at 0.42 (the strength of the IF-side). The truncated output fuzzy set is shown in Figure 2.15.

- Correlation Product

  Correlation Product Inference method uses the truth value of the IF-side to scale the consequent fuzzy set.

  By using the *Correlation Product* inference method, the fuzzy output of the Rule 1 with respect to crisp inputs of temperature 78° humidity 40% is shown

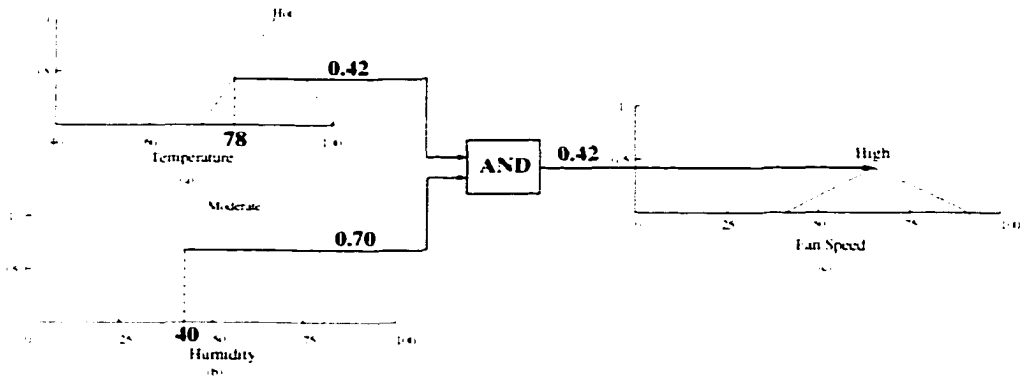


Figure 2.16: Rule 1 inference by using correlation product method.

in Figure 2.16, in which the output fuzzy set **High** for fan speed has been scaled by 0.42.

- MIN-MAX

  MIN refers to the use of the minimum intersection operator. MAX means, for a given consequent membership function, only the strongest(MAX) rule consequent is used.

2.3.3.3 Aggregation

Usually, we use several rules instead of one rule to describe a fuzzy logic system. These rules considered together is called a rulebase. All the knowledge about a fuzzy model is embedded in the rulebase and the membership functions. We can use the inference methods discussed above to derive strength of each rule, then aggregation method can be used to determine an overall fuzzy region that indicates the total effect of all rules based on the strength of each rule. Two commonly used aggregation methods are *Additive Aggregation* and *MAX Aggregation*.

- Additive Aggregation

  Additive Aggregation sums up the results produced by each rule in the rulebase.

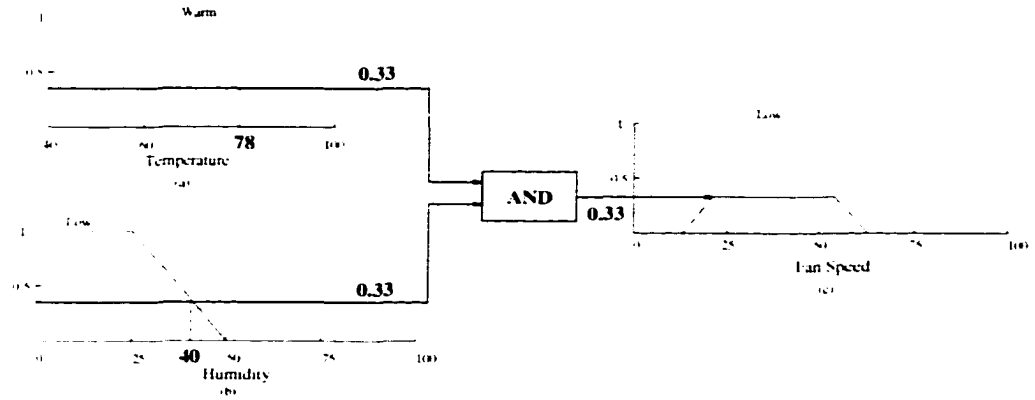


Figure 2.17: Rule 2 inference by using correlation minimum method.

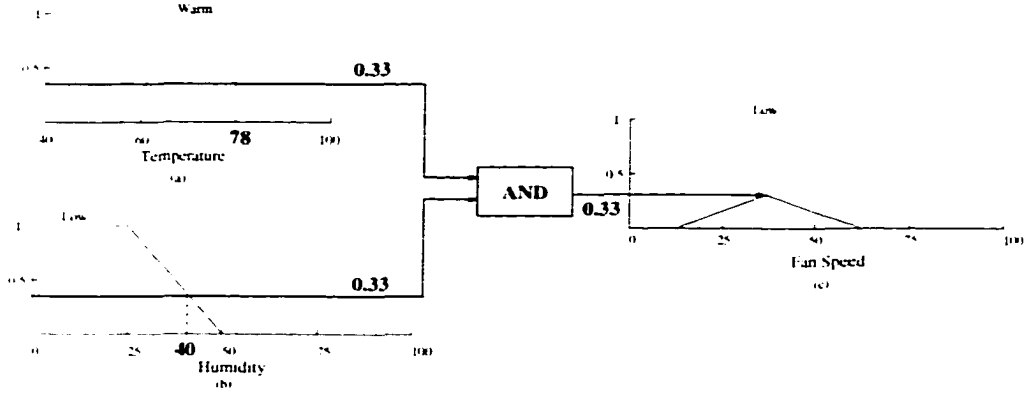


Figure 2.18: Rule 2 inference by using correlation product method.

- MAX Aggregation

  MAX Aggregation only take the consequents with the highest truth value.

To illustrate the aggregation method more clearly, the previous fuzzy fan-speed control system again is used here. Following is an additional rule which is used together with the Rule 1 to describe the fuzzy logic system.

Rule 2: IF Temperature IS Warm AND Humidity IS Low THEN Fan_speed is Low.

The rule inference results by using correlation minimum and correlation product method for the Rule 2 are shown in Figure 2.17 and 2.18, respectively.

The aggregation of Rule 1 and Rule 2 is shown in Figure 2.19 and 2.20 by using *Additive Aggregation* method and *MAX Aggregation* method, respectively.

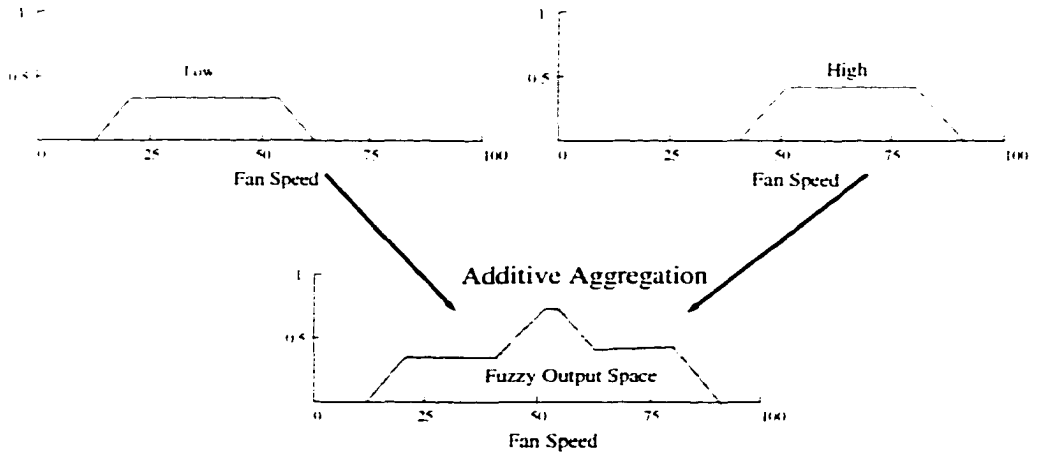


Figure 2.19: Additive aggregation of consequents produced by MIN intersection and correlation minimum inference.

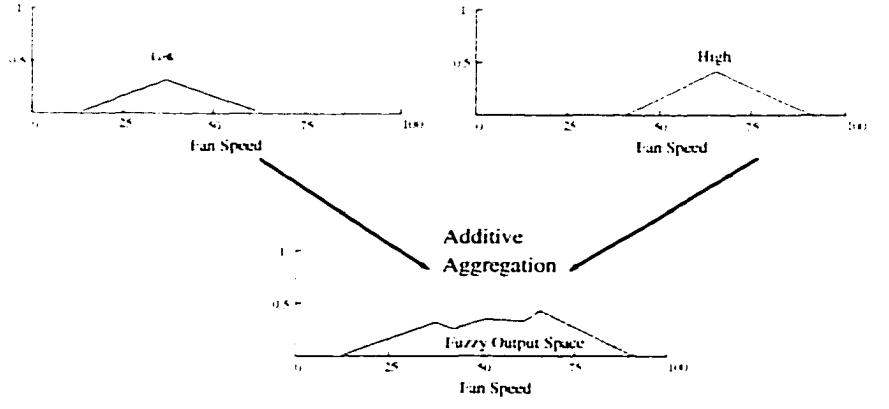


Figure 2.20: Additive aggregation of consequents produced by MIN intersection and correlation product inference.

2.3.3.4 Defuzzification

The output of aggregation process is a fuzzy set, but in many applications it is necessary to find a crisp value which best represents the fuzzy output set, and this process is called defuzzification. Two commonly used defuzzification methods are *Center of Area (COA)* and *Mean of Maximum (MOM)*.

- Center of Area (COA)

  In Center of Area method, the centroid of the output set is chosen as the crisp output.

$$Centroid = \frac{\int x\mu(x)dx}{\int \mu(x)dx} \tag{2.42}$$

  The centroids that correspond to Figure 2.19 and 2.20 are 52.03 and 52.66, respectively. Therefore, the system output fan speed is 52.03 or 52.66 (as shown in Figure 2.21).

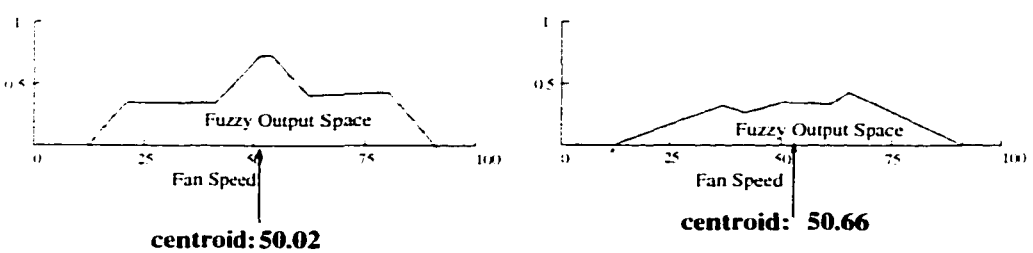


Figure 2.21: defuzzification result by using COA method.

- Mean of Maximum (MOM)

  In Mean of Maximum method, the output value corresponding to the maximum membership value is chosen as the crisp output.

Chapter 3

DEVELOPMENT OF A FUZZY LOGIC, NEURO-FUZZY SYSTEM FOR HYDROMETEORS CLASSIFICATION

The development of a fuzzy logic system for hydrometeor classification is described in this chapter. This chapter will cover the considerations for determining the inputs (features useful for classification) and outputs (potential classifiable hydrometeor types) for the system, and the definition of the fuzzy sets and fuzzy rulebase for the hydrometeors classification problem. In addition, automatic fine-tuning the parameters of the system is made available by the neuro-fuzzy learning algorithm introduced in this chapter. The fuzzy logic classification algorithm is implemented by using an object-oriented programming method. This implementation is suitable for modification and expansion of the designed system without significant the effort, which provides a convenient way to improve performances of the system with accumulation of more empirical and theoretical knowledge about the classification issue.

3.1 General System Description

The fuzzy logic system discussed here is intended to infer the possible hydrometeor types based on the available polarimetric radar observations by using fuzzy logic reasoning. The fuzzy logic reasoning is different from the conventional reasoning, where it is an approximate reasoning method which can derive a decision

from imprecise (or biased) or incomplete information from inputs. The polarimetric radar measurements reflect the average characteristics of the hydrometeors in the radar resolution volume; therefore, the inferred result is the bulky hydrometeor type instead of point-wise hydrometeor type. The fuzzy logic system for hydrometeor type classification is described based on the inputs and outputs of the system.

3.1.1 Output of the System

In this work, the goal of the fuzzy logic system is to derive hydrometeor types from remotely sensed radar data. There are a variety of hydrometeors, which can be categorized generally as liquid water and frozen ice. Liquid water can be further classified as cloud droplet, drizzle, medium rain, heavy rain, and supercooled rain drop. The frozen ice can be more specifically discriminated based on shape and density as dry snow, ice crystal, wet snow dry graupel, wet graupel, small hail and large hail. These are single hydrometeor types. In practice, however, precipitation often is a mixture of different hydrometeor types. This increases the possible mixture states of hydrometeors, therefore, the complexity of the classification problem is enhanced greatly.

In this study, 10 categories have been chosen to be the outputs of the classifier (as shown in table 3.1). From the fuzzy logic system view, the outputs are singletons. There are more possibilities of hydrometeor states, but this fuzzy logic classification system provides only the dominant hydrometeor type in mixed state precipitation.

Different hydrometeors are different in many aspects, such as size, shapes, preferred orientation, density, dielectric constant, etc.. Typical shapes for the four basic hydrometeors, rain, snow, graupel and hail, are shown in Figure 3.1(a)-(d), respectively.(Pruppacher, 1997)

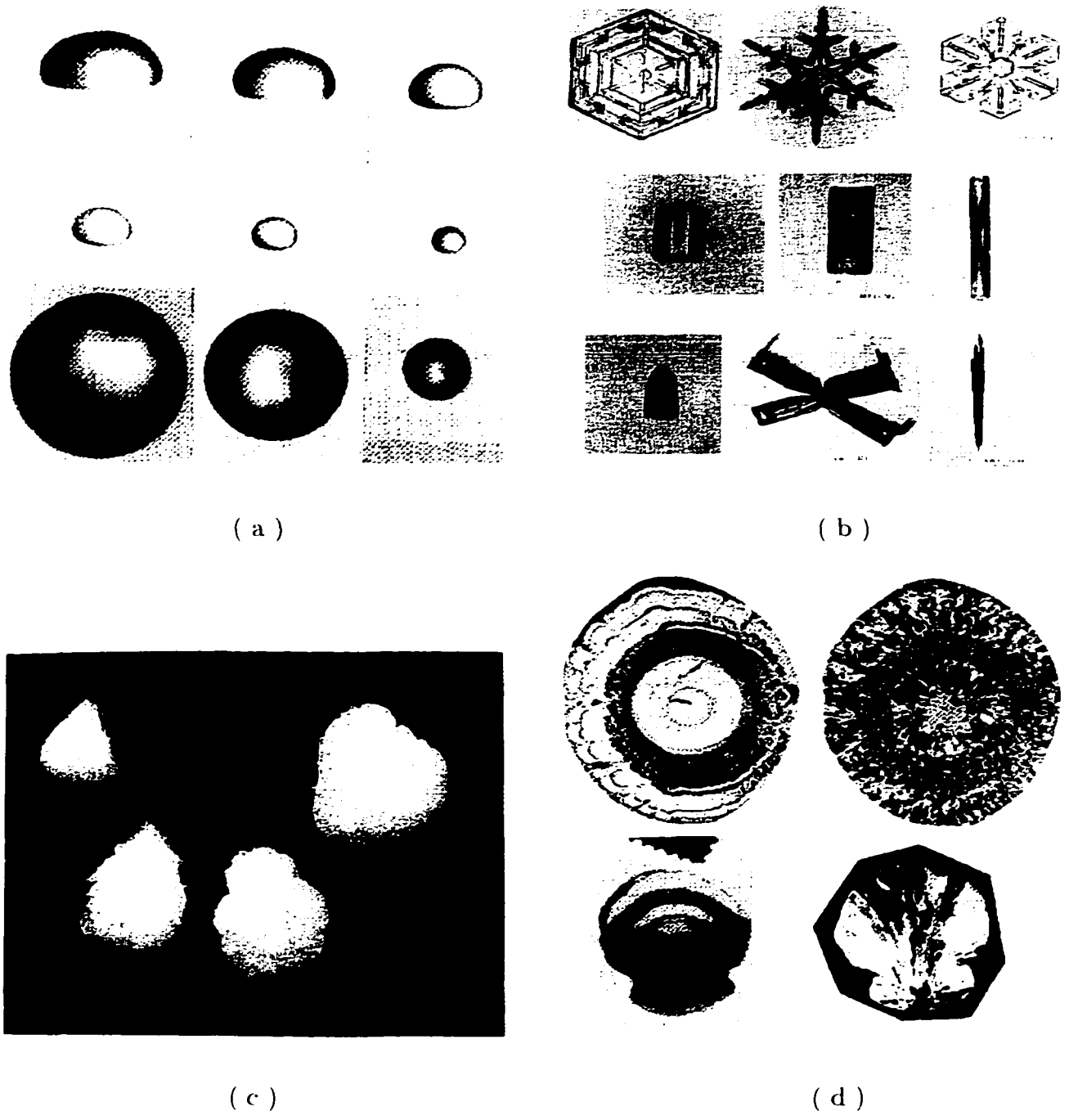


Figure 3.1: Shapes of the four basic hydrometeor types (a) shape of raindrops (b) Major shapes of snow crystals (c) graupel particles (d) hailstones (Pruppacher, 1997)

Table 3.1: Output of the fuzzy classifier system

Hydrometeor Type	Classifier Output C
drizzle	1
rain	2
dry ice crystal	3
high density dry ice crystal	4
wet ice crystal	5
dry graupel	6
wet graupel	7
small hail	8
large hail	9
rain + hail	10

Figure 3.1 (a) shows the shape of raindrops. Equivalent radius of drops is given by top row, from left to right: 4.00, 3.68, 2.90 mm; second row, from left to right: 2.65, 1.75, 1.35 mm; third row, from left to right 393, 354, 155 μm. Drops in third row were printed comparatively large to show sphericity. The shape of small raindrops is spherical. When rain dropsize increases, they become oblate. Figure 3.1 (b) shows the major shapes of snow crystals: top row, from left to right: simple plate, dendrite, crystal with broad branches; second row, from left to right solid column, hollow column, sheath; third row, from left to right bullet, combination of bullets, combination of needles. Figure 3.1 (c) show graupel particles of various sizes. Distance between lines on collection plate is 2 mm scale in figure. Figure 3.1 (d) shows thin sections of hailstones. Maximum diameter of hailstone on top row: 4cm; maximum diameter of hailstone on second row: 1.8 cm. We can see that with increasing size, hailstones become more oblate in vertical direction.

The difference in size, shapes, preferred orientation, canting angle, density, dielectric constant, etc. among different hydrometeors may result in different radar

Table 3.2: Values of polarimetric measurands for various hydrometeor types

	Z_h (dBZ)	Z_{dr} (dB)	$\lvert\rho_{hv}(0)\rvert$	K_{dp} (deg km^{-1})	LDR_{hv} (dB)
Drizzle	<25	0	>0.99	0	<-34
Rain	25 to 60	0.5 to 4	>0.97	0 to 10	-27 to -34
Dry Snow	<35	0 to 0.5	>0.99	0 to 0.5	<-34
Dense Snow	<25	0 to 5	>0.95	0 to 1	-25 to -34
Wet Snow	<45	0 to 3	0.8 to 0.95	0 to 2	-13 to -18
Dry Graupel	40 to 50	-0.5 to 1	>0.99	-0.5 to 0.5	<-30
Wet Graupel	40 to 55	-0.5 to 3	>0.99	-0.5 to 2	-20 to -25
Small Hail <2cm	50 to 60	-0.5 to 0.5	>0.95	-0.5 to 0.5	<-20
Large Hail >2cm	55 to 70	<-0.5	>0.96	-1 to 1	-10 to -15
Rain & Hail	50 to 70	-1 to 1	>0.9	0 to 10	-20 to -10

signatures. As a consequence, classification based on radar measurements potentially is possible.

3.1.2 Input of the System

There are many polarimetric radar measurements which are highly related to the bulky hydrometeor type. For linear polarization measurements, the five parameters, Z_h, Z_{dr}, K_{dp}, LDR, and ρ_{hv} contain information about hydrometeors in a radar resolution volume. The thresholds of these parameters related to the 10 hydrometeor types are given by Doviak and Zrnic (1993) are listed in Table 3.2. Using both linear and circular basis measurements is a potentially useful framework for hydrometeor type classification (Bringi and chandrasekar, 2000). However, the current available radar parameters from CSU-CHILL radar are only linear polarization measurements. Therefore, we mainly chose the five linear polarized radar parameters (Z_h, Z_{dr}, K_{dp}, LDR and ρ_{hv}) as the inputs to the fuzzy logic system for hydrometeor classification. The following provides discussion of potential contributions of these parameters to the classification issue.

(1) Reflectivity factor at horizontal polarization Z_h:

Z_h was defined in section 2.1.1.3. Z_h is proportional to the received power at h-port when transmitted signal is horizontally polarized. The received power is proportional to the mean cross-section of the hydrometeors in the radar resolution volume. To some degree, Z_h contains the information about the hydrometeors size (on average). Approximately, Z_h varies between 25 dBZ and 60 dBZ for rain; for snow, Z_h is typically less than 35 dBZ; for graupel, Z_h varies in the range 40 to 55 dBZ; for hail, Z_h varies in the range 50 to 70 dBZ (Doviac and Zrnic, 1993). It is very clear Z_h alone cannot discriminate between different hydrometeors, because there is extensive overlapping of Z_h among different hydrometeor types.

(2) Differential Reflectivity Z_{dr}:

As defined in section 2.1.1.3. Z_{dr} is the ratio of Z_h and Z_v, where Z_h is the reflectivity factor at horizontal polarization, Z_v is the reflectivity factor at vertical polarization. Z_{dr} can be written as,

$$Z_{dr} = 10log(Z_h/Z_v) \tag{3.1}$$

Z_{dr} is a good indicator of mean shape of precipitation particles and can be used to indicate the region of frozen precipitation, and to identify large rain drops, small hail, and graupel, as well as large hail.

- Z_{dr} signature for raindrop:

  For raindrops with equivalent diameter greater than 1mm, their shapes can be approximated by oblate spheroids with major axes in the horizontal. Therefore, Z_h is greater than Z_v, which results in Z_{dr} greater than zero dB. For small raindrops, their shape is very close to spherical, then Z_{dr} is approximately zero.

- Z_{dr} signature for hail:

  On the average, small hail is spherical. As a result, Z_{dr} is approximately zero dB. For larger hail, even though their shapes may not be spheres, they tumble and assume random orientaion and on the average Z_{dr} is close to zero. When their shapes are oblate with their major axis in the vertical, Z_{dr} is negative. The dielectric constant and the size of the hail particles can affect the value of Z_{dr}. Large hailstones with the same shape and orientation as smaller ones have different differential reflectivity if the scattering is in the Mie region.

- Z_{dr} signature for ice particles:

  Z_{dr} is close to zero for most ice particles, normally falls between -0.5 and $+0.5$ dB. However, high Z_{dr} can be found in the region of melting aggregates, or pristine plate-like crystals, or supercooled water drops above freezing level.

(3) Differential Propagation Phase Shift K_{dp}:

When a plane wave propagates through precipitation, the phase of the wave shifts as the propagation distance increase. The difference of the phase shift between propagation through precipitation and clear air is called the propagation phase shift. The differential propagation shift (ϕ_{dp}) is defined as the difference between the propagation phase shift for horizontal and vertical polarization. K_{dp} is the derivative of ϕ_{dp} with respect to the range.

- K_{dp} signature for rain:

  K_{dp} is very sensitive to nonisotropic medium such as rain, therefore, the range of K_{dp} for rain can be as large as 10 deg/km depending on the frequency. K_{dp} is close to zero for small spherical drops.

- K_{dp} signature for ice particle, hail, and graupel

  K_{dp} is insensitive to istropic media such as ice particle, hail, and graupel, so K_{dp} is normally very small with respect to these hydrometeor types. However, when hydrometeor particles start melting, they are coated with water outside, which cause the increases of K_{dp} dramatically.

(4) Correlation Coefficient at zero lag $\rho_{hv}(0)$:

The correlation coefficient between horizontally and vertically polarized echoes ($\rho_{hv}(0)$) is affected mainly by the variability in the ratio of the vertical-to-horizontal size of individual hydrometeors, and the orientation of hydrometeors. Many factors can decrease ρ_{hv} , such as, eccentricity distribution, drop oscillations, canting angle variation, change in differential phase shift on backscatter, etc.(Bringi and Chandrasekar, 2000).

- $\rho_{hv}(0)$ signature for rain:

  The $\rho_{hv}(0)$ for pure rain normally is high (> 0.97) because of the small variation in orientation and shape. (Brinig and Chandrasekar, 2000)

- $\rho_{hv}(0)$ signature for graupel and hail:

  The $\rho_{hv}(0)$ for graupels and hail is generally small due to the irregular shape and variable orientation.

- $\rho_{hv}(0)$ signature for mixed phase precipitation:

  Due to different dielectric constant, wide variation of size and shape distribution, scattering characteristics of particles are not correlated, and as a result $\rho_{hv}(0)$ in mixed phase precipitation is significantly less.

(5) Linear Depolarization Ratio LDR:

The factors that result in LDR_{hv} make it useful in identifying bulk hydrometeors and region of mixed phase hydrometeor types. The lowest values of $LDR_{hv}(< -34\text{dB})$ generally indicated dry, low density snow.Slightly larger values of LDR_{hv}(-27 to -34 dB) are found in rain with larger values for deformed or canted drops. Unaligned or non-spherical of mixed phase precipitation, wet snow and hail, could result in higher LDR values such as -18 to -12 dB.

From the above analysis, the five radar polarimetric measurements intimately are tied with hydrometeor types, and each of them individually can not be used to derive the hydrometeor types. Five of them together provide more information about the hydrometeors, therefore these five radar measurements are chosen as inputs for the fuzzy classifier system. The complexity of the hydrometeor classification problem is far beyond what a simple thresholding decision method can deal with. This can be seen clearly in Figure 3.2 where the thresholds in table 3.2 is re-represented by four 2-Dimensional space diagrams (i.e., Z_h vs. Z_{dr}, Z_h vs. K_{dp}, Z_h vs. LDR, Z_h vs. ρ_{hv}).

3.2 Architecture of a fuzzy hydrometeor classifier

To implement the hydrometeor classification using fuzzy logic (fuzzy hydrometeor classifier, hence forth referred as FHC), the four general blocks (*fuzzification*, IF-THEN rule *inference*, *aggregation* and *defuzzification*) need to be specified. The block diagram of the FHC is shown in Figure 3.3. Z_h, Z_{dr}, K_{dp}, LDR, ρ_{hv}, and altitude of the observation (H), are the six inputs, and the hydrometeor Class, C, is the output. The FHC will infer the hydrometeor type C from 6 inputs based on a rulebase defined for this specific classification problem. Table 3.1 lists the 10 classes used in inference of summer storms. The detailed block diagram of the Fuzzy Hydrometeor Classifier is shown in Figure 3.4.

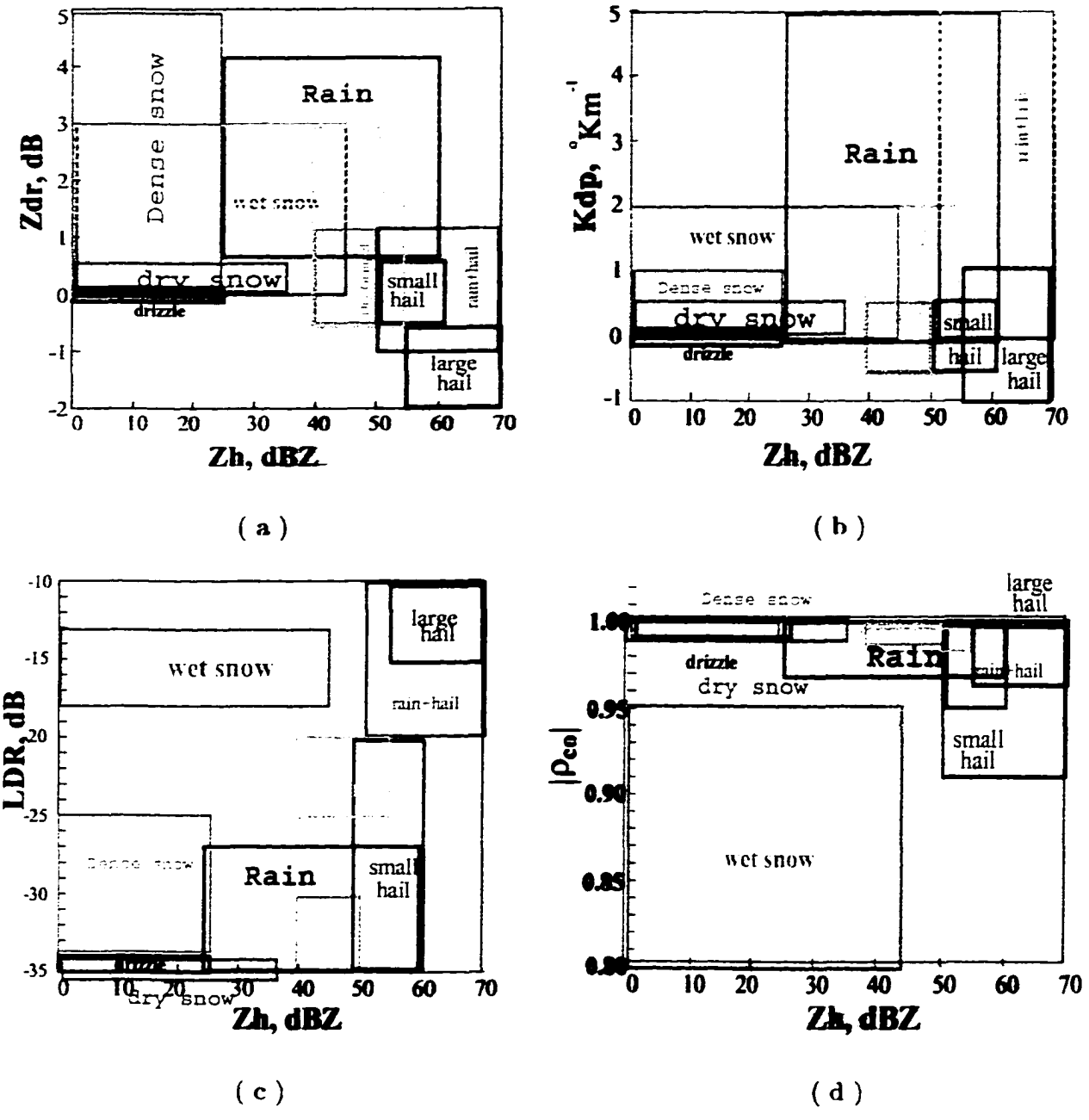


Figure 3.2: (a) Z_h - Z_{dr} domain for different hydrometeors (b) Z_h - K_{dp} domain for different hydrometeors (c) Z_h - LDR domain for different hydrometeors (d) Z_h - ρ_{hv} domain for different hydrometeors

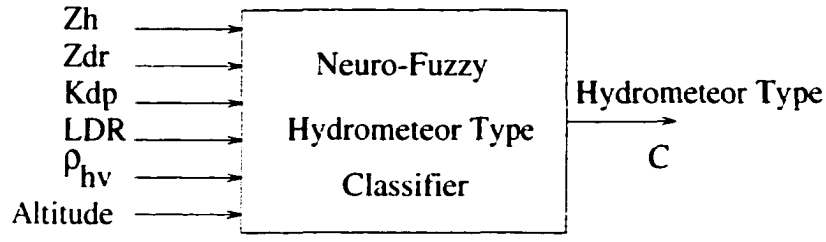


Figure 3.3: Block scheme of the fuzzy classifier

The classification procedure of FHC shown in Figure 3.4 can be described as the following: Firstly, the 5 radar measurements and altitude are fuzzified by using membership functions (MBFs). There are 10 MBFs for each of the input variables in the system. After fuzzification, the IF-THEN rule inference is carried out based on the rulebase for the classification system. To achieve the total effect of all the rules, rule aggregation is applied. The last step is defuzzification, which can convert aggregation result to a single hydrometeor type.

3.3 Membership Function Determination

The purpose of fuzzification is to convert precise input measurements to fuzzy sets with corresponding membership degree. The specification of membership functions is critical to classification performance. . Two different sets of membership functions are used, one for summer storms and the other winter storms. Ten fuzzy sets corresponding to the 10 hydrometeor types are specified for each of the six input variables for summer storms. Similarly, five fuzzy sets corresponding to the hydrometeor types (namely, drizzle, rain, dry ice crystal, oriented ice crystal, and wet ice crystal) are specified for each of the six input variables for winter storms. Each fuzzy set is represented by a membership function, denoted as MBFi_j, where the index i corresponds to the six inputs, the index j corresponds to the fuzzy sets. The index j takes values 1 through 10 for summer storms and 1 to 5 for winter storms.

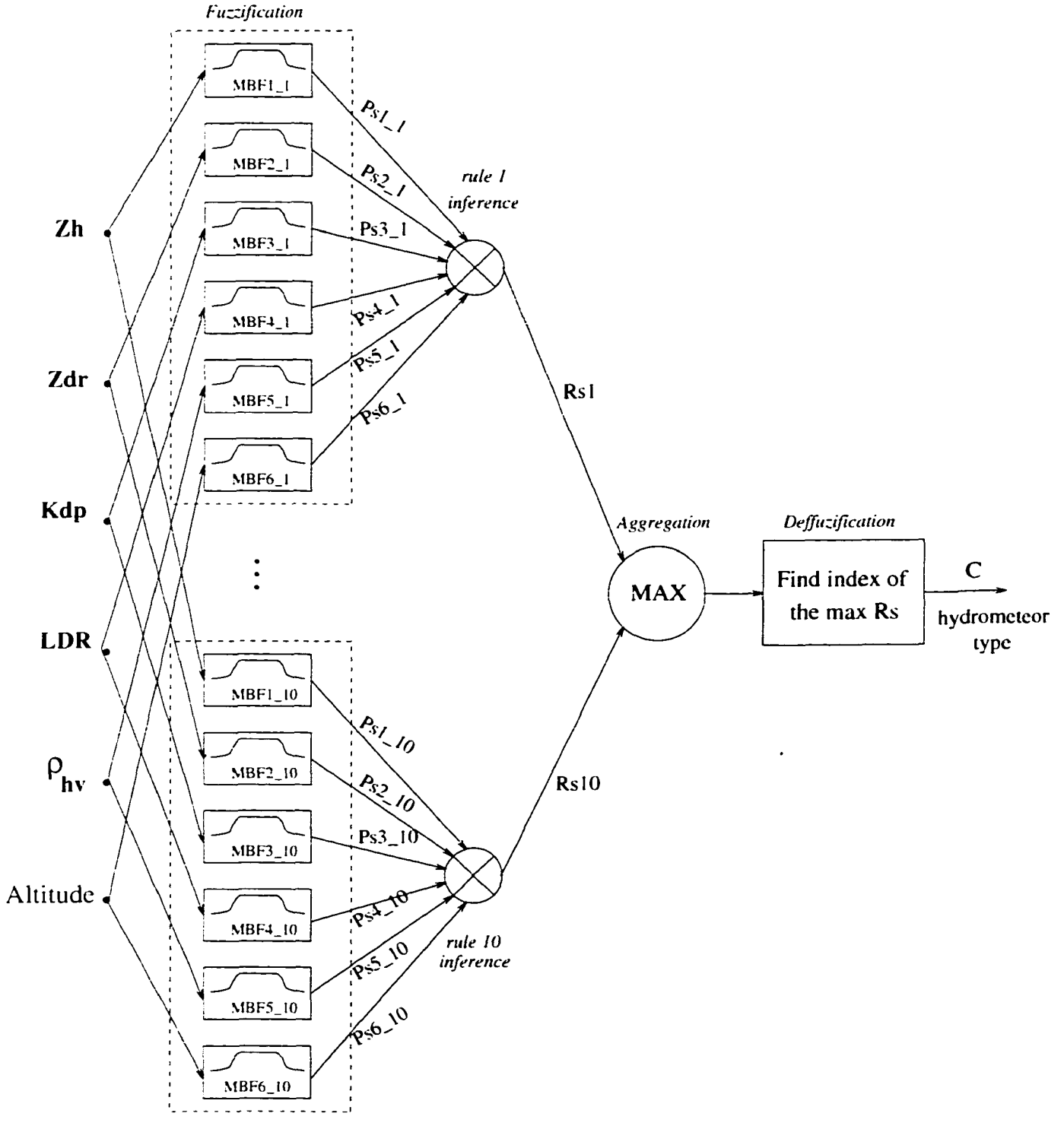


Figure 3.4: A Fuzzy System for Hydrometeor Type Classification

3.3.1 1-D Membership Functions

Several functional forms can provide adequate representation of membership functions, such as triangular, trapezoidal, Gaussian shapes, S and Z curves, and Beta functions. In this study, a Beta function is chosen to describe membership function for following reasons: a) In the hydrometeor classification problem, we expect most membership functions to have a wide, flat region in which the maximum value is 1. One hydrometeor type such as rain can have a wide range of reflectivity. In other words, there is no preferred or unique value of reflectivity for rain, but a preferred region such as 25 to 60 dBZ. The best MBF to represent this is by means of a flat function over the preferred region that tapers off outside the preferred range. The Beta function has the desired characteristics: therefore, it is chosen as the form of the membership functions. In addition, a Beta function has a long tail, which improves the robustness of FHC. The derivative of the Beta function is continuous and this feature is useful for automatic adjustment of the parameters that will be needed for development of a neuro-fuzzy system.

