



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT



Lehrstuhl für Technische Elektronik

Lehrstuhl für Technische Elektronik

Prof. Dr.-Ing. Dr.-Ing. habil. Robert Weigel

Prof. Dr.-Ing. Georg Fischer

Masterarbeit

im Studiengang

“Elektrotechnik, Elektronik und Informationstechnik (EEI)”

von

Zhengnan Lin

zum Thema

**2-Dimensional Direction of Arrival Estimation for
FMCW mm Wave Radar using Artificial Neural
Networks**

Betreuer:

Anand Dubey
Jonas Fuchs

Beginn:

18.12.2019

Abgabe:

23.07.2020

Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde.

Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Erlangen, den 30.06.2020

Zhengnan Lin

Abstract

This thesis implements and evaluates different models of artificial neural networks for 2-dimension DOA estimation for linear FMCW radar systems. Firstly, a 77GHz FMCW radar simulator is implemented in MATLAB platform, then the maximum detectable number of targets for specific radar configuration using MUSIC algorithm is discussed. Then the different type neural network based efficient 2D DOA estimation algorithms for 1 to 3 random number of targets using MATLAB simulated single chirp signal of 1-transmitter(rx) 3x3-receiver(rx) 77GHz automotive radar sensors is developed. Finally a few performance comparison between different neural network models and traditional approaches are shown.

Contents

List of Abbreviations	ix
1 Introduction	1
2 problem formulation	3
3 Theoretical Basis	5
3.1 FMCW radar	5
3.2 Radar signal description	6
3.3 Target range detection	7
3.4 Target velocity detection	9
3.5 Range-Doppler map	11
3.6 Direction of arrival estimation	13
3.6.1 Phase-comparison monopulse	13
3.6.2 MUSIC algorithm	15
3.7 Artificial neural networks	17
3.7.1 Perceptron	17
3.7.2 Activation function	17
3.7.3 Multilayer perceptron	20
3.7.4 Back propagation	20
3.7.5 Artificial neural network training	22
3.7.6 Convolutional neural network	22
3.7.7 Recurrent neural network	27
4 2D DOA Estimation using MUSIC Algorithm	33
4.1 FMCW mm Wave Radar Simulator establishing in MATLAB	33
4.2 Determining the maximum detectable targets number using MUSIC algorithm under given radar configuration	35
5 Data Pre-processing for Artificial Neural Network	37
5.1 Convert complex data into real data	37
5.2 Data normalization	38
5.3 Label design	38
6 2D DOA Estimation using Artificial Neural Network	41
6.1 ANN based 2D DOA angle estimation using data with value label	41
6.1.1 Single target 2D DOA angle estimation	41
6.1.2 Double targets 2D DOA angle estimation	46
6.1.3 Three targets 2D DOA angle estimation	49

6.1.4	Random number targets 2D DOA angle estimation	50
6.2	ANN based 2D DOA angle estimation using data with one hot label	57
7	Summary and Outlook	61
Bibliography		63
List of Figures		65

List of Abbreviations

ADAS	Advanced Driver Assistance Systems
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
DOA	direction of arrival
FFT	Fast Fourier Transformation
FMCW	frequency modulated continuous wave
GRU	gated recurrent unit
MUSIC	multiple signal classification
RCS	Radar cross-section
RNN	Recurrent neural network
SNR	signal noise ratio

1 Introduction

One of the background of the article is the necessity to improve radar technology. Radar has been widely used for a long time in almost every field of transportation including aerospace, nautical, and car driving. In recent years, the autonomous driving has received more and more attention. It is expected that Level 4 and 5 autonomous cars will become a large market by 2030 with size of 60 billion U.S. dollar. The radar has already occupied an important position in area of ADAS, with the growing of the autonomous driving market size, efficient radar sensing technology will be strongly demanded.

The environmental perception sensors in self-driving car are shown in figure 1.1. The radars in self-driving sensor system are used to detect the velocity, distance and direction of the objects in the environment of driving. In traditional approach, for example in ADAS, object direction detecting using radar is limited in azimuth angle, which can detect the existing of objects but not the exact shape of that. But with the developing of ADAS and self-driving systems, more effective performance in environment perception is required. Therefore 2D DOA estimation using radar should be paid more attention, if both azimuth and elevation angles of objects are detected, it is possible to get object shape information. As the next step, the shape information can be combined with that of other sensor to serve the object identify in driving environment, for example pedestrian behavior prediction.

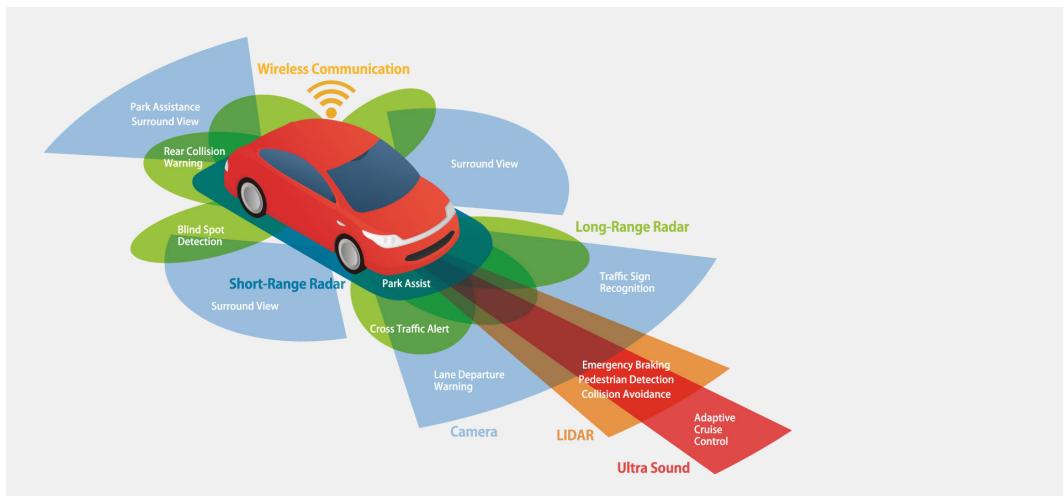


Figure 1.1: autonomous driving sensors

As for DOA estimation in traditional approach, there have been several approaches like maximum likelihood method of Capon, Burg's maximum entropy method, MUSIC method and so on. According to the conclusion of Massachusetts Institute of Technology's Lincoln Laboratory, among the algorithms, MUSIC is the most promising and a leading candidate for further study though has a high cost in computation[1]. Until now, MUSIC is one of the most popular

algorithms in DOA estimation. In this article, the result of neural network approach is compared with MUSIC to confirm the effectiveness of the new approach in compare with traditional approach.

Another background of the article is machine learning. In recent years machine learning gets widely attention in combining with traditional fields. It is seen as a subset of artificial intelligence, machine learning algorithms use specific mathematical model to find out the relationship between data or that of data and its corresponding labels using a certain amount of sample data, this process is called training. After training, machine learning model can be used to predict the unknown part of data. Machine learning is used widely in many areas, such as computer vision, natural language processing and many other areas. The advantage of machine learning over traditional approach is, there are no necessity to find out the theoretical explanation for the algorithm, with enough sample data and suitable neural network architect it is possible to develop machine learning based algorithm to solve the problem.

In area of DOA estimation, there are several works show the solution using neural networks [2–4], in these papers work on single target DOA estimation was proposed, which pointed out the potential of neural network in DOA estimation. However in terms of 2D DOA estimation for multiple targets, there are still no too much result using machine learning, and the shown results mainly focus on neural network form of fully connected layers. In addition to fully connected layers, there are also other types of neural networks with good potential in data processing. One of them is CNN, which widely used in computer vision. RNN shows good performance in time continuous signal processing, which is in field of natural language processing very popular. In this article it is tried to implement and evaluation single and multiple targets 2D DOA estimation based on several different architect types of neural networks, especially CNN and RNN.

In this work, neural network models are trained with simulated data. The simulations assumed a 3x3 rectangular array composed of isotropic antenna elements and ideal propagation environment.

The article is organized as follows. In chapter 2, simulation scenario and specific radar system configuration will be given. chapter 3 introduces theoretical basis for radar signal model, traditional DOA estimation algorithms, and ANN models. In chapter 4, the maximum number of detectable targets for specific radar configuration will be discussed, then the target number to be detected using neural network is decided, then follows the simulated data pre-processing procedure for neural network training. Then chapter 5 follows, which describes neural network based 2D DOA estimation. Besides a comparison between different neural network models and MUSIC will be shown in chapter 6. Finally is chapter 7, the summary and outlook.

2 Problem Formulation

This chapter shows the application scenario of the algorithm and specific simulation parameters.

2D DOA estimation has other significance in compare with 1D DOA estimation. In a ADAS system, a high resolution 1D DOA estimation mainly works to detect existence of objects, but high resolution multi targets 2D DOA estimation can be used to get the shape information of objects. In other words, 2D DOA estimation provides the possibility, that to achieve target classification using radar. In a ADAS or self-driving system, one application field of target classification is to identify the objects nearby vehicle. Figure 2.1 shows the typical environment perception requirement in autonomous driving: every things including other vehicles, pedestrian, obstacles who affect driving safety should be identified in environment. Until now camera and LIDAR sensors played the main role in this field, but it is also possible if radar can get the shape information of objects using 2D DOA estimation. From this perspective, multi targets 2D DOA estimation has a good potential in autonomous driving environment perception.

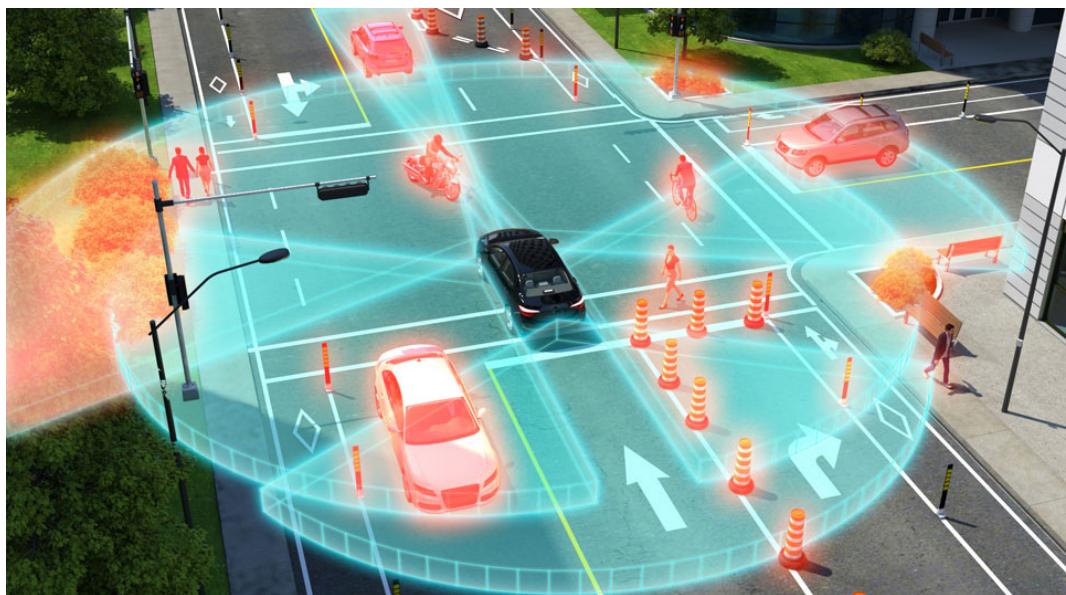


Figure 2.1: environment perception in autonomous driving

In this article, it is tried to implement a neural network based 2D DOA estimation algorithm of short ranged object.

In terms of targets, target range is limited in 2-5m, RCS of targets is 0.04m^2 , target azimuth angle is limited in from -30° to 30° , target elevation angle is limited in from -20° to 20° .

As to radar configuration, an upper linear FMCW waveform is simulated, for FMCW waveform is widely used in ADAS systems, frequency range is from 76GHz to 77GHz, chirp dura-

tion time is $288\mu\text{s}$, sample rate is $256/\text{s}$. Specific concept of the FMCW waveform is presented in following chapter. A 1 transmitter 3x3 rectangular receiver array antenna is used, whose spacing is 0.5 wavelength.

Since the neural network based algorithm uses single chirp of radar signal, velocity of radar platform and targets has no influence to the estimation result. Therefore a stable radar platform is used, target velocity can be randomly set, here assumed as 2m/s in random direction, in order to imitate the low velocity targets in driving environment such as pedestrian.

3 Theoretical Basis

3.1 FMCW radar

FMCW radar is abbreviation of Frequency-Modulated Continuous Wave radar, which can radiate continuous transmission power with the changing operating frequency.

Usually the operating frequency of FMCW increases linear, the figure 3.1 shows a FMCW signal in terms of amplitude vs time.

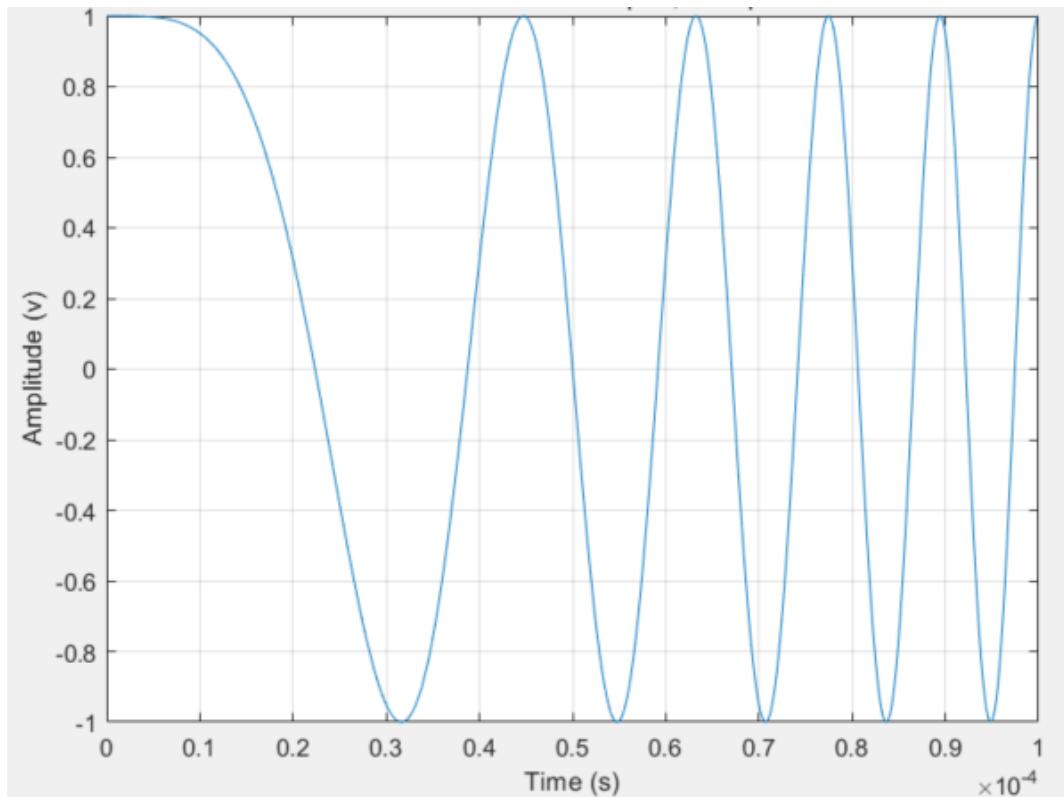


Figure 3.1: FMCW signal in terms of amplitude-time

Figure 3.2 presents FMCW signal in terms of frequency vs time, whose starting frequency is 77GHz, bandwidth is 1GHz, and chirp duration is $288\mu s$. In this article, this kind of signal is used to estimate the DOA angles. The reason why the signal frequency is 77GHz is that in recent years more and more in ADAS system used radars take 77GHz as its operating frequency.