The Beta membership function is defined as

$$Beta(x, m, a, b) = \frac{1}{1 + ((\frac{x-m}{a})^2)^b}. \tag{3.2}$$

It can be seen from (3.2) that three parameters define the shape of a Beta function, namely, center of the function m, the width a and the slope b (shown in Figure 3.5). Typical one dimensional membership functions for Z_h are shown in Figure 3.6 for summer storms and Figure 3.7 for winter storms. These 10 membership functions are representations of the 10 Fuzzy Sets of Z_h. The corresponding three parameters (m, a, b) are listed in Table 3.3. For each curve of membership function, the horizontal axis is the value of Z_h, and the vertical axis stands for the membership degree of Z_h corresponding to that fuzzy set. In a fuzzy logic system, fuzzy sets, instead of precise values, are used to represent input variables. For example, if we

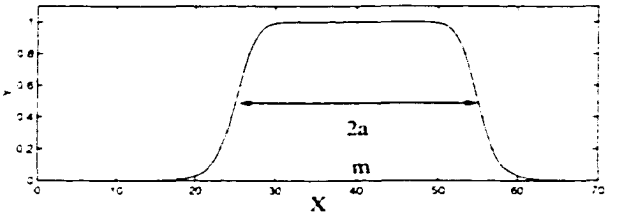


Figure 3.5: Beta function

Table 3.3: Table for Z_h MBF Coefficients

MBF	m	a	b
MBF1-1	0.00000	25.69000	25.49000
MBF1-2	42.50000	18.17000	18.32000
MBF1-3	0.00000	35.70000	35.04000
MBF1-4	0.00003	25.69309	25.48988
MBF1-5	0.00000	45.70000	44.59000
MBF1-6	44.99999	5.58037	6.28998
MBF1-7	47.50000	8.10922	8.72278
MBF1-8	55.00000	5.58000	6.29000
MBF1-9	70.00000	15.67000	15.93000
MBF1-10	70.00000	20.68000	20.71000

have Z_h with 40 dBZ, it belongs to the *drizzle* set with membership degree 0. to the *rain* set with membership degree 1. ... . to the *DryGraupel* set with membership degree 0.8. ... . to the rain and hail mixture with membership degree 0.

One dimensional membership functions are used to represent K_{dp}, LDR and ρ_{hv} for all hydrometeor type fuzzy sets. Tables 3.4, 3.5 and 3.6 list the parameters for their respective membership functions. Figure 3.8, 3.9, and 3.10 show the curves of the membership functions. It is easy to notice that the slope of membership functions for LDR is small compared with other radar measurement variables. This is based on the fact that LDR is more noisy compared to other measurements. Decreasing the slope is equivalent to increasing the robustness of the parameter. The membership function of altitude is dependent on location and season. The most important parameter here is the melting level. In addition, the altitude of the

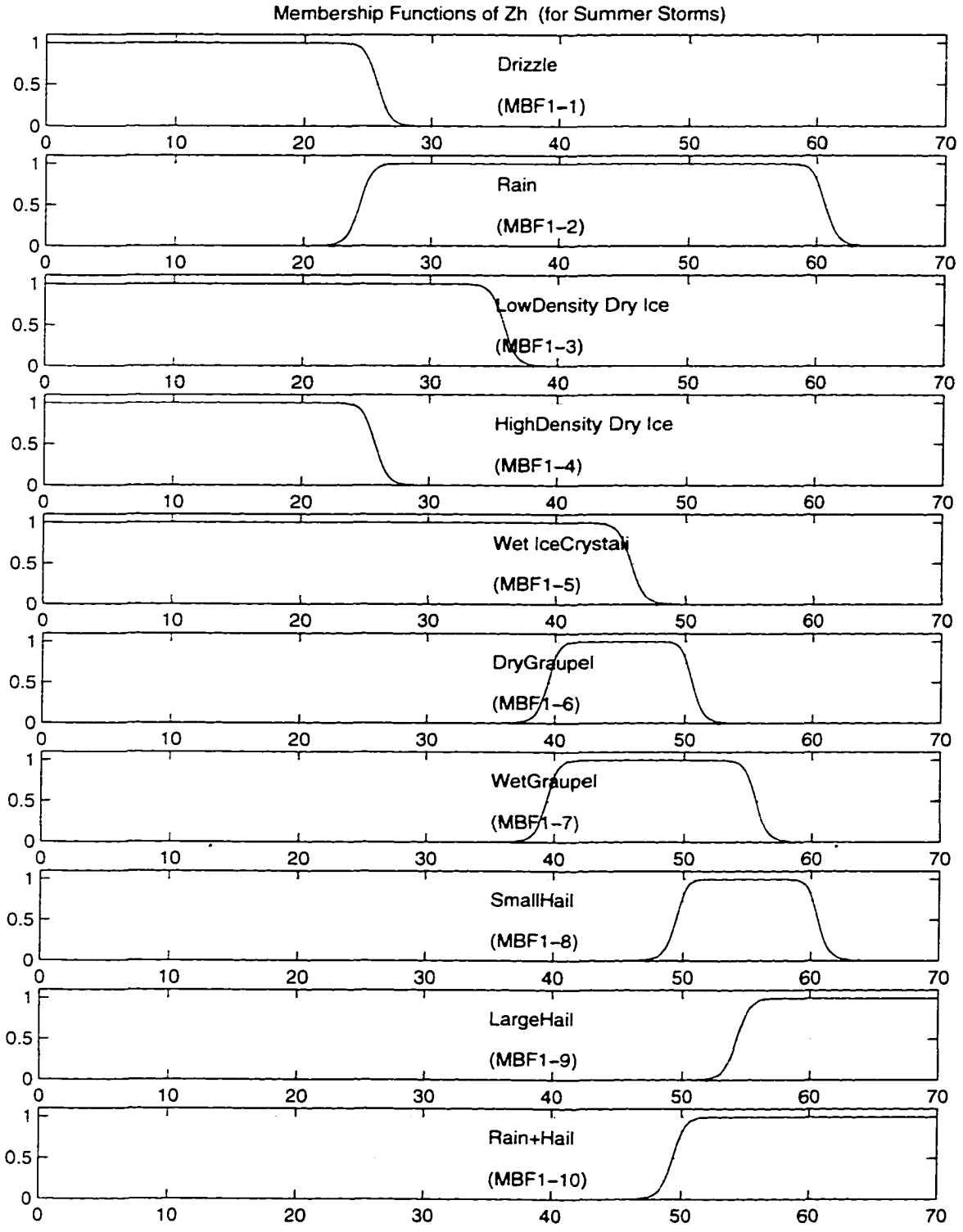


Figure 3.6: Membership functions for Fuzzy Variable Zh (in dBZ), and illustration of the fuzzification of Z_h to its 10 fuzzy sets.

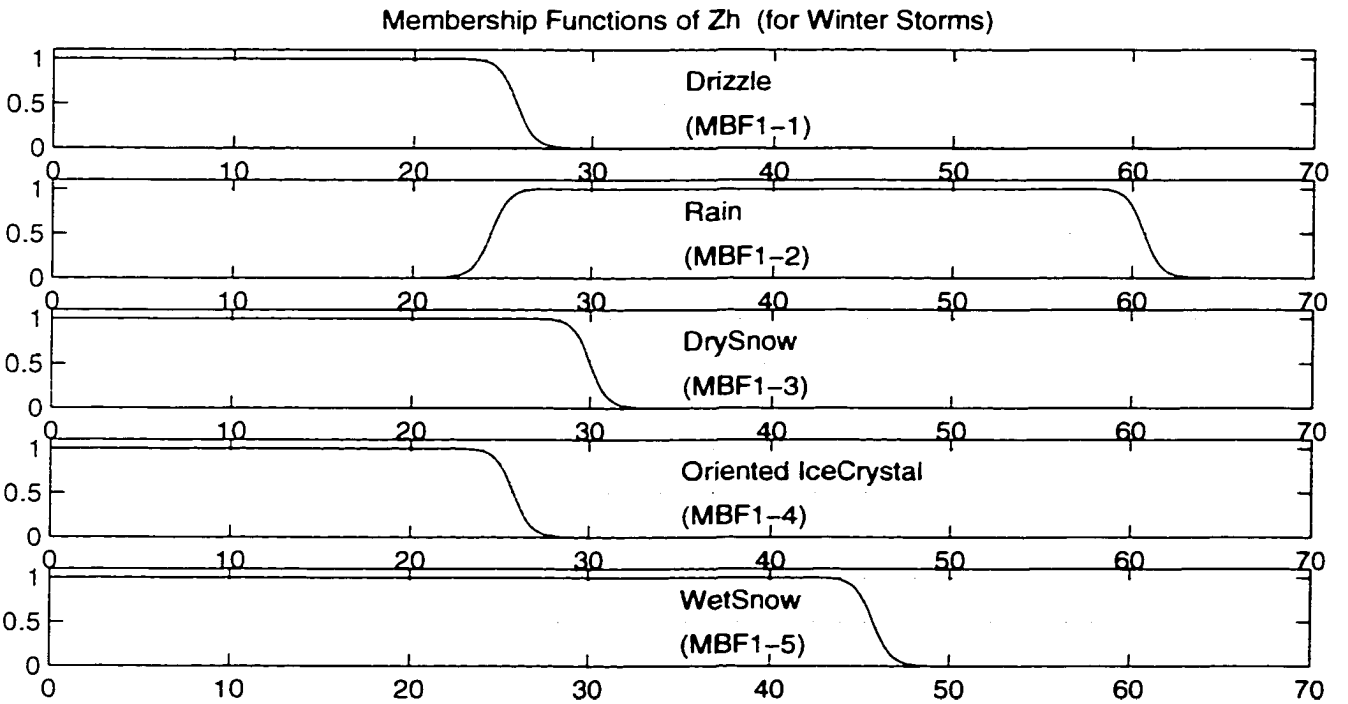


Figure 3.7: Membership functions for Fuzzy Variable Z_h, and illustration of the fuzzification of Z_h to its 6 fuzzy sets for winter storms.

melting level depends on the season. There are a total of 90 membership functions (60 for summer, 30 for winter). The membership functions will get fine tuned over time when more in-situ data are observed.

3.3.2 2-D Membership Functions

The membership functions for Z_{dr} need to be specified with Z_h. If the measurements of Z_{dr} are physically independent of other radar measurements such as Z_h, then one dimensional membership functions are adequate to represent the fuzzy sets. The corresponding 1-D membership functions are shown in Figure 3.11, and their parameters are listed in table 3.7. Multi-dimensional membership functions can be used to represent combinational fuzzy sets. Let f_{rain_zh} be the one

Table 3.4: Table for K_{dp} MBF Coefficients

MBF	m	a	b
MBF3-1	0.00000	0.15000	1.66000
MBF3-2	11.00000	11.13000	58.17000
MBF3-3	0.25000	0.33000	2.61000
MBF3-4	0.50000	0.59000	4.01000
MBF3-5	1.00000	1.11000	6.67000
MBF3-6	0.00000	0.59000	4.01000
MBF3-7	0.75000	1.36000	7.98000
MBF3-8	0.00000	0.59000	4.01000
MBF3-9	0.00000	1.11000	6.67000
MBF3-10	11.00000	11.13000	58.17000

Table 3.5: Table for LDR MBF Coefficients

MBF	m	a	b
MBF4-1	-38.00000	4.82000	3.73000
MBF4-2	-30.50000	3.99504	3.69757
MBF4-3	-38.00000	4.83609	3.71487
MBF4-4	-29.50000	5.34000	4.03000
MBF4-5	-15.50000	3.21564	2.77366
MBF4-6	-38.00000	8.94409	6.13533
MBF4-7	-22.50000	3.26213	2.75770
MBF4-8	-38.00000	19.07000	11.96000
MBF4-9	-12.50000	3.21000	2.78000
MBF4-10	-15.00000	5.87000	4.34000

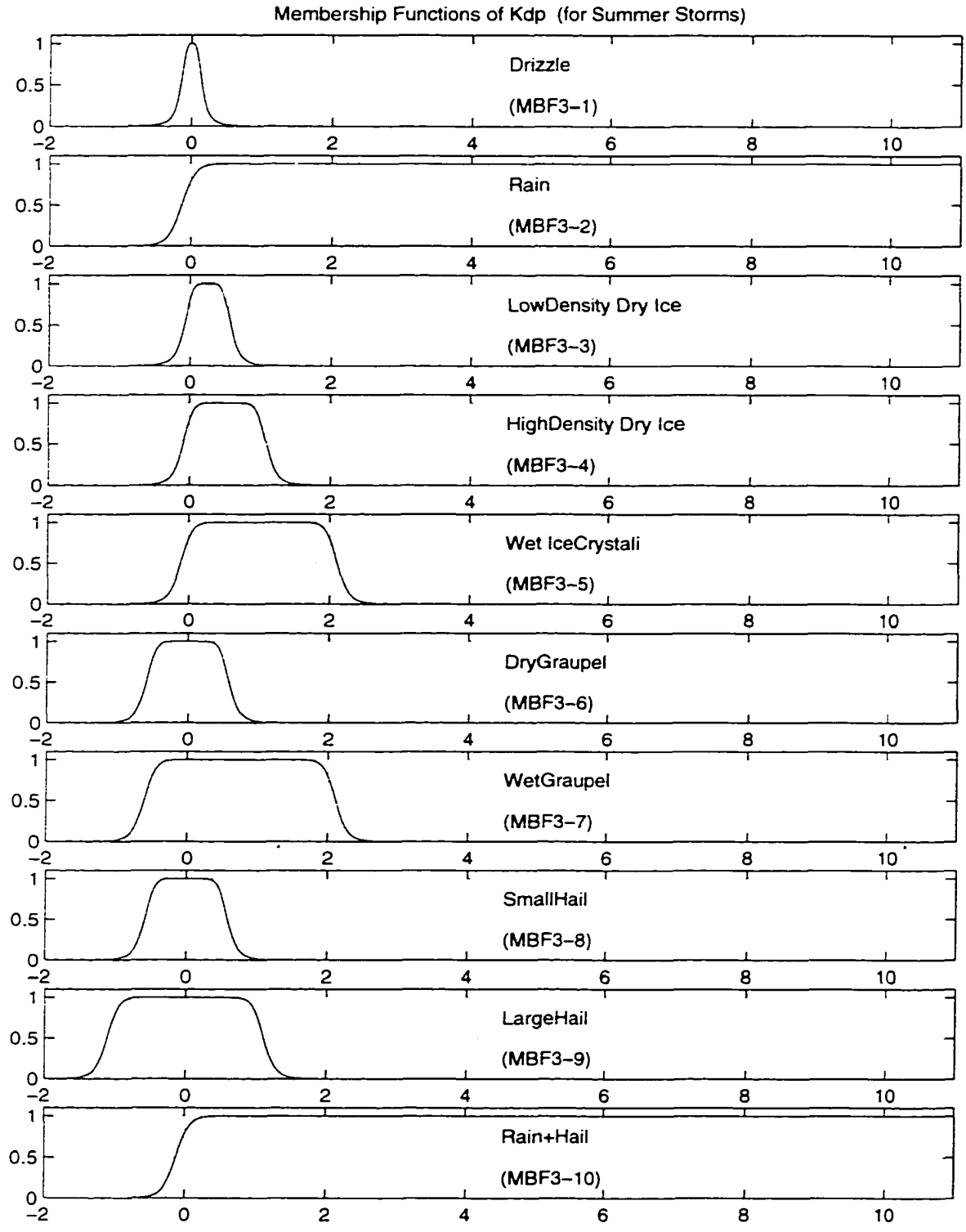


Figure 3.8: Membership functions for Fuzzy Variable K_{dp} (in deg/km), and illustration of the fuzzification of K_{dp} to its 10 fuzzy sets.

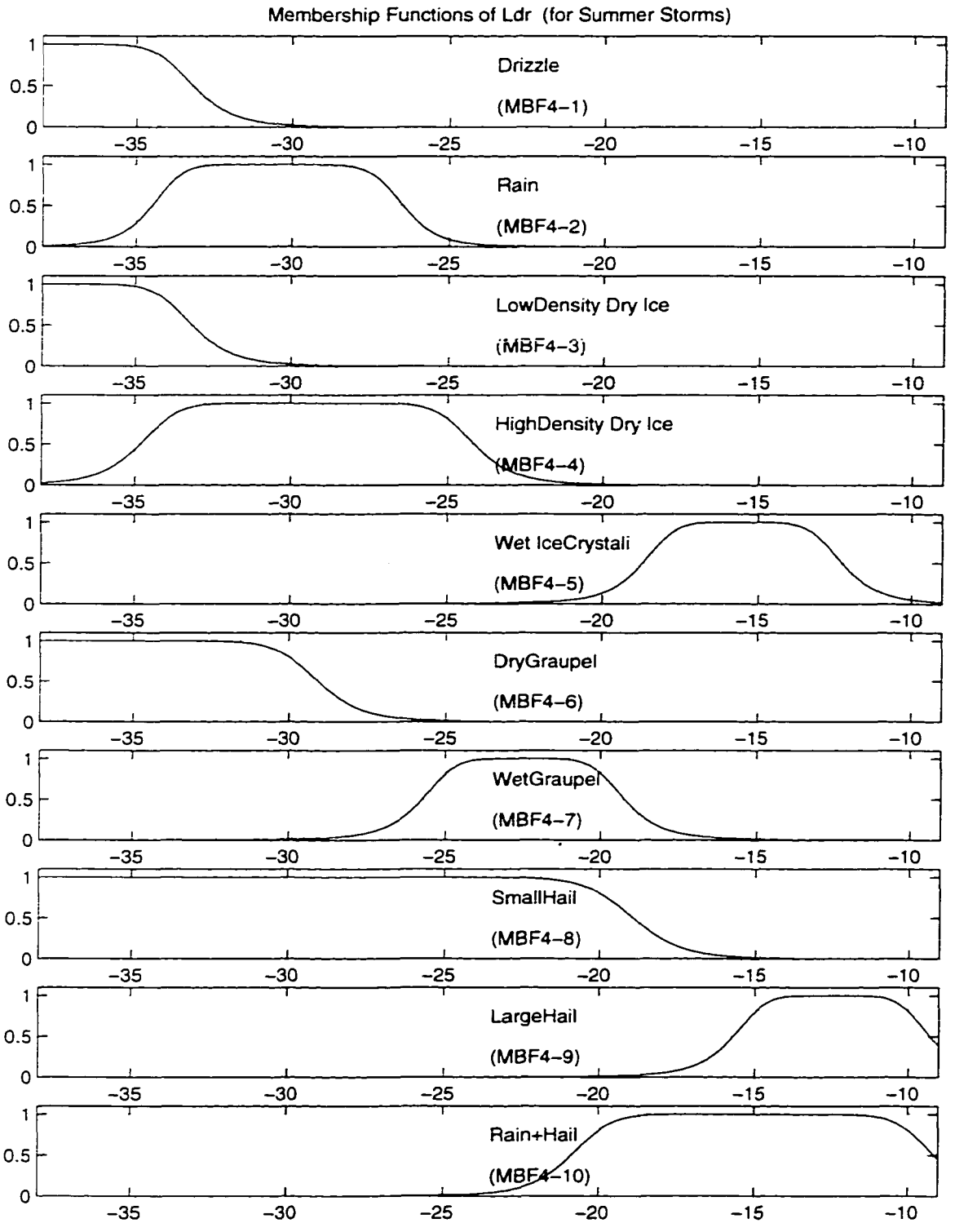


Figure 3.9: Membership functions for Fuzzy Variable *LDR*, and illustration of the fuzzification of LDR to its 10 fuzzy sets.

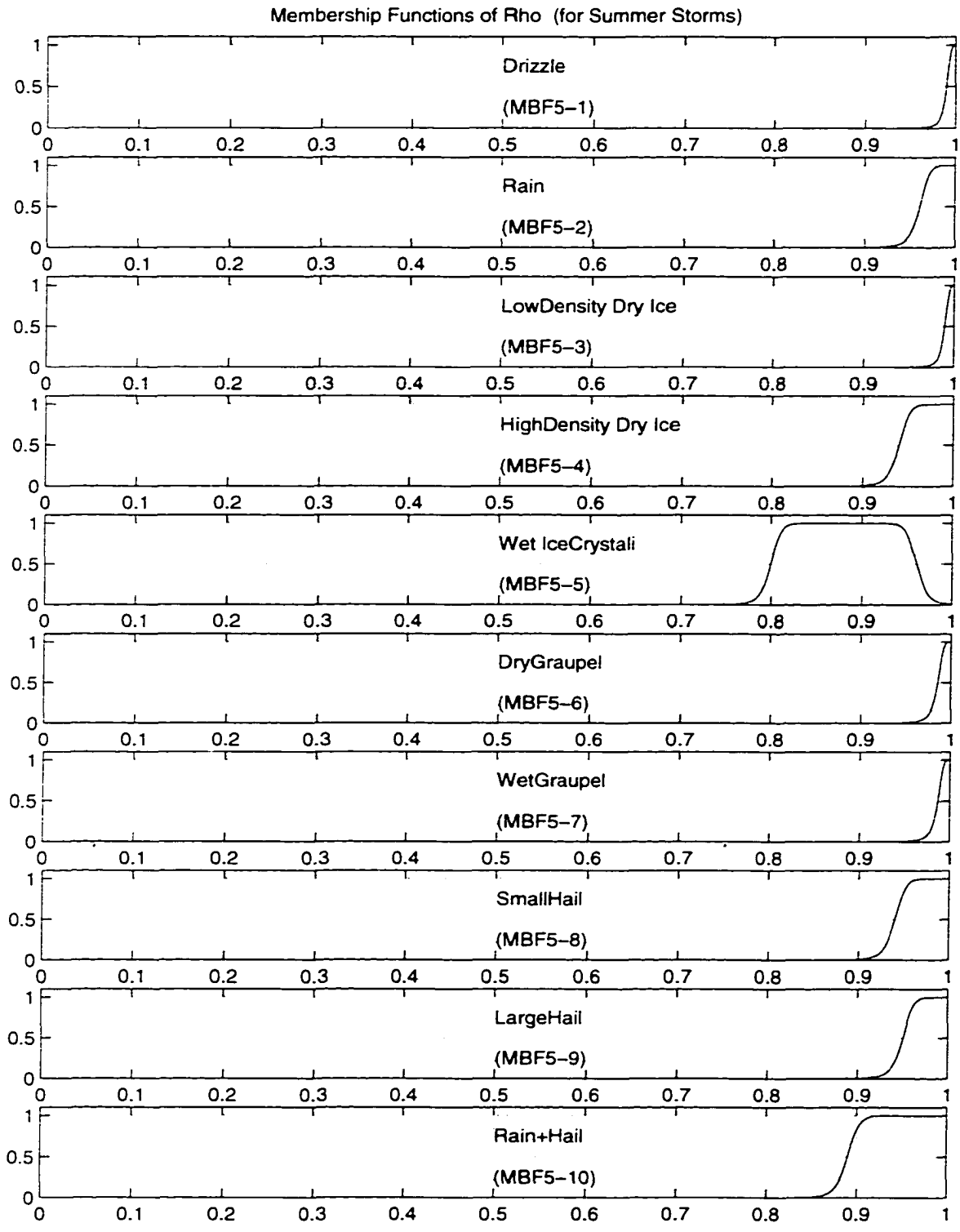


Figure 3.10: Membership functions for Fuzzy Variable ρ_{hv}, and illustration of the fuzzification of ρ_{hv} to its 10 fuzzy sets.

Table 3.6: Table for ρ_{hv} MBF Coefficients

MBF	m	a	b
MBF5-1	1.00000	0.01000	1.86000
MBF5-2	1.00009	0.03914	3.41000
MBF5-3	1.00000	0.01000	1.86000
MBF5-4	1.00000	0.06000	4.82000
MBF5-5	0.88010	0.08069	6.54000
MBF5-6	0.99924	0.01377	1.85998
MBF5-7	0.99952	0.01288	1.85999
MBF5-8	1.00000	0.06000	4.82000
MBF5-9	1.00000	0.05000	4.12000
MBF5-10	1.00000	0.11000	8.24000

dimensional membership function for rain set with respect to input variable Z_h. f_{rain_zh} can be written as.

$$f_{rain\_zh}(Z_h) = \frac{1}{1 + ((\frac{Z_h - 42.5}{18.17})^2)^{18.32}} \tag{3.3}$$

Similarly, f_{rain_zdr}, the one dimensional membership function for rain set with respect to Z_{dr}, can be written as.

$$f_{rain\_zdr}(Z_{dr}) = \frac{1}{1 + ((\frac{Z_{dr} - 2.25}{1.83})^2)^{16.22}} \tag{3.4}$$

If Z_h and Z_{dr} are treated as independent variables for rain, then in a 2 dimensional Z_h - Z_{dr} space, the membership function for rain set with respect to Z_h and Z_{dr} can be expressed as, the product of f_{rain_zh} and f_{rain_zdr}.

$$f_{rain\_zhzdr}(Z_h, Z_{dr}) = f_{rain\_zh}(Z_h) \cdot f_{rain\_zdr}(Z_{dr}) \tag{3.5}$$

$f_{rain\_zhzdr}(Z_h, Z_{dr})$ is shown in Figure 3.12(a) and its contours are shown in Figure 3.12(b).

The radar measurements Z_h and Z_{dr} are not independent in rainfall. For example, the maximum excursion of Z_{dr} is related to the value of Z_h for rain. This feature can be seen from the scatter plot of Z_{dr} vs. Z_h from rain data shown in

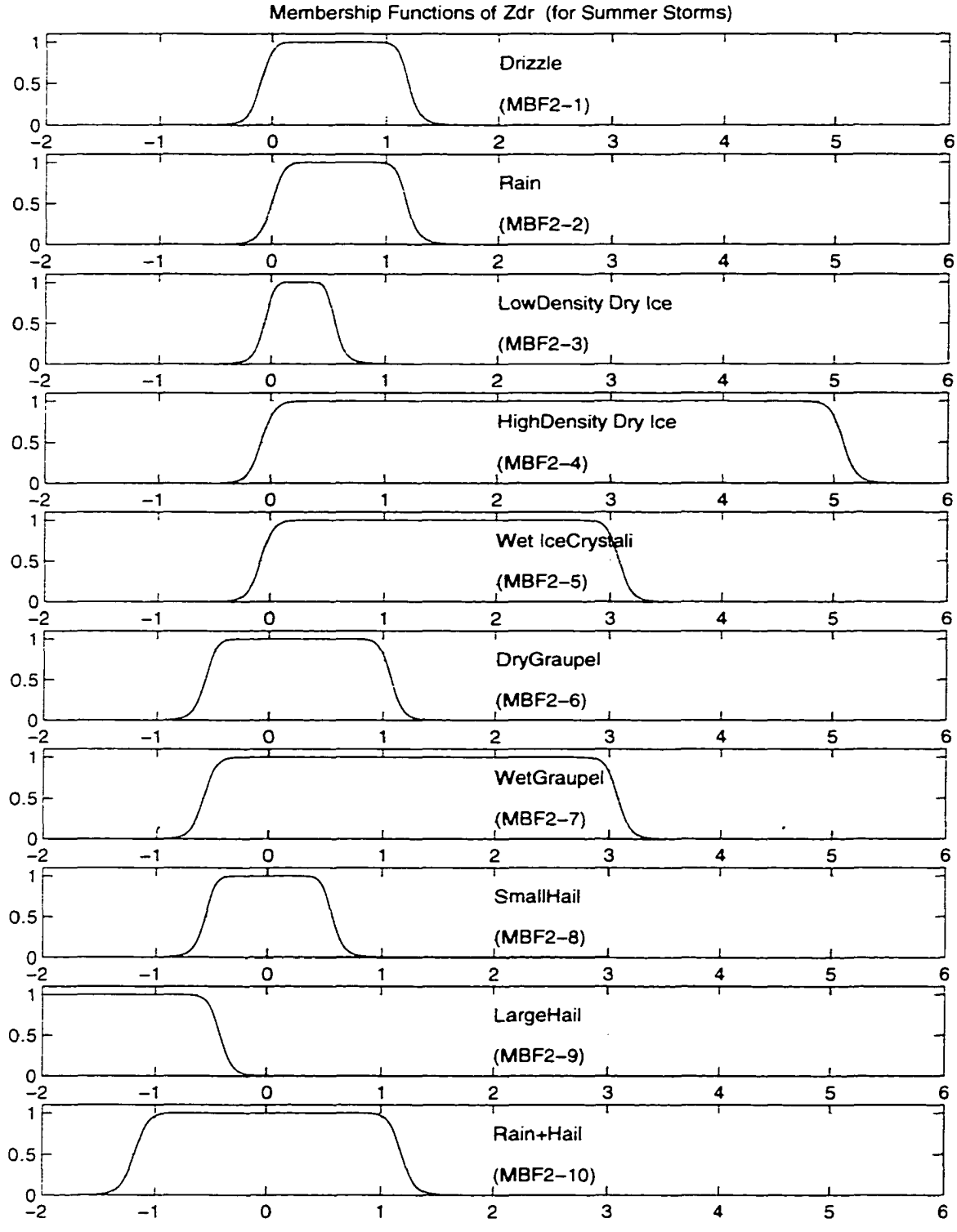


Figure 3.11: Fuzzy Sets For Differential Reflectivity Z_{dr}

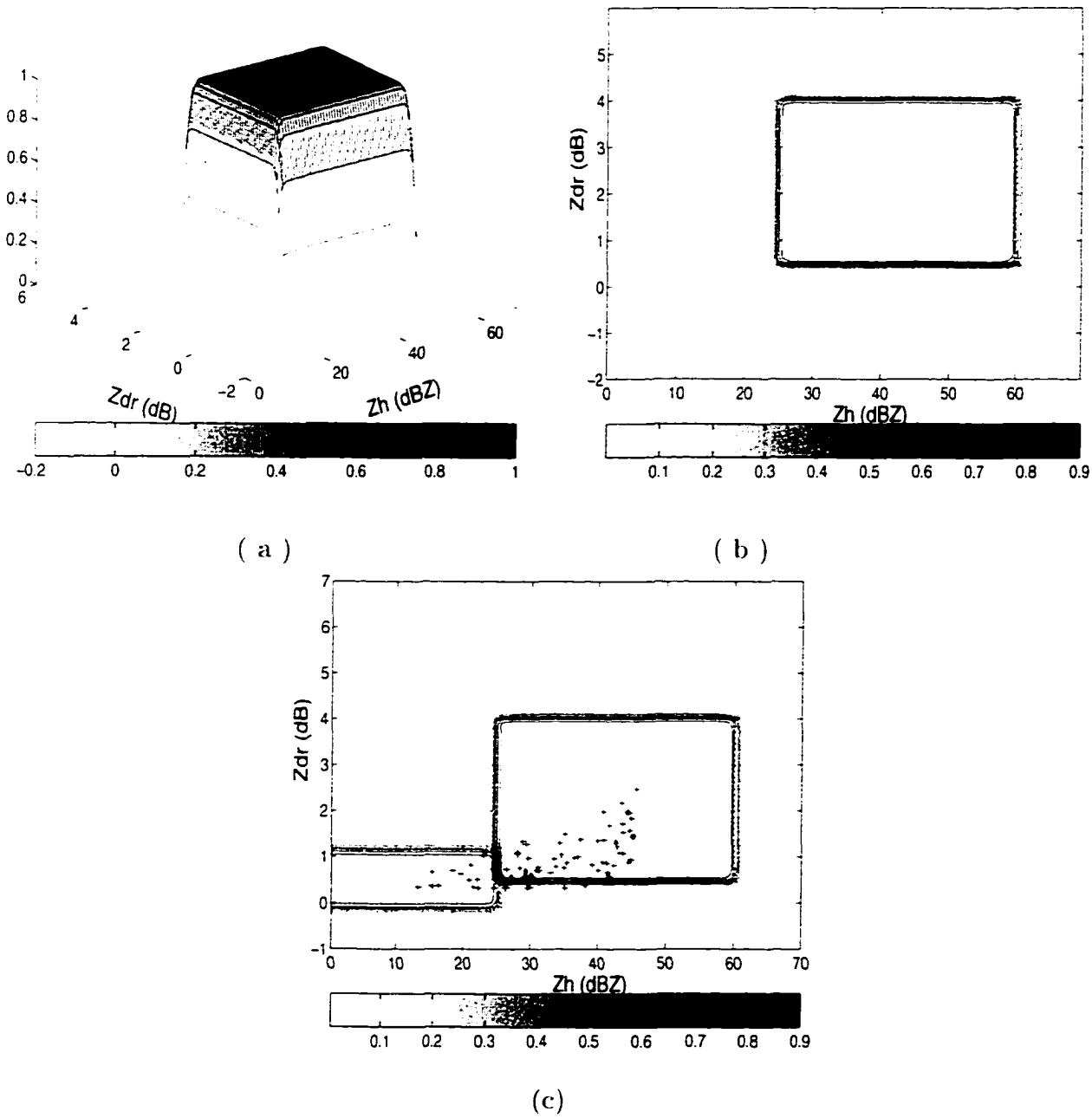


Figure 3.12: (a)2D membership function of Z_h and Z_{dr} for rain set (b) contour of the 2D membership function (c) Contour of 2D membership functions of drizzle and rain with scatter plot of Z_{dr} vs. Z_h for rain

Table 3.7: Table for Zdr MBF (1-D) Coefficients

MBF	m	a	b
MBF2-1	0.5500	0.6500	6.2480
MBF2-2	0.5950	0.5950	5.6082
MBF2-3	0.2498	0.3074	3.4985
MBF2-4	2.5000	2.5800	22.5046
MBF2-5	1.5000	1.5800	14.1307
MBF2-6	0.2500	0.8200	7.8093
MBF2-7	1.2500	1.8300	16.2198
MBF2-8	0.0000	0.5600	5.6802
MBF2-9	-2.0000	1.5800	14.1307
MBF2-10	0.0019	1.1829	9.9322

Figure 3.12(c) as "+" (Bringi et al. 1991). All the (Z_h, Z_{dr}) pairs lie in a small portion of the two dimensional Z_h, Z_{dr} domain of Figure 3.12(a). Therefore, we have to modify the 2D membership function given by(3.5).

To build the 2D membership function for the "Rain" member in the 2D fuzzy set Z_h and Z_{dr}, contour curves of 0.99 and 0.50 membership degrees in the 2D (Z_h, Z_{dr}) domain need to be specified. However, no theoretical method exists for defining these contour curves. An experimental method is adopted here, which is based on rain sample distribution in the 2D plane (as shown in Figure 3.12(c)). A group of discrete data pairs on the 0.99 membership degree contour curve and 0.50 degree contour curve is chosen, which is listed in the Table 3.8. In this table, X_1's are the points on the bottom part of the 0.99 degree contour, X_4's are the points on the top part of the 0.99 degree contour, $X_1$2's and X_3's are the points on the bottom and top part of the 0.50 degree contour, respectively.

To obtain a closed form of the 0.50 degree contour, a polynomial fitting method is applied to the points of X_1 and X_4. The fitting curve is shown in Figure 3.14, in which the circles represent those X_1 and X_4 defined in Table 3.8. The close form

Table 3.8: List of discrete points on 0.99 and 0.50 degree membership function contour

Z_h (dBZ)	X_1	X_2	X_3	X_4
20	0.0000	0.2000	0.9900	1.1900
25	0.0000	0.2000	1.2400	1.4400
30	0.1000	0.3000	1.4900	1.6900
35	0.1500	0.3500	1.7900	1.9900
40	0.2500	0.4500	2.1900	2.3900
45	0.3500	0.5500	2.6900	2.7900
50	0.4500	0.6500	3.1900	3.3900
55	0.5500	0.7500	3.5900	3.7900
60	0.6500	0.8500	4.1900	4.3900

obtained from the polynomial fitting can be expressed as following two equations. equation 3.6 and equation 3.7.

$$Z_{dr} = -0.002Z_h^2 + 0.0097Z_h^3 - 0.2031Z_h^4 + 1.5338Z_h^5 \tag{3.6}$$

$$Z_{dr} = 0.0018Z_h^2 - 0.0662Z_h^3 + 1.2315Z_h^4 - 7.8194Z_h^5 \tag{3.7}$$

Based on the 0.50 and 0.99 membership contours above, we can define a new 2D membership function of rain set with respect to Z_h and Z_{dr} as shown in Figure 3.15(a), its contours in Z_h - Z_{dr} space are shown in Figure 3.15(b). From Figure 3.15(c), we can see that the new two dimensional membership functions restrict the Z_h - Z_{dr} domain for rain set, incorporating the correlation of these two radar measurements. The membership functions of Z_{dr} for other hydrometeor type sets are represented by using simple one dimensional membership functions and their waveforms are shown in Figure 3.11.

3.4 Rule Base of the Fuzzy Hydrometeors Classifier (FHC)

Prior knowledge about the hydrometeor classification problem is incorporated in the fuzzy logic system in the form of IF-THEN rules and membership functions.