Block diagram 3.3 shows how does radar signal work in the radar system. At first, the signal generator generates FMCW signal, then it is sent to TX antenna and mixer. Tx antenna transmits the signal, and signal propagates in the space, when the signal reaches the target surface,

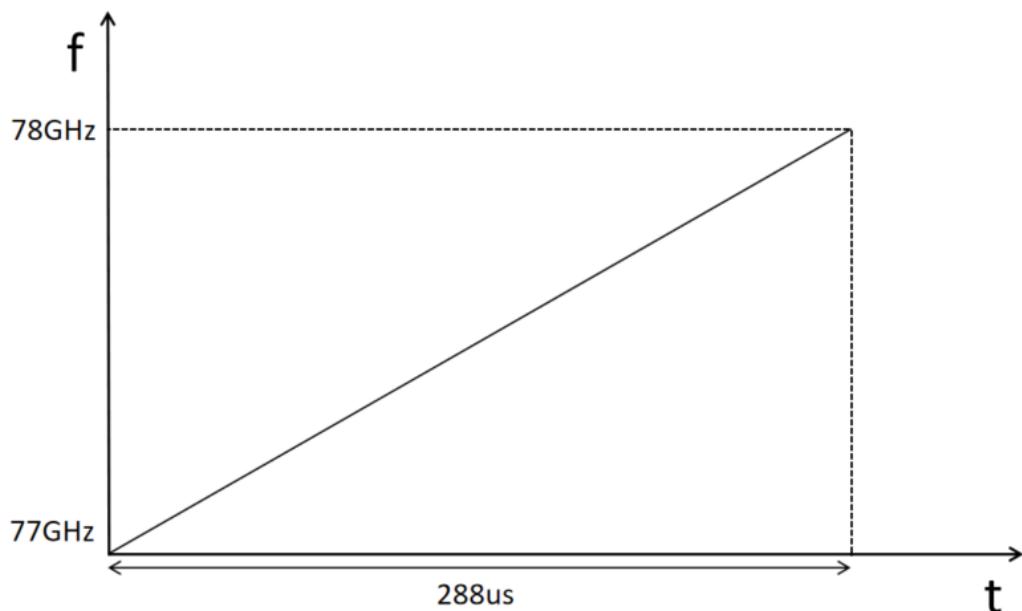


Figure 3.2: FMCW signal in terms of frequency-time

some of the radar signal is reflected by the target and back propagates to the radar, and captured by the Rx antenna. The received signal can be mixed with Tx signal or directly used to be analyzed. The mixed signal of Tx antenna and Rx antenna is called Intermediate signal, can be shortly written as IF signal.

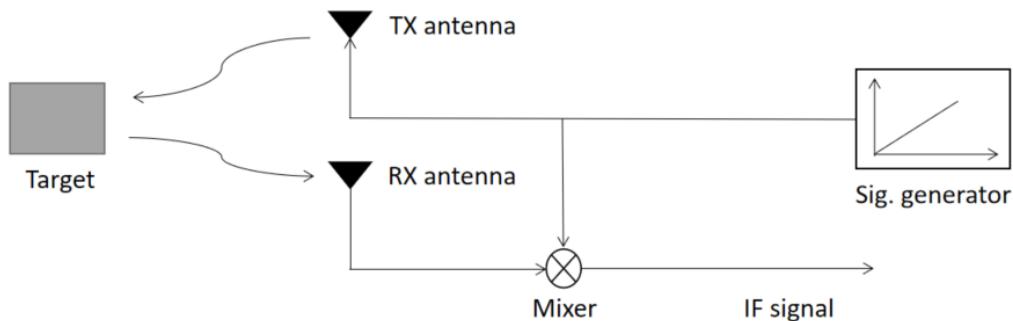


Figure 3.3: radar signal block diagram

3.2 Radar signal description

In mathematical aspect, from Tx antenna transmitted Radar signal can be describe as a cosine wave as in (3.1), where A, f_0, k, ϕ_0 are amplitude, starting frequency, increasing slope of frequency, starting phase of radar signal respectively. In this article, they can be 1, 77GHz, 1GHz/288μs, and 0°.

$$y = A(t)\cos(2\pi f_0 t + \pi kt^2 + \phi_0) \quad (3.1)$$

After the signal transmitted by Tx antenna, it propagates in the space, if there are targets on the way signal spreading, signal wave can be reflected by the surface of targets and back to the radar, then the Rx antenna can capture the back propagating signal. We assume that $M \times N$ Rx rectangular antenna array sensors with element spacing $d = \lambda/2$ (λ is the wavelength of radar signal waves) receive signal from K targets with azimuth angles $\alpha_1, \alpha_2, \dots, \alpha_K$, elevation angles $\beta_1, \beta_2, \dots, \beta_K$. Radar sensors are located at origin of the coordinate system, and the azimuth, elevation angles are described relative to the radar position. With above assumption, the received signal wave can be described as in (3.2).

$$x = \sum_{K=1}^K \mathbf{a}(\alpha_k, \beta_k) s_k + \mathbf{n} \quad (3.2)$$

In above equation, s_k represents signal received by Rx antenna elements, \mathbf{n} is circular complex white Gaussian noise, \mathbf{a} is steering vector for the k th target. Steering vector for k th target can be further described as in (3.3), considering the $M \times N$ antenna sensors, it is $M \times N \times 1$ matrix. Steering vector \mathbf{a} represents the transformation relationship of signal location and coordinate origin, it depends on the azimuth angle and elevation angle of targets.

$$\mathbf{a}(\alpha_k, \beta_k) = \begin{bmatrix} 1 \\ e^{j(2\pi f/c)(dm \cos \beta_k \sin \alpha_k + dn \sin \beta_k)} \\ \vdots \\ e^{j(2\pi f/c)(d(M-1) \cos \beta_k \sin \alpha_k + d(N-1) \sin \beta_k)} \end{bmatrix} \quad (3.3)$$

Using Rx antenna received radar signal, the target range, velocity and direction can be detected. Target range detection depends on the total propagation time of radar signal, when the radar signal transmitted time point and received time point is clear, the target range can be calculated. As for target velocity, multiple radar signal chirps are required, since the interval of chirps are very small (in most cases are tens of micro seconds), the moving of target influence the radar wave phase, in terms of propagation time it is not obvious. Therefore the received radar signal phase is analyzed to detect the targets velocity. As in 3.3 described, target direction influence the total shape of signal wave, then multiple Rx antenna sensors are required to analyze the direction information. In the following sections, the three detection methods are presented.

3.3 Target range detection

In terms of graphic approach, the transmitted and received radar signal can be described as in figure 3.4. It is assumed that there are 3 different range targets in the propagation direction of transmitted radar signal. The influence of target velocity are very small to the frequency of signal, then it is ignored in the figure 3.4. Since the targets are at different range, the total propagation time of radar signal of different targets differs from each other, then the signal receiving time of each targets depends on the range of targets. As the result of the different receiving time, each Rx signal of different targets has unit frequency difference with the Tx signal.

As introduced in the figure 3.3, the received Rx signal is mixed with the transmitted signal in the mixer, mixed signal is called intermediate signal. the IF signal which derived from figure

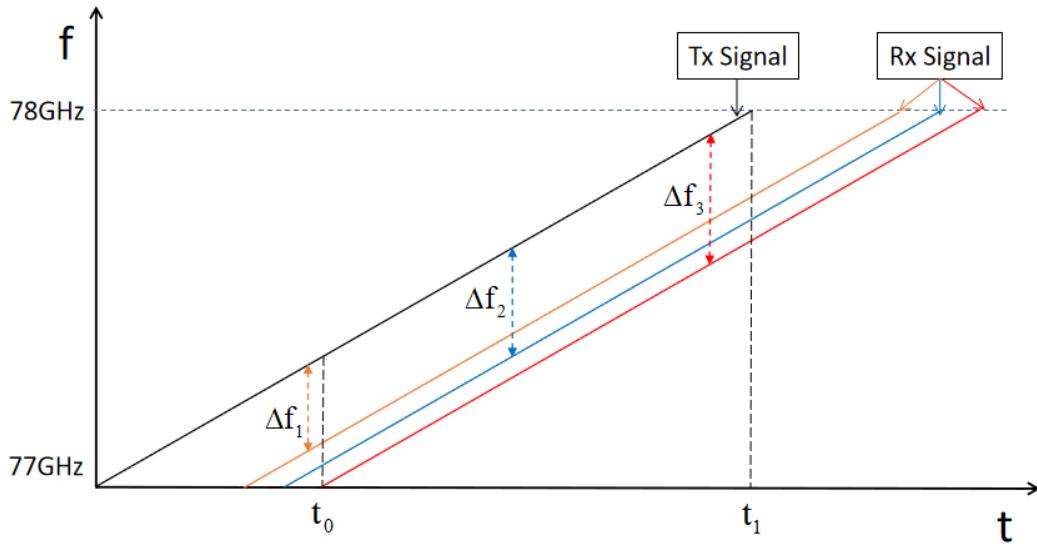


Figure 3.4: Transmitted Tx signal and received Rx signal from 3 targets in different range

3.4 is shown in figure 3.5. After a Fast Fourier transformation which called fast-time FFT, the mixed IF signal can be divided into several single IF signals with its own unit frequency as in figure 3.5.

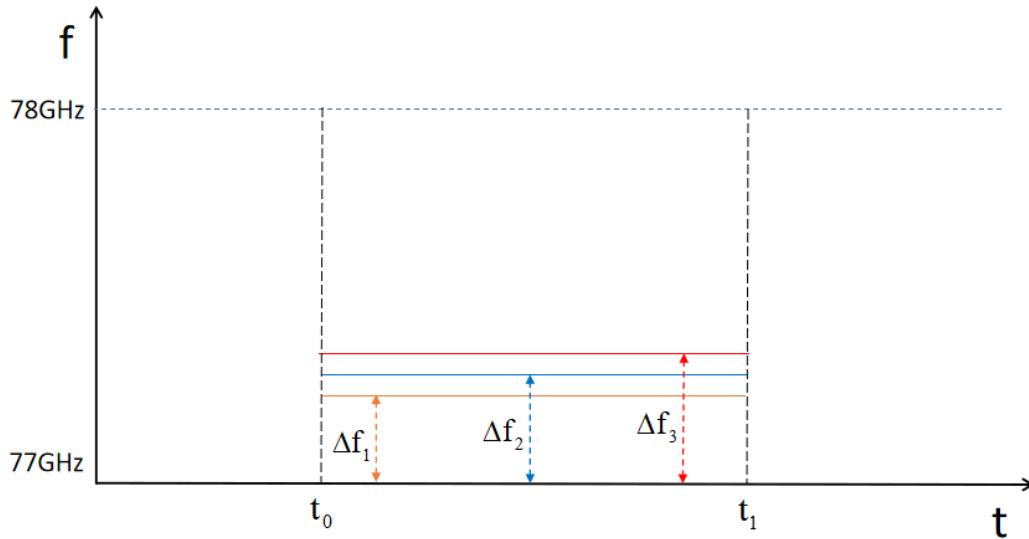


Figure 3.5: IF signal for 3 targets in different range

Using the IF signal frequency, the target range can be calculated. For IF signal with frequency Δf_3 , it is easily prove that

$$\frac{78\text{GHz} - 77\text{GHz}}{t_1} = \frac{\Delta f_3}{t_0} \quad (3.4)$$

and

$$r = \frac{ct_0}{2} \quad (3.5)$$

, where r is target range for If signal frequency Δf_3 , c is propagation speed of radar signal. Then the range-If signal frequency equation can be written as:

$$r = \frac{c\Delta f}{2S} \quad (3.6)$$

In equation 3.6, c is wave speed in the space, Δf is If signal frequency, S is chirp slope. In such a way, the target range can be detected.

From the theoretical aspect it is clear that the frequency resolution of FFT has direct influence on range resolution. Since the frequency resolution in Discrete Fourier Transform is

$$f_0 = \frac{f_s}{N} = \frac{1}{Nt_s} = \frac{1}{T} \quad (3.7)$$

, where f_0 is FFT frequency resolution, f_s is sampling frequency, N is number of samples, T is sampling time. The assumed application situation of radar in this article is short range target detection, therefore the sampling time $t_1 - t_0$ of If signal in figure 3.5 can be seen as chirp duration t_1 . After combining 3.6 and 3.7, the frequency resolution of radar signal FFT can be described as:

$$\frac{1}{t_1} = \frac{S2r_{res}}{c} \quad (3.8)$$

, where r_{res} is range resolution of radar. Considering $B = 78GHz - 77GHz = St_1$, the range resolution of radar target can finally be shown as:

$$r_{res} = \frac{c}{2B} \quad (3.9)$$

From equation 3.9 can be derived that the radar target range resolution is inversely proportional to bandwidth of radar signal.

3.4 Target velocity detection

To measure the velocity of targets, multi chirps with fixed launch interval T_c should transmitted. The figure 3.6 shows 2 continuous chirp signal with interval duration T_c . If targets has velocity

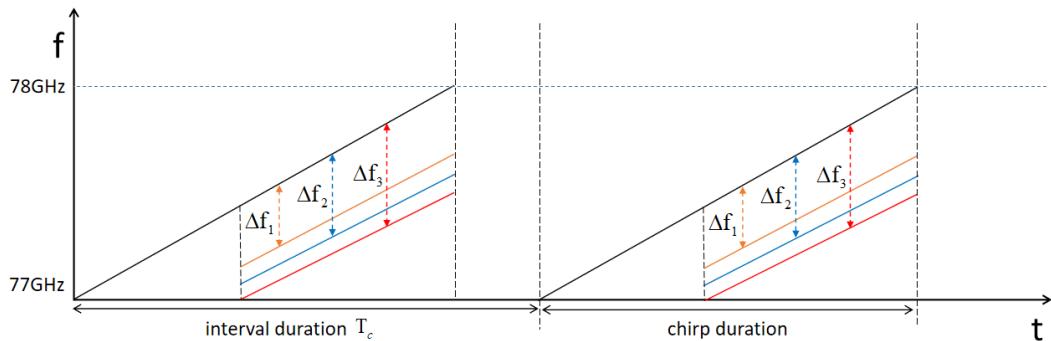


Figure 3.6: 2 continuous chirp signal and Rx signal from 3 targets in different range

relative to the radar, since there are chirp interval durations, the range from targets to radar are

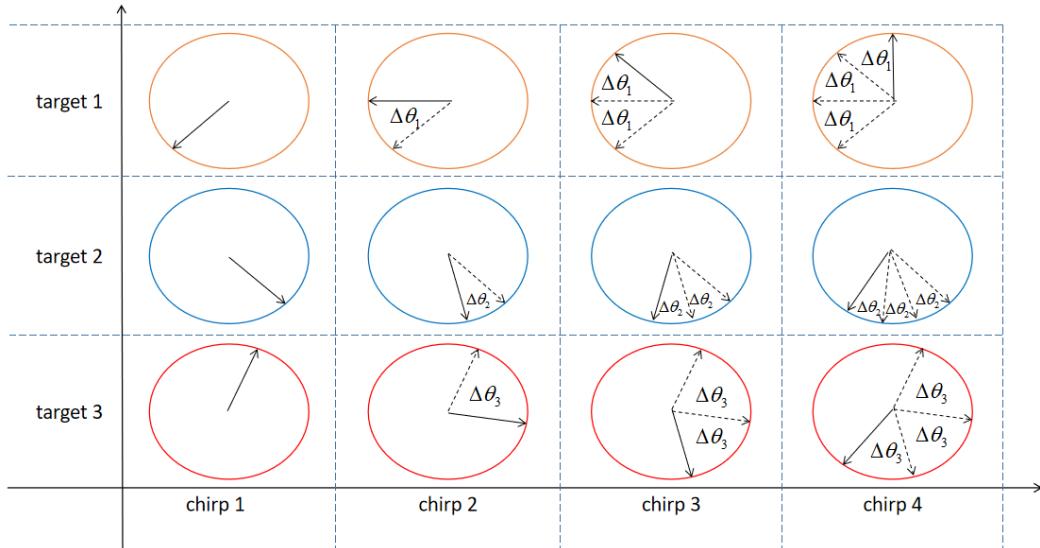


Figure 3.7: Phase changing of 3 target signals over chirps

different in every time point when radar signal reach the targets, that results to the different propagation range of different chirp signals. But the moved range of target in chirp duration is much smaller than the total propagation distance of signal, the target moving influences mainly on the phase of reflected signal.

As to Doppler effect, since the speed of radar signal can be seen as light speed, and target velocity is much smaller than light speed in driving environment, Doppler effect on radar signal can be ignored.

Then the target velocity detecting can be easily measured. As the first step, the data should be processed as in range detection, using FFT the IF signal of same chirp can be separated into several signal waves with its unit certain frequency, each frequency corresponds to a target with certain range. Then the range of targets can be calculated.

On the other hand, each separated signal wave with certain frequency has its own phase. Since the radar signal chirp duration and interval duration are tens to hundreds microseconds, it is assumed that the targets' velocities is stable during the measurement. The result of FFT processing to the IF signal in terms of phase can be described like in the figure 3.7, where solid line represent the current chirp signal phase when dashed line represents the phase of past chirp signal. It shows the phase changing of different target signals over chips. The velocity information of targets can be derived from the phase changing of signals.

The phase difference is caused by the additional propagation distance, which equal to the 2 times of target moved range during the interval duration as described in following equation:

$$\Delta r = 2vT_c \quad (3.10)$$

, where Δr is the additional propagation distance, v is target velocity, T_c is chirp interval duration.

Using following equation, the relationship of phase shift and propagation distance is presented:

$$\Delta r = \lambda \frac{\Delta\theta}{2\pi} \quad (3.11)$$

, where λ is wavelength of signal, $\Delta\theta$ is phase difference between chips.

Then the target velocity calculation can be derived as in equation 3.12.

$$v = \frac{\lambda \Delta\theta}{4\pi T_c} \quad (3.12)$$

There are more two things that need attention in target velocity detection. One is phase shift $|\Delta\theta|$ shouldn't be larger than π , or there will be confuse when calculating target velocity, then combined with equation 3.12 can be derived that:

$$v_{max} = \frac{\lambda}{4T_c} \quad (3.13)$$

, where v_{max} is maximum detectable velocity of radar target.