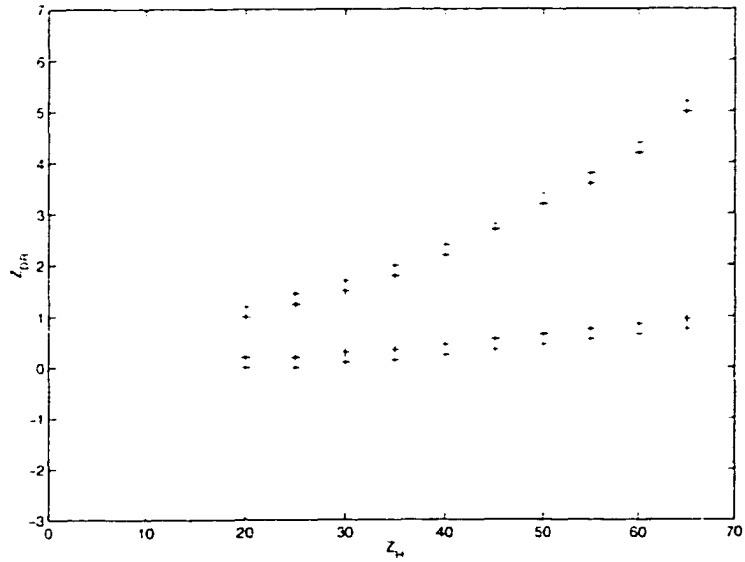


Figure 3.13: discrete points on 0.99 and 0.50 degree membership function contour

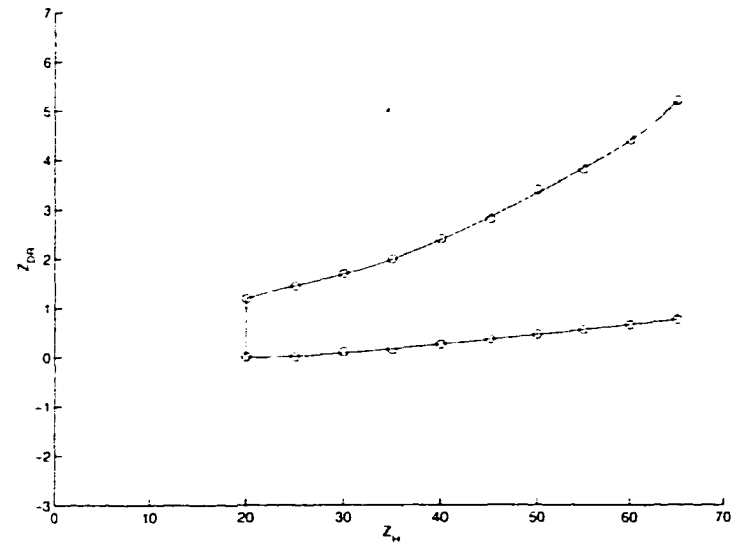
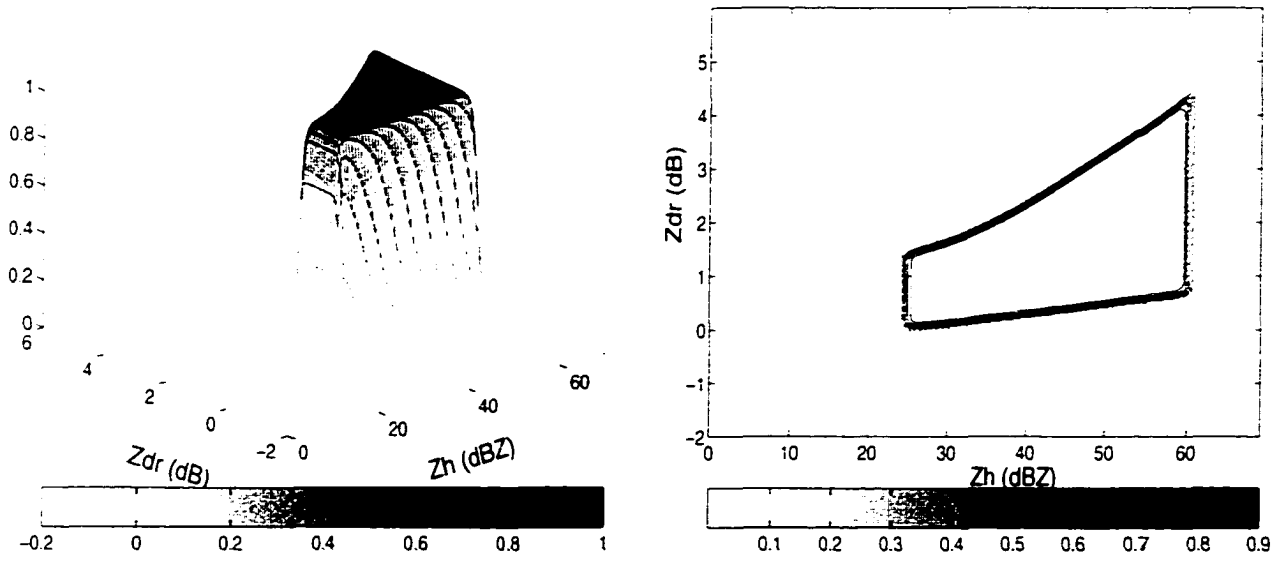
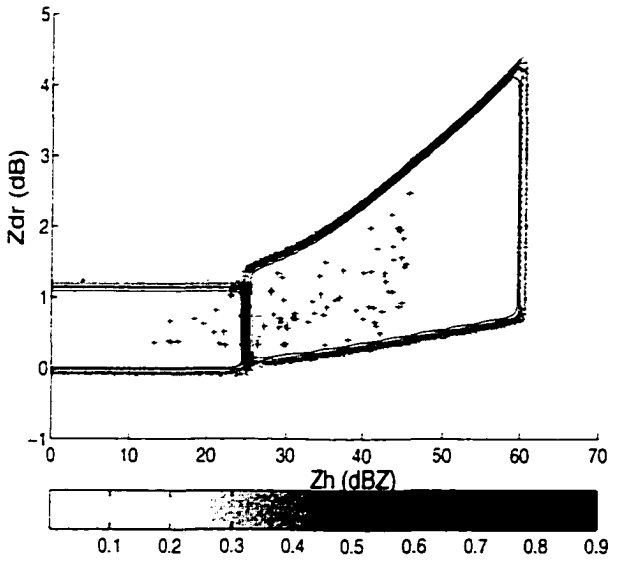


Figure 3.14: the fitting curve of the 0.50 degree contour



(a) (b)



(c)

Figure 3.15: (a) Modified 2D membership function of Z_h and Z_{dr} for rain set (b) contour of the 2D membership function (c) Scatter plot of Z_{dr} vs. Z_h for rain with the contour of the membership functions.

Table 3.9: RuleBase for the Hydrometeor Classification

IF	Zh IS MBF1_1 AND Zdr IS MBF2_1 AND Kdp IS MBF3_1 AND LDR IS MBF4_1 ρ_{hv} IS MBF5_1 AND Height IS MBF6_1	THEN	Precipitation IS Drizzle
IF	Zh IS MBF1_2 AND Zdr IS MBF2_2 AND Kdp IS MBF3_2 AND LDR IS MBF4_2 ρ_{hv} IS MBF5_2 AND Height IS MBF6_2	THEN	Precipitation IS Rain
IF	Zh IS MBF1_3 AND Zdr IS MBF2_3 AND Kdp IS MBF3_3 AND LDR IS MBF4_3 ρ_{hv} IS MBF5_3 AND Height IS MBF6_3	THEN	Precipitation IS SnowDry
IF	Zh IS MBF1_4 AND Zdr IS MBF2_4 AND Kdp IS MBF3_4 AND LDR IS MBF4_4 ρ_{hv} IS MBF5_4 Height IS MBF6_4	THEN	Precipitation IS Crystal
IF	Zh IS MBF1_5 AND Zdr IS MBF2_5 AND Kdp IS MBF3_5 AND LDR IS MBF4_5 ρ_{hv} IS MBF5_5 AND Height IS MBF6_5	THEN	Precipitation IS SnowWet
IF	Zh IS MBF1_6 AND Zdr IS MBF2_6 AND Kdp IS MBF3_6 AND LDR IS MBF4_6 ρ_{hv} IS MBF5_6 AND Height IS MBF6_6	THEN	Precipitation IS GraupelDry
IF	Zh IS MBF1_7 AND Zdr IS MBF2_7 AND Kdp IS MBF3_7 AND LDR IS MBF4_7 ρ_{hv} IS MBF5_7 AND Height IS MBF6_7	THEN	Precipitation IS GraupelWet
IF	Zh IS MBF1_8 AND Zdr IS MBF2_8 AND Kdp IS MBF3_8 AND LDR IS MBF4_8 ρ_{hv} IS MBF5_8 AND Height IS MBF6_8	THEN	Precipitation IS HailSmall
IF	Zh IS MBF1_9 AND Zdr IS MBF2_9 AND Kdp IS MBF3_9 AND LDR IS MBF4_9 ρ_{hv} IS MBF5_9 Height IS MBF6_9	THEN	Precipitation IS HailLarge
IF	Zh IS MBF1_10 AND Zdr IS MBF2_10 AND Kdp IS MBF3_10 AND LDR IS MBF4_10 ρ_{hv} IS MBF5_10 AND Height IS MBF6_10	THEN	Precipitation IS RainHail

The IF-THEN rules for this hydrometeor type classification can be written as follows:

IF (Zh IS MBF1_j AND Zdr IS MBF2_j AND Kdp IS MBF3_j AND LDR IS MBF4_j AND ρ_{hv} IS MBF5_j AND Height IS MBF6_j) THEN Hydrometeor Class is j

where j=1.2. ... . 10 correspond to the classes listed in 3.1. The rule base for the fuzzy hydrometeor classifier is listed in Table 3.9.

3.5 Rule Inference Method

The strength for the six antecedent propositions could be obtained from the fuzzification block as PS_{i_j} where the index i represents the five measurements and the altitude. index j represents the classes. For example, PS_{1_2} is the strength of reflectivity for rain. In this paper. "*product intersection operation*" is used to find the strength of the IF-SIDE, and "*Correlation Product*" inference method is used to find the rule strength. The truth value of the IF-SIDE is used to scale the consequent fuzzy set. In this case, the strength of the rule is equal to the strength of the IF-SIDE because the output is singleton. Therefore, the strength of rule j (RS_j) can be obtained as the product of the strength of individual propositions as.

$$RS_j = \prod_{i=1}^{6} PS_{i\_j}. \tag{3.8}$$

3.6 Aggregation and Defuzzification

"MAX Aggregation" method is used to determine net fuzzy result from the individual rule inference results. Max Aggregation procedures take only the consequent with the highest truth value. Therefore the aggregation result is the maximum rule strength of the various RS_j defined in(3.8).

For the FHC, the output is singleton. A simple method is used to defuzzify the output in order to get a singleton result, namely the index of the rule with maximum rule strength.

3.7 Neuro-Fuzzy hydrometeor type classifier

The performance of the FHC depends critically on the shape of the membership functions. A combination of empirical and theoretical knowledge of current state of the art was used to construct the membership functions. Manually adjusting the functions is tedious and inefficient. It would be very useful to develop a system that has the ability to learn from data and adjusts the membership functions automatically. This is achieved by the neuro-fuzzy hydrometeor type classifier developed in this section.

3.7.1 Configuration of neuro-fuzzy hydrometeor types classifier

There are many types of combination of neural network and fuzzy logic classification systems. Essentially, the combinations can be categorized into *cooperative neuro-fuzzy models* and *hybrid neuro-fuzzy models*. In the category of *cooperative neuro-fuzzy models*, the neural network method and fuzzy logic system are used separately, neural network technique is used offline/online to determine the parameters of the fuzzy logic classification system. The parameters of the fuzzy logic classification system include the parameter in fuzzy sets, the fuzzy rules, the weight factors of each rule, therefore, the *cooperative neuro-fuzzy network* can be further categorized as the four types shown in the Figure 3.16.

In panel (a), the neural network deduces the optimum parameters of membership functions in the the fuzzy set, this parameter optimization problem belongs to function approximation problem for the neural network to learn and the learning process is offline. In panel (b), the neural network derives the fuzzy classification

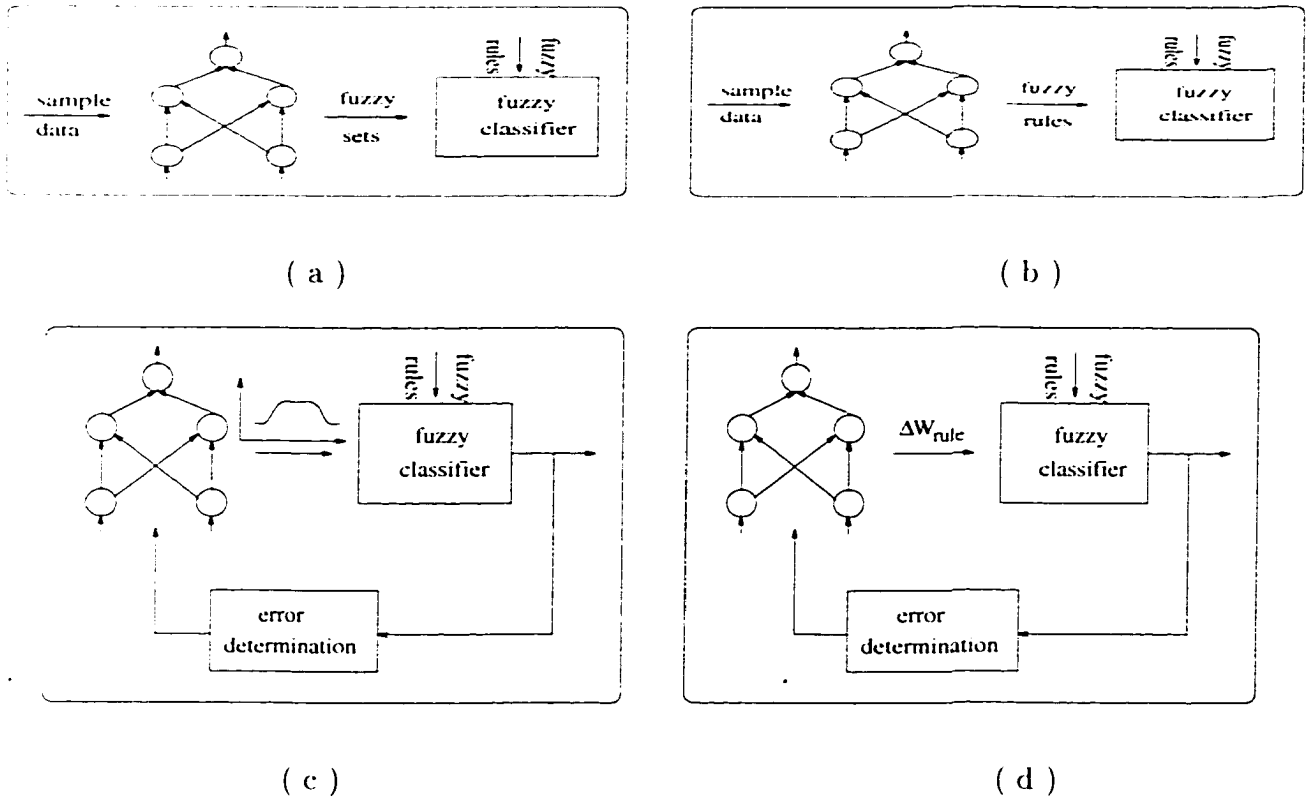


Figure 3.16: Cooperative Neuro-Fuzzy Models (Detlef Nauchk, 1997)

rules for the system offline, the precondition is the fuzzy sets are known. Usually, unsupervised learning method such as SOM learning algorithm is used to cluster the data and therefore to derive the fuzzy rules. In panel (c), the neural network learns the fuzzy set online, and in the panel (d), the neural network learns the optimum weight factor for rules offline.

The other combination of a neural network and fuzzy logic system is called hybrid neuro-fuzzy model, its basic configuration for classification is shown in Figure 3.17. In this model, the Fuzzy system can be viewed as a multilayer feedforward neural network. Usually, it has five layers: the input layer (consisting of input variables), the IF layer (or called fuzzification layer), the THEN layer (or called RULE inference layer), the aggregation/defuzzification layer, and finally, the output layer. In Figure 3.17, the aggregation/defuzzification and output layer are combined as one layer, such that, the resulting architecture looks like a four layer neural network. The neuro-fuzzy network can be regarded as a special neural network with special neurons and connections between layers. Unlike a fuzzy system, it uses a learning algorithm derived from or inspired by neural network concept to learn its parameters (fuzzy sets and fuzzy rules).

For the specific application of hydrometeor classification, the fuzzy rulebase is designed already, and very reliable. The only parameters that need to be adjusted are the fuzzy sets. The hybrid neuro-fuzzy classification model is chosen for fine-tuning the parameters of the membership functions designed in section 3.3.

In this system, the fuzzy logic part can be modeled as a multilayer feedforward neural network, with five layers, namely, the input layer (consisting of input variables), the IF layer (fuzzification layer), the THEN layer (RULE inference layer), the aggregation/defuzzification layer, and the output layer. Under this model, the neural network learning algorithms can be used to learn the parameters of the system. One implementation of the neuro-fuzzy hydrometeor classifier (NFHC)

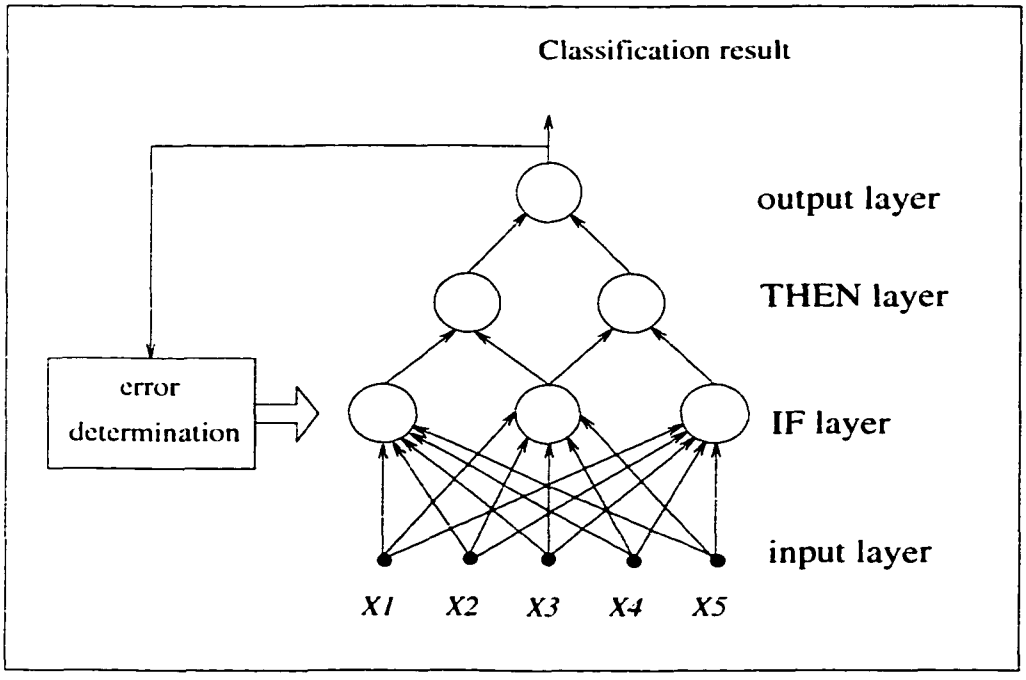


Figure 3.17: Hybrid Neuro-Fuzzy Classification Model(Detlef Nauchk, 1997)

is shown in Figure 3.18. In Figure 3.18, if we ignore the blocks shown by dotted lines, it is the same configuration as the FHC described in section 2. The solid lines form the feedforward path, and the dotted lines form the backpropagation path. The mis-classification error is back-propagated to the IF-layer and is used to adjust the parameters of the membership functions. This is a very effective and efficient procedure to build a good NFHC over time.

3.7.2 Fuzzy Set Learning Algorithm

The learning algorithm for the fuzzy logic system is as follows: let **P** be the vector of radar measurements and altitude (Z_h, Z_{dr}, K_{dp}, LDR, ρ_{hv}, and H) and let C_T be the known class of hydrometeor for this measurement set denoted as target class.

(i) Apply the set **P** to the FHC and it produces an output C.

(ii) Determine the output error as $\delta = C_T - C$

(iii) Go to (i) if $\delta = 0$, otherwise, adjust the membership functions according to the procedure given in (iv).

(iv) The adjustment of the membership functions is performed only on the fuzzy sets corresponding to class C_T and C, which are the target output class and the current output class, respectively, and their rule strength relationship is given by $RS_{C_T} < RS_C$. The adjustment of membership functions should be in the direction that makes $RS_{C_T} > RS_C$. '*Gradient Descent Learning*' method is used to tune the parameters for the membership functions related to Class C_T and C. The details of the procedure used in the adjustment of membership functions are described in the following.

(v) If the error goal is not met, go back to step (i), otherwise, terminate the learning process.

3.7.2.1 Procedure for Adjusting the Membership Function

(i) From the set of $MBFi\_C_T$ ($i \in [1,6]$), find the MBF with the smallest truth degree,

$$PS_{k\_C_T} < PS_{i\_C_T}, \ where \ i \in [1,6] \tag{3.9}$$

For the membership function $MBF_{k\_C_T}$, determine the delta values for its parameter m, a, b, as

$$\delta_m = l_m(-1)(\frac{\partial RS_{C_T}}{\partial PS_{k\_C_T}})(\frac{\partial PS_{k\_C_T}}{\partial m}) \tag{3.10}$$

$$\delta_a = l_a(-1)(\frac{\partial RS_{C_T}}{\partial PS_{k\_C_T}})(\frac{\partial PS_{k\_C_T}}{\partial a}) \tag{3.11}$$

$$\delta_b = l_b(-1)(\frac{\partial RS_{C_T}}{\partial PS_{k\_C_T}})(\frac{\partial PS_{k\_C_T}}{\partial b}) \tag{3.12}$$

where l_m, l_a and l_b are the learning rate for the three parameters $m.a.b$; RS_{C_T} is the rule strength for rule C_T, and the derivatives for $\frac{\partial PS_{k\_C_T}}{\partial m}$, $\frac{\partial PS_{k\_C_T}}{\partial a}$ and $\frac{\partial PS_{k\_C_T}}{\partial b}$ can be obtained by the following equations.

$$\frac{\partial PS_{k\_C_T}}{\partial m} = (PS_{k\_C_T})^2 \frac{2b}{a}[(\frac{x_k - m}{a})^{2b+1}] \tag{3.13}$$

$$\frac{\partial PS_{k\_C_T}}{\partial a} = (PS_{k\_C_T})^2 \frac{2b}{a}[(\frac{x_k - m}{a})^2]^b \tag{3.14}$$

$$\frac{\partial PS_{k\_C_T}}{\partial b} = (-1)(PS_{k\_C_T})^2[(\frac{x_k - m}{a})^2]^b log[(\frac{x_k - m}{a})^2]^b \tag{3.15}$$

where x_k is the kth crisp input variable. Note that these equations are based on 1-dimensional membership functions. For 2-dimensional membership functions, we can get similar expression for these derivatives by converting two dimensional membership functions to one dimensional membership functions.

After having all the delta values, the three parameters in the membership function can be updated as followings.

$$m^{new} = m^{old} + \delta_m \tag{3.16}$$

$$a^{new} = a^{old} + \delta_a \tag{3.17}$$

$$b^{new} = b^{old} + \delta_b \tag{3.18}$$

(ii) Repeat the same procedure as (i) for the membership functions related to the current output class C, therefore the updating of membership functions for class C is performed in the same way, but in the direction which makes RS_C decrease.

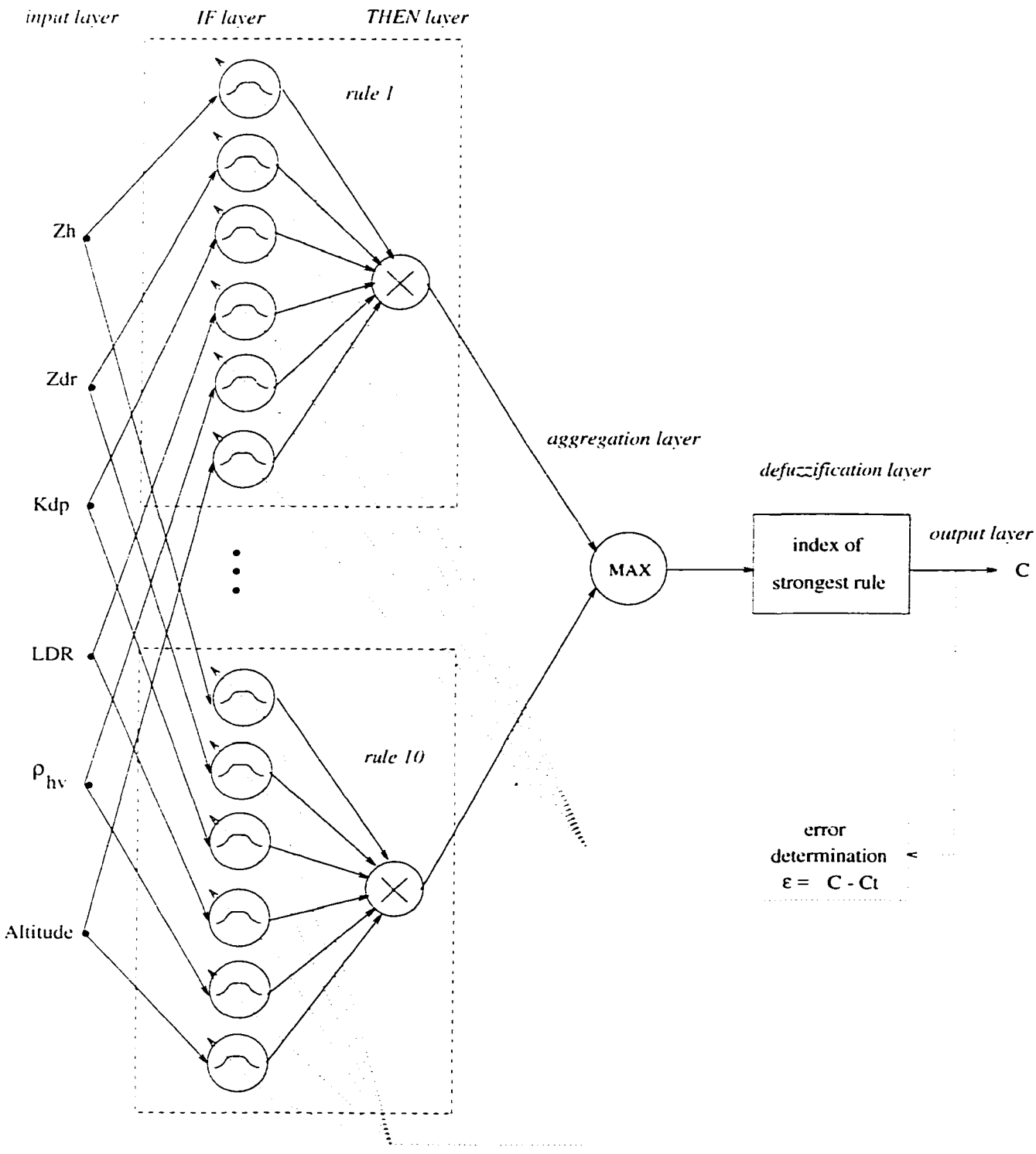


Figure 3.18: A neuro-fuzzy System for Hydrometeor Type Classification

Chapter 4

IN-SITU VERIFICATION OF NEURO-FUZZY HYDROMETEOR CLASSIFICATION

4.1 Data Sources and Instrumentation

(i) CSU-CHILL Radar

The radar data used in this study were collected by the CSU-CHILL radar. CSU-CHILL is a S-band, dual polarization radar, which can measure a full set of polarimetric measurements. Most of the data analyzed in this paper were at ranges less than 70 km from the radar.

(ii) In-Situ Observations

Hail Chase Van

An instrumented hail chase van with a roof-mounted net was constructed to intercept the storms. This chase van is equipped with a rain/hail separator and a Young capacitance raingage. Hail was collected and quenched in chilled hexane and then stored in dry ice. The hail stones were eventually photographed using digital camera for post analysis. A detailed description of the hail chase van is given in Hubbert et al (1998).

T-28 aircraft

High Volume Particle Spectrometer (HVPS) Images were collected by using a HVPS probe mounted on the T-28 aircraft, operated by south Dakota school of

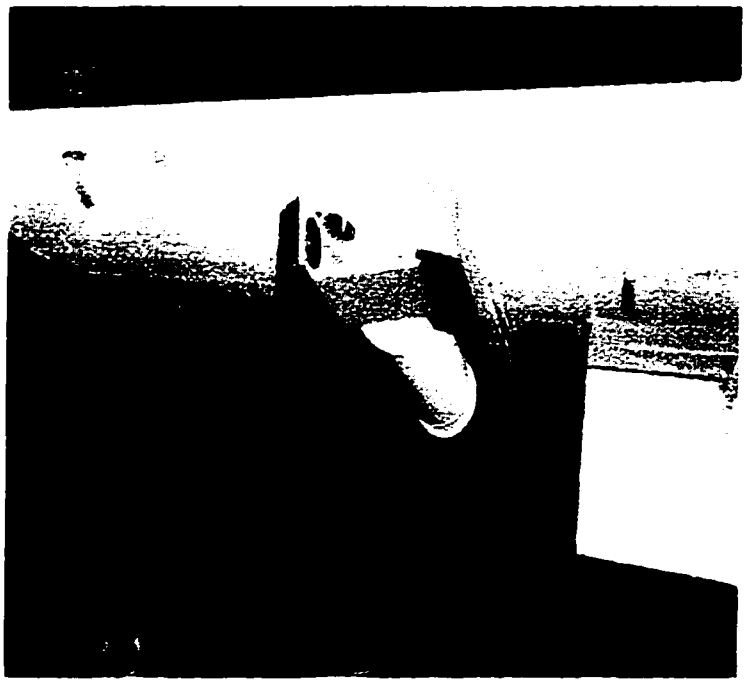


Figure 4.1: HVPS probe installed under the right wing of the T-28 aircraft

Mines and Technology. The HVPS is capable of measuring particles size up to 4.5 cm. by taking two-dimensional digital pictures of hydrometeors that pass through a $4.5cm \times 20cm$ plane which is normal to the direction of aircraft flight. This plane is a curtain of light that is projected onto a 256 pixel, linear array which is sampled at a rate proportional to the speed of the aircraft. The pixel spacing in the sample plane is 0.2 mm. As particles pass through the light plane, they create shadows on the linear array that are converted by a one-bit analog to digital converter. Thus, as a particle passes through the sample plane, the sequential slides produces a two-dimensional image of the particle. Figure 4.1 shows the high volume spectrometer installed under the right wing of the T-28.

4.2 In-situ Verification of Hydrometeor Classification

(i) The severe hail storm on June 7, 1995

On 7 June 1995, the CSU-CHILL radar observed a supercell structure, and a chase van with a roof-mounted hail collector net was sent to intercept the storm core and collect in-situ measurements. The 7 June 1995 storm turned out to be a severe hail storm. Figure 4.2 shows the radar measurements Z_h, Z_{dr}, K_{dp}, LDR, and ρ_{hv}, and Figure 4.2 also shows the the classification result from the neuro-fuzzy classification system. We can see from the hydrometeor classification result that hail and rain mixture, wet graupel, and rain are found on the ground as well as at low altitude, whereas ice crystals were inferred at high altitude. The vertical structure of the storm can be seen fairly well from the classification result. The in-situ ground observations were consistent with the result of the neuro-fuzzy Hydrometeor classifier. The hail chase van collected some hailstones, and also observed rain mixed with in the storm. The in-situ ground observations for this case are given in Hubbert et al (1998). Hubbert et al (1998) have presented detailed dual Doppler analysis of this case. The Fuzzy classification results agree well with inference of Hubbert et al (1998). Figure 4.3 shows the time series of the measurements at the hail chase van location ($y = -4.0$km) for 50 minutes. Figure 4.3 also shows the hydrometeor classification inferred by the Hydrometeor Classification System, and the "ground truth" observed by the hail chase van. It can be seen from the results of Figure 4.3 that the automatic classification result agree fairly well with in the limits of comparison between radar and ground observations.

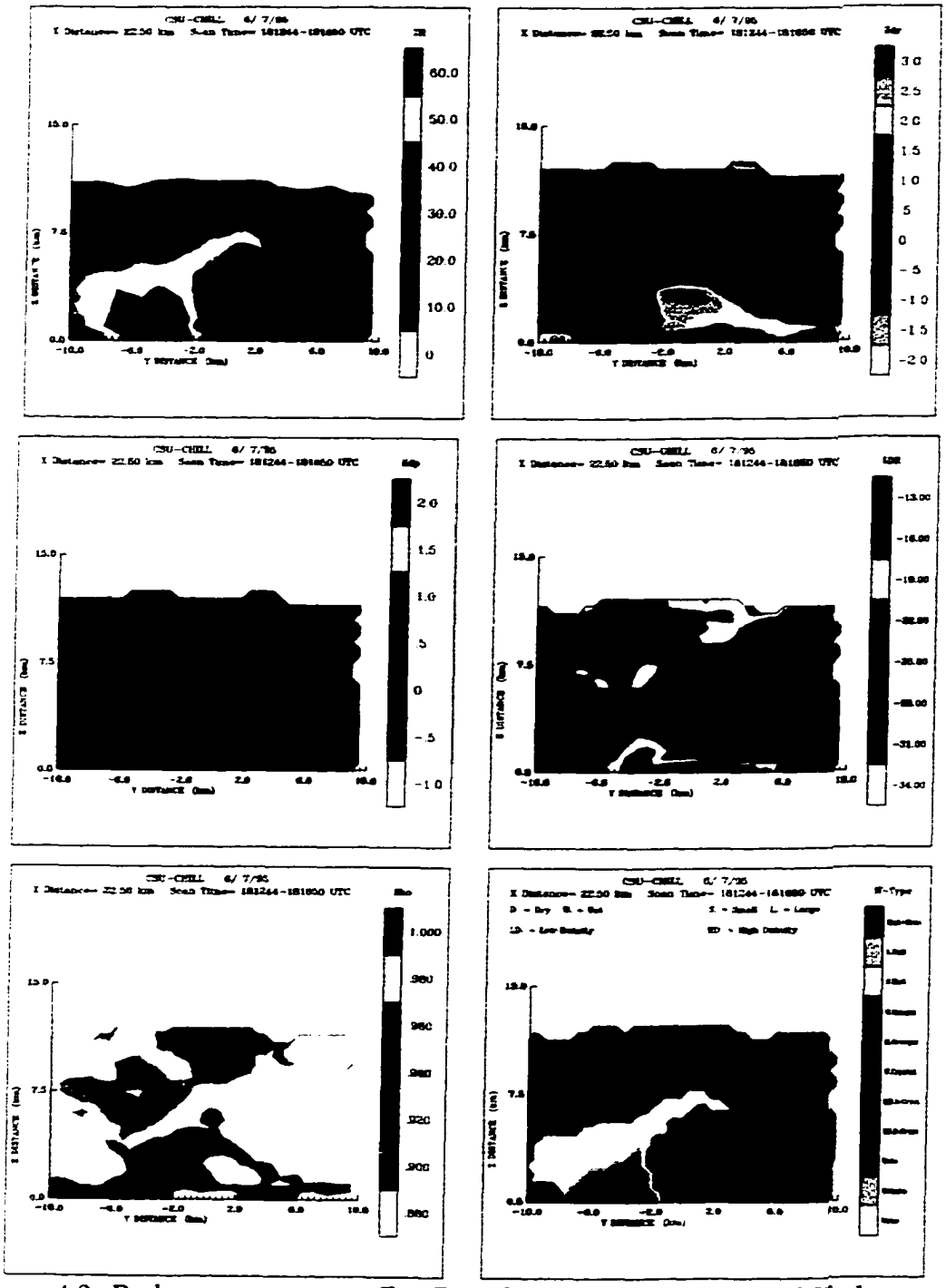


Figure 4.2: Radar measurements Z_H, Z_{DR}, K_{DP}, LDR and ρ_{HV}, and Hydrometeor Type Classification result corresponding to the case of June 7, 1995

(ii) The Convective Storm on June 22, 1995

A severe Hailstorm occurred on June 22, 1995 near Fort Collins, Colorado. This storm grew to a height of 12.5 km and upon collapsing produced heavy rain and hail of maximum sizes 3-4 cm. The intense part of the storm was located at distance of 45 to 50 km to the North East of the CSU-CHILL radar around 17:30:29 local time. The radar continuously scanned the storm approximately with 2 minute resolution for about an hour. At the same time the T-28 aircraft made several penetrations through the storm collecting samples of hydrometeors. The flights were at altitudes between 2.5 and 3.5 km above ground level to collect data in small hail region. The storm was characterized by heavy rain mixed with hail. A CAPPI (Constant Altitude PPI) of the radar measurements Z_h, Z_{dr}, K_{dp}, LDR, and ρ_{hv} at the height of the aircraft is shown in Figure 4.4. Also shown in Figure 4.4 is the automatic hydrometeor classification result. The solid line on Figure 4.4 is the flight track of T-28 aircraft. It can be seen from Figure 4.4 that T-28 track was mostly in the region of small hail and graupel. Figure 4.5 shows time series of automatic hydrometeor classification encountered along the T-28 flight penetration shown in Figure 4.4. Note here that the time series correspond to the distance along the aircraft track (from (x:14.1 km, y: 39.7 km) to (x:16.96km, y:42.01 km)). Figure 4.6 and Figure 4.7 show sample HVPS images for the data shown in Figure 4.5. Figure 4.8 shows the comparison of automatic hydrometeor classification and "in-situ" observations from T-28 HVPS data. From Figure 4.8, We can see fairly good agreement between automatic classification and in-situ observations.

(iii) The Convective Storm on June 20, 1995

On June 20, 1995, the CSU-CHILL radar observed a storm cell that formed 20 km east of the radar. A T-28 aircraft carrying HVPS probe penetrated through

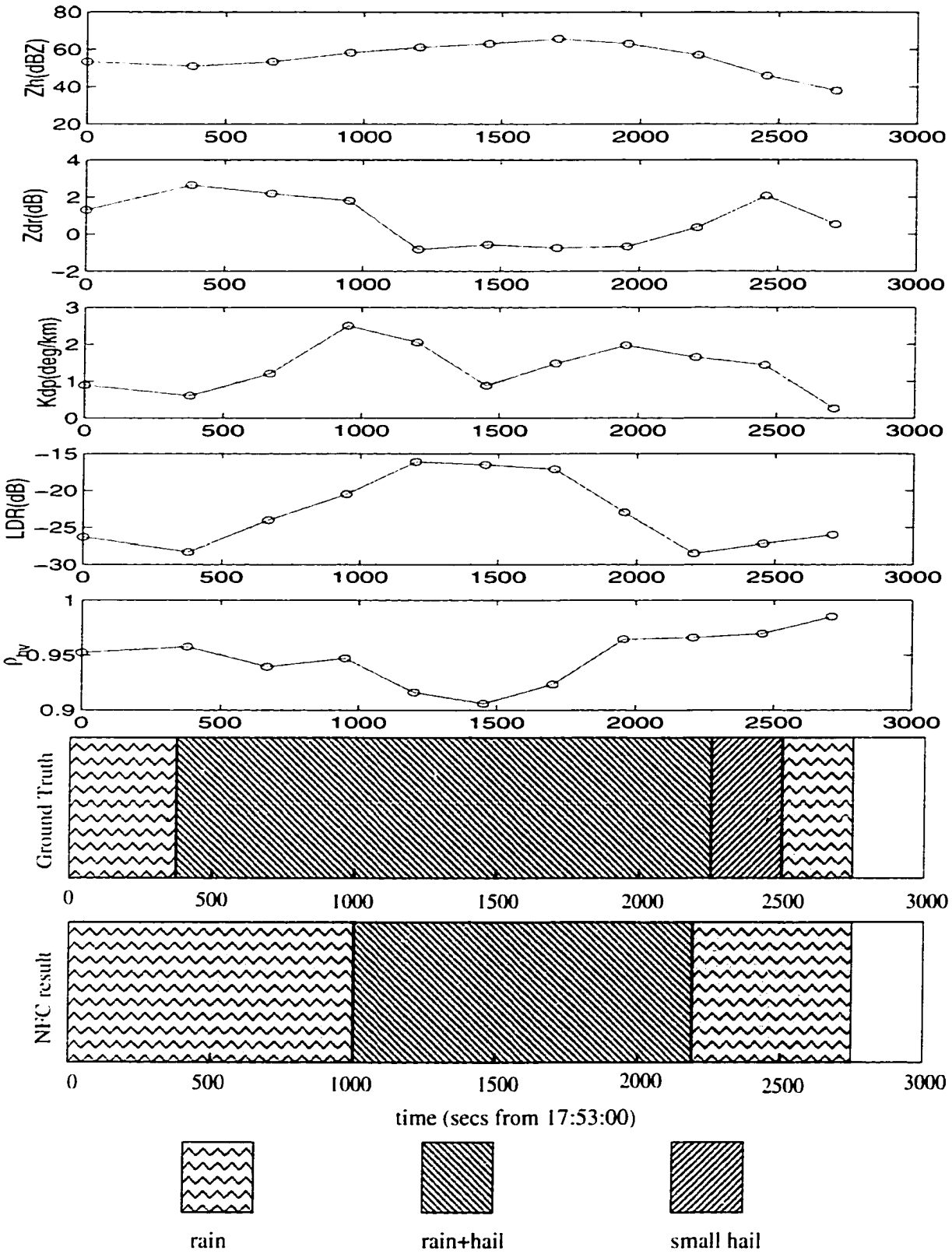


Figure 4.3: Comparison classification result from fuzzy logic system with ground truth

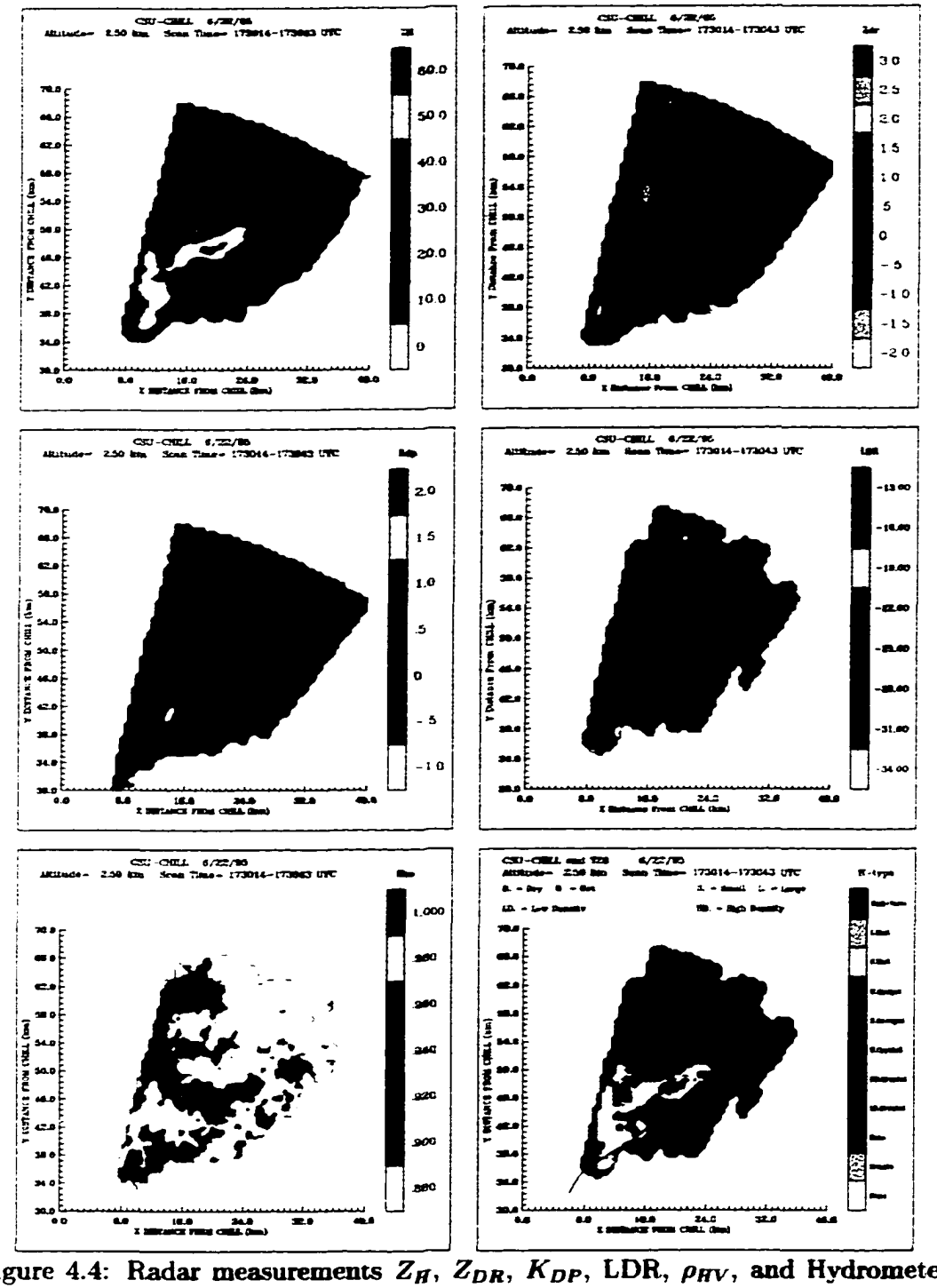


Figure 4.4: Radar measurements Z_H, Z_{DR}, K_{DP}, LDR, ρ_{HV}, and Hydrometeor Type Classification result corresponding to the case of June 22, 1995

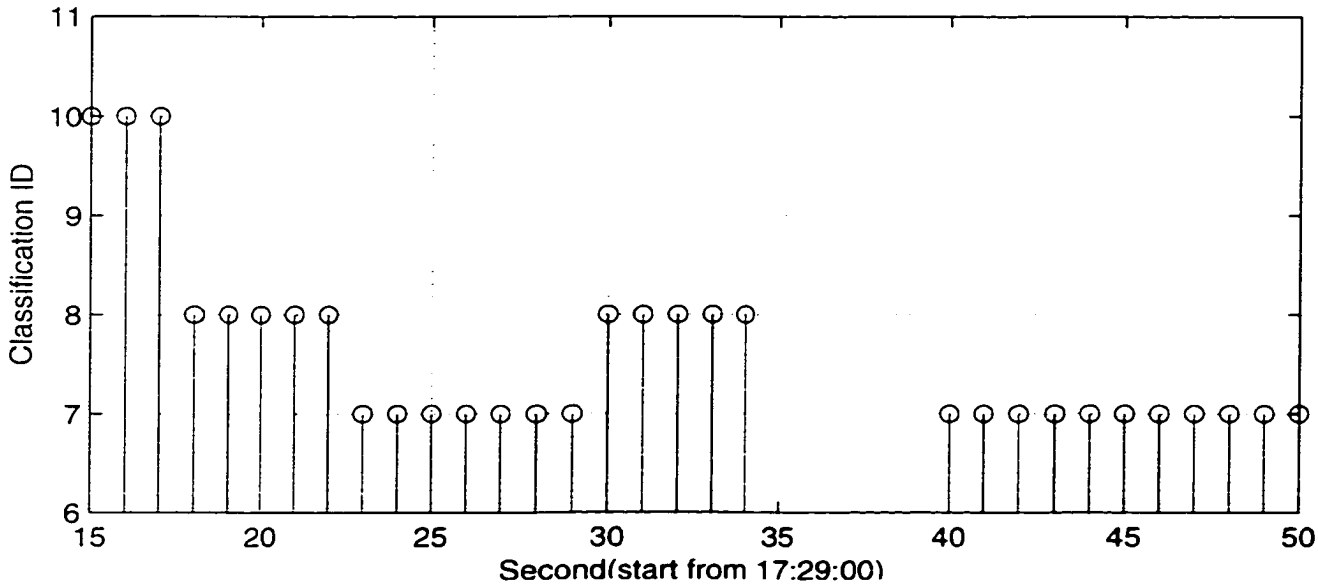


Figure 4.5: classification result along the aircraft track (from (x:14.1km, y:39.7km) to (x:16.96km, y:42.01km) on Figure 15) during the first penetration in 22 June 1995 storm.

the core of the storm to collect in-situ observations. Most of the aircraft flights through the storm were at a constant altitude of 4 km above ground. During the flight the HVPS collected samples of hydrometeors along the path. The Figure 4.9 shows a CAPPI of the radar measurements at an altitude of 4km (radar scan time is from 16:38:17 to 16:39:14 UTC). The five radar polarimetric fields is shown in Figure 4.9, and the hydrometeor classification result from the NFHC are shown in Figure 4.9. the solid line is the T-28 aircraft track. From Figure 4.9, it can be seen that the aircraft tracks are in predominantly graupel region and a region of small hail. This inference agrees very well with what the T-28 in-situ aircraft observations obtained from the HVPS probe. Sample images of hydrometeors from HVPS for the flight at time 16:38:34 UTC are shown in Figure 4.10, the corresponding T-28 aircraft location is at 30.33 km in x direction and 11.80 km in y direction with respect to the location of the CSU-CHILL radar. We can see the presence of conical graupel particles in the images.

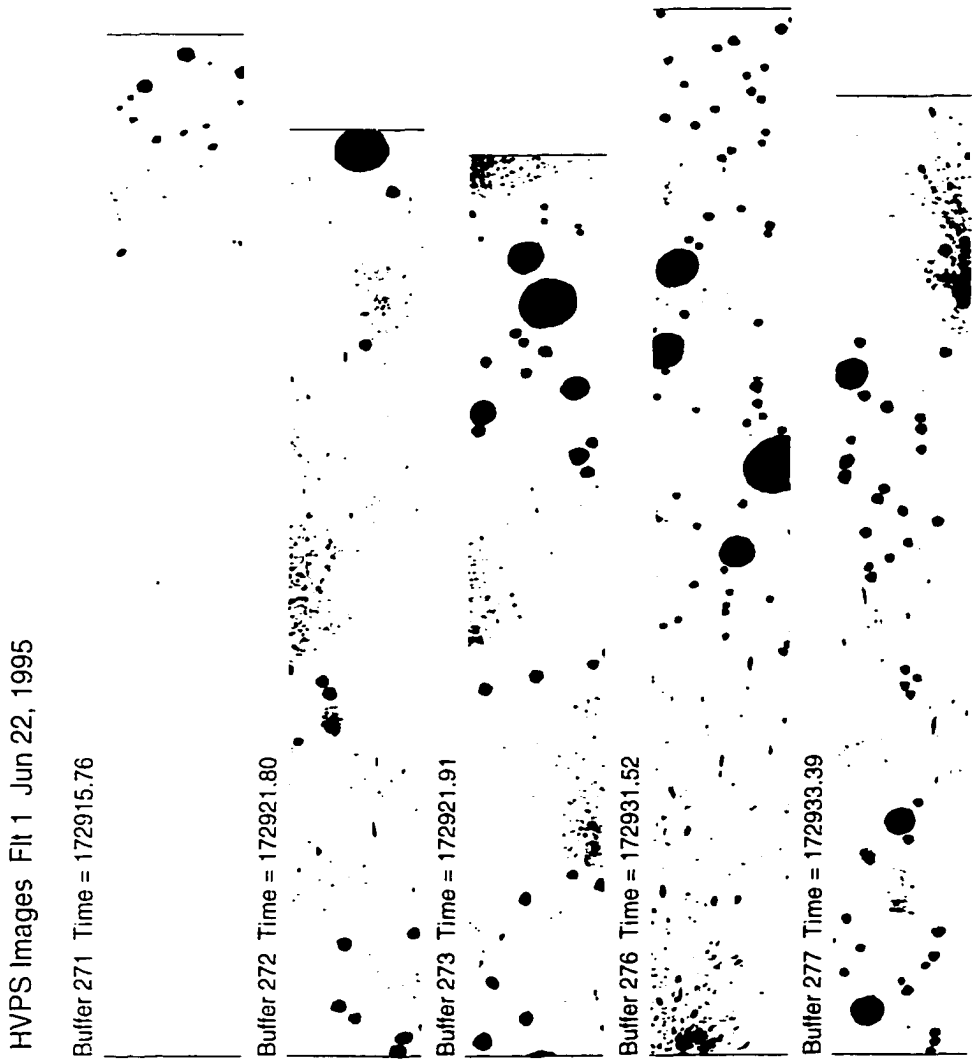


Figure 4.6: HVPS images from 17:29:15 to 17:29:33 along the aircraft track during the first penetration in 22 June 1995 storm.

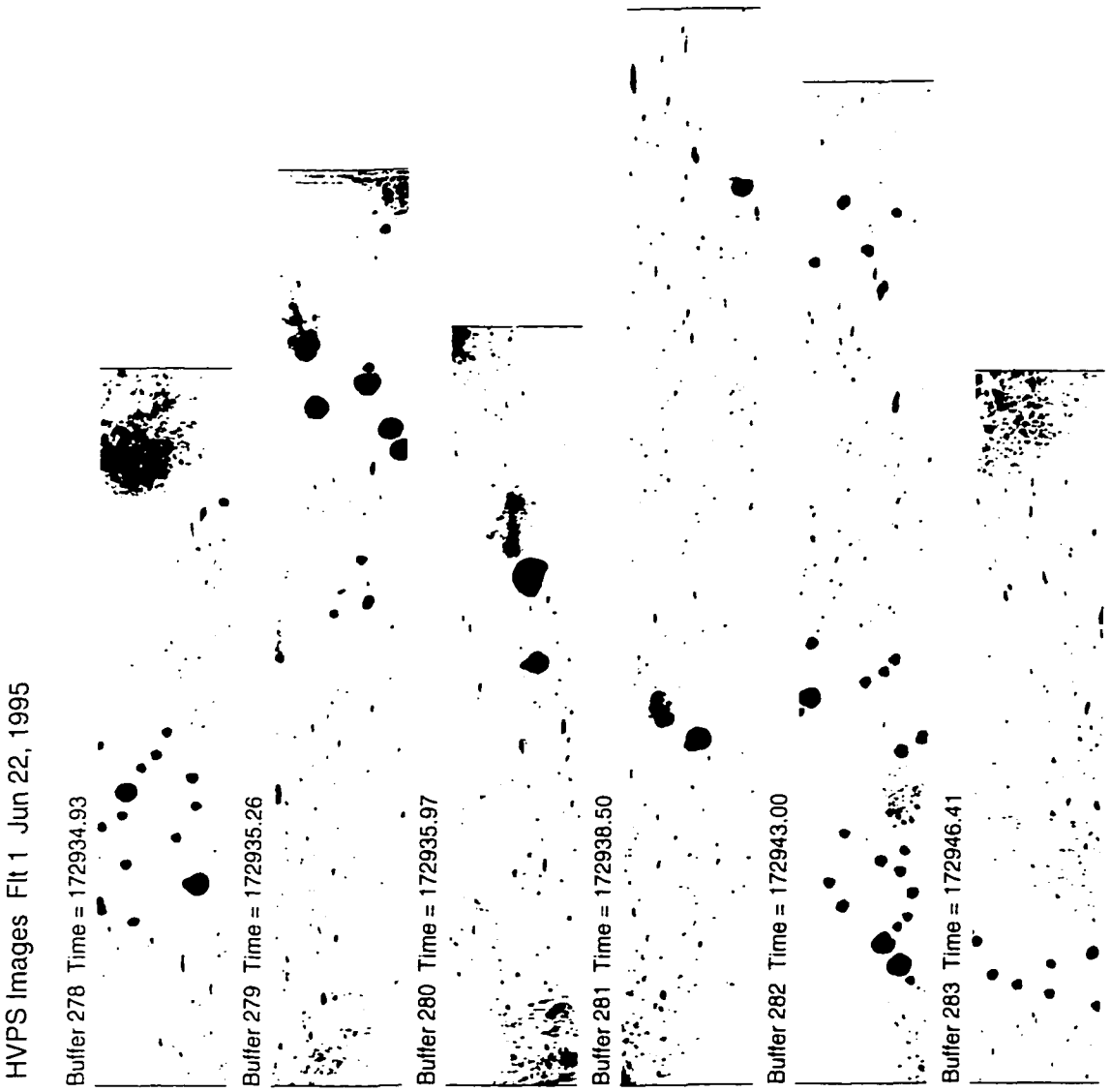


Figure 4.7: HVPS images from 17:29:34 to 17:29:50 along the aircraft track during the first penetration in 22 June 1995 storm.

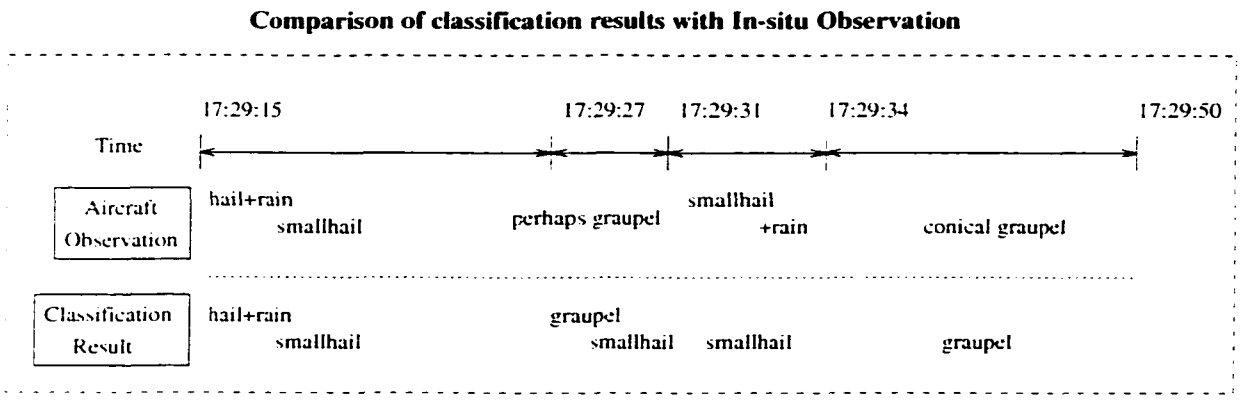


Figure 4.8: Comparison of the classification results with in-situ observation along the aircraft track during the first penetration in 22 June 1995 storm. The time series corresponding to the aircraft track data shown in Figure 4.4.

(iv) The Snow Storm on February 18, 1997

On February 18, there was a light snow event in the vicinity of the CSU-CHILL radar. Figure 4.11 and 4.12 shows a vertical section of the CSU-CHILL radar measurements through this storm. The radar measurements of Z_H, LDR and ρ_{HV} and the hydrometeor classification result are shown in Figure 4.11 and Figure 4.12. The hydrometeor classification indicates wet snow below 1 km, dry snow and oriented ice crystal above 1 km. We can see a transition from rain to snow on the ground. This feature of rain to snow transition was observed on the ground in excellent agreement with radar based NFHC inferences.

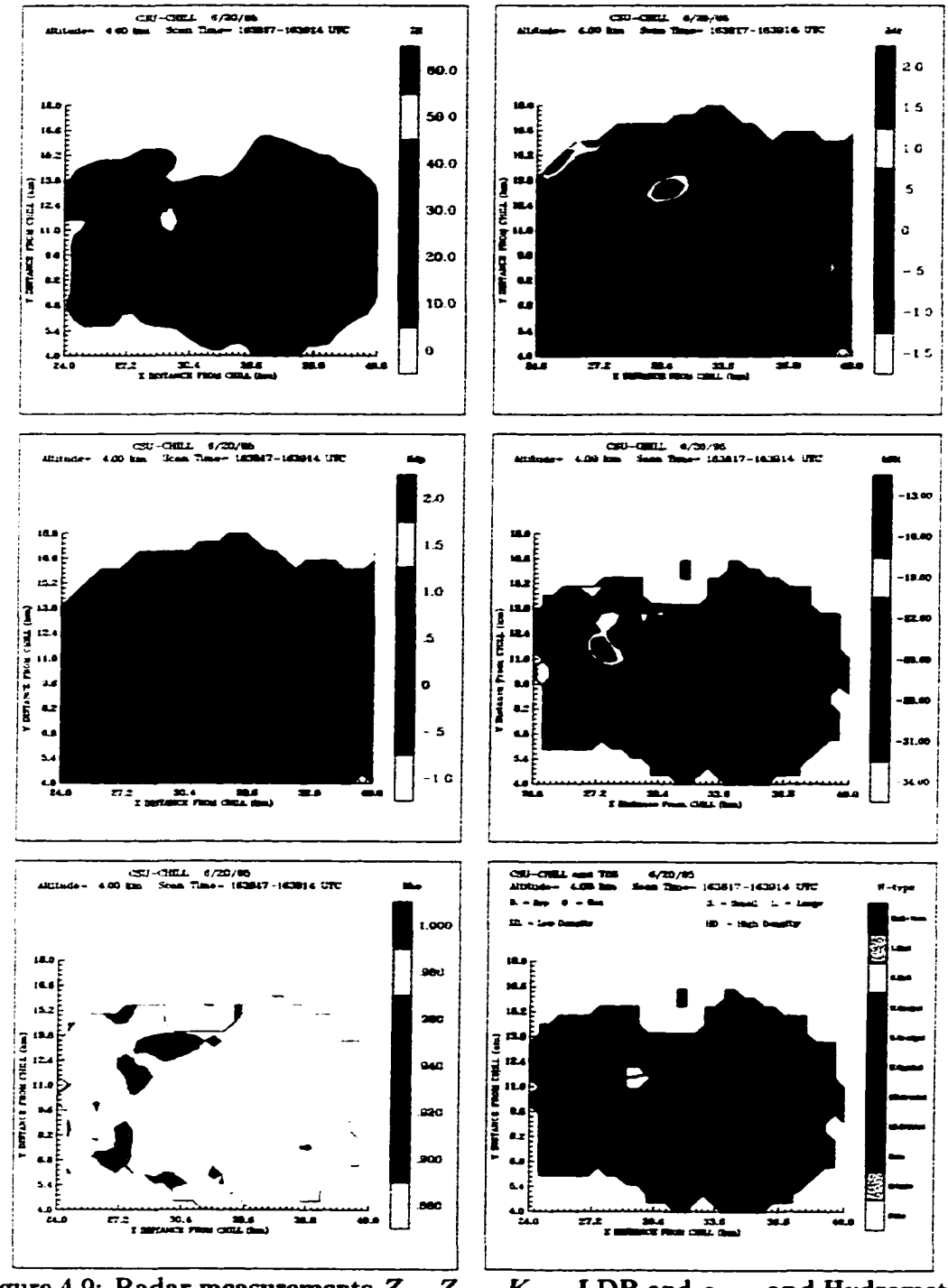


Figure 4.9: Radar measurements Z_H, Z_{DR}, K_{DP}, LDR and ρ_{HV}, and Hydrometeor Type Classification result corresponding to the case of June 20, 1995

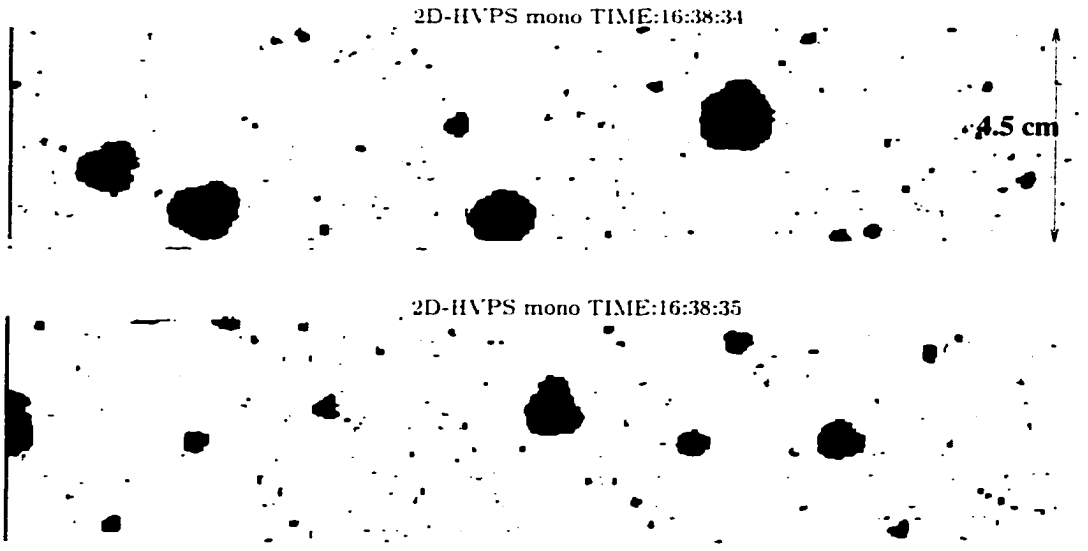


Figure 4.10: HVPS Images of hydrometeors at 16:38:34 in the 20 June 1995 storm. The aircraft was located at x=30.33km, y=11.80km with respect to the radar during that time

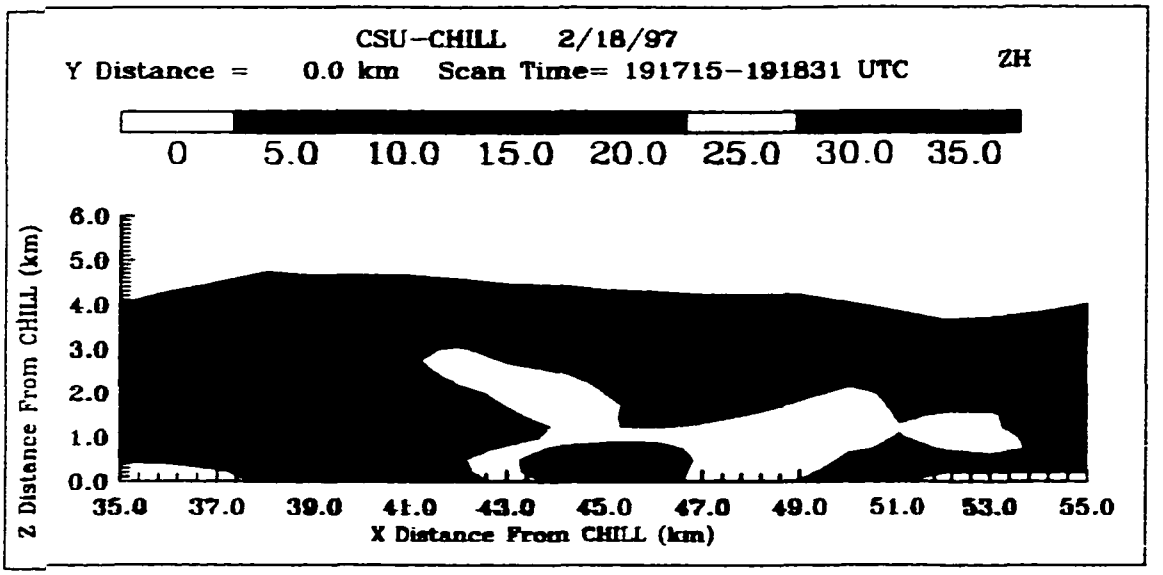


Figure 4.11: (a)Radar measurements Z_H for the storm on Feb. 18, 1997

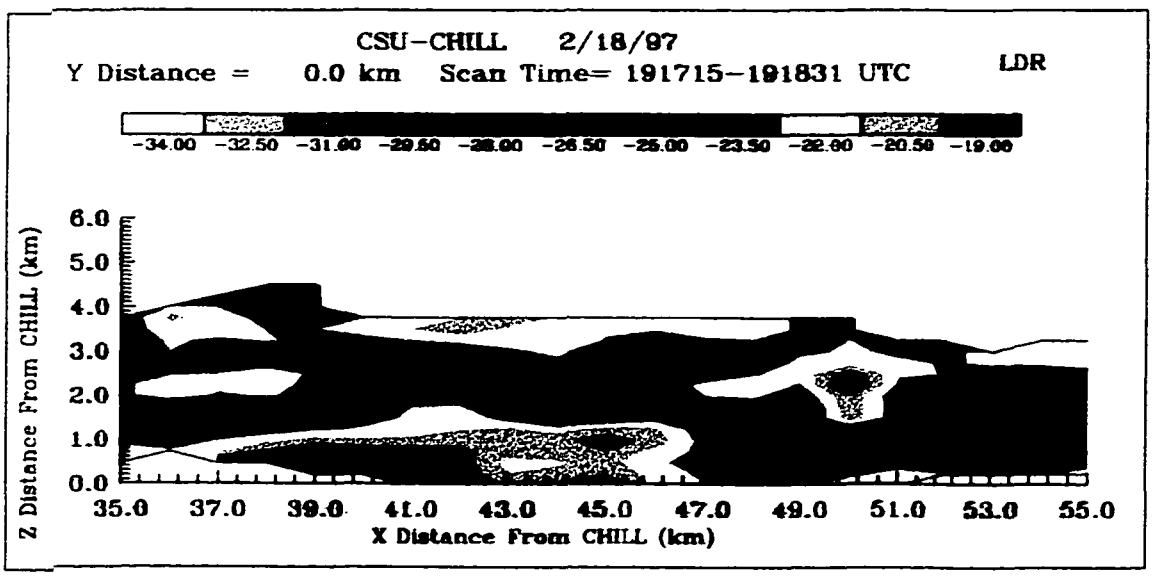


Figure 4.11: (b) Radar measurements LDR for the storm on Feb. 18, 1997

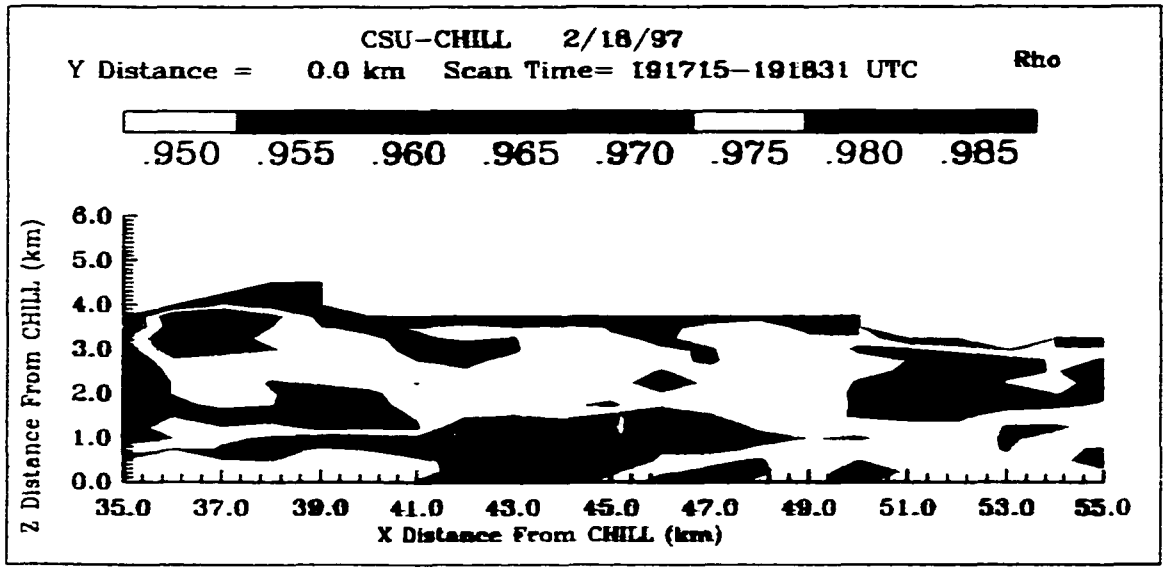


Figure 4.11: (c)Radar measurements ρ_{hv} for the storm on Feb. 18, 1997

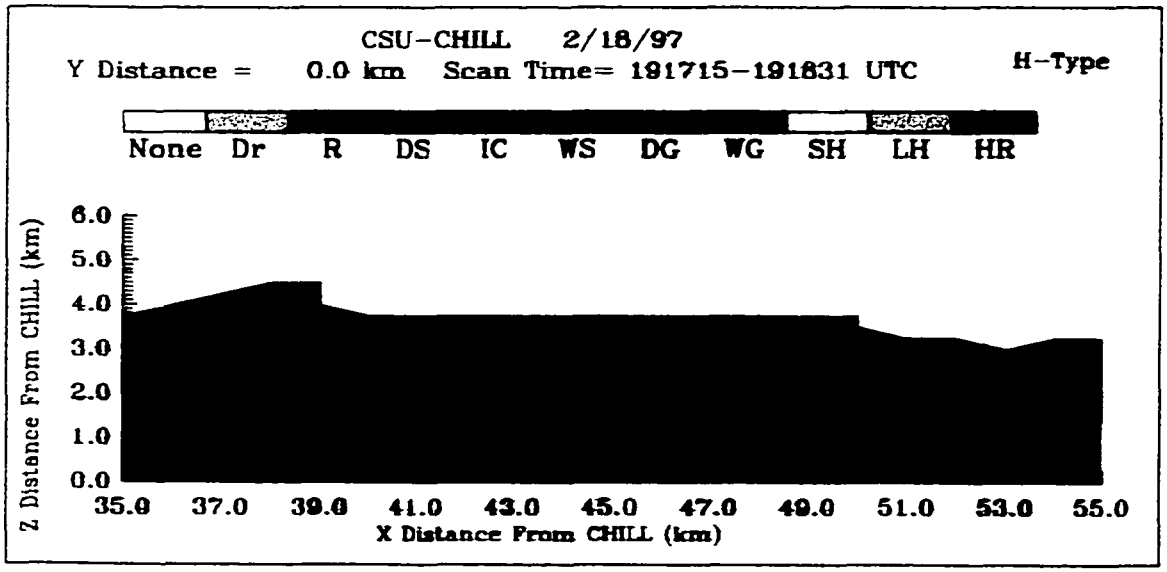


Figure 4.12: Hydrometeor type classification result, note that Dr. stands for Drizzle, R. for Rain, DS. for Dry Snow, IC for Oriented Ice Crystal, WS for Wet Snow, DG for Dry Graupel, WG. for Wet Graupel, SH for Small Hail, LH for Large Hail, HR for mixture of Hail and Rain.

Chapter 5

DETECTION OF RAIN/NO-RAIN CONDITION ON THE GROUND BASED ON RADAR OBSERVATIONS

5.1 Introduction

One of the important issues in radar rainfall estimation is to first determine if there is rain on the ground. Many modern weather radar systems are equipped with high power transmitters and high sensitivity receivers, so that even weak electromagnetic waves reflected by few hydrometeors can be detected. Significant radar reflectivity observations can be obtained from precipitation particles aloft even if there is no precipitation on the ground. However, these reflectivity values may be erroneously converted to precipitation on the ground based on some radar rainfall algorithm. Therefore, it is important to have a detection/decision scheme to determine if there is precipitation on the ground prior to computing rainfall using radar measurements. It would be useful to develop an automatic classification scheme that can detect the presence and absence of rain on the ground. One of the goals of this chapter is to develop a neural network based technique to determine presence and absence of rain from three dimensional radar reflectivity structure. Several factors such as rain evaporation, non-precipitating echo and very light rain that is below the observing capability of raingages can result in a radar observed echo, with zero rainfall observation on a ground instrument such as raingage. This may also happen due to anomalous propagation. This research

presents a technique to infer the presence/absence of rain on the ground from radar observations. this procedure does not consider anomalous propagation.

This chapter is organized as follows: The development of a Radial Basis Function (RBF) neural network for determination of rain/no-rain condition on ground is discussed in section 5.2. The performance evaluation of the network for radar data collected by the WSR-88D radar system are provided in section 5.3. Finally, important conclusions are summarized in section 5.4.

5.2 Classification of vertical reflectivity profiles with neural network

The three dimensional structure of reflectivity aloft is strongly correlated to the conditions of presence and absence of rain on the ground. In order to reduce the volume of data that is input to the network, the reflectivity measurements on a horizontal grid of 3 km × 3 km area are averaged to obtain an averaged vertical profile of reflectivity, which is then used as input to a neural network. Figure 5.1 shows the schematic describing the computation of average vertical profile. In this paper, we use average vertical profile in rain region (up to the melting layer) as input to the neural network. Let $Z_h(1), \ldots, Z_h(4)$ be the average reflectivity measurements at altitudes 1.5 km, 2.5 km, 3.5 km and 4.5 km, respectively. These four reflectivities are used as input to determine the presence and absence of rain on the ground. The data set used in this study was collected by the WSR-88D radar at Melbourne, Florida and several raingage networks are located in the vicinity of the radar. Data from a total of 59 raingages are used in this study. The vertical profiles were extracted from radar scans to construct averaged vertical reflectivity profiles.

Determination of the presence of rainfall on the ground based on radar data can be viewed as a simplified pattern recognition problem. Radial Basis Function neural network is well suited for the the problem of determinating rain/no-rain on the ground.

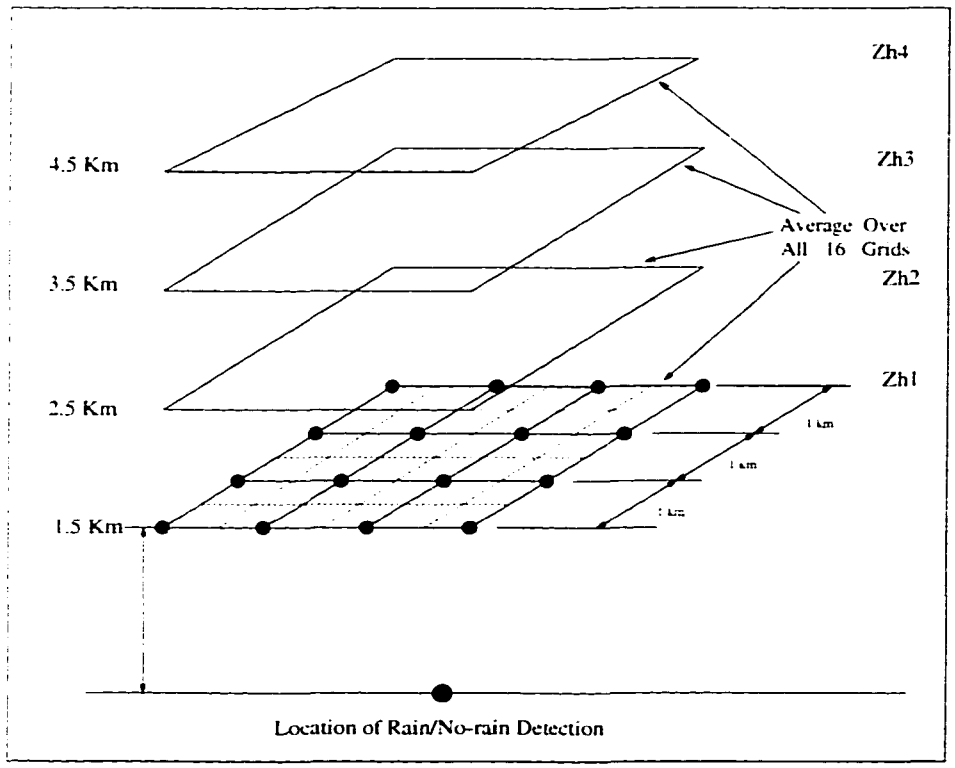


Figure 5.1: Construction of averaged vertical profiles

To develop a RBF network for rain/no-rain classification, two basic issues need to be addressed, namely, i) choice of the right size of the neural network, and ii) finding the optimum parameters for the corresponding network by using a representative training data set. The input layer consists of the four averaged reflectivities namely Z_{h1},Z_{h2}, Z_{h3}, Z_{h4}. In this research, we use average vertical profile in rain region (up to the melting layer) as input to the neural network. Mean reflectivities at the altitudes of 1.5, 2.5, 3.5 and 4.5 km are used as input to determine the presence and absence of rain on the ground. There is only one output "Y" for the network, which represents the rain/no-rain condition. Specifically, "0" is used for representing *no-rain* and "1" is for *rain*. The output of a RBF network is not binary. A decision threshold 0.5 is used to convert the output Y to the binary format C (shown in Figure 5.2).