Another one is target velocity resolution, from the equation 3.12 can be drawn that the radar target velocity resolution is inversely proportional to the phase shift resolution between chirps. Since phase shift can be distinguished when $\Delta\theta \geq 2\pi/N$, where N is number of chirps, then combined with equation 3.12 can the radar target velocity resolution be described as in equation 3.14.

$$v_{res} = \frac{\lambda}{2NT_c} \quad (3.14)$$

From equation 3.14 can be derived that the radar target velocity resolution is inversely proportional to number of chirps and chirp interval duration of radar signal.

3.5 Range-Doppler map

In traditional radar signal process approach, the range and velocity detection result is shown in a range-doppler map. As described in above sections, range detection can be implemented by frequency result of FFT of one chirp signal, and velocity detection is implemented by phase shift result of FFT of multi chirp signal.

Range and velocity detection of targets in with different range and velocity is possible using only one Rx antenna sensor, radar signal samples of one Rx antenna sensor can be written as matrix form as in figure 3.8, which contains M chirp signal with N samples in each chirp signal, in column direction is called fast-time-sampling, in row direction is called slow-time-sampling.

After sampling the radar signal, the signal matrix can be processed using FFT, then the original received signal is divided into several signals with certain frequency and phase as shown in figure 3.9.

It is assumed in figure 3.9 that there are k targets in different range, the FFT in matrix column direction is called fast-time-FFT, which focuses on signal frequency, then the received signal is divided into k different signals with frequency f_k . With frequency f_k it is possible to calculate target range. The FFT in matrix row direction is called slow-time-FFT, which focuses on phase shift of signal, using the phase shift a_k in every chirp duration is the velocity of target with signal frequency f_k calculated. Since each signal frequency and phase shift corresponds to range and velocity respectively, the signal matrix after FFT can be directly mapped into range-velocity coordinate then the range-velocity relationship of targets can be described as in figure 3.10, which shows 3 targets range-velocity relationship as an example.

Radar Signal Matrix	Slow time sampling				
	Chirp 1	Chirp 2	Chirp 3	...	Chirp M
Fast time sampling	Sample 1-1	Sample 2-1	Sample 3-1	...	Sample M-1
	Sample 1-2	Sample 2-2	Sample 3-2	...	Sample M-2
	Sample 1-3	Sample 2-3	Sample 3-3	...	Sample M-3

	Sample 1-N	Sample 2-N	Sample 3-N	...	Sample M-N

Figure 3.8: Radar signal in matrix form

Radar Signal After FFT	Slow time FFT(phase focused)				
	Chirp 1	Chirp 2	Chirp 3	...	Chirp M
Fast time FFT (frequency focused)	Signal 1-1 (Frequency:f1 Phase:p1)	Signal 2-1 (Frequency:f1 Phase:p1+a1)	Signal 3-1 (Frequency:f1 Phase:p1+2*a1)	...	Signal m-1 (Frequency:f1 Phase:p1+(m-1)a1)

	Signal 1-k (Frequency:fk Phase:pk)	Signal 2-k (Frequency:fk Phase:pk+ak)	Signal 3-k (Frequency:fk Phase:pk+2*ak)	...	Signal m-k (Frequency:fk Phase:pk+(m-1)*ak)

Figure 3.9: Radar signal after FFT

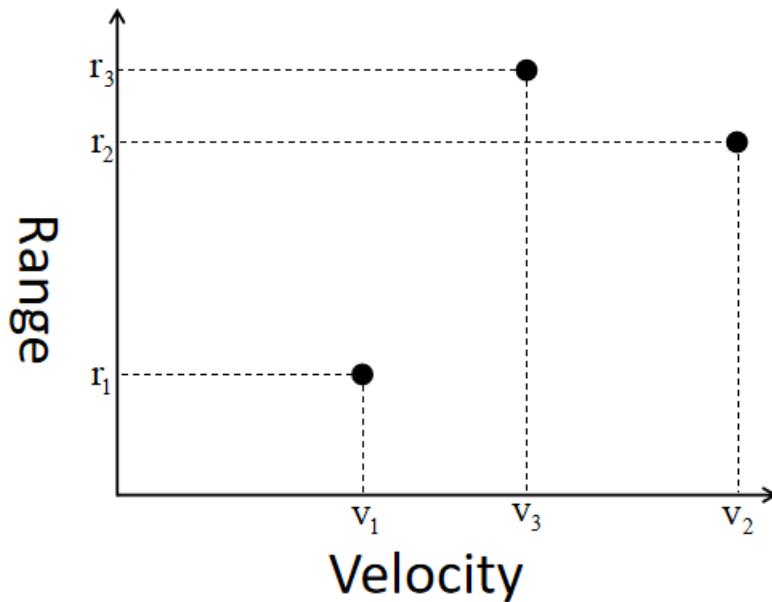


Figure 3.10: range doppler map of 3 targets

In practical radar application, there are many factors that influence the radar signal quality. Also in engineering aspect some measurement error are not avoidable. For example the distortion of radar signal which caused in sampling process has bad influence in target detection, especially when the target number is big. Then though using 1 Rx antenna sensor is theoretically enough, in practical approach are more Rx antennas needed.

3.6 Direction of arrival estimation

3.6.1 Phase-comparison monopulse

Using FMCW radar, not only the range and velocity but also the direction of targets can be detected. In this section is basis of DOA estimation introduced. Firstly is a method called phase-comparison monopulse[5].

To implement DOA estimation, multiple Rx antenna sensors are needed. The DOA estimation principle is similar with that of velocity estimation. Firstly, one dimensional DOA estimation of one target is considered, as shown in figure 3.11, by different Rx antenna element received signals propagate different ranges. The differences are caused by the antenna element spacing and DOA angle of target signals, where Δr is target distance difference between different Rx antenna elements.

Because Rx antenna element spacing is normally half a wavelength with several millimeters length, its influence on DOA angles can be ignored when considering targets on at least few meters range, then it is assumed that target angles are same for different Rx antenna elements. Under such a assumption, the figure 3.12 shows propagation range differences Δr of single target signals for different Rx antenna elements with element spacing d , where target DOA angle is α . Considering element spacing d is smaller than signal wavelength, Δr is also smaller than signal wavelength, then the received IF signal of one target has no frequency difference

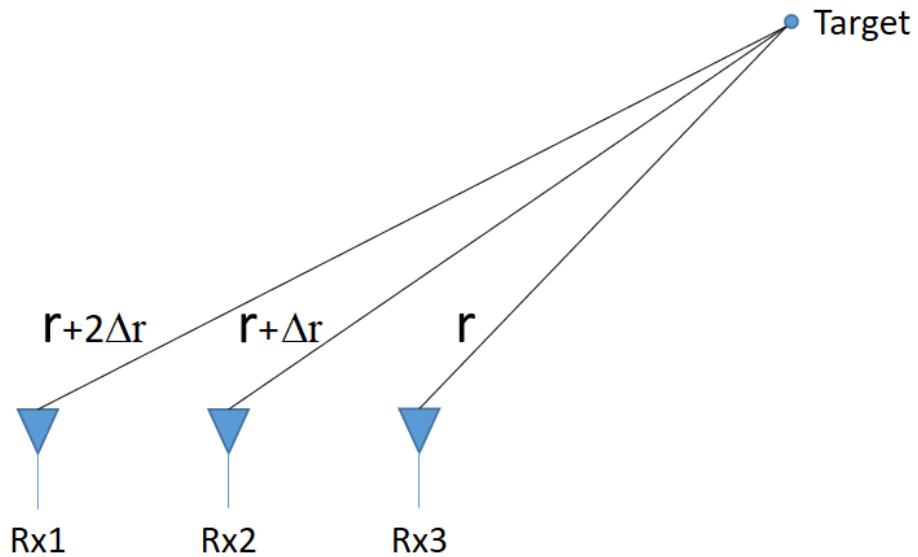


Figure 3.11: single target signals arrive on different Rx antenna elements

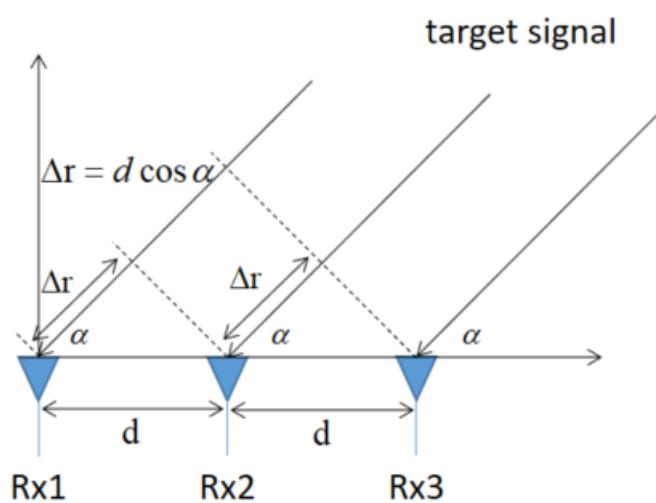


Figure 3.12: propagation range differences Δr of single target signals with DOA angle α for different Rx antenna elements with element spacing d

Phase shift in azimuth domain					
Phase shift in elevation domain	Frequency:fk Phase:pk	Frequency:fk Phase:pk+bk	Frequency:fk Phase:pk+2*bk	...	Frequency:fk Phase:pk+(n-1)*bk
	Frequency:fk Phase:pk+ak	Frequency:fk Phase:pk+ak+bk	Frequency:fk Phase:pk+ak+2*bk	...	Frequency:fk Phase:pk+ak+(n-1)*bk

	Frequency:fk Phase:pk+(m-1)*ak	Frequency:fk Phase:pk+(m-1)*ak+bk	Frequency:fk Phase:pk+(m-1)*ak+2*bk	...	Frequency:fk Phase:pk+(m-1)*ak+(n-1)*bk

Figure 3.13: signal phase shift of K th target in azimuth and elevation domain, where  and  correspond elevation and azimuth angle respectively

between different Rx antenna elements, the range difference is reflected by the phase differences of received IF signals.

Now consider multi targets 2D DOA estimation using one chirp signal from $M \times N$ rectangular Rx antenna element array. As the first step, the signal of each Rx element is analyzed by the FFT and divided into several frequencies which correspond to different target in different range. Then the signal phase shift can be described in a $M \times N \times K$ matrix form, where $M \times N$ corresponds Rx array, K corresponds number of target. $M \times N$ signals of K th target can be written as in table 3.13, where fk is frequency of the target signal, ak is phase shift in elevation domain, bk is phase shift in azimuth domain.

The difference of signal propagated range for difference Rx antenna element can be written in 2 ways, first one is description with phase shift as in 3.11, second one is description as in figure 3.12 that $\Delta r = d \cos \alpha$, after combining the two description the DOA angle of target is like in 3.15, where α is DOA angle of target, $\Delta\theta$ is phase shift of signal between two adjacent Rx antenna element. Then phase shift $\Delta\theta$ in azimuth domain of signal reflects azimuth target angle, that in elevation domain of signal reflects elevation target angle.

$$\alpha = \arccos \frac{\lambda \Delta\theta}{2\pi d} \quad (3.15)$$

Under ideal circumstances this method works in the 2D DOA angle detection for targets in different distance, but in practical implementation the signal noise must be considered, the accuracy of the angle detection drops sharply when the signal-noise ratio is low. Therefore an alternative and more effective method is needed. Then an algorithm called MUSIC can be considered, since MUSIC algorithm can separate radar signal data into target signal and noise and concentrate on target signal.

3.6.2 MUSIC algorithm

MUSIC is abbreviation of multiple signal classification, which uses orthogonality of signal and noise subspaces to implement the DOA estimation.

Radar received signal equation 3.2 can be in matrix form written as

$$\mathbf{X} = \mathbf{AS} + \mathbf{n} \quad (3.16)$$

where \mathbf{A} is a $MN \times K$ steering matrix

$$\mathbf{A} = [a(\alpha_1, \beta_1), \dots, a(\alpha_k, \beta_k)] \quad (3.17)$$

and \mathbf{S} is signal vector

$$\mathbf{S} = [s_1, \dots, s_k]^T \quad (3.18)$$

Then the signal matrix should be used to estimate the signal covariance matrix, which is defined as

$$\mathbf{R}_x = E\{\mathbf{XX}^H\} \quad (3.19)$$

where \mathbf{X}^H is conjugate transpose of complex matrix \mathbf{X} , E is expectation. Combining equation 3.16, 3.19 can be written as

$$\mathbf{R}_x = \mathbf{AR}_s \mathbf{A}^H + \mathbf{R}_n \quad (3.20)$$

where

$$\mathbf{R}_s = E\{\mathbf{SS}^H\} \quad (3.21)$$

is source signal covariance matrix,

$$\mathbf{R}_n = \sigma^2 \mathbf{I} \quad (3.22)$$

is noise covariance matrix with noise power σ^2 , \mathbf{I} is $M \times N$ identity matrix.

Here assume that signal and noise are uncorrelated, noise is zero-mean Gaussian white noise, therefore the received signal matrix has full rank. Considering the rank of \mathbf{R}_x is $M \times N$, there are $M \times N$ eigenvalues, where the larger eigenvalues correspond signal, while the smaller eigenvalues belong to noise.

The explanation to the MUSIC algorithm can be found in many references, the following is one of the explanation.

Let the noise eigenvector matrix be as following:

$$\mathbf{E} = [v_1, \dots, v_{MN-K}] \quad (3.23)$$

then

$$\mathbf{R}_x \mathbf{E} = \sigma^2 \mathbf{E} \quad (3.24)$$

place 3.20 into 3.24, acquire that

$$\sigma^2 \mathbf{E} = (\mathbf{AR}_s \mathbf{A}^H + \sigma^2 \mathbf{I}) \mathbf{E} \quad (3.25)$$

then obtained following equation:

$$\mathbf{AR}_s \mathbf{A}^H \mathbf{E} = 0 \quad (3.26)$$

because \mathbf{AA}^H has full rank, $(\mathbf{AA}^H)^{-1}$ exists, $(\mathbf{R}_s)^{-1}$ also exists, then multiply $\mathbf{R}_s^{-1}(\mathbf{AA}^H)^{-1}\mathbf{A}^H$ to both sides of equation 3.26, get

$$\mathbf{A}^H \mathbf{E} = 0 \quad (3.27)$$

Above equation proved the orthogonality of the signal and noise subspaces, which means that every eigenvector of noise eigenvalue is perpendicular to each column of matrix \mathbf{A} . Because

column vector of matrix \mathbf{A} contains 2D DOA information of target signal, 2D DOA estimation can be implemented by searching target angle which fit the equation 3.27. Then the spatial spectrum $P(\alpha, \beta)$ can be

$$P(\alpha, \beta) = \frac{1}{\mathbf{a}^H(\alpha, \beta) \mathbf{E} \mathbf{E}^H \mathbf{a}(\alpha, \beta)} = \frac{1}{\|\mathbf{E}^H \mathbf{a}(\alpha, \beta)\|^2} \quad (3.28)$$

The 2D DOA angles can be searched in above equation 3.28, in practical circumstances 2D DOA estimation is implemented by finding the peak of spatial spectrum $P(\alpha, \beta)$.

MUSIC algorithm is a efficient DOA estimation method with high resolution, which is widely used in many radar system. To verify the performance of the neural network based 2D DOA estimation model, the DOA estimation result of the neural network based model is compared with that of MUSIC in the final chapter.

3.7 Artificial neural networks

Artificial neural network which is called ANN develops rapidly and widely in recent years, especially in computer vision and natural language processing. In many other fields it is also tried to make an approach using ANN instead of traditional method and acquired notable results. The ANN architect is inspired from structure of human brain, human brain consists of neurons or nerve cells which transmit and process the information.

3.7.1 Perceptron

Multilayer perceptron is a basic type of ANN, and perceptron is basic unit of multilayer perceptron which looks like in figure 3.14, whose input and output relationship can be described with equation 3.29, where x_i, w_i, b, f, n are input data, weight, bias, activation function, number of input data respectively. The equation also can be presented in form of matrix as in 3.30, where W and X are weights and input data in form of matrix.

$$y = f\left(\sum_{i=1}^n x_i w_i + b\right) \quad (3.29)$$

$$y = f(W * X + b) \quad (3.30)$$

3.7.2 Activation function

One function of activation function is to make the artificial neuron network be able to fit nonlinear function. Artificial neuron networks can be seen as a mapping function between input data and output data, the goal of training artificial neuron network is to find out the suitable parameters of network, so that let the mapping function of network can fit the real mapping function between data and it contained information. In most cases the mapping function is nonlinear, that means artificial neuron network has to be able to fit nonlinear functions. As can be seen in equation 3.29, without the activation function, mapping function between input and output is linear, only after adding the activation function, perceptron has the ability to fit the nonlinear function. Regarding the perceptron is the linearly linked only basic unit of multilayer perception, it can be say that activation functions bring nonlinear fitting ability to ANN.