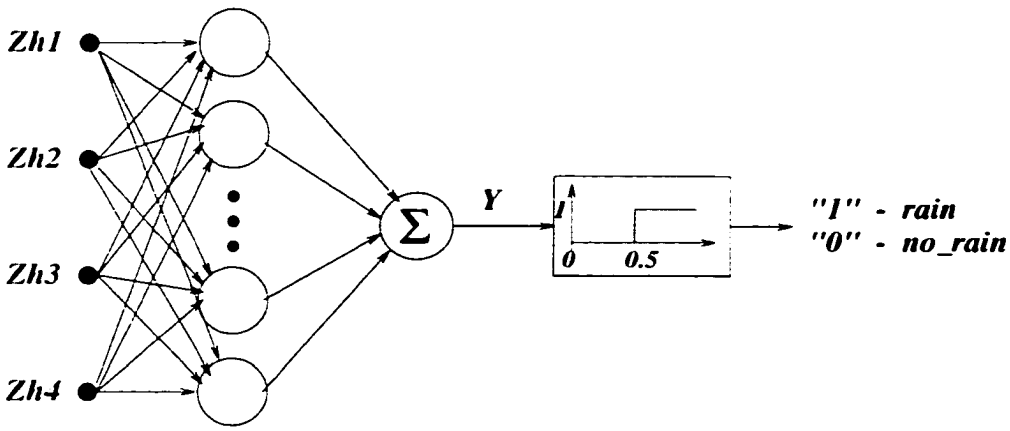


Figure 5.2: Diagram of RBF network for rain/no-rain determination

The data set used in this study was collected by the WSR-88D radar at Melbourne, Florida and several raingage networks are located in the vicinity of the radar. The vertical profiles were extracted from radar scans to construct averaged vertical reflectivity profiles. Five minute accumulations of raingage data are used to determine the rain/no-rain condition. To get the optimum size of hidden layer and its parameters, a representative training data set is needed for training the network. The data set that is used in the training process was collected during July 1993. This data has a total 17,023 averaged vertical reflectivity profiles, out of those, 15,671 vertical profiles were associated with no precipitation on the ground (zero gauge measurements) and 1352 vertical reflectivity profiles were associated with non-zero rainfall on the ground. A small portion of this data was used for purpose of training the network, and the rest was used as testing data set. Figure 5.3 shows the location of these tipping-bucket raingages (total 59) relative to the radar. Note the distances from the raingages to the radar range from less than 10 km to about 200 km. Figure 5.4 shows an example of the ground rainfall measurements at all 59 gage sites during a storm event on July 2, 1993. We can see from Figure 5.4 this storm lasted for 4 hours.

Training is the most important process in the set-up of the neural network, which critically influences the performance of the network. 1000 data samples

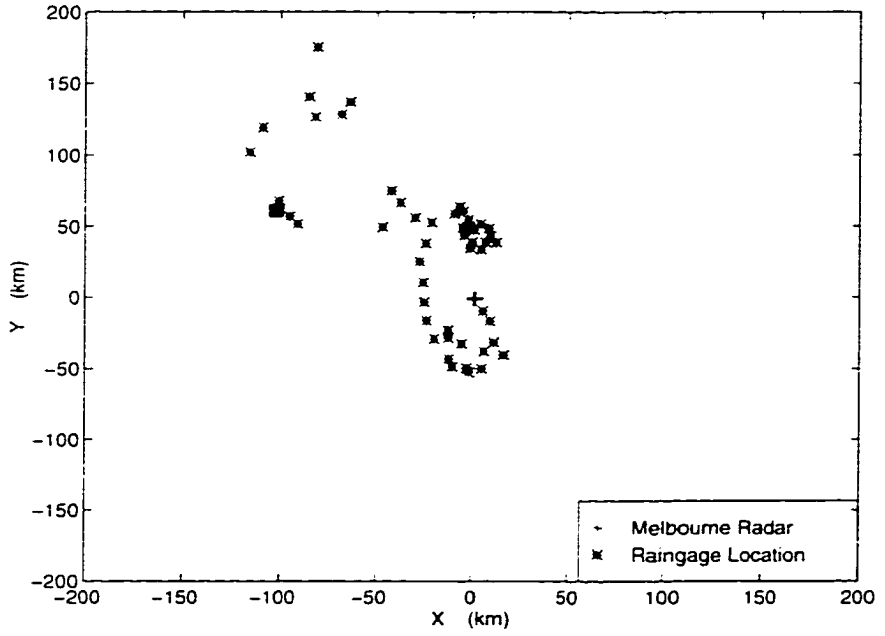


Figure 5.3: Location of the 59 tipping-bucket raingages relative to the Melbourne radar. The radar is at (0.0).

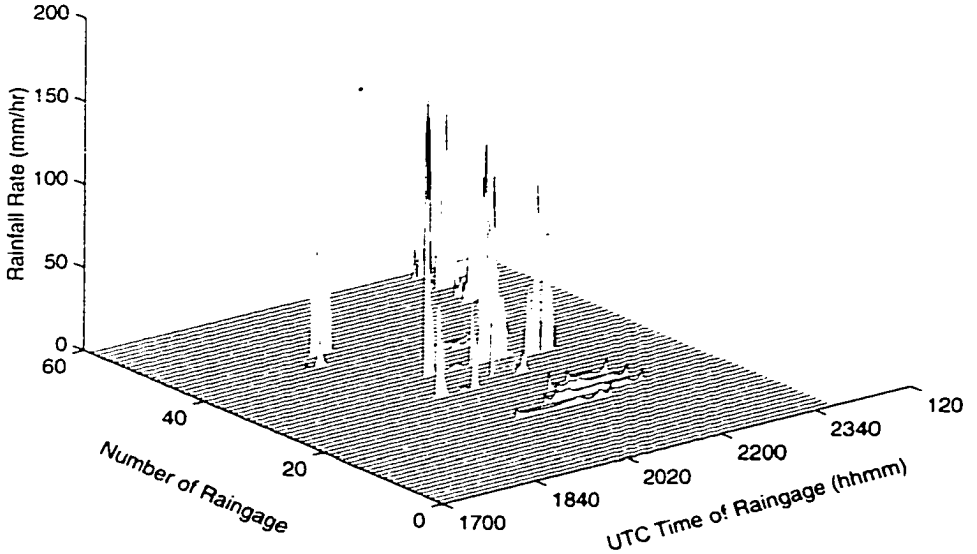


Figure 5.4: Raingage measurements at 59 gauge sites during a storm event on July 2, 1993

Table 5.1: Classification Performance of the RBF network for July 1993 data set

	data samples	correct samples	incorrect samples	correct rate
rain	1352	1149	203	85%
no-rain	15,671	12,999	2672	82%

were chosen from the rain data set and 1000 data samples from the no-rain data set to form the training data set. The averaged radar reflectivity vertical profiles in the training data set are applied to the RBF network sequentially. "Orthogonal Least Square" supervised learning method is used to train the network, and the Generalized Cross-Validation (GCV) is used as the convergence. To obtain the classification performance of the RBF network with respect to July 1993 data set, the original data set (17,023 data) with 1352 "rain cases" and 15,671 "no-rain cases" were applied to the trained network. 85% of the rain cases and 82% of the no-rain cases were classified correctly. The classification results of July 1993 data are listed in table 5.1.

5.3 Performance Evaluation of the RBF Network for rain/no-rain Classification

To evaluate the classification performance of the RBF networks developed above by using the data set of 1993, two evaluation data sets are used to test the network, one set was collected in August 1995, and the other one in September 1995. These two data sets were applied to the RBF network obtained in section II, separately. Table 5.2 shows the performance of the test results for the RBF network. It can be seen that the correct classification rate for 'rain' cases are 70% and 74% for August and September data set, respectively, the correct classification rate for 'no-rain' cases are 87% and 95%, respectively.

To investigate the impact of the classification scheme on the application of rainfall estimation, the rainfall accumulation by using the best Z-R algorithm is

Table 5.2: Classification Performance of the RBF network for August/September 1995 data set

	data samples	correct samples	incorrect samples	correct rate
rain (Aug./1995)	3127	2195	932	70%
no-rain (Aug./1995)	9480	8243	1237	87%
rain (Sept./1995)	442	328	114	74%
no-rain (Sept./1995)	7188	6846	342	95%

obtained under the conditions of applying or not-applying the classification scheme. The rainfall accumulation for the months of August and September 1995 over the 19 gages were obtained. The gauge data over the 19 gauges resulted in an accumulation of 2059 mm, whereas the corresponding accumulation from radar over the gauge sites yielded an accumulation of 2571 mm. However, if the rain/no-rain classification neural network was applied prior to rainfall estimation, it yielded an accumulation of 2199 mm, resulting 18% improvement with the gauge data as reference. This evaluation demonstrates the practical use of the rain/no-rain classification scheme developed.

The procedure developed in this paper was compared against other classification methods, namely simple classification using a reflectivity threshold and the combined learned vector quantization (LVQ) network and self-organization mapping (SOM).

The combined LVQ and SOM method did not perform as well as the RBF network, especially for the rain cases. A reflectivity threshold based decision for rain/no-rain classification is attractive because it is very simple. However, the optimum threshold that minimized the optimum threshold that minimized the probability of erroneous decision kept changing with the data set. In addition, the optimum threshold based on one data set yield substantially higher errors in rainfall accumulation for other data sets.

Table 5.3: The performance of a neural network rain/no-rain classifier based on data collected in August, 1995

	data samples	correct samples	incorrect samples	correct rate
rain (Aug./1995)	3127	2304	823	74%
no-rain (Aug./1995)	9480	8496	984	90%
rain (Sept./1995)	442	333	109	75%
no-rain (Sept./1995)	7188	6902	286	96%

The RBF network for rain/no-rain classification are constructed by using data collected in July, 1993. This neural network classifier was tested on the data collected two years later, i.e., August/September, 1995. During these two years, many factors related to radar may have changed, such as the radar calibration. To evaluate this we developed a rain/no-rain classification neural network based on the data collected in August 1995 and tested it on the data collected in September 1995. The classification results are presented in the table 5.3. A comparison of classification result for September 1995 data in table 5.2 and table 5.3 shows a very slight improvement if training based on the data from the same year. Perhaps the radar characteristics did not change much. In addition, this test demonstrates the robustness of the procedure. The corresponding improvement in rainfall estimation was also similar (about 18%).

5.4 Summary and Conclusion

A RBF neural network based scheme for the determination of presence/absence of precipitation (rain/no-rain) on the ground is presented in this chapter. The RBF network is set up by using the radar data and associated raingage data collected in the vicinity of the radar. Data collected during July 1993 was used to train and test the RBF network. Data collected in August and September of 1995 are used to evaluate the performance of the RBF network for rain/no-rain detection

on ground. The averaged correct classification rate for the evaluation data is about 84%. By using the classification scheme before applying Z-R algorithm, the rainfall estimation accuracy is improved by 18%. The performance was also compared with other classification method namely learned vector quantization. The result shows that the RBF network performs better than the learned vector quantization method. The RBF network developed in this paper can be used in automated computation of precipitation on the ground from radars.

Chapter 6

DEVELOPMENT OF NEURAL NETWORKS FOR RADAR RAINFALL ESTIMATION

6.1 Introduction

Rainfall estimation by using a neural network has been introduced as a convenient, efficient and robust method in the weather radar system application. Xiao and Chandrasekar(1997) proposed a framework (Xiao, 1996, Xiao and Chandrasekar, 1997) to develop neural network estimates of rainfall/snowfall and investigated the potential and improvement of estimation accuracy by using the multiperceptron backpropagation neural network. The neural network technique involves training the network based on a representative data set consisting of radar measurements from 3-Dimensional precipitation region and corresponding ground based raingage observations. Radar data are applied to the network as the input and the corresponding raingage data are used as the desired output to guide the learning of the parameters of the MLP neural network during the training process. The recursive least square (RLS) backpropagation learning algorithm has been used to build a three layer BPN for rainfall estimation based on data collected by CP-2 multiparameter radar and WSR-88D systems, respectively. Once the training process is complete, the network is ready for application. The performance of the network is evaluated by using a separate data set which was not used in the training process. The schematic illustration of training of an MLP for rainfall

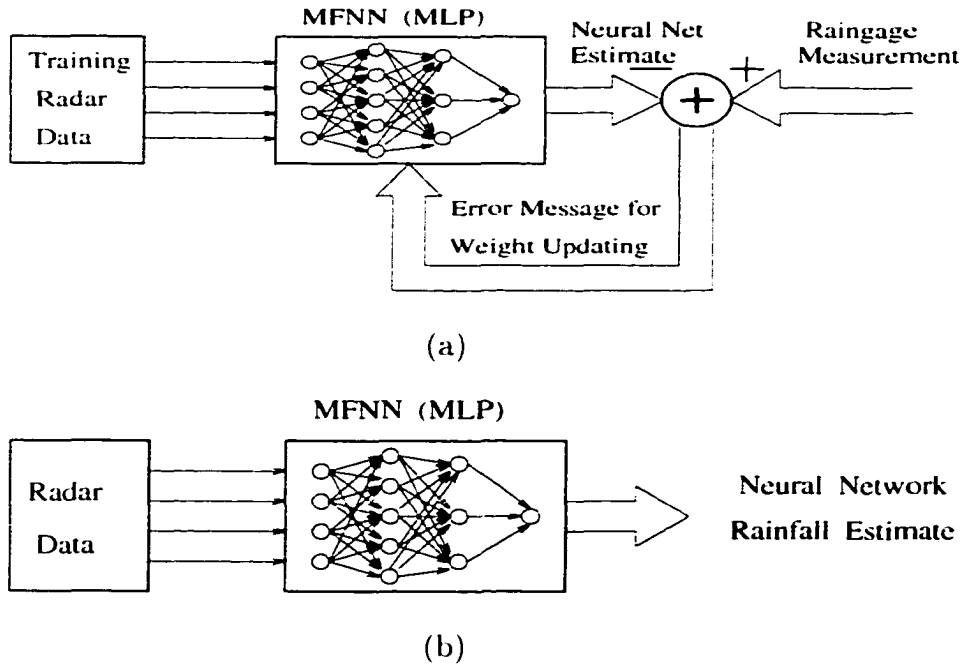


Figure 6.1: Development of multilayer perceptron (MLP) for rainfall estimation. (a) Training of an MLP for rainfall estimation. (b) Application of MLP for rainfall estimation. [From xiao's dissertation]

estimation and application of an MLP for rainfall estimation, as proposed by xiao and Chandrasekar (1997), is shown in Figure 6.1.

Based on this previous study, we can further study the factors that influence the performance of the neural network for rainfall estimation, so that we can design a neural network that can be used to estimate rainfall in real-time. In this chapter, we study further the application of neural networks for rainfall estimation problems. In order to put the technique into a real operation for rainfall estimation, some important issues for successful application, such as choosing the right structure of neural network, examining the most appropriate and efficient learning algorithm, determining the optimum combination of input variables polarimetrically and spatially, appropriate training data set selection and pre-processing and post-processing, studying the methods of improving the generalization capability, are investigated. Successfully solving all these issues is important for building a high-performance neural network for rainfall estimation, however, this is still not

sufficient for building a feasible network for rainfall estimation. Considering there is wide variability of the relationships between rainfall rate recorded on the ground and the radar measurements aloft, a neural network with fixed structure and parameters may not perform well in all cases. For a better performance, the network can be fine-tuned with every rainfall regime. However retraining the network can be a tedious task. An adaptive neural network whose parameters can be updated in an adaptive manner based on the most recent information is a good compromising solution to the dilemma of accuracy and generalization. An adaptive scheme is developed in this chapter for dynamically updating the structure and parameters of the neural network which enables the network to handle the non-stationary relationship between radar measurements and precipitation estimation that occur with change of season, location and other environment conditions.

Section 6.2 describes and compares potential neural network topologies for radar rainfall estimation. In addition, some considerations for choosing the right type of neural network for rainfall estimation are discussed in this section. In section 6.3, choice of input is investigated. Training data preparation is covered in section 6.4. Finally, an adaptive neural network scheme for rainfall estimation is proposed in section 6.5. The application of the adaptive scheme is represented in the following chapter.

6.2 Neural Network Topology for Radar Rainfall Estimation

Generally, rainfall estimation based on polarimetric radar measurements can be modelled as a multivariate function approximation problem. The objective to find a non-linear complex mapping function from radar measurements to the rainfall rate. It has been shown that neural networks can approximate any given continuous function based on a compact subset of data cases to any degree of accuracy, provided that a sufficient number of hidden layer neuron is used. There

are two types of neural network that are most suitable for function approximation, i.e., MultiLayer Perceptron Neural Network (MLPNN) and Radial Basis Function Neural Network (RBFNN). The structures and learning algorithm relevant to these two neural network were introduced in chapter 2.

Both MLPNN and RBF are Multilayer Feedforward neural networks, but there some difference between these two networks. The differences are listed as following:

(1) Difference in Architecture

MLP neural network may have several hidden layers, and MLPNN with more than 2 hidden layers can map any arbitrary function from input space to the output space. A conventional RBF neural network has only one hidden layer, but it can be very useful to include a linear layer for dimensionality reduction or a non-linear transformation of each input. The purpose of the dimensionality reduction is to avoid the problem caused by "Curse of Dimensionality", and non-linear transformation of each inputs will make the learning easier for the network.

(2) Difference in Combination Function

Each neuron always uses some type of combination function to combine the inputs to the neuron. For MLPNN, linear combination function the form of $w_{111}x_1 + w_{112}x_2 + ... + w_{11n}x_n$, which is the *inner product* of the input vector $[x_1, x_2, ... x_n]$ and the synaptic weight vector $[w_{111}, w_{112}, ... w_{11n}]$ of neuron, is used. As for the RBFNN, the combination functions are based on the Euclidean distance between the input vector and center vector $[c_{11}, c_{12}, ... c_{1n}]$ as the form of $\frac{(x_1-c_{11})^2+(x_2-c_{12})^2+...+(x_n-c_{1n})^2}{r_1^2}$.

(3) Difference in Activation Function

After the inputs are combined by a combination function, the activation function acts on the combined result of each neuron to produce the neuron's output. For MLPNN, the most commonly used forms of the activation function are sigmoid,

tangent, hard limit, and linear. For RBFNN, the most commonly used activation function is exponential.

(4) Difference in Parameters

The parameters that are needed to learn from training depend on the combination function and activation function, and can be different for MLPNN and RBFNN. Weights and biases are two parameters used in MLPNN, whereas widths and centers for the hidden layer neuron and weights from hidden layer to the output layer, are the parameters used in RBFNN.

(5) Difference in Training Algorithm

The training algorithm is the method that neural networks use to adjust their parameters. The Back-Propagation (BP) is the most widely used training algorithm for MLPNN. Usually this method is very *slow*, especially, for large-scale problems. For conventional RBFNN, "hybrid training" methods are often used, in which the centers and widths for the hidden layer neurons are first obtained by unsupervised learning and the weights are computed by a superwised learning method. The supervised learning is quite straightforward and very efficient because the computation reduces to a linear or generalized linear model. Therefore, the training process is very fast. Even though the "hybrid training" method is very fast, it can not guarantee to find the optimum locations of the center, therefore, Orthogonal Least Square (OLS) supervised learning methods are recommended for obtaining the centers of the hidden neurons. However this method is not as fast as "unsupervised learning", but it is still faster than the BP learning algorithm (S. Chen, 1991).

(6) Difference in Response to Redundant Inputs

Both MLPNN and RBFNN can handle redundant inputs very well, because the redundant inputs have little effect on the effective dimensionality of the data.

If the test cases follow the same redundant patterns as the training data set the networks can provide very accurate outputs, otherwise extrapolation is needed for the generalization, which will probably cause the generalization to fail.

(7) Difference in Response to Irrelevant Inputs

The MLPNN is good at ignoring irrelevant inputs. The first hidden layer of the MLP can perform linear transformation of the inputs and reduce the dimensionality. Adding irrelevant inputs to the training data does not increase the number of hidden units required, although it increases the amount of training data required. Conventional RBF networks are not good at ignoring the irrelevant inputs. The number of hidden units required grows exponentially with the number of inputs, regardless how many inputs are relevant.

(8) Difference in Generalization Capability

The MLPNN tends to over generalize. This is evident in the case of pattern classification. An MLP can be trained to have high accuracy in classifying patterns from a set of known categories, but will also classify any out-of-category pattern as one of the trained categories. However, the RBFNN would probably classify the out-of-category pattern as an unknown category (tao, 1992). Usually, the RBFNN has good generalization in the trained subspace, any generalization that requires an exptrapolation tends to fail.

(8) Difference in Approximation Characteristics

The difference between MLPNN and RBFNN in the aspect of function approximation, or so-called input-output mapping, is that the MLPNN construct *global* approximations while RBF networks use exponentially decaying localized nonlinearities (e.g., Gaussian functions) to construct *local* approximations. Therefore, the MLPNN may require a smaller number of parameters than the RBF network

for the same degree of accuracy for the approximations to nonlinear input-output mapping.

From the above comparison between MLPNN and RBFNN, we know that these two neural networks have their own advantages and disadvantages. We cannot simply conclude which one outperforms the other. The specific application drives the decision of which network to choose. Following are some considerations for choosing a neural network between the MLPNN and RBFNN:

- In the case of sufficient training data available, RBFNN is a good choice because RBFNN is fast and the adverse effect caused by "Curse of Dimensionality" is not serious (when there is adequate data.)
- If a dynamic neural network is needed in application, RBFNN is preferred because it has a "local receptive field", the local change in the network will not affect the other part of the network. It is relatively easy to implement the adaptive RBFNN, however, it is very difficult to implement dynamic MLPNN, since it requires complete re-training when new data becomes available.
- In the case of scarcity of data, or in the case of many inputs variables, which we are not sure are relevant to the output, we should choose MLPNN for better generalization.

6.3 Input Optimization and Pre-processing

There are many factors that would influence the performance of the neural network for rainfall estimation. First of all, the inputs to the network should contain sufficient information pertaining to the rainfall on the ground, so that there exists a nonlinear mapping function relating the rain rate on the ground with radar measurements.

Several polarimetric radar measurements that can be used for rainfall estimation are: reflectivity factor (Z_h), differential reflectivity (Z_{dr}) and specific differential phase shift (K_{dp}). A case study demonstrated that, when rain rate is small ($R < 10 mmhr^{-1}$, R represents the rain rate), it is appropriate to use Z_h (horizontal reflectivity) alone to estimate rainfall; the combination of Z_h and Z_{dr} can provide good estimation when rain rate is moderate ($20\, mmhr^{-1} < R < 70 mmhr^{-1}$). When rain rate is heavy ($R > 70\, mmhr^{-1}$), using K_{dp} provides a good estimate (Bringi and Chandrasekar, 2000).

Z_h, Z_{dr} and K_{dp} can be candidate input variables for rainfall estimation, but not many radars can measure these variables. Hence forth in this chapter, Z_h, Z_{dr} and K_{dp} are refered to as *candidate input variables (CIV)*.

6.3.1 Spatial selection of Input variables

To estimate the rainfall at a location on the ground (denoted as EPOG: Estimation Point On the Ground, as shown in Figure 6.2), radar measurements over a region around the EPOG are needed to apply to the neural network as inputs.

6.3.1.1 Spatial Structure of Radar Data

A pulsed Doppler radar operates in either PPI (Plan Position Indicators) or RHI (Range Height Indicators) mode. The PPI mode is shown in Figure 6.3. For each PPI scan, the radar beams keep elevation angles constant, and azimuth angles may varies from 0^o to 360^o. The RHI mode is illustrated in Figure 6.4. For each RHI scan, the radar beams keep the azimuth angle constant, the elevation angle can change from 0^o to 90^o (or less). A radar volume scan consists of several PPI or RHI scans. Radar measurements are collected along the range as equi-spaced samples. For CSU-CHILL radar, the range sampling distance is 150 m. Therefore, the radar data are distributed spatially on the grids as shown in Figure 6.5.

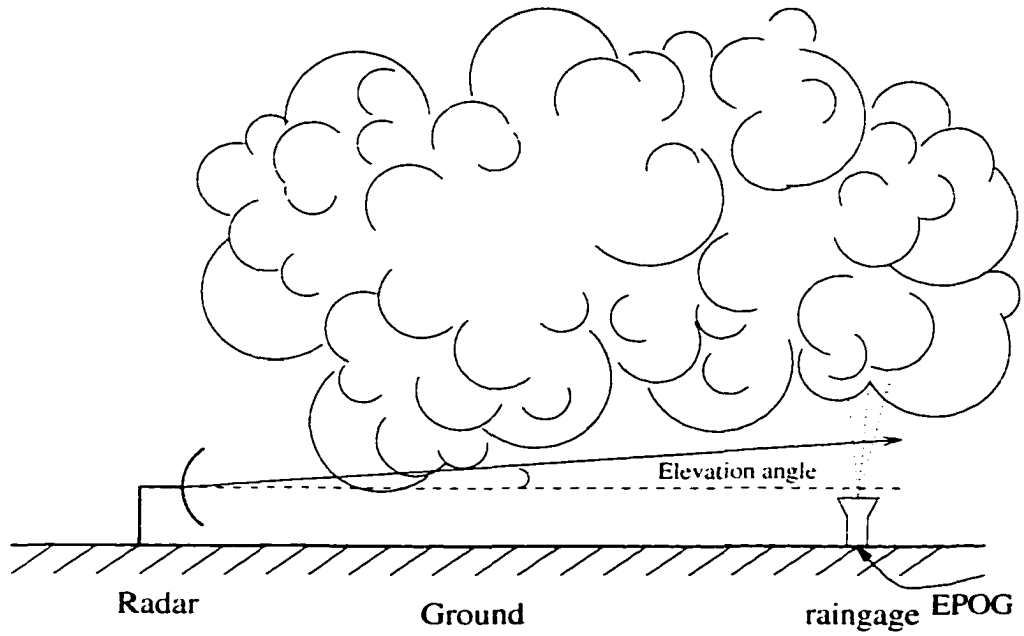


Figure 6.2: illustration of rain rate estimation for ground site EPOG by a radar

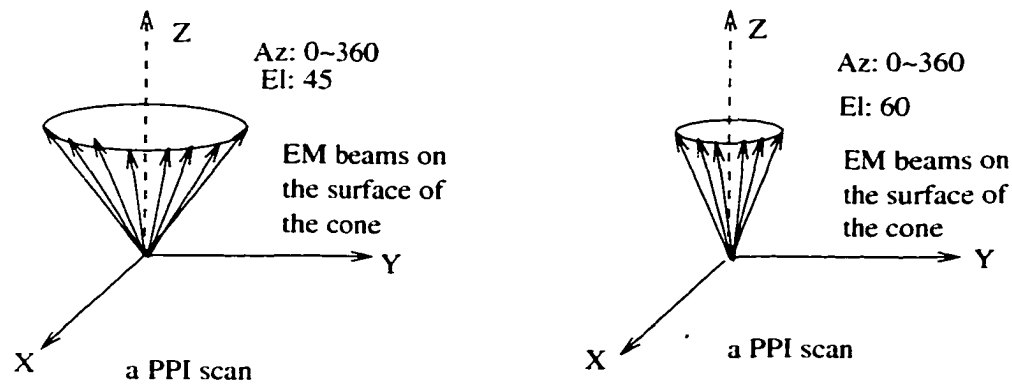


Figure 6.3: PPI scan mode

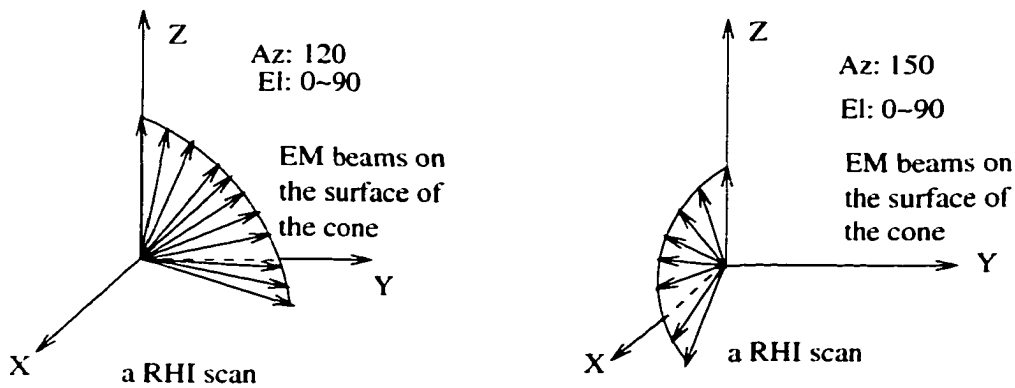


Figure 6.4: RHI scan mode

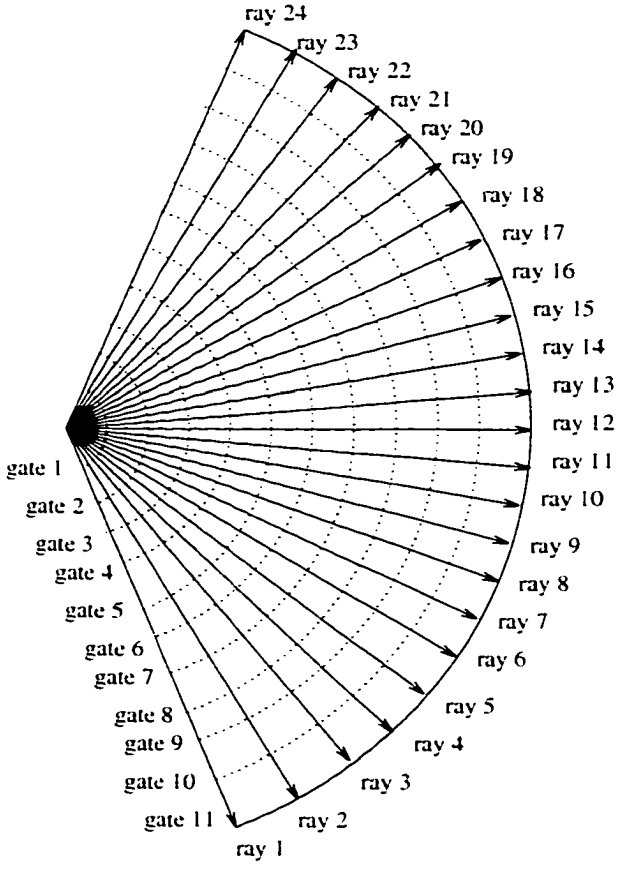


Figure 6.5: illustration of data grids in a PPI or RHI scan

Usually, it takes about 5 minutes for a radar to perform a volume scan. The radar measurements are discrete in time domain and spatial domain. For any ground site in the range of radar scan, rain rate estimation can be carried out in the time interval of radar volume scan time.

6.3.1.2 Feature/Data Extraction

From volume scan data, a subset of needs to be extracted for the purpose of estimating the rain rate corresponding to a specific ground site. This data subset should be most correlated to the rain rate at the ground site.

(1) Searching the most correlated radar-scattering volumes

The rainfall rate at time t is most correlated to radar measurements in radar-scattering volumes at time $t - t_\delta$, during when the rainfall field decends to the location of ground. These most correlated radar-scattering volumes could be different storms. Aydin et al (1987) investigated the method of finding the most correlated radar-scattering volumes by studying the horizontal and vertical motion of raindrops. In this method, estimates of the speed and direction of the storm were obtained by observing the reflectivity contours from PPI scans at different times during the rainfall event. Then along the storm path direction, several columns (called swath) are examined to find the cells that correlate best with the rain rate at the ground location. These cells define the region in space that most likely contributes to the rain rate estimation at EPOG. Z_h, Z_{dr}, and K_{dp} (if they are all available) are extracted from the most correlated radar-scattering volume.

(2) Obtain the surface profile and the vertical profile

Intuitively, the farther the radar measurements are from the EPOG, the less information they can provide for ground rain rate estimation. As a result, radar measurements from a three dimensional region just above the EPOG contribute

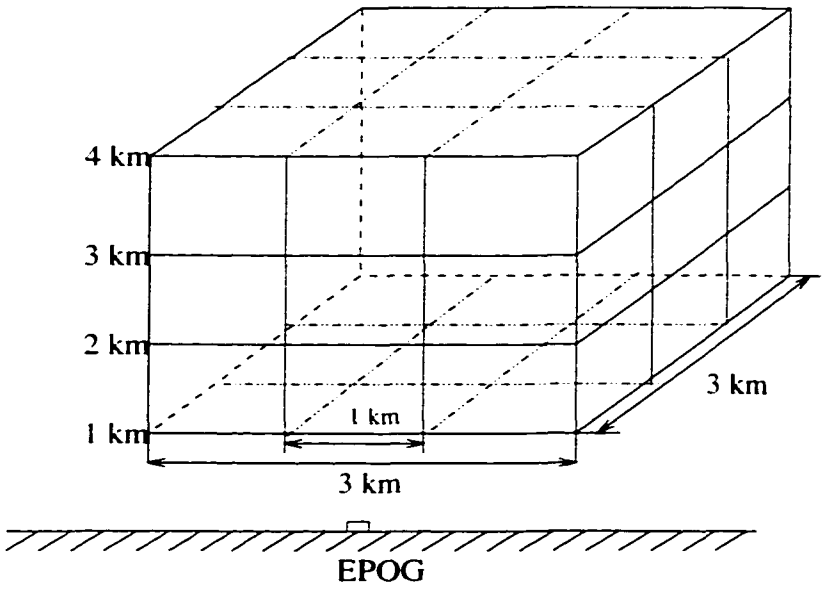


Figure 6.6: 3D region above the EPOG

most to the rain rate estimation: The larger the region, the more data it will provide for input and hence the input space will be of a higher dimensionality. This aggravates the problem of "curse of dimensionality". The optimum dimension of the region depends on the storm characteristics, such as storm cell size, the storm's moving speed and direction, etc. A common 3-D region should be defined so as to apply the neural network to various storms for general application. Therefore, a 3-D region with 3 km x 3 km surface size and 4 km height is used (as shown in Figure 6.6). From this 3-D region, a surface profile and a vertical profile are extracted (as shown in the Figure 6.7), further decreasing the input dimensionality.

The surface profile can be obtained as follows: The lowest surface (at 1 km altitude) in the 3-D region is segmented into 9 equally partitioned square areas, radar measurements are averaged in each of the small area. 9 averaged data can be obtained on the surface, they are called surface profiles. The three potential surface profiles are Z_h surface profile, Z_{dr} surface profile and K_{dp} surface profile. The vertical profiles are obtained by averaging radar measurements on the surface with height of 1 km, 2 km, 3 km and 4 km, respectively. Similarly, there are three

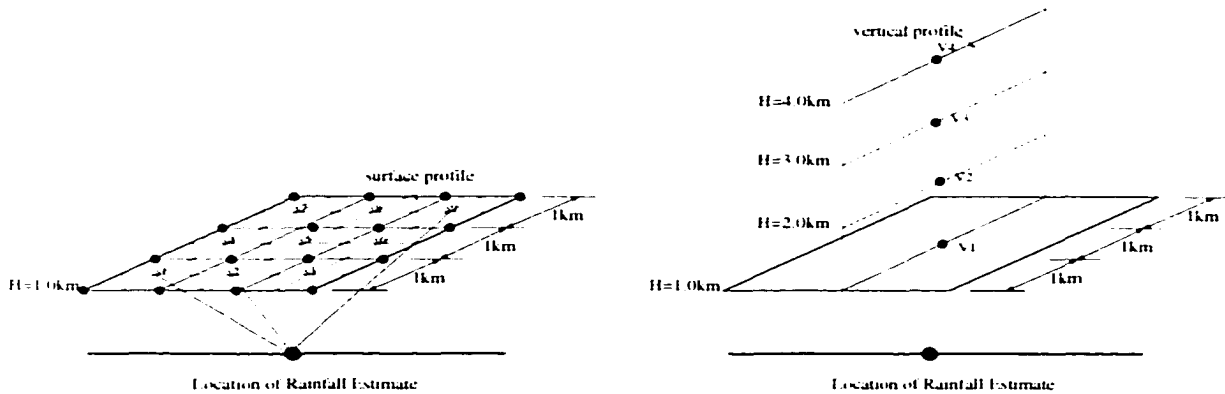


Figure 6.7: configuration of surface profile and vertical profile

possible vertical profiles, they are Z_h vertical profile, Z_{dr} vertical profile and K_{dp} vertical profile, respectively.

The inputs selection method by searching the most correlated radar-scattering volumes and extracting the CIV from the volume is complicated. It requires significant computation to obtain the storm speed and direction at different instants. Comparatively, the inputs extracted just from surface and vertical profile are much simpler. The spatial locations of the inputs are fixed, therefore, the process is faster and more practical than the first method.

(3) Optimum combination of candidate variables spatially

After extraction of the surface and vertical profile from the original data set, the resulting subset still consists of 9 points from the surface profiles and 4 points from the vertical profiles for each of the input variable candidates. If all three input radar parameters are available, there are a total of 39 input candidates. Building a neural network with 39 inputs reasonable, but there still may be some redundancy present in the data. The possibility of further reducing the size of the input subset by searching for the optimum combination of candidate variables is investigated next.

The experiment is based on data from the CaPE program conducted in 1993, and a RBF neural network is chosen. From this CaPE data set, only two of

Table 6.1: Performance comparison of neural networks with different inputs (1)

Experiment no.	input variables	inputs number	FSE
1	Z_h_S	9	0.222
2	Z_h_S, Z_h_V	13	0.202
3	Z_h_S, Z_{dr}_S	18	0.203
4	Z_h_S, Z_{dr}_V	13	0.182
5	Z_h_S, Z_h_V, Z_{dr}_S	22	0.207
6	Z_h_S, Z_h_V, Z_{dr}_V	17	0.207
7	Z_h_S, Z_{dr}_S, Z_{dr}_V	22	0.221
8	Z_h_S, Z_h_V, Z_{dr}_S, Z_{dr}_V	26	0.207

the input variable candidates are available, which are Z_h and Z_{dr}. Data from 20 raingages are available in addition to radar data. Details of this data set are described in Xiao (1997). This leads to two surface profiles and two vertical profiles. The eight combinations are 1) Z_h surface profile 2) Z_h surface profile and Z_h vertical profiles 3) Z_h surface profile and Z_{dr} surface profile 4) Z_h surface profile and Z_{dr} vertical profiles 5) Z_h surface profile, Z_h vertical profiles and and Z_{dr} surface profile 6) Z_h surface profile, Z_h vertical profiles and Z_{dr} vertical profiles 7) Z_h surface profile, Z_{dr} surface profile and Z_{dr} vertical profiles 8) Z_h surface profile, Z_h vertical profiles, Z_{dr} surface profile and Z_{dr} vertical profiles.