Figure 3.15 shows four kinds of common used nonlinear activation functions.

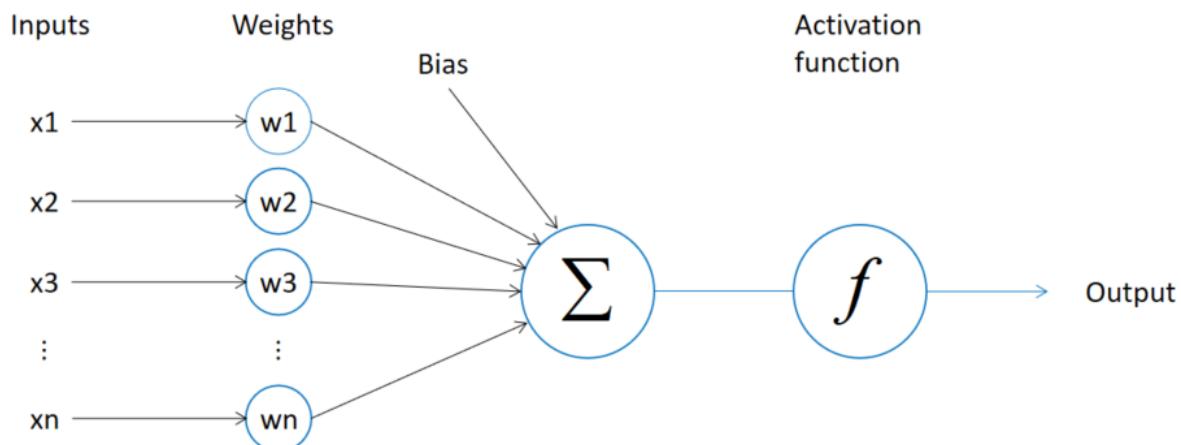
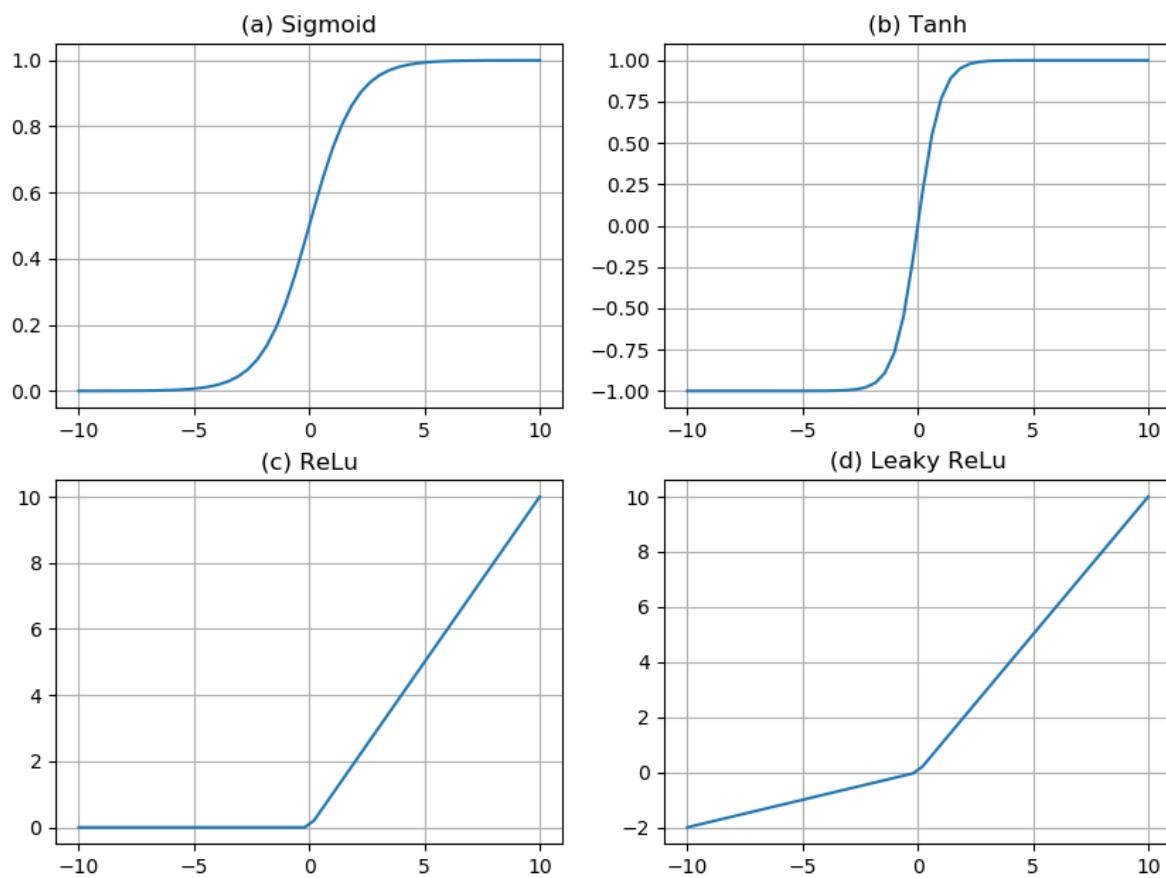


Figure 3.14: structure of an perceptron

Figure 3.15: activation functions with input value range $[-10,10]$

sigmoid First one is sigmoid activation function, sometimes also called logistic function, which can be written by

$$y = \frac{1}{1 + e^{-x}} \quad (3.31)$$

Sigmoid activation function is often used in ANN output. Because its output range is [0,1], sigmoid is used to predict the probability based output and also widely applied in binary classification problems. Besides, it has simple derivative which brings advantage to network learning.

Though sigmoid is still popular in present, but its disadvantage is also obvious. First of all, it needs large computation resource since has e^x in the function. Secondly the function saturates at the both end of function curve, then the gradient at the regions are almost zero, then the network can't fix its weight which means network can't learn anymore. Another one is the output value of sigmoid is all positive, this causes reduced neural network learning efficiency.

tanh Another one activation function is hyperbolic tangent function, which is known as tanh function. It has a similar curve as sigmoid but range is [-1,1], which is given by

$$y = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.32)$$

Tanh is much popular than sigmoid though has similar function curve as sigmoid, since its output range has minus part, which brings better training efficiency than sigmoid. That means tanh not only has all the advantages of sigmoid, but also overcame the disadvantage of sigmoid.

However there are still problems of sigmoid, big computation pressure, gradient vanishing are still the problem to this activation function.

ReLU ReLU is abbreviation of rectified linear unit, which is most popular activation function in deep learning area. The working principle of ReLU is simple, set the values less than zero to zero, which is given by

$$y = \max(0, x) \quad (3.33)$$

The most obvious advantage of ReLU is that it can provide faster compute than other activation functions since there are no exponential and division in the ReLU function. Besides, ReLU provides better performance and generalization than sigmoid and tanh activation functions[6].

The disadvantage of ReLU is caused by the zero value output, since the gradient is zero in minus x-axis, the weights of perceptron don't update when network is learning.

Leaky ReLU The leaky ReLU provides a small negative slope to the minus x-axis to avoid zero value of gradient, so that the weights can be updated during the network learning. The leaky ReLU function is given by

$$y = \max(\alpha x, x), \alpha < 1 \quad (3.34)$$

Softmax The Softmax function is an activation function which is often used in output layer to compute probability. The function is defined by

$$y_i = \frac{e^{x_i}}{\sum_{k=1}^K e^{x_k}} \quad (3.35)$$

where y_i is output value, $i \in 1, \dots, K$ where K is dimension of output layer.