Case 1: Using data from gauges 1, 4, 8, 12, 16, 20 as training data

In this case, data from gauges 1, 4, 8, 12, 16, 20 are used as training data. When only the 9 points belonging to the Z_h surface profile are used as inputs, the average fractional standard error (FSE) the network can achieve is 0.222. If the inputs are increased the inputs by adding Z_h vertical profiles (4 more inputs), the FSE can be decreased to 0.202; In addition, the FSE decreased by adding Z_{dr} surface profiles to the inputs. There is significant performance improvement when the inputs are taken from combined Z_h surface profile and Z_{dr} vertical profile. In this situation, the FSE reaches a value of 0.182, which is substantially lower than

Table 6.2: Performance comparison of neural networks with different inputs (2)

Experiment no.	input variables	inputs number	FSE
1	Z_h_S	9	0.210
2	Z_h_S, Z_h_V	13	0.210
3	Z_h_S, Z_{dr}_S	18	0.195
4	Z_h_S, Z_{dr}_V	13	0.171
5	Z_h_S, Z_h_V, Z_{dr}_S	22	0.200
6	Z_h_S, Z_h_V, Z_{dr}_V	17	0.196
7	Z_h_S, Z_{dr}_S, Z_{dr}_V	22	0.196
8	Z_h_S, Z_h_V, Z_{dr}_S, Z_{dr}_V	26	0.215

using the Z_h surface profile alone as inputs. Because the training data is insufficient in this case, it very easily suffers from the problem of "Curse of Dimensionality" if the number of inputs are increased. This trend is shown very obviously within table 6.1.

Case 2: Using data from gauges 1, 4, 8, 11, 12, 13, 16, 18, 20 as training data

It is shown (as in table 6.2) in this case also that increasing the inputs can not always improve the performance of the network. And the best performance is reached when the Z_h surface profile and Z_{dr} vertical profiles are the combined inputs. This is consistent with what was obtained in case 1.

6.3.2 Input variables and target output transformation

In principle, once the input variables are determined, the raw data from these input variables can be used to build the neural network. For improved efficiency, input variables and target outputs are often passed through some transformation before the neural network is trained. An appropriate transformation can speed-up the convergence of the network, minimize the network size, and most importantly, improve the network's accuracy. This is based on the fact that it is easier to learn a simpler function for a neural network. The prior knowledge about the function

can be incorporated into the neural network by using appropriate transformations of the input variables. In this section, the influence of the input and output transformations to the performance of the rain rate estimation neural network will be studied.

The simple forms of transformation include standardization and rescaling of the input and output data. More complicated transformations include non-linear mapping of the data to new values, in which case, the most important issue is finding the best non-linear mapping function. The form of the transformation function depends on the intrinsic characteristics of the data. The following are among the many factors which need consideration: intrinsic dimensionality of the input data, the correlation of the input data, the potential relation between the target output and the input variables, and the neural network architecture chosen.

For the rainfall rate estimation problem, the inputs are Z_h and Z_{dr} (and possibly K_{dp}). The typical range of Z_h is from 0 dBZ to 60 dBZ, and the typical range for Z_{dr} is from -1 dB to 5 dB. It is clear that their range is not in the same level and the range of Z_h is significantly larger than that of Z_{dr}. If a multilayer perceptron (MLP) neural network is chosen as the network type, it is not strictly necessary to standardize the inputs. However, there are still a variety of practical reasons that standardization can make training faster and reduce the chance of getting stuck in local optimum. If a radial basis function (RBF) neural network is chosen to be the network structure, then the standardization becomes imperative because the input variables are combined via a distance function (such as Euclidean distance). If standardization or rescaling is not used, then the input with bigger range will contribute more than that which is associated with a smaller range. As a result, the effect of the small range inputs may be swamped by the bigger range inputs. It is essential to rescale the inputs so that the input variability reflects their importance, or at least is not in an inverse relation to their importance. If

it is unclear which input is more important, then everything should be rescaled to the same range.

Transformation is performed on the input variables for rainfall estimation. First, a logarithm operation is taken for Z_h and Z_{dr}, then they are rescaled to the range of [0 1] and [-1 1] by dividing them with their possible maximum values in rain, as shown in the following two equations.

$$Z'_h = \frac{Z_h}{60} \tag{6.1}$$

$$Z'_{dr} = \frac{Z_{dr}}{6} \tag{6.2}$$

The logarithm transformation for the inputs is derived from their relation with the output (rain rate). The scatter plot of rain rate vs. horizontal reflectivity (Z_h) is shown in the following figure. Many previous studies have provided many expressions for the R-Z_h relationship. The more typical ones are given as follows.

$$R = 3.65 \times 10^{-2} Z_h^{0.625} \qquad (Marshall\ \ Palmer\ \ relation) \tag{6.3}$$

$$R = 0.0017 \times Z_h^{0.714} \qquad (WSR-88D\ \ Z-R\ \ relationship) \tag{6.4}$$

Even though, the relationships have wide variability because of the wide variations of the raindrop distribution, the relation can be expressed in the following common form.

$$R = C_1 {Z_h}^{C_2} \tag{6.5}$$

where C_1 and C_2 are constants. Taking the logarithm on both sides of the equation, we arrive at,

$$log_{10}(R) = log_{10}(C_1) + C_2 log_{10}(Z_h) \tag{6.6}$$

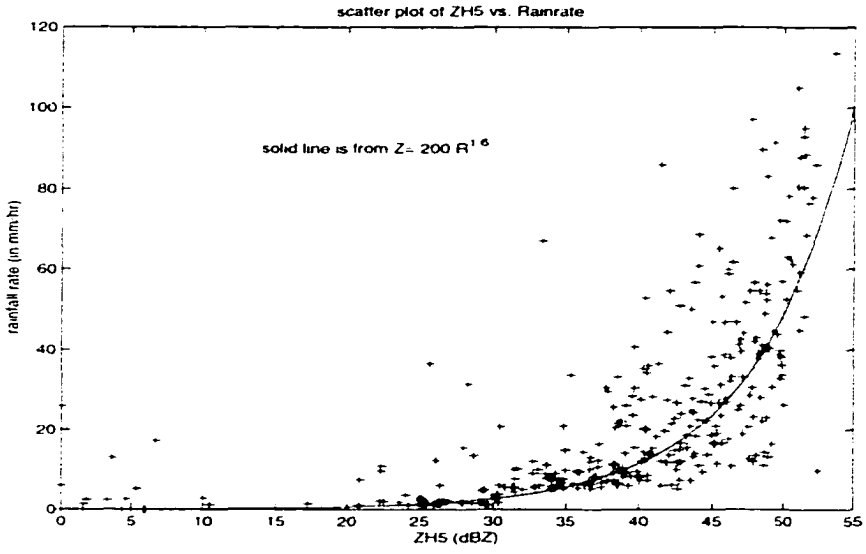


Figure 6.8: Scatter plot of Rain rate (R) vs. $log_{10}(Z_h)$

We can see from this equation that $log_{10}(R)$ is approximately related to $log_{10}(Z_h)$ linearly. Therefore, after the logarithm transformation, the relation becomes simple, and thus it is easier for the neural network to learn. Similarly, other expressions for rain rate are:

$$R(Z_h, Z_{dr}) = 0.0117 \times Z_h^{0.91} \times Z_{dr}^{-0.403} \tag{6.7}$$

$$R(K_{dp}) = 40.5 \times (K_{dp})^{0.85} \tag{6.8}$$

$$R(K_{dp}, Z_{dr}) = 52.0 \times K_{dp}^{0.956} \times Z_{dr}^{-0.447} \tag{6.9}$$

After the transformation, the scatter plots of $log_{10}(R)$ vs. Z_h and Z_{dr} are shown in the Figure 6.9.

To avoid taking logrithm of zero, the logarithm is taken with (R+1), then the output R is rescaled by the following two equations. With the transformation discussed above, neural network models are built with different input combination based on the same data set (CaPE 1993 data) used before. The results are shown in the table 6.3. From this table, we can see that the performance of the network

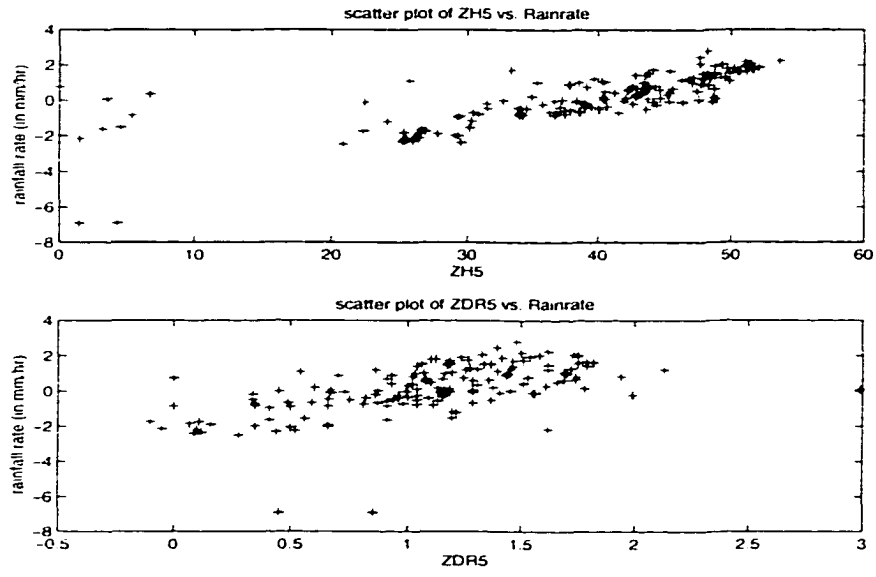


Figure 6.9: Scatter plots of Rain rate $log_{10}(R)$ vs. $log_{10}(Z_h)$ and $log_{10}(Z_{dr})$

with transformation 2 (TF2) is better than that associated with transformation 1 (TF1).

$$TF1(R) = \frac{log_{10}(R+1)}{log_{10}(max(R)+1)} \tag{6.10}$$

$$TF2(R) = \frac{log_{10}(R+1)}{max(R)} \tag{6.11}$$

In summary, the transformation for the inputs and output of the rainfall neural network can be shown schematically in the Figure 6.10.

6.4 Training Data Set Preparation

Another important issue for building a neural network is to choose the appropriate training data set (collection of input data and its associated desired output data). Careful selection of the training data set is so crucial that the performance of the neural network will be affected directly. This is based on the fact that training is accomplished passively in most neural network learning algorithm. A learning algorithm uses whatever training data is presented to it and modifies its

Table 6.3: Comparison the effects of different transforms to the inputs and output

input variables	inputs number	FSE (with TF1)	FSE (with TF2)
Z_h_S	9	0.210	0.193
Z_h_S. Z_h_V	13	0.210	0.198
Z_h_S. Z_{dr}_S	18	0.195	0.189
Z_h_S. Z_{dr}_V	13	0.171	0.170
Z_h_S. Z_h_V. Z_{dr}_S	22	0.200	0.174
Z_h_S. Z_h_V. Z_{dr}_V	17	0.196	0.173
Z_h_S. Z_{dr}_S. Z_{dr}_V	22	0.196	0.169
Z_h_S. Z_h_V. Z_{dr}_S. Z_{dr}_V	26	0.215	0.160

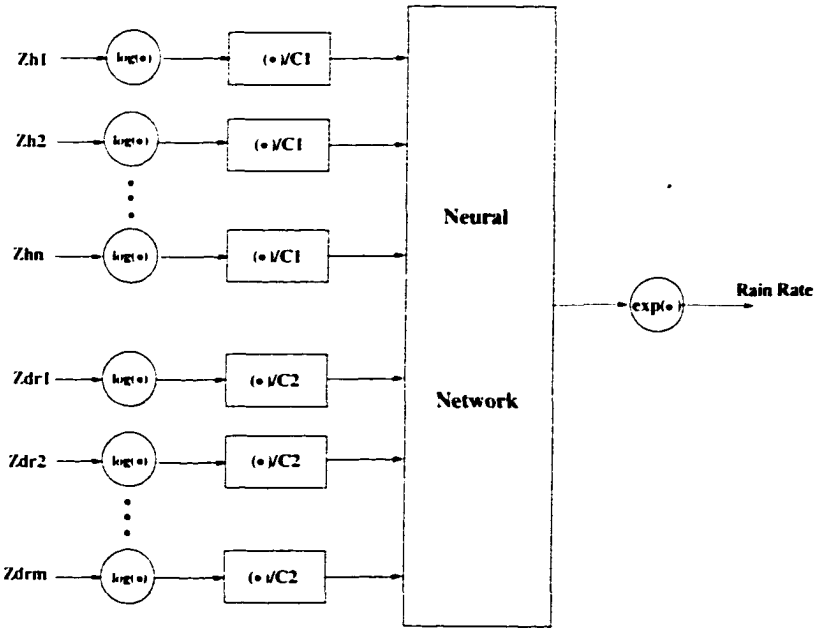


Figure 6.10: configuration of pre-transformation and post-transformation

parameters (weights, biases, centers) so that its output matches the desired output vectors as much as possible. Therefore, the training cases should provide sufficient information to represent the target function. Two important issues related to training data preparations are the representativity and sufficiency of a training data set.

6.4.1 Representative training data set

The final goal of training a neural network is to get valid generalization to unknown cases, i.e., the usefulness of a trained network depends on the prediction accuracy for the future data cases which are not in the training set. There are two different types of generalization: *interpolation* and *extrapolation*. Usually, interpolation can be done reliably, but, extrapolation is not always a reliable form of generalization. Often, it is risky for a neural network to perform extrapolation; and it usually leads to an unusable result. Therefore, to get reliable generalization in the whole input domain, it is essential to choose training cases which are evenly distributed throughout the input space if sufficient data are available. If the training data set is in a small subspace of the entire input space, the neural network established based on the training data may perform well only while inputs are near the subspace, it would probably perform badly if the inputs are far from the subspace because much extrapolation has to be done. Usually, the following three methods can be used to choose training data set if sufficient data are available.

(1) Choose a training data set such that the joint distribution of input variables is close to uniform.

(2) If Condition 1 cannot be achieved, choose a training data set such that the distribution of each input variable itself is uniformly distributed.

(3) Else, choose a training data set such that the target output is uniformly distributed.

The first method is reasonable but very strict. It is complicated to select the training data set with tight requirement of joint distribution of input variables to be uniform when input dimensionality is high. The requirement can be loosened by using method (2), i.e., individual distribution of the inputs is uniformly distributed. Furthermore, it can be simplified to the method (3), only target output distribution is required to be uniform. This method is practical when the number of input variables is very large, and the total available data set is not sufficient enough (as is the case with the rain rate estimation problem because of the insufficiency of raingage measurements), it is impossible to satisfy the first two constraints.

For the problem of neural network rainfall rate estimation, usually, a training data set is chosen randomly from the available data set, and the histogram of the rain rate corresponding to this training data set is drawn. If the histogram is not uniform, then we go back to the available database and pick more or discard some of the data from the training data set according to the current histogram of rain rate. This process continues until the histogram is close to uniform.

6.4.2 Sufficiency of a training data set

Many factors influence the necessary size of the training data set, such as the dimensionality of the input space (or the number of input variables), the free parameters in the network, the quality of the training data itself, and the complexity of the target input-output mapping functions.

First, the training cases should be sufficient to fix the free parameters in the network. Considering a network for function approximation, the parameters of the function is registered in the the weights and biases for a MLPNN, or centers and widths for a RBFNN. To uniquely determine these parameters, the number of training cases (denoted as $N_{TrainingCases}$) should be at least greater than the number of free parameters in the network (denoted as $N_{FreeParameters}$), i.e.,

$$N_{TrainingCases} > N_{FreeParameters}. \tag{6.12}$$

$$N_{FreeParameters} = N_{input} * N_1 + N_1 * (N_{input} + 1) + (N_1 + 1) (for RBFNN) \quad (6.13)$$

$$N_{FreeParameters} = N_{input} * N_1 + N_1 + N_1 * N_2 + N_2 + N_2 + 1 (for MLPNN) \quad (6.14)$$

For a RBFNN with N_{input} inputs and N_1 neurons in the hidden layer, the computation of $N_{FreeParameters}$ can be expressed by equation 6.13. For a 3-layer MLPNN with N_{input} inputs, N_1 neurons in the first hidden layer and N_2 neurons in the second hidden layer, and one output, the computation of $N_{FreeParameters}$ can be expressed by equation 6.14. Usually, the number of training cases necessary is much greater than the limit provided in the above inequality. Using sufficient training data cases is an efficient way to avoid over-fitting. Overfitting is a major problem for a neural network modeling complex systems. To avoid overfitting, the first approach is to increase the size of the training sample. In theory, the solution to the problem of noise is to increase to the sample size. Regardless of how complex a function the network is capable of filtering, it can be prevented from overfitting by making the training sample large enough to compensate for the level of noise in the data. Usually, when using a large number of training cases such that it is about 30 times the number of free parameters, the network is unlikely to suffer serious over-fitting (from experience). Further more, if the training data are noise-free, then only five times the number of free parameters is sufficient. These constrains are shown in the following two expressions.

$$N_{TrainingCases} > 30 \times N_{FreeParameters} \text{ (for noisy data)}$$

$$N_{TrainingCases} > 5 \times N_{FreeParameters} \text{ (for noise-free data)}$$

In practice, there are some severe constraints on sample size. As the target function becomes more complex or the level of noise increases, the statistically desirable sample size goes up rapidly. There may be practical limits to sample size, because resources are required to collect and store the data. To improve

generalization, we should use a sample size that is as large as possible while still being relevant and manageable.

Curse of dimensionality (Bellman 1961) refers to the exponential growth of a hypervolume as a function of dimensionality. Two aspects of curse of dimensionality affecting the neural network are as follows: 1) The number of training cases needed for constructing a neural network (with good generalization capability) increases exponentially with the dimensionality of the input space. However if many inputs are irrelevant, the network will probably behave very poorly. 2) The number of hidden layer neurons needed increases exponentially with the dimensionality of the input space. So high dimension of input space would increase the complexity of the network. Unsupervised learning algorithms and conventional RBFs are prone to this problem.

$$N_{TrainingCases} \propto Dimensionality\ of\ input\ space$$

An experiment is designed in this section to investigate the minimum training data set size needed for radar rainfall estimation by using neural network function approximation technique. As we discussed above, the training data size depends mainly upon the three factors, as follows: 1) The complexity of the target function: to maintain a given accuracy, sample size needs to increase as the target function becomes more complex. 2) the dimensionality of the input vectors: as the dimension of the input space increases, the sample size needed may increase exponentially. 3) the quality of the training data set: to maintain a given accuracy, sample size needs to increase as noise in the data increases. In the following, several scenarios with respect to different number of input variables and different levels of noise contamination have been studied in context of minimum size of training data for radar rainfall estimation. The performance of all these RBF networks for

rain rate function approximation are tested on the same testing data set which is extracted from the input space by computing the normalized error.

(1) Mapping one radar reflectivity factor to rain rate on the ground with noise-free training data samples

In this case, the simplest R-Z relationship ($R_{WSR-88D}(Z_h) = 0.017Z_h^{0.714}$) is assumed to be learned by a RBF neural network from a simulation data set. The simulation data set is obtained by randomly choosing Z_h from a uniformly distributed (from 0 to 60 dBZ) data and find its corresponding rain rate R without adding any noise. The purpose of this, is to find the minimum size of the training data set for learning the specific reflectivity-rain curve. Several examples of the training data set and its corresponding curve learned from this data set by the RBF neural network are shown in Figure 6.11. The number of training data samples in Figure 6.11(1)-(4) are 3, 5, 6 and 7, respectively. From these figures, we can see that the learned function matches the target function very well when the number of training data samples is 7. Figure 6.12 shows the relationship of the normalized error (between the learned function and the target function) over the number of training data samples. When the training samples exceed 7, the normalized error can be as small as 0.0083. Therefore, we can conclude that **7** training samples are sufficient for learning the one dimensional reflectivity-rain function based on the condition of normalized error less than 0.01.

(2) Mapping one radar reflectivity factor to rainfall rate on the ground with noise-contaminated training data samples

The simulation presented in (1) is an ideal case, i.e., the training data are noise-free. In practice, this case does not apply. For the issue of radar rain rate estimation, both radar measurements and rain gauge measurements (which are used as target outputs while training neural network in real case) can NOT be

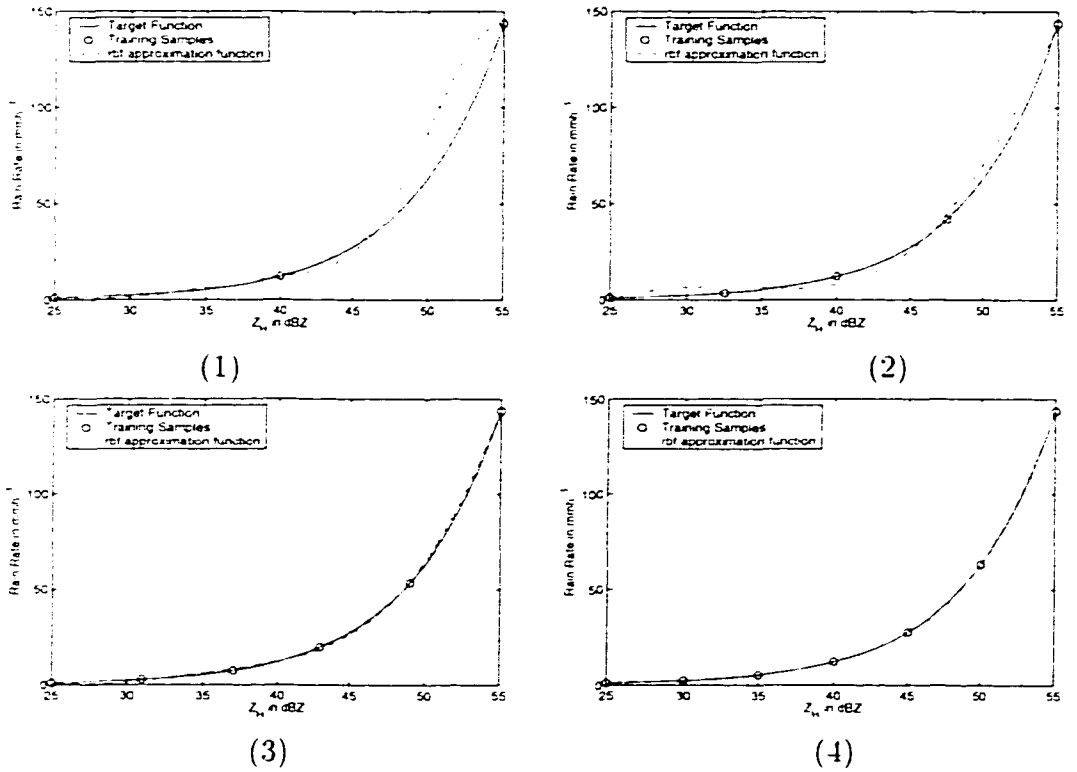


Figure 6.11: Curve fitting of the target function: WSR-88D rainfall algorithm with different training data set size (with noise free) (1) 3 training data samples (2) 5 training data samples (3) 6 training data samples (4) 7 training data samples

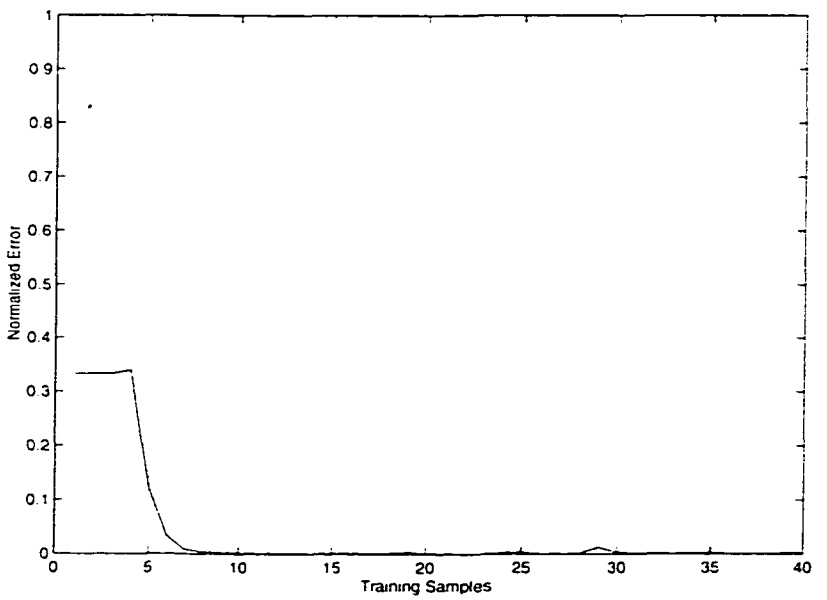


Figure 6.12: the relationship between normalized error and training data sample number

noise-free. Figure 6.13 shows several cases of the rain rate function learned by a RBF neural network. The training data samples are 10. 20. 30. 50 and 60. respectively, for the six cases listed in the figure. All the training data samples are contaminated by noise, the noise for Z_h is assumed uniformly distributed in the range of [-2.5 2.5] dB. From this figure. we can see that many more training data samples are needed for noisy training data set. With 60 training data samples. the learned curve and the target curve are still not matched well. To study the sufficiency of a training data set (with noise) for the 1-D rain rate curve fitting. it is helpful to look the relationship of the normalized error and the number of training data samples. However. the normalized error for training data sets with a fixed number of samples may be different because the noise added to the original training data is random. therefore. 1000 simulations of a fixed number of training samples are carried out so that the probability of normalized error is less than 0.1 can be obtained (as shown in Figure 6.14). From this figure. we can conclude that the probability of normalized error less than 0.1 is over 90% while the number of training data samples is over 73. Therefore. the training data samples needed for the noisy training data (noise in [-2.5 2.5] dBZ range) set is more than 10 times larger than the training data samples needed for noise-free case. this is based on the assumption that normalized error less than 0.1 is enough for this function approximation problem.

(3) Mapping one radar reflectivity (Z_h) and one differential reflectivity (Z_{dr} to rainfall rate on the ground with noise-free training data samples

It is clear that using Z_{dr} and Z_h for rainfall estimation will increase the accuracy compared with using Z_h alone most of time. If both Z_{dr} and Z_h are used as the inputs to a radar rainfall neural network. the problem of rainfall estimation becomes a 2D surface fitting. Intuitively. the number of training samples needed will be larger than a 1-D curve fitting. To study the sufficiency of the training data

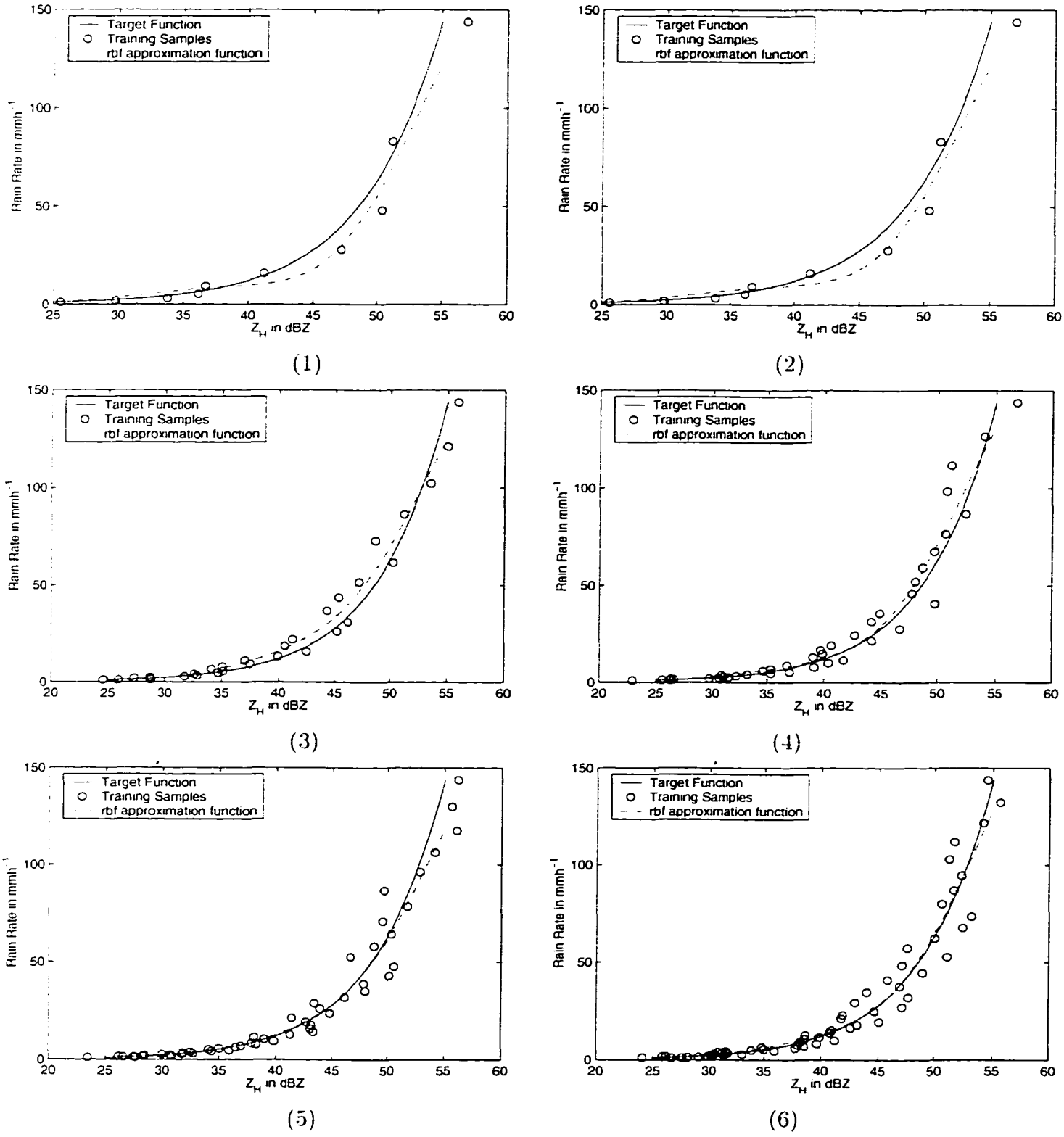


Figure 6.13: Curve fitting of the target function: WSR-88D rainfall algorithm with different training data set size, the training data is contaminated with noise, assume that the noise is Gaussian distributed with variance 1. (1) 10 training data samples (2) 20 training data samples (3) 30 training data samples (4) 40 training data samples (5) 50 training data samples (6) 60 training data samples

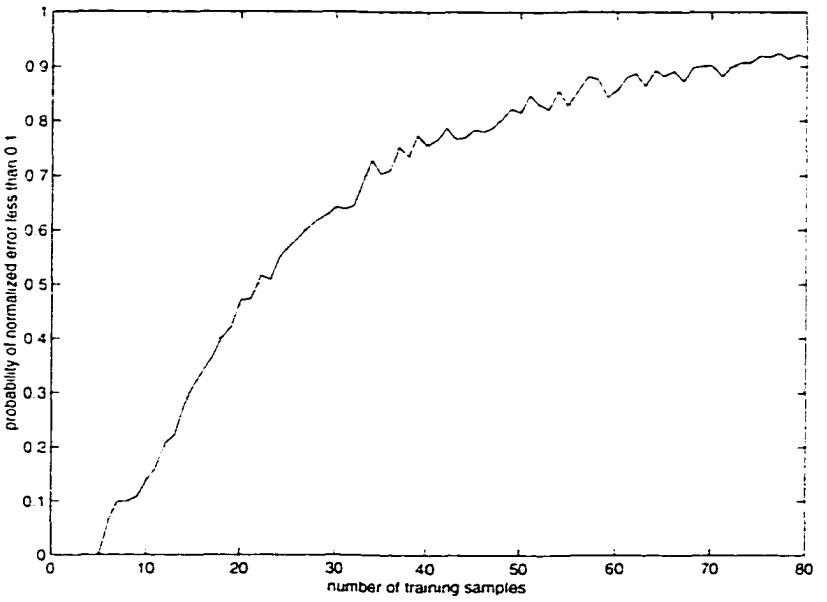


Figure 6.14: the relationship between the probability of normalized error less than 0.1 and training data sample number

set, a target rainfall surface is assumed as $R = 0.0117 Z_h^{0.91} Z_{dr}^{0.403}$, which is shown in Figure 6.15(1). By using (Z_h, Z_{dr}) on 3x3 evenly distributed grids on Z_h - Z_{dr} plane as training data set, the learned surface (represented by a trained RBF network) is shown in Figure 6.15(2). Obviously, the 9 points training data samples is not enough for learning the target rainfall surface. The Figure 6.15 also shows the learned 2D surface from 5x5, 7x7, 8x8 and 20x20 training data samples in (3)-(6). From the figure, we can see that the RBF network learned very well from the 8x8 (total 64) training data samples. The relationship of the normalized error and the number of training data samples is shown in Figure 6.16. The normalized error is around 0.0238 when using 8x8 (total 64) evenly distributed training data samples. If 0.01 normalized error is the goal, then at least 16x16 grids, i.e., total 256 training data samples, can be said sufficient for learning R-(Z_h,Z_{dr}) relationship for noise-free training data.

The simulation results are summarized in table 6.4.

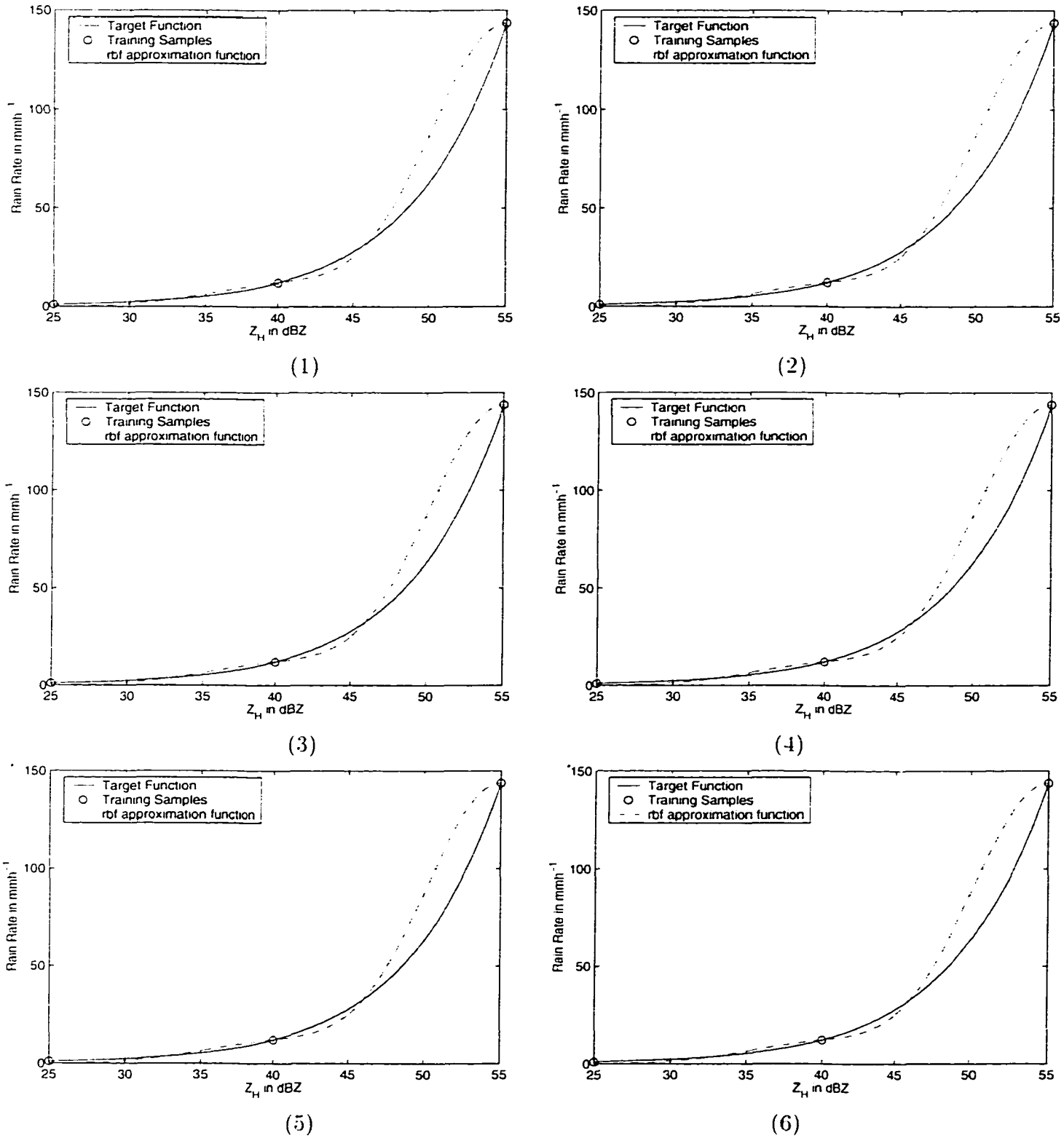


Figure 6.15: Surface fitting of the target function: $R(Z_h.Z_{dr})$ rainfall algorithm with different training data set size. the training data is noise-free. (1) the target rainfall surface (2) 3x3 training data samples (3) 5x5 training data samples (4) 7x7 training data samples (5) 8x8 training data samples (6) 20x20 training data samples

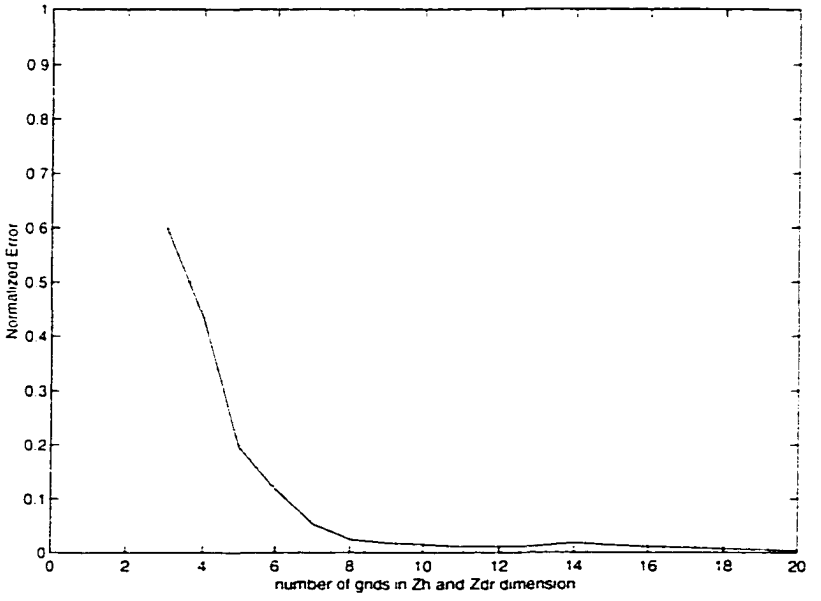


Figure 6.16: the relationship between normalized error and training data sample number

Table 6.4: Summary of minimum sample size for different cases

Target function	$R(Z_h)$	$R(Z_h)$	$R(Z_h, Z_{dr})$	$R(Z_h, Z_{dr})$
Quality of training data	noise-free	noise-contaminated	noise-free	noise-contaminated
Size of training data	7	73	256	2560

6.5 An Adaptive Neural Network Scheme for Precipitation Estimation

The goal of an adaptive neural network is that, the network can adjust itself whenever new raingage data are available (as shown in Figure 6.17). To start, using all the available data so far, the network can be built by initial training. Subsequently, the network can be switched to the application mode. Once new raingage data are collected, the network switches into an updating mode. By using an adaptive updating algorithm, the network adjusts some of its parameters, adds or removes some neurons so as to fine-tune its structure with the new information. The scheme not only provides a fast and an efficient way to build a new neural

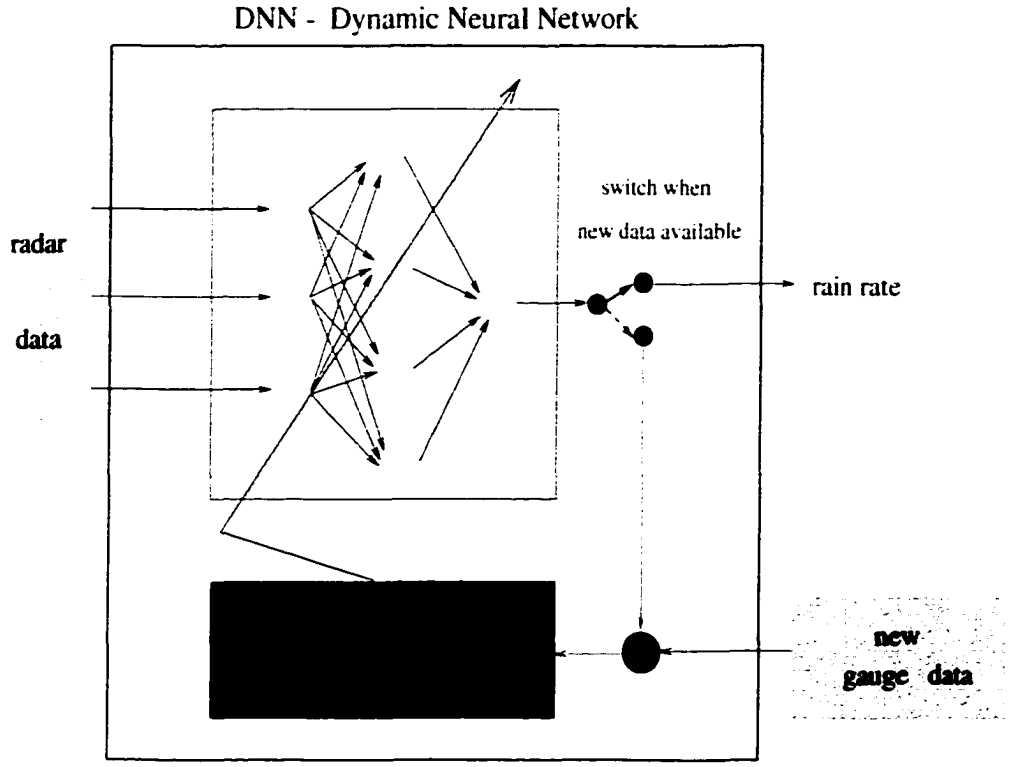


Figure 6.17: Basic structure of the adaptive neural network for rainfall estimation

network rainfall estimation model, but also can provide a way to maintain an existing neural network rainfall estimation model and make it evolve gradually.

Multi-layer feedforward neural network (MLFNN) can be used successfully for the radar rainfall estimation (Xiao and Chandrasekar, 1997). Xiao and Chandrasekar (1997) showed that a Backprogation Propagation Neural Network (BPN) which is a class of the MLFNN that can be used for the radar rainfall estimation. One of the disadvantages of BPN is that the training process is computationally demanding and the learning process is tedious. However once trained, BPN can be successfully used for radar rainfall estimation. The structure and learning algorithm of a BPN make it difficult for implementing an adaptive rainfall estimation algorithm. One of the alternatives suited for rainfall estimation is a Radial Basis Function (RBF) neural network. The RBF network has a unique structure which will make it conducive for adaptive radar rainfall estimation. RBF neural networks

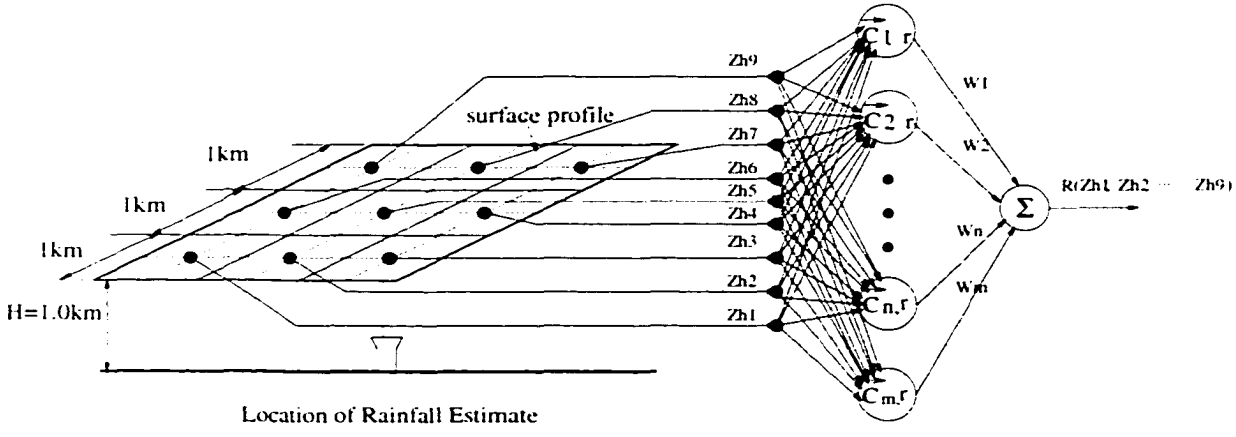


Figure 6.18: Radar rainfall estimation by using a RBF neural network

have local receptive field, therefore the changing of the parameters related to one specific hidden neuron affects the output of the network only in a neighborhood of the center of the hidden unit. Therefore, it is feasible to realize dynamically or adaptively updating its structure and parameters.

6.5.1 Development of a RBF Network for rainfall estimation

In order to understand how the adaptive neural network scheme works, it is essential to introduce the procedure of developing a RBF neural network for radar rainfall estimation. Figure 6.18 shows one of RBF neural network which is used for radar rainfall estimation. In this figure, the inputs are reflectivities extracted at the Z_h surface profile, i.e., the 9 reflectivities Z_{H1}, Z_{H1}, ... , Z_{H9}. There are m neurons (radial basis functions) in the hidden layer. The number m can be determined in the training process, which depends on the training data set. The weighted summation of the radial basis functions is the estimated rainfall rate $R(Z_{H1}, \ldots, Z_{H9})$.

The free parameters of the radial basis function are the center vector $\vec{C}_i$ and width r, which are related to the radial basis functions for the hidden layer neurons, as well as weight vector $\vec{W}_j$ (W1, W1, ... , W9), which are the weights of the connection from the hidden layer to the output neuron. The goal of training the

network is to learn the optimum values for these free parameters from the training data set.

(i) Determination of Size Vectors $\vec{r}_i$

According to the three learning algorithms introduced in Chapter 2 for RBF neural network, r may be set as a constant as in equation 1.17 in the *fixed centers selected at random* method and the *recursive hybrid learning procedure*, or, r may be learned by supervised method in the *stochastic gradient approach*. In this research, the supervised learning algorithm is chosen for the problem of radar rainfall estimation. The available programs for this learning algorithm the functions called "**newrb**" available in MATLAB and "**rbfDesign**". Both methods need the width (or called spread factor, or called size vector) supplied as known inputs. Therefore, the value r has to be determined experimentally based on a specific problem. Large r implies a lot of neurons will be required to fit a fast changing function. Small r means many neurons will be required to fit a smooth function, and the network may not generalize well. We know that there is no sharp change in the function $R(Z_{H1}, \ldots, Z_{H9})$, so we should not choose too small r. Many different r (from 2 to 40, step 0.2) are used in the radar rainfall RBF network, the result shows that the best value for r is 10.

(ii) Determination of Center Vectors $\vec{c}_j$ and Weight Vectors W

In both "**newrb**" and "**forwardSelect**" (Mark, 1997) functions, initially, there is no neurons in the hidden layer. The orthogonal least squares (OLS) method (Chen, 1991) is an effective and efficient algorithm for selecting most significant RBF centers from a given training data set. Therefore, OLS forward selection method is utilized to choose subset of input training set $(\vec{x}_i)_{i=1}^{p}$ to be the centers of the RBFs. It starts with an empty subset, and adds one basis function at a time, tat will yield the largest sum-squared-error according to the OLS method. The criterion used to determine the time for stopping addition of more radial

basis function is based on prediction error. The network has the lowest prediction error when the optimum subset of radial basis functions are chosen. Several standard criteria can be used to measure prediction error, such as, generalized cross-validation (GCV), unbiased estimation of variance (UEV), final prediction error (FPE), or Bayesian information criterion (BIC). When one of these measures stop decreasing, then there should be no more radial basis functions added to the hidden layer(Mark, 1998).

Once the center vectors and width vectors are determined, the computation of weights is straight-forward by using equation 1.19 or the Least-Mean-Square (LMS) algorithm and the Recursive-Least-Square (RLS) algorithms.

6.5.2 Adaptive updating scheme for the RBF network

We can use the procedure introduced above to develop the RBF neural network for rainfall estimation. As days go by, more data become available. Some of the data are unknown to the network, some of the data are known to the network but with different output. To incorporate the information from the unknown data, it is necessary to refine the network by adding or removing some neurons in the hidden layer. For the known data but with greatly different desired output 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,but keep the same structure of the network.

One way to incorporate the new information (from the new data), into the network is simply combining the new data with old training data set to form a larger training data set and retrain from beginning (as shown in Figure 6.19).The most important part in the re-training process is searching for the optimum center set from the new training data set, and this process is tedious. This is not a convenient and practical solution. Another disadvantage is that simple retraining

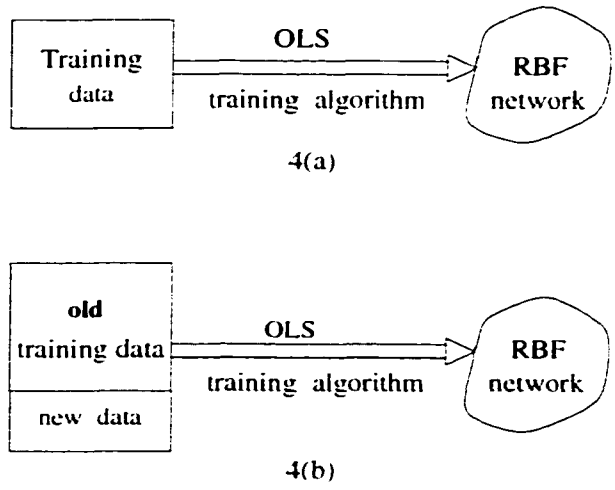


Figure 6.19: Simple combination of new data with previous training data and form the new training data set and the center vectors are selected from it by using Orthogonal Least Square forward selection method

process will not give higher priority to the latest data and cannot make sure the new network can trace any changes in the mean relationship between input and output. Based on these reasons, it is better to use an adaptive RBF neural network for rainfall estimation.

In this adaptive learning scheme developed here, the new network is based on the existing network, but the existing network is modified according to the new data. One of the simplest ways to modify the network for new data is to add or remove neurons as well as change the center vector. The schematic of altering the center vectors is shown in Figure 6.20. The procedure used to modify the RBF network is as follows. First the standard OLS method that was used to build the network from the beginning can be applied to the new data set to come up with new center vector C_2. Forward selection OLS method is used to choose some center vectors from the new data (called center vector *set 2*,C_2): The existing model has center vector set (called center vector *set 1*, C_1), new center vector set can be constructed from these two center vector sets C_1 and C_2.

For $\vec{C_i} \in \mathbf{C_1}$ and $\vec{C_j} \in \mathbf{C_2}$.

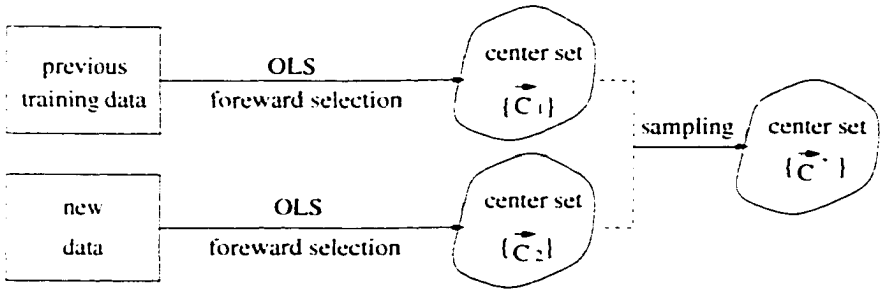


Figure 6.20: A new center subset $\{\vec{C}_2\}$ is obtained by using OLS foreward selection from the new available data. From $\vec{C}_2$ and center set $\{\vec{C}_1\}$ from the existing network, a new center vector set $\{\vec{C}\}$ is selected.

if

$$\| \vec{C}_i - \vec{C}_j \| \leq T \quad for\ i \neq j \tag{6.15}$$

then remove $\vec{C}_i$ ($\| \|$ indicates the Euclidean norm and T is threshold). If the distance from a center vector in the C_2 to any of the center vector in the C_1 is greater than a threshold, then this center vector is added to the RBF, one more neuron is added to the hidden layer. The next parameter to be determined is the set of weights w_j. W_j is determined from (2.19). In this process it is important to determine how different the new data are compared to 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 6.21 shows the schematic diagram of the adaptive updating scheme for RBF-NN rainfall estimation.

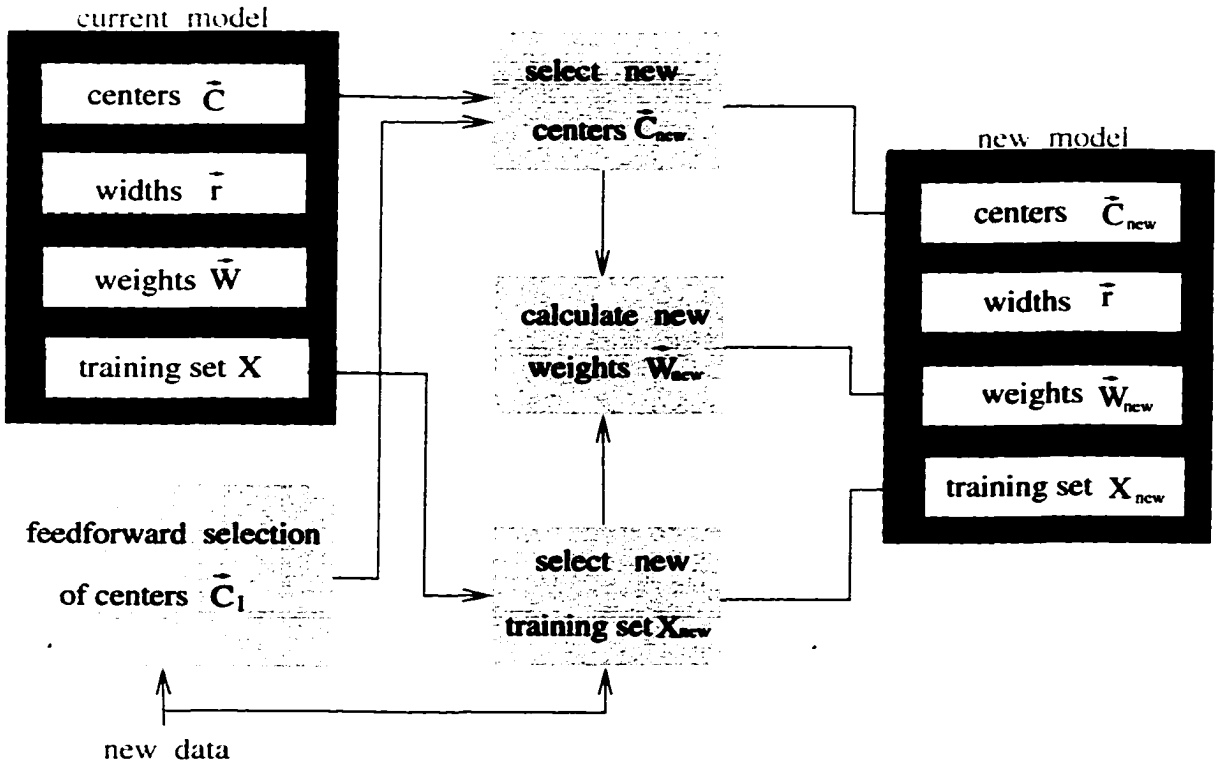


Figure 6.21: Scheme of adaptive RBF neural network for Rainfall Estimation

Chapter 7

APPLICATION OF THE ADAPTIVE RADAR RAINFALL NEURAL NETWORK (RRN)

This chapter describes the application and performance evaluation of the adaptive neural network scheme proposed in Chapter 6 for TRMM (Tropical Rainfall Measurement Mission)(1995).

7.1 Data Source

Radar data used in this study were collected by the Melbourne WSR-88D radar in the summer of 1995 over central Florida. The period of two consecutive months (August and September in 1995) of radar data and the corresponding raingage measurement records were used for this study. The radar volume scan is performed every 10 minutes, and 19 rain gauges in the radar vicinity within 200 km radius provide rainfall accumulation of every 5 minutes. The locations of the raingages are shown in Figure 7.1. Three groups of rain gauges, KSC, SFL, and STJ are marked by "o", "+", and "*", respectively.

7.2 Performance Evaluation

7.2.1 Comparison of a fixed network with an adaptive network for rainfall estimation

To show the benefit of using the adaptive updating scheme, rainfall estimation by using an adaptive neural network for the period of August 21 to 30 is compared

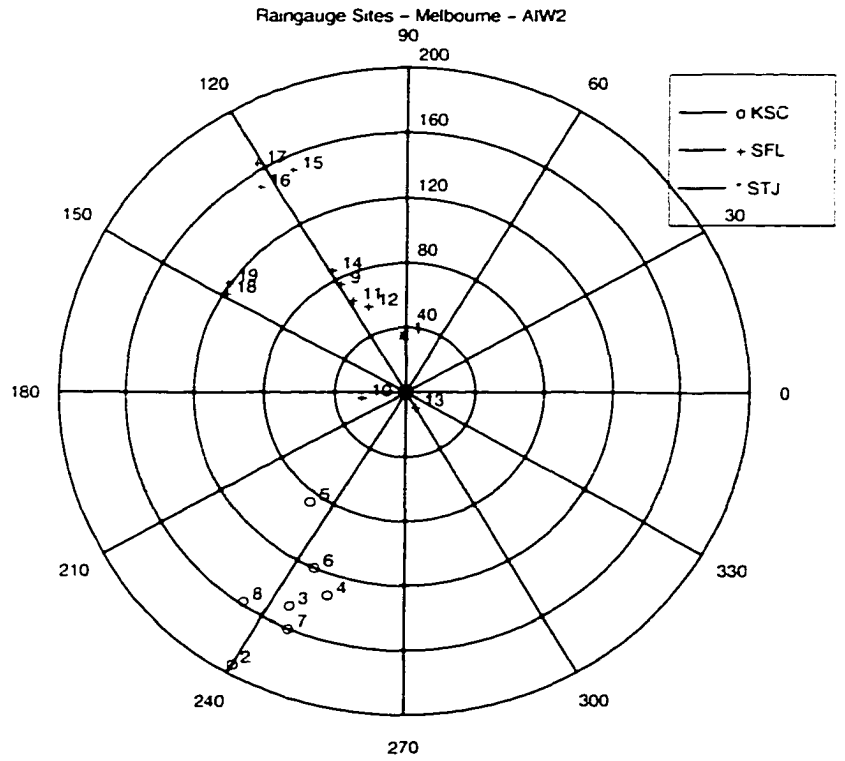


Figure 7.1: The locations of the raingages

against a fixed network. The fixed network is set up based on the radar data and raingage measurement during the period of August 1 to 20. The RBF network is constructed using the algorithm described in section II. This fixed network is denoted "RRN[20]", where RRN stands for Radar Rainfall Neural Network and the number 20 indicates that it is based on the data collected in these twenty days. The adaptive network is based on the initial model (RRN[20]), and adaptively updated everyday when new raingage data become available. The adaptive networks are denoted as RRN[21], RRN[22], RRN[23], ... , in Figure 7.2, which is used to estimate rainfall for the successive days.

The mean hourly rain rate estimation from the fixed network and the adaptive network over the gauge are shown in Figure 7.3 during the period of August 21 to 30, 1995 (240 hours). There are 3 panels in this figure. Panel (b) shows the hourly rain rate estimated by using the fixed network (in dashed line) against the ground

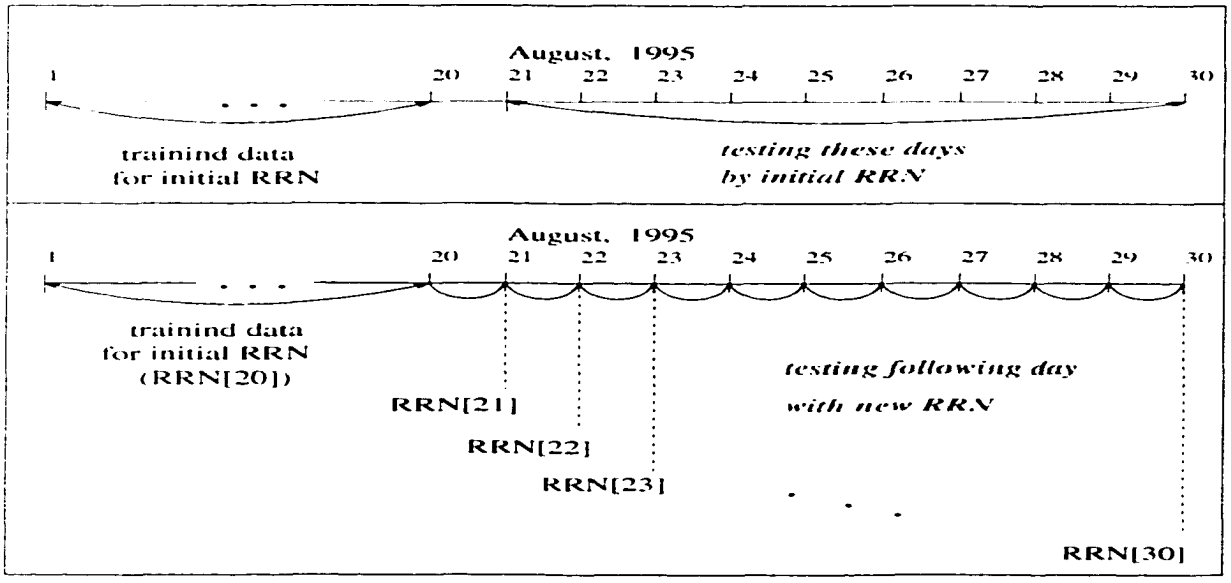


Figure 7.2: The scheme of comparison between a fixed network with an adaptive neural network for rainfall estimation for the 10 day period (August 21 to 30 in 1995).

observation (in solid line). Panel (c) shows the hourly rain rate derived by using the adaptive network compared against the corresponding ground observations. For the convenience of evaluating these two neural network algorithms, the hourly rain rate obtained from the WSR-88D Z-R algorithm is provided in panel (a). The mean daily rainfall accumulation results based on these three rainfall algorithm are shown in Figure 7.4. The statistical analysis of the rainfall estimation results are listed in table 7.1. From this table, it can be seen that the hourly rain rate estimation bias corresponding to WSR-88D Z-R algorithm, the fixed RRN and the adaptive RRN are 0.1875mm, 0.0353mm, and 0.0258mm, respectively; the corresponding FSE (Fractional Standard Error) (Xiao, 1997) are 0.50, 0.29, and 0.27, respectively. As for the daily rainfall accumulation, the estimation bias with respect to the three algorithm are 4.991mm, 0.8416mm, and 0.6226mm, respectively, the FSE are 0.49, 0.16, and 0.10, respectively. It is obvious that both the fixed RRN and the adaptive RRN perform better than the WSR-88D Z-R algorithm, the adaptive

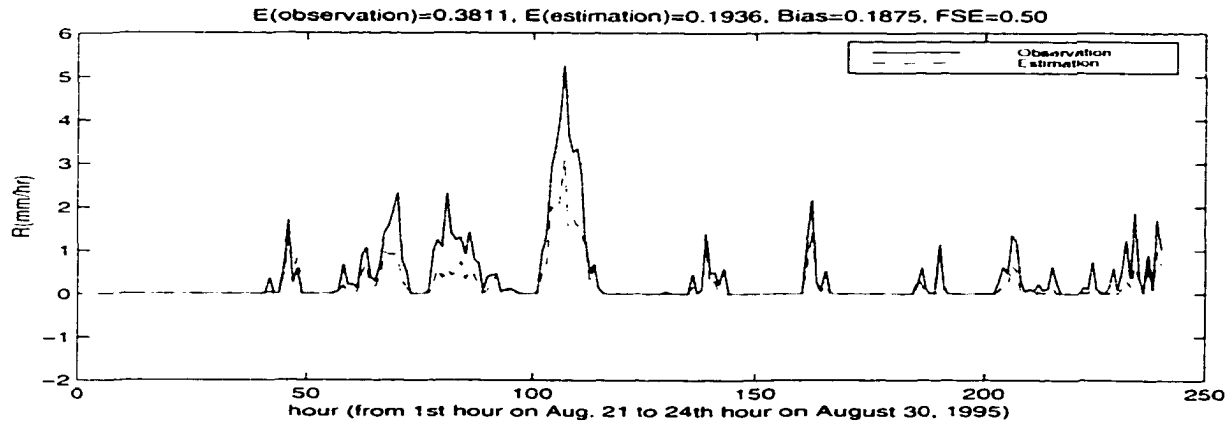
neural network outperforms the fixed neural network. The performance of the network is improved when using adaptive scheme.

Table 7.1: Mean rainfall estimation comparisons between 3 Algorithms during a 10 day period (Aug. 21 - 30, 1995). $\overline{RF}_g$ is the averaged ground observation from all the gauges. $\overline{RF}_e$ is the averaged estimation from NN or WSR-88D R-Z algorithm at all the gauges' locations. CORR is the correlation coefficient. FSE is the Fractional Standard Error. Values of $\overline{RF}_g$, $\overline{RF}_e$, and bias are in millimeters

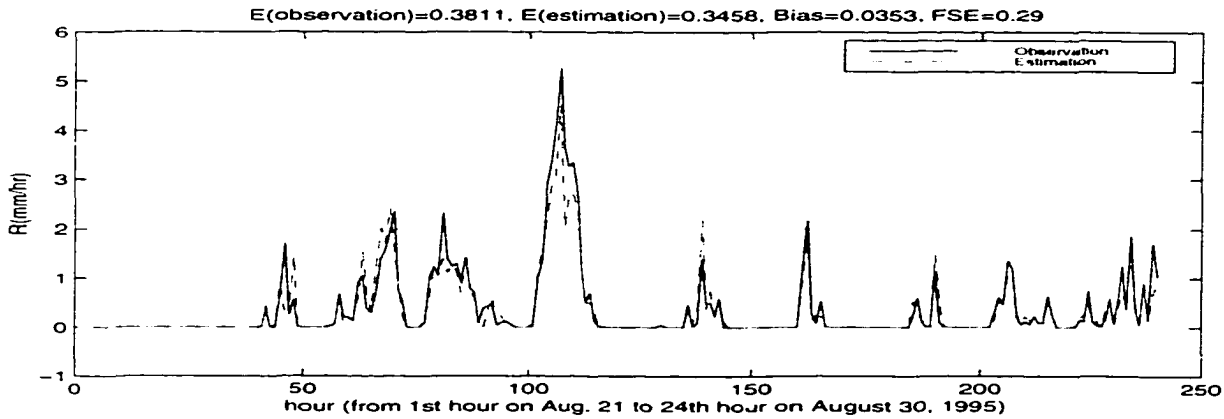
networks	hourly rain rate estimation				
	$\overline{RF}_g$	$\overline{RF}_e$	bias	CORR	FSE
WSR-88D Z-R algorithm	0.3811	0.1936	0.1875	0.95	0.50
the fixed RRN	0.3811	0.3458	0.0353	0.95	0.29
the adaptive RRN	0.3811	0.3553	0.0258	0.95	0.27
networks	daily rainfall accumulation				
	$\overline{RF}_g$	$\overline{RF}_2$	bias	CORR	FSE
WSR-88D Z-R algorithm	9.1463	4.6472	4.4991	0.88	0.49
the fixed RRN	9.1463	8.3026	0.8416	0.89	0.16
the adaptive RRN	9.1463	8.5237	0.6226	0.90	0.10

7.2.2 Comparison of the adaptive RRN network with the completely re-trained network

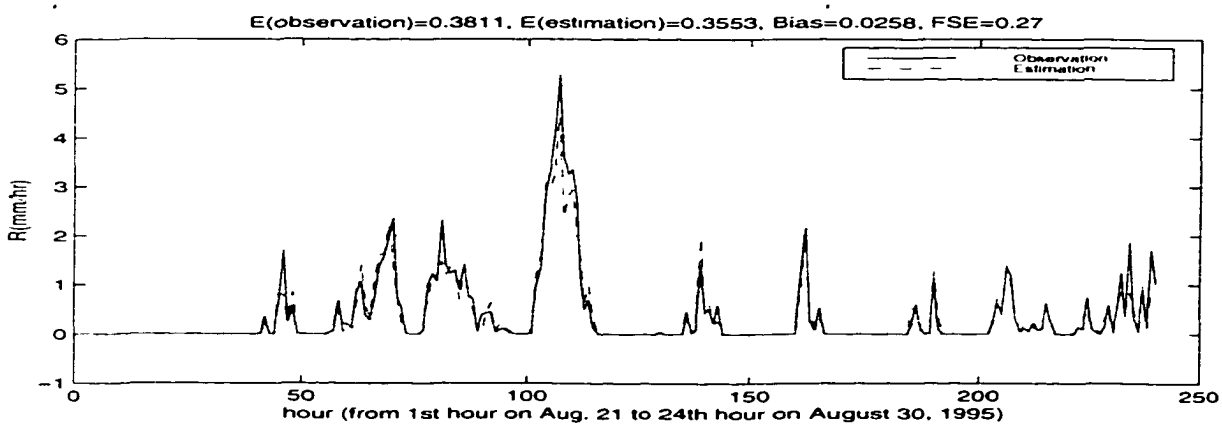
To show the adaptive updated neural network can reach nearly the same level of accuracy as the network which is completely re-trained with all the available data, the 5 day (from August 21 to 25) rainfall estimation are compared by using the adaptive updating scheme and the completely re-training scheme. This exercise also reveals one of the practical advantages of the adaptive scheme over simple re-training scheme. The schematic of the experiment is shown in Figure 7.5. In Figure 7.5, the meaning of RRN[21], RRN[22] etc. is the same as that explained in section 7.2.1. Let RRN_F[21] indicate the Fixed Rainfall Rate Neural Network obtained by combining data from the first 21 days to form a training data set, and then simply train without using the adaptive updating scheme introduced in the



(a) estimation from WSR-88D Z-R algorithm

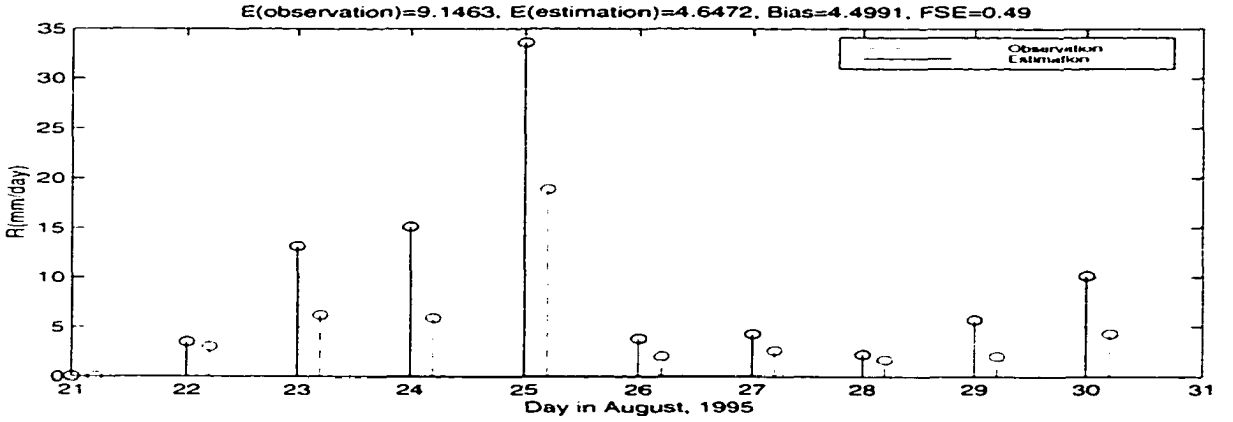


(b) estimation from non-adaptive NN

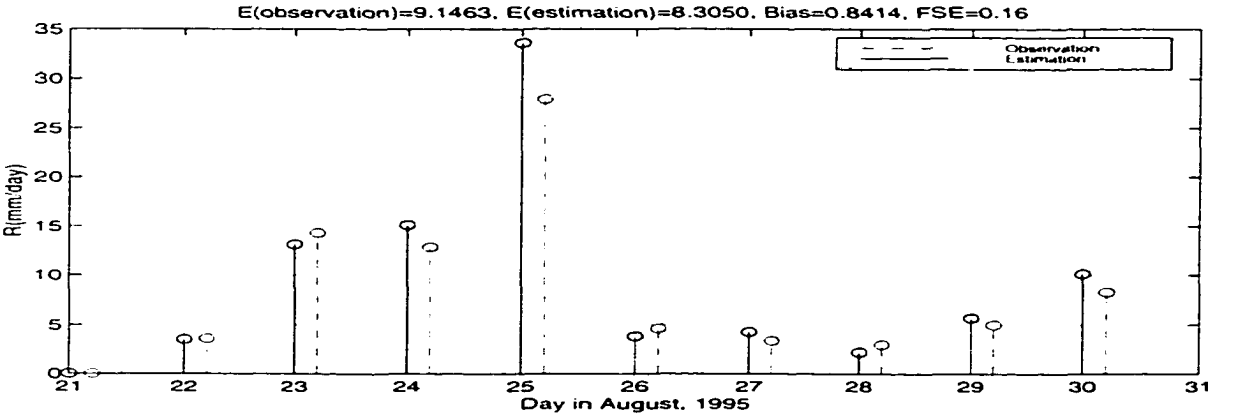


(c) estimation from adaptive NN

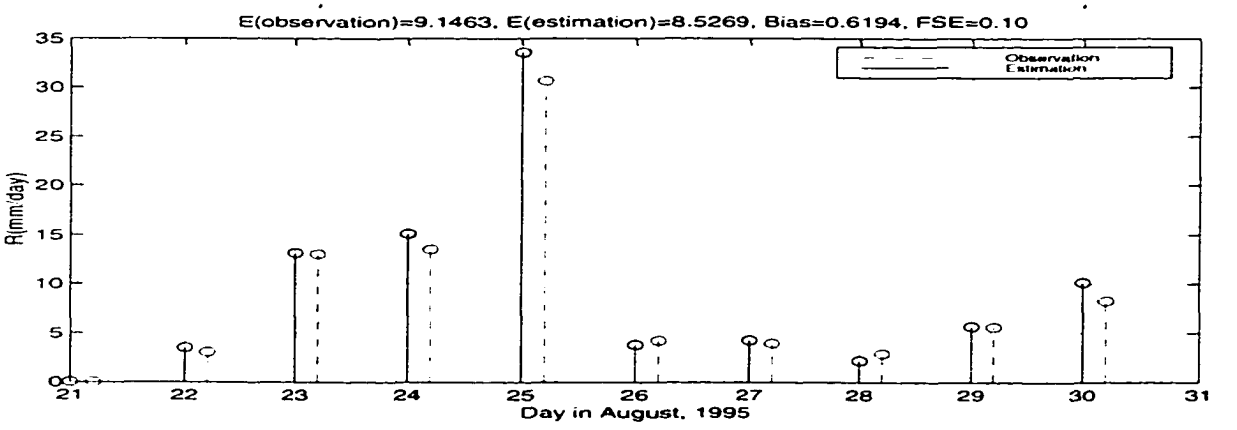
Figure 7.3: Hourly rain rate estimation results comparison between a fixed NN and a adaptive NN (for Aug.21-30, 1995)



(a) estimation from the WSR88D algorithm



(b) estimation from the non-adaptive NN



(c) estimation from the adaptive NN

Figure 7.4: Daily rainfall accumulation estimation results comparison between a fixed NN and the adaptive NN (for Aug.21-30, 1995)

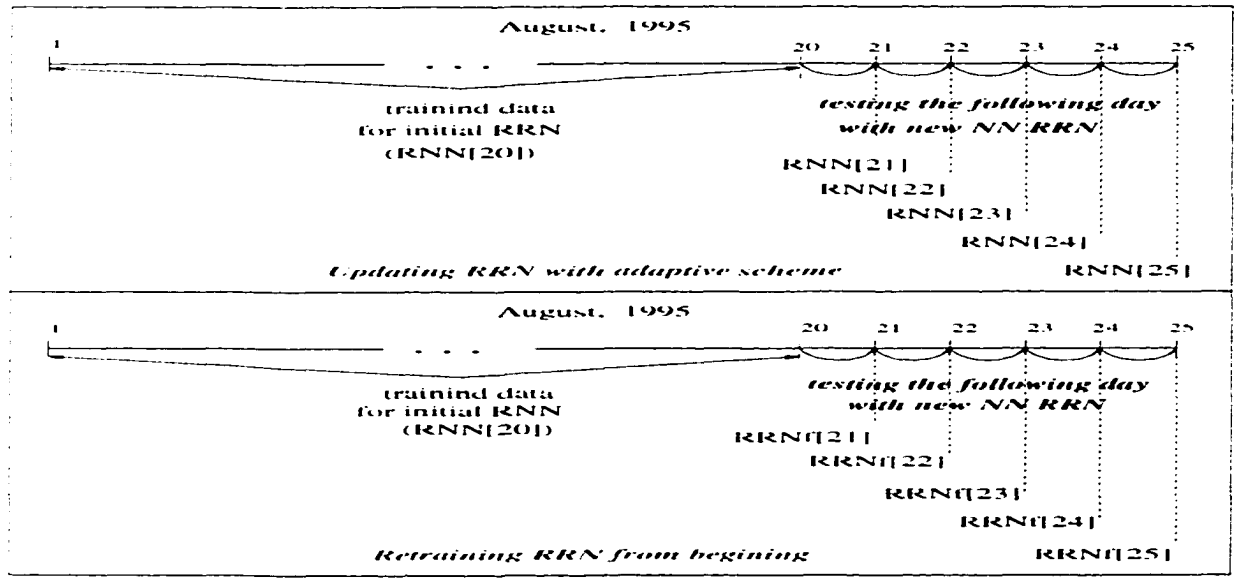
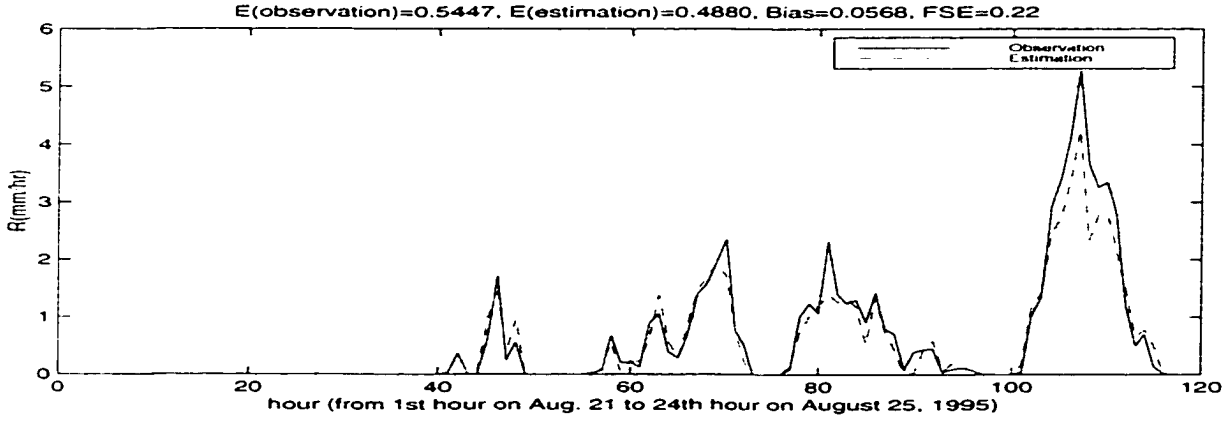


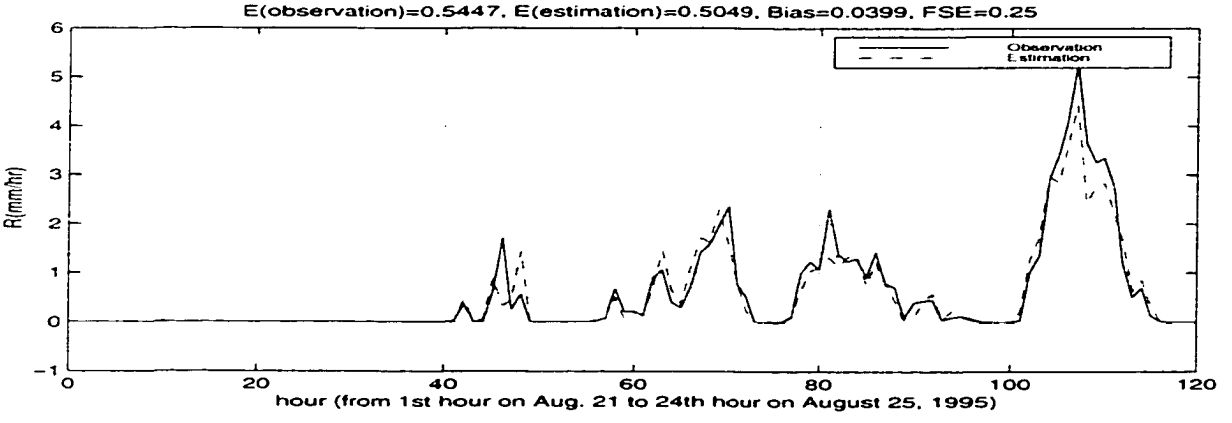
Figure 7.5: The scheme of comparison between a simply re-trained network with a dynamic network for rainfall estimation for the day of August 21 to 25 in 1995.

section II. During the process, the advantage of the adaptive scheme became obvious when the complete training process became computationally very expensive as new data become available. This is an important practical problem. However, by using the adaptive updating scheme introduced above, this process is simplified. This experiment was done only for 5 days' rainfall accumulation due to computational complexity for complete re-training. We can compare the performance of the two schemes. The mean hourly rain rate estimation comparison over the gauge locations is shown in Figure 7.6 and the mean daily rainfall accumulation comparison is shown in Figure 7.7.

Table 7.2 lists the statistical evaluation of this experiment. From the result of Table 7.2, it can be seen that the performances of the simply re-trained RRN and the adaptive updated RRN are similar. Therefore, we can conclude that from this analysis that the adaptive RRN can perform as well as the completely re-trained RRN in all aspects of estimation accuracy, such as bias and percentage error. Furthermore, the adaptive RRN is much simpler, faster and easier to train, and never computationally grows out of control.

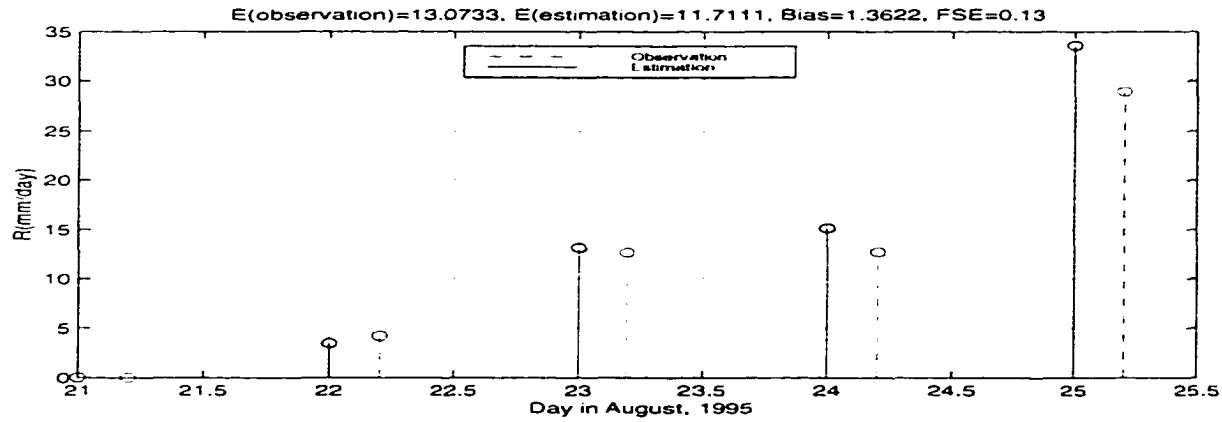


(a) hourly rain rate from ground observation and the simply re-trained NN

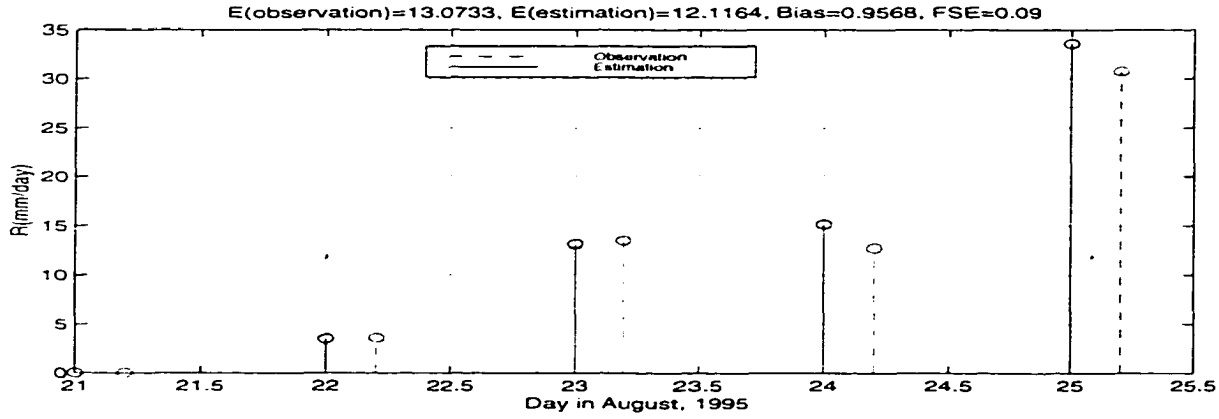


(b) hourly rain rate from ground observation and the adaptive NN

Figure 7.6: 5 days (Aug. 21-25) hourly rain rate estimation results comparison between a simply re-trained NN and the dynamic NN



(a) daily rainfall estimation from ground observation and the simply re-trained NN



(b) daily rainfall estimation from ground observation and the adaptive NN

Figure 7.7: 5 days (Aug. 21-25) daily rainfall accumulation estimation results comparison between a simply re-trained NN and the dynamic NN

Table 7.2: Mean rainfall estimation comparison between simply re-trained network and the adaptive neural network during the 5 day period (Aug. 21-25, 1995). $\overline{RF}_g$ is the averaged ground observation from all the gauges, $\overline{RF}_n$ is the averaged estimation from NN at all the gauges' locations. CORR is the correlation coefficient. FSE is the Fractional Standard Error. Values of $\overline{RF}_g$, $\overline{RF}_n$, and bias are in millimeters

networks	hourly rain rate estimation				
	RF_g	RF_e	bias	CORR	FSE
simply re-trained RRN	0.5447	0.4880	0.0568	0.97	0.22
the adaptive RRN	0.5447	0.5049	0.0399	0.95	0.25
networks	daily rainfall accumulation				
	RF_g	RF_2	bias	CORR	FSE
simply re-trained RRN	13.0732	11.7111	1.3621	0.80	0.13
the adaptive RRN	13.0732	12.0721	1.0011	0.80	0.09

7.2.3 Further comparison of the fixed network with the adaptive network over longer period

To investigate the long-term stability (month to month) of the adaptive RRN scheme, the adaptive RRN is used continuously to estimate rainfall for the next 40 days (Aug. 21 - Sept. 30, 1995). The schematic of this analysis is shown in Figure 7.8.The hourly rain rate and daily rainfall accumulation for the 40 days by using the fixed RRN and adaptive RRN are computed.

Table 7.3 gives a summary of the statistical evaluation of the two schemes for this analysis. Also shown in Table 7.3 are comparison with WSR-88D algorithm. From Table 7.3, it can be seen that by using adaptive RRN rather than the fixed RRN, the bias of hourly rain rates decreased substantially from 6.1% to 2.05%, and the bias of daily rainfall accumulation decreased from 6.18% to 2.23%; Also, FSE for hourly rain rate is decreased from 0.39 to 0.34, and the FSE for daily rainfall accumulation is decreased from 0.21 to 0.17. Table 7.4 shows rainfall estimation

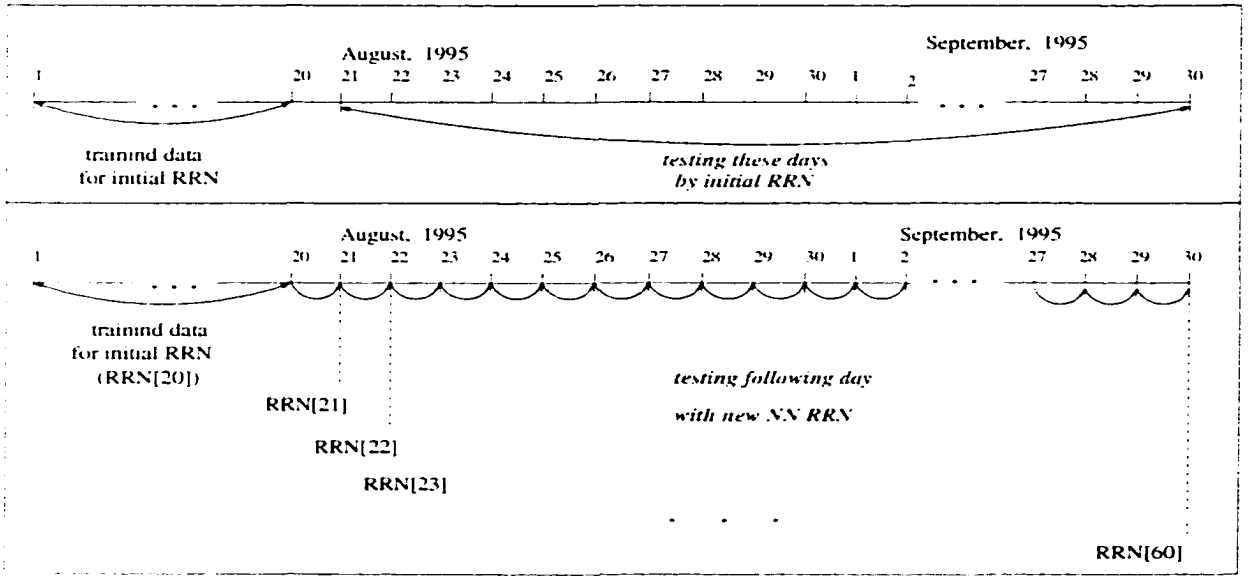


Figure 7.8: The scheme of comparison between a fixed network with a dynamic network for rainfall estimation for the day of August 21 to September 30 in 1995.

comparison between the adaptive RRN and WSR-88D algorithm for single point in space for one month period (September, 1995). In this table, the statistical analysis of rainfall accumulation over 1-day, 5-day, and 25-day (almost one month, no-rain days are removed from the accumulation) are listed. The bias for 1-day, 5-day, and 25-day rainfall accumulation by using the adaptive RRN algorithm is 3.2%, 4.06%, and 3.8%, respectively; the related FSE is 37%, 32%, and 13%, respectively. The bias using WSR-88D Z-R algorithm is 38.91%, 44.82%, and 39.29%, respectively, and the corresponding FSE is 52%, 50%, and 39%, respectively. Therefore, we can conclude that rainfall estimation bias by using the adaptive RRN algorithm is only $\frac{1}{10}$ of that by using WSR-88D Z-R algorithm, and also FSE is decreased by approximately 16%.

7.3 Summary and Conclusion

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

Table 7.3: Mean rainfall estimation comparisons between 3 Algorithms for the period of 40 days' (Aug. 21 - Spet. 30, 1995). $\overline{RF}_g$ is the averaged ground observation from all the gauges. $\overline{RF}_e$ is the averaged estimation from NN or WSR-88D R-Z algorithm at all the gauges' locations. CORR is the correlation coefficient. FSE is the Fractional Standard Error. Values of $\overline{RF}_g$, $\overline{RF}_e$, and bias are in millimeters

networks	hourly rain rate estimation				
	$\overline{RF}_g$	$\overline{RF}_e$	bias (%)	CORR	FSE
WSR-88D Z-R algorithm	0.1465	0.0895	38.57	0.89	0.49
the fixed RRN	0.1465	0.1375	6.1	0.92	0.39
the adaptive RRN	0.1465	0.1435	2.05	0.94	0.34
networks	daily rainfall accumulation				
	$\overline{RF}_g$	$\overline{RF}_2$	bias (%)	CORR	FSE
WSR-88D Z-R algorithm	3.519	2.1530	38.82	0.91	0.44
the fixed RRN	3.519	3.3015	6.18	0.96	0.21
the adaptive RRN	3.519	3.4405	2.23	0.96	0.17

network built for radar rainfall estimation can be gradually modified over time without retraining the network from the beginning. This was achieved by using a Radial Basis Function neural network where new neurons were added by picking up potential new information from the new data and adjusting the center vectors and weights of the RBF. The algorithm also provided priority to new data in the training process. The performance of the adaptive neural network is evaluated by using two months of WSR-88D data collected for the TRMM program. The analysis indicated that the adaptive RRN outperforms the fixed neural network for rainfall estimation. In addition the adaptive network also reach the same estimation accuracy as a completely re-trained fixed RRN with all the available data. When compared with a completely re-trained neural network, the adaptive neural network is easier and faster to set-up because its construction is based on the current existing RRN rather than starting from the beginning.

Table 7.4: Rainfall estimation comparison between adaptive RRN and WSR-88D algorithm for single point in space for almost a month period (September, 1995). $\overline{RF}_g$ is the averaged ground observation from all the gauges, $\overline{RF}_n$ is the averaged estimation from NN at all the gauges' locations. CORR is the correlation coefficient. FSE is the Fractional Standard Error. Values of $\overline{RF}_g$, $\overline{RF}_n$, and bias are in millimeters

Accumulation over	$\overline{RF}_g$	Adaptive neural network			
		$\overline{RF}_n$	Bias (%)	CORR	FSE
1 day	9.3844	9.0773	3.27	0.9111	0.3763
5 day	27.4696	26.3555	4.06	0.9019	0.3237
25 day ($\approx monthly$)	137.7549	132.4481	3.8	0.8838	0.1356
Accumulation over	$\overline{RF}_g$	WSR-88D R-Z Algorithm			
		$\overline{RF}_{r-z}$	Bias (%)	CORR	FSE
1 day	9.3844	5.7331	38.91	0.8413	0.5294
5 day	27.4696	15.1581	44.82	0.8659	0.5083
25 day ($\approx monthly$)	137.7549	83.634	39.29	0.853	0.3929

Chapter 8

SUMMARY, CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

8.1 Summary

The main goal of this research is to develop algorithms for automatic identification/classification of hydrometeor types based on polarimetric radar data, and quantitatively to estimate precipitation such as rainfall rate or rainfall accumulation. In this dissertation, fuzzy logic and neural network techniques are applied to address these issues. Specifically, a neuro-fuzzy network is developed to fulfill the task of automatic classification of the hydrometeor types, an adaptive neural network scheme is developed to estimate rainfall rate and to maintain/update the radar rainfall estimation network on-line over time. The major contributions of this dissertation are summarized as followings:

i) A fuzzy logic hydrometeor type classification has been developed. Five radar parameters, namely, horizontal reflectivity (Z_h), differential reflectivity (Z_{dr}), differential propagation phase shift (K_{dp}), correlation coefficient ($\rho_{hv}(0)$), and linear depolarization ratio (LDR), and corresponding altitude have been used as input variables to the fuzzy logic system. The system can discriminate 10 different hydrometeor types based on the 10 fuzzy rules. According to the fuzzy rule base, 50 membership functions are used to represent the 10 fuzzy sets for the five input variables. The membership functions play an important role in the system.

Both one dimensional and two dimensional beta functions are used as the form of membership functions. This fuzzy logic hydrometeor classification algorithm is implemented by using an objective oriented programming method. both the rule base and the membership function description files are not coded in the programs. instead. they are all external files. which increases the flexibility of the algorithm. In other words. the system can be adjusted simply by modifying the rule base and membership function files without changing the programs or re-compiling the whole program package.

ii) A learning algorithm is developed for fine-tuning the parameters of the membership functions in the system based on in-situ data. This learning algorithm is inspired by regarding a fuzzy logic system as a multi-layer perceptron neural network with the different form of neuron and inter-layer connections, but with the similar multi-layer structure. Essentially, the learning algorithm is a modified back-propagation learning algorithm.

iii) Performance evaluation and in-situ verification of the fuzzy logic hydrometeor classification algorithm have been carried out by applying the algorithm to the radar data collected by the CSU-CHILL radar and compared with the HVPS images gathered by T-28 aircraft, manual hail collection by hail-chase-van instrumented with hail-collection net, and 2D-distrometeors. Many case studies show the effectiveness and high accuracy of the inference.

iv) A RBF neural network algorithm for detection of rain/no-rain condition on the ground is developed. The classification of rain/no-rain is based on the vertical profile of reflectivity above the detection point on the ground. The training of the neural network is carried out by using the data collected by the WSR-88D radar at Melbourne, Florida and several raingage networks located in the vicinity of the radar in July, 1993. The testing and evaluation of the network is studied

by applying the network on two evaluation data sets. one set was collected in August. 1995. and the other one in September. 1995. The effect of using the classification scheme before applying any radar rainfall algorithm is studied also in this dissertation: it has shown that the rainfall accumulation accuracy can be improved as much as 10%.

v) An adaptive neural network radar rainfall algorithm is developed to enable the network to be self-maintaining and self-improving by incorporating new data into the network without going through the tedious and time-consuming re-training process from the beginning. One of the main advantages of the adaptive learning algorithm is that the network fine-tunes the parameters and the structure of the network by using the current neural network as an initial network. A by-product of the algorithm is a procedure for selecting the most representative training data. which is a crucial step that influences the performance of network directly. The performance of the adaptive neural network is evaluated by using two months of WSR-88D data collected for the TRMM program in Melborne (Florida). The analysis indicated that the adaptive network outperforms the fixed neural network for rainfall estimation. In addition. the adaptive network can reach the same estimation accuracy as a completely re-trained RRN with all the available data. Furthermore. the adaptive neural network is easier and faster to set-up because its construction is based on the current RRN rather than starting from the beginning.

8.2 Recommendations for future work

The following areas of investigation could follow on from the work presented in this dissertation:

- The preliminary form of rule base and membership functions describing the fuzzy sets appeared in the rule base for fuzzy hydrometeor classification pro-

posed in this dissertation are based upon the empirical and theoretical experience. Further improvement of the rule base and the membership functions is necessary. This includes adding more possible mixtures of hydrometeor states for increasing the discrimination capability, thus enhancing the representation of the membership functions.

- The learning algorithm of the neuro-fuzzy hydrometeor type classification has to be applied to practical cases as more and more in-situ measurements become available.

- The feasibility of the adaptive neural network needs to be validated by further application to data over many geographical regions and climatilogical regimes.

Bibliography

[1] A. V. Ryzhkov and D. S. Zrnic. Comparison of dual-polarization radar estimators of rain. *Journal of Atmosphere and Oceanic Technology*. 12:249–256. April 1995.

[2] Adam Krzyzak, and Tamas Linder. Radial basis function networks and complexity regularization in function learning. *IEEE Transactions on Neural Networks*. 9(2):247 - 256. March 1998.

[3] Andras Bardossy and Lucien Duckstein . *Fuzzy Rule-Based Modeling with Applications to Geophysical, Biological and Engineering Systems*. Annabooks. first edition. 1995.

[4] Ari Visa, and Jukka Iivarinen. Evolution and evaluation of a trainable cloud classifier. *IEEE Transaction on Geoscience and Remote Sensing*. 35(5):1307 - 1315. September 1997.

[5] Atlas. D. and C. W. Ulbrich . Early foundations of the mesurement of rainfall by radar. In David Atlas, editor, *Radar in Meteorology*. pages 86–97. American Meteor. Soc., 1990. Battan Memorial Conference.

[6] Atlas. D., R. C. Srivastava and R. S. Sekhon. Doppler radar characteristics of precipitation at vertical incidence. *Rev. Geophys. Space Phys.*, 2:1–35. 1973.

[7] Aydin,K., Zhao,Y., and Seliga,T.A. A differential reflectivity radar hail measurement technique: observations during the denver hailstorm of 13 june 1984. *Journal of the Atmospheric and Oceanic Technology*, 7(1):104 - 113, February 1990.

[8] Azimi-Sadjadi, M. R. and R. J. Liou. Fast learning process of multilayer neural networks using recursive least squares me thod. *IEEE Trans. Signal Processing*, 40(2):446-450, 1992.

[9] Bringi, V.N. and A. Hendry . Technology of polarization diversity radars for meteorology. In David Atlas, editor, *Radar in Meteorology*, pages 153-190. American Meteor. Soc., 1990. Battan Memorial Conference.

[10] Bringi, V.N., Seliga,T.A., and Aydin, K. Hail detection with a differential reflectivity radar. *Science*, 225(4667):1145 - 1147, Sept. 1984.

[11] Bringi,V.N., Chandrasekar,V., Meischner,P., Hubbert,J. and Golestani,Y. Polarimetric radar signature of precipitation at s- and c-bands. *IEE PROCEEDINGS-F*, 138(2), April 1991.

[12] Bringi,V.N., Vivekanandan,J., and Tuttle,J.D. Multiparameter radar measurements in colorado convective storms, pt.2, hail detection studies. *Journal of the Atmospheric Sciences*, 43(22):2564-2577, Nov. 1986.

[13] Calleiros, R. V. and I. Zzawadzki. Reflectivity-rain rate relationships for radar hydrology in brazil. *J. Climate and Appl. Meteorol.*, 26:118-132, 1987.

[14] Carpenter, G. A. and S. Grossberg . *Pattern recognition by self-organizing networks*. The MIT Press, Cambridge, Massachusetts, 1991.

[15] Chandrasekar, V., V. N. Bringi, N. Balakrishnan and D. S. Zrnic. Error structure of multiparameter radar and surface measurements of rainfall. part iii: specific differential phase. *J. Atmos. Oceanic Tech...* 7:621-629, 1990.

[16] Christopher M. Bishop . *Neural Networks for Pattern Recognition.* Oxford University Press, first edition, 1995.

[17] David J. Garbrick. *Classification of Random Hydrometeor Images Using Neural Network Technique.* PhD thesis, Colorado State University, Fort Collins, CO, 1994.

[18] Detlef Nauck. Beyond neuro-fuzzy: Perspectives and directions. In *Proc. of the 3rd European Congress on Intelligent Techniques and Soft Computing,* pages 1159–1164, 1995.

[19] Detlef Nauck and Rudolf Kruse. A fuzzy neural network learning fuzzy control rules and membership functions by fuzzy error backpropagation. In *Proc. IEEE Int. Conf. on Neural Networks 1993,* pages 1022–1027, 1993.

[20] Detlef Nauck and Rudolf Kruse. Choosing appropriate neuro-fuzzy models. In *Proc. EUFIT'94,* pages 552–557, 1994.

[21] Detlef Nauck and Rudolf Kruse. Nefclass - a neuro-fuzzy approach for the classification of data. In ACM press, editor, *Proc. of the 1995 ACM Symposium on Applied Computing,* 1995.

[22] Detlef Nauck, Frank Klawonn, Rudolf Kruse. *Neuro-Fuzzy Systems.* Wiley, 1997.

[23] Donald Specht. Probabilistic neural networks. *Neural Networks,* 3:109 – 118, 1990.

[24] Doneaud, A. A., S. Ionescu-Niscov, D. L. Priegnitz and P. L. Smith. The area-time integral as an indicator for convective rain volumes. *J. Climate Appl. Meteorol.,* 23(4):555–561, 1984.

[25] Doviak, R.J. and D.S. Zrnić . *Doppler radar and weather observations.* Academic Press, second edition, 1994.

[26] Eklund, P., F. Klawonn, and D. Nauck. Distributed errors in neural fuzzy control. In *Proc. 2nd Int. Conf. on Fuzzy Logic and Neural Networks IIZUKA'92.* pages 1139-1142, 1992.

[27] Foote, G. B. *Scientific overview and operations plan of CaPE.* National Center for Atmospheric Research, 1991.

[28] Frank Klawonn, Detlef Nauck, and Rudolf Kruse. Generating rules from data by fuzzy and neuro-fuzzy methods. In *Proc. of the 3rd German GI-Workshop: Fuzzy-Neuro-Systems'95,* 1995.

[29] Funahashi, K. On the approximate realization of continuous mappings by neural networks. *Neural Networks,* 2:183-192, 1989.

[30] G. J. Borse . *Numerical Methods with MATLAB.* PWS publishing company, 1997.

[31] G. Scarchilli, E. Gorgucci, V. Chandrasekar, and A. . Self-consistency of polarization diversity measurement of rainfall. *IEEE Transaction of Geoscience and Remote Sensing,* 34(1):22 - 26, 1996.

[32] Gorgucci, E., G. Scarchilli and V. Chandrasekar. A robust estimator of rainfall rate using differential reflectivity. *J. Atmos. Oceanic Tech.,* 11:586-592, 1994.

[33] Gorgucci, E., V. Chandrasekar and G. Scarchilli. Radar and surface measurement of rainfall during cape. *J. Appl. Meteor.,* 1995. In press.

[34] Grezgorz J. Ciah and Witold F. Krajewski. Statistical detection of anomalous propagation in radar reflectivity patterns. *Journal of Atmosphereic and Oceanic Technology,* (11):1026 - 1034, 1994.

[35] Grzegorz J. Ciach, Witold F. Krajewski, Emmanouil N. Anagnostou, etc. Radar rainfall estimation for ground validation studies of the tropical rainfall measurement mission. *J. Appl. Meteor.*, 36:735 - 747, 1997.

[36] Gwo-Jong Huang. *The two transmitters/two receivers CSU-CHILL radar system: Calibration issues and sample data analysis.* PhD thesis, Colorado State University, Fort Collins, CO, 1997.

[37] H. Holler, V.N. Bringi, J. Hubbert, M. Hagen, and P. F. Meischner. Life cycle and precipitation formation in a hybrid-type hailstorm revealed by polarimetric and doppler radar measurements. *Journal of the Atmospheric Sciences*, 51(17):2500 - 2522, 1994.

[38] Hall,M.P.M., Goddard,J.W.F., and Cherry, S.M. Identification of hydrometeors and other targets by dual-polarization radar. *Radio Science*, 19(1):132-140, Jan./Feb. 1984.

[39] Hans Henrik Thodberg. A review of bayesian neural networks with an aapplication to near infrared spectroscopy. *IEEE Transactions on Neural Networks*, 7(1):56 -72, January 1996.

[40] Hartmur Holler. Radar-derived mass-concentrations of hydrometeors for cloud model retrievals. In *27th Conf. On Radar Met.*, pages 453–454. Amer. Meteor. Soc., 1995.

[41] Hongping Liu and V. Chandrasekar. Radar precipitation estimation using neural network. In *28th Conf. On Radar Met.*, pages 202–203. Amer. Meteor. Soc., 1997.

[42] Hongping Liu and V. Chandrasekar. An adaptive neural network scheme for precipitation estimation from radar observation. In *Proceedings, IEEE GARSS'98*, 1998.

[43] Hongping Liu and V. Chandrasekar. Classification of hydrometeor types based on multiparameter radar measurements: Development of a neuro-fuzzy system. In *Proceedings. Conference on Cloud Physics*. pages 253-256. 1998.

[44] Hongping Liu. V. Chandra. Hydrometeor classification based on multiparameter radar measurements: Development of a neuro-fuzzy systems. *Journal of Atmospheric and Oceanic Technology*. 1999. in press.

[45] J. Hubbert. V.N. Bringi. L.D. Carey and S. Bolen. Csu-chill polarimetric radar measurements from a severe hail storm in eastern colorado. *Journal of Applied Meteorology*. 37:749 - 775. August 1998.

[46] J. Park and I. W. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computation*. 3:246–257. 1991.

[47] James A. Leonard and Nark A. Kramer. Radial basis function networks for classifying process faults. *IEEE Control Systems*. pages 31 - 1315. April 1991.

[48] John Yen and Reza Langari . *Fuzzy Logic - Intelligence. Control. and Information*. Prentice Hall. 1999.

[49] Jurgen Schurmann . *Pattern classification A Unified view of statistical and neural approaches*. John Wiley and Sons. Inc.. first edition. 1996.

[50] K. Mike Tao. A closer look at the radial basis function (rbf) networks. *IEEE*. pages 401–405. 1993.

[51] Ken-Ichi Funahashi. On the approximate realization of continuous mapping of neural networks. *Neural Networks*. 2:183 - 192. 1989.

[52] Kou-Lin Hsu. Xiaogang Gao. Soroosh Sorooshian. and Hoshin V. Gupta. Precipitation estimation from remotely sensed information using artificial neural networks. *Journal of Applied Meteorology*. 36:1176–1190. 1997.

[53] Krzysztof J. Cios. George Y. Baaklini. Alex Vary, and Robert E. Tjia. Radial basis function network learns ceramic processing and predicts related strength and density. Technical Report E-7795. NASA. 1993.

[54] Kultegin Aydin. Yuan-Ming Lure. and Thmos A. Seliga. Polarimetric radar measurements of rainfall compared with ground-based rain gauges during maypole'84. *IEEE Transactions on Geoscience and Remote Sensing.* 28(4):443-449. July 1990.

[55] Kurt Hornik. Multilayer feedforward networks are universal approximatiors. *Neural Networks.* 2:359 - 366. 1989.

[56] Lippmann. R. P. . An introduction to computing with neural nets.

[57] M. J. Patyra and D. M. Mlynek . *Fuzzy Logic Implementation and Application.* Wiley. first edition. 1995.

[58] Martin Hagen. Hartmut Holler and Peter F. Meischner. Multiparameter radar characterization of precipitation particles with in situ measurements. In European Comission. editor. *COST 75 Weather Radar Systems International Seminar.* pages 519-526. Brussels. Belgium. 1994.

[59] Matrosov. Sergy Y., Reinking. Roger F., Kropfli. Robert A. and Bartram. Bruce W. Estimation of ice hydrometeor types and shapes from radar polarization measurements. *Journal of Atmospheric and Oceanic Technology.* 13(1):85-96. Feb. 1996.

[60] Moody. J. E. and C. J. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation.* 1:281-294. 1989.

[61] N. K. Bose. and P. Liang . *Neural Network Fundamentals with Graphs. Algorithms. and Applications.* McGraw-Hill. Inc.. first edition. 1996.

[62] Park. J. and I. W. Sanberg . Universal approximation using radial-basis-function networks. *Neural Computation*, 3:246–257, 1991.

[63] Pruppacher, H.R., and Klett, J.D. *Microphysics of Clouds and Precipitation*. Kluwer Academic Publishers, 1997.

[64] Richard P. Lippmann. An introduction to computing with neural nets. *IEEE ASSP Magazine*, pages 4 – 22, April 1987.

[65] Rongrui Xiao and V. Chandrasekar. Development of a neural network based algorithm for rainfall estimation from radar observations. *IEEE Transactions on Geoscience and Remote Sensing*, 35(1):160–171, 1997.

[66] Rongrui Xiao, V. Chandrasekar and H. Liu. Development of a neural network based algorithm for radar snowfall estimation. *IEEE Transaction on Geoscience and Remote Sensing*, pages 716–724, May 1998.

[67] Rongrui Xiao, V. Chandrasekar, H. Liu and E. Gorgucci. Detection of rain/no rain condition based on radar data and neural network. In *IEEE GARSS'98*, pages 159–161, 1998.

[68] S. Chen. Nonlinear time series modelling and prediction using gaussian rbf networks woth enhanced clustering and rls learning. *Electronic Letters*, 31(2):117 – 118, January 1995.

[69] S. Chen, C. F. N. Cowan, and P. M. Grant. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks*, 2(2):302–309, March 1991.

[70] S. Chen, S. McLaughline and B. Mulgrew. Complex-valued radial basis function network, part i: Network architecture and learning algorithms. *Signal Processing*, 35:19 – 31, 1994.

[71] Sachidanada, M. and D. S. Zrnić. Rain rate estimates from differential polarization measurements. *J. Atmos. Oceanic Tech.*, 4:588–598, 1987.

[72] Saleem A. Kassam and Inhyok Cha. Radial basis function networks in nonlinear signal processing. *IEEE*, pages 1021 – 1025, 1993.

[73] Seliga, T. A. and V. N. Bringi. Potential use of radar differential reflectivity measurements at orthogonal polariza tions for measuring precipitation. *J. Appl. Meteor.*, 15:69–76, 1976.

[74] Seo D. J. Real-time estimation of rainfall fields using radar rainfall and rain gauge data. *Journal of Hydrology*, 108(1-2):37 – 52, 1998.

[75] Simon Haykin . *Adaptive Filter Theory*. Prentice Hall, 3rd edition, 1996.

[76] Simon Haykin . *Neural Networks*. Prentice Hall, 2rd edition, 1998.

[77] Smith Paul L. Equivalent radar reflectivity factors for snow and ice particles. *Journal of Climate and Applied Meteorology*, 23(8):1258–1260, 1984.

[78] Straka, J. M., and D. S. Zrnic. An algorithm to deduce hydrometeor types and contents from multi-parameter radar data. In *Preprints 26th Radar Meteor. Conf.*, pages 513–515, Boston, 1993. Amer. Meteor. Soc.,.

[79] Stuart Geman, Elie Bienenstock, and Rene Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1 – 58, 1992.

[80] T. Kohonen . *Self-Organizing Maps*. Springer, 1995.

[81] Ted Heske and Jill Neporent Heske . *Fuzzy Logic for Real World Design*. Annabooks, first edition, 1996.

[82] Tong, H. . Chandrasekar,V.. Knupp, K.R. and Stalker,J. Multiparameter radar observations of time evolution of convective storms: Evaluation of water budgets and latent heating rates. *Journal of Atmospheric and Oceanic Technology*, pages 1097–1109, 1998.

[83] V. N. Bringi and V. Chandrasekar . *CSU-CHILL Weather Radar*. 2000.

[84] Valluru Rao and Hayagriva Rao . *C++ Neural Networks and Fuzzy Logic*. MT Books, 2nd edition, 1995.

[85] Xiao, R. *Neural Network Applications In Weather Radar Systems*. PhD thesis, Colorado State University, Fort Collins, CO, 1996.

[86] Zadeh, L. A.. . Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. Syst. Man Cybern.*, SMC-3:28–44, 1973.

[87] Zadeh, L.A. A computational approach to fuzzy quantifiers in natural languages. *Computers and Mathematics*, 9(1):49–184, 1983.