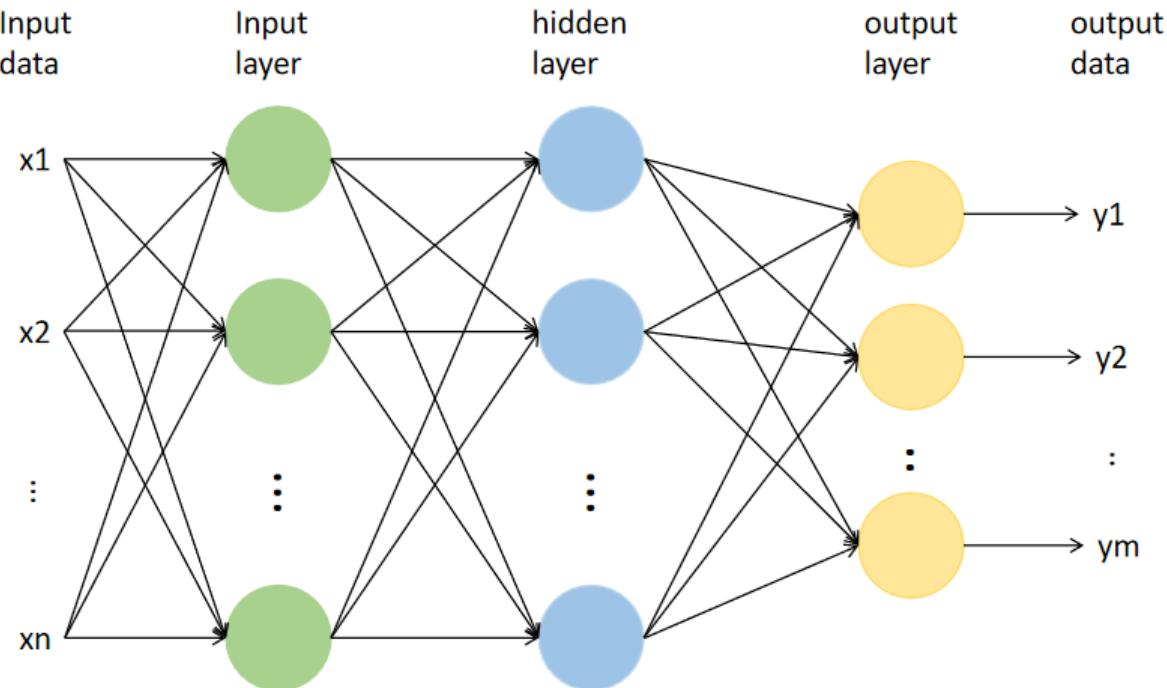


Figure 3.16: structure of a multilayer perceptron

3.7.3 Multilayer perceptron

Multilayer perceptron consist of one input layer, specific number of hidden layers and one output layer. Each layer is composed of one neuron or multi neurons. Input layer receives the input signal, output layer outputs processed data, which shows the final result of multilayer perceptron, hidden layers process data to find out the ideal output. Figure 3.16 shows the structure of multilayer perceptron with one hidden layer.

3.7.4 Back propagation

The weights and bias should be continuously updated to make the multilayer perceptron output to fit the desired value, this process is called training. The training consists of two steps, feed-forward and back propagation. In feedforward, the input data goes forward from input layer to output layer, then the multilayer perceptron get output data, which is shown in figure 3.16. Then the output data is compared with the desired value to determine the error value.

In back propagation, the contribution of every single weight and bias to the error is calculated, then the weights and bias are updated according to the calculation result. In terms of multilayer perceptron, the weight update equation is defined by

$$w_{ij}^l * = w_{ij}^l - \eta \frac{\partial E}{\partial w_{ij}^l} \quad (3.36)$$

where \$w^*\$ is new weight, \$w_{ij}^l *\$ is updated weight from \$l - 1\$th layer \$i\$th perceptron to \$l\$th layer \$j\$th perceptron when \$w_{ij}^l\$ is old value of that, \$\eta\$ is learning rate, \$E\$ is output error. When it is

assumed that I_j^l is input value of l th layer j th perceptron, O_j^l is output value of l th layer j th perceptron, the gradient $\frac{\partial E}{\partial w}$ can be according to the chain rule given by

$$\frac{\partial E}{\partial w_{ij}^l} = \frac{\partial E}{\partial O_j^l} \frac{\partial O_j^l}{\partial I_j^l} \frac{\partial I_j^l}{\partial w_{ij}^l} \quad (3.37)$$

when let

$$\delta_j^l = \frac{\partial E}{\partial O_j^l} \frac{\partial O_j^l}{\partial I_j^l} \quad (3.38)$$

it is clear that the gradient $\frac{\partial E}{\partial w_{ij}^l}$ is decided by output of last perceptron O_i^{l-1} and variable δ_j^l . According to the chain rule, $\frac{\partial E}{\partial O_j^l}$ can be rewritten as

$$\frac{\partial E}{\partial O_j^l} = \sum_{k=1}^{n_{l+1}} \left(\frac{\partial E}{\partial O_k^{l+1}} \frac{\partial O_k^{l+1}}{\partial I_k^{l+1}} \frac{\partial I_k^{l+1}}{\partial O_k^l} \right) \quad (3.39)$$

where n_{l+1} is number of perceptrons in layer $l + 1$, then considering

$$\frac{\partial I_k^{l+1}}{\partial O_k^l} = w_k^{l+1} \quad (3.40)$$

and

$$\delta_k^{l+1} = \frac{\partial E}{\partial O_k^{l+1}} \frac{\partial O_k^{l+1}}{\partial I_k^{l+1}} \quad (3.41)$$

Further, assume that

$$O_j^l = f(I_j^l) \quad (3.42)$$

where f is non-linear activation function, then

$$\frac{\partial O_j^l}{\partial I_j^l} = \frac{\partial f(I_j^l)}{\partial I_j^l} \quad (3.43)$$

which can be computed with activation function on each perceptron, then equation 3.38 can be redefined as

$$\delta_j^l = \sum_{k=1}^{n_{l+1}} (\delta_k^{l+1} w_k^{l+1}) \frac{\partial f(I_j^l)}{\partial I_j^l} \quad (3.44)$$

and it is also clear that

$$\frac{\partial I_j^l}{\partial w_{ij}^l} = O_i^{l-1} \quad (3.45)$$

then the weight update equation 3.36 can be rewritten as

$$w_{ij}^l * = w_{ij}^l - \eta \delta_j^l O_i^{l-1} \quad (3.46)$$

especially considering the output layer has no next layer, its δ can be computed by directly using 3.37.

Using chain rule the bias update equation can be derived as

$$b_j^{l*} = b_j^l - \eta \sum_{k=1}^{n_l} \delta_k^l \quad (3.47)$$

where b_j^{l*} is updated bias, b_j^l is old bias, η is learning rate, n_l is number of perceptrons in l th layer.

3.7.5 Artificial neural network training

Training a neural network needs training data with desired output value. At the beginning of network training, the weights and bias are randomly initialized. Then all the sample data is inputted into the network one by one, and feedforward and backpropagation are executed, in each back propagation weights and bias are updated, when the output error is small enough, the train is finish. After the training, some new data should be provided to evaluate the performance of network model, since there are potential risk of overfitting [7], if the model performance by evaluation data is as good as that of training data, the network model can be used in practical prediction.

The desired output of corresponding data is called data label.

3.7.6 Convolutional neural network

Convolutional neural network with abbreviation CNN is the most popular type of ANN in recent years which mainly applied in visual analyzing.

2 dimensional convolutional layer In this article the main attention is focused on 2 dimensional convolutional layer, since it is the most widely used convolutional layer in CNN model, which is suitable to process 2D or 3D data, also is the only convolutional layer type that used in this article, in the following paragraph 2 dimension convolutional layer is shortly called convolutional layer.

As the first step of feedforward, the input data is processed by convolutional layer, where the input data is convolutional calculated by filter. Assume that matrix A has dimensions (M_a, N_a) and matrix B has dimensions (M_b, N_b) , then discrete convolution result C of matrix A and B is given by

$$C(i,j) = \sum_{m=0}^{M_a-1} \sum_{n=0}^{N_a-1} A(m,n)B(i-m, j-n) \quad (3.48)$$

where $0 \leq i \leq M_a + M_b - 1$ and $0 \leq j \leq N_a + N_b - 1$, with which the matrix C is called full convolution of matrix A and matrix B or full padding in CNN. The graphical explain to the above equation 3.48 is:

- 0) Assume that $A(0,0)$ is located at $[0,0]$.
- 1) Flip $B(m,n)$ horizontally and vertically to get new matrix $B(-m, -n)$.
- 2) Move $B(-m, -n)$ to the location, where $B(0,0)$ is at coordinate $[i,j]$.

- 3) Compute $\sum_{m=0}^{M_a-1} \sum_{n=0}^{N_a-1} A[m,n]B[m,n]$, where $A[m,n]B[m,n]$ is value of matrix on corresponding coordinate.

In convolutional layer in CNN, A is the input data to the layer and $B(-m, -n)$ is so called filter of the layer, therefore in CNN the convolutional layer is in fact cross-correlation layer. The size and number of filter can be flexible chosen so that to control the shape and number of layer output matrix C . By choosing value range of i and j in 3.48, the layer output shape also can be controlled. Especially, it is called same padding when $M_b \leq i \leq M_a - M_b$ and $N_b \leq j \leq N_a - N_b$, whose layer output data shape is same as input data shape, if $\frac{M_b}{2} \leq i \leq M_a - \frac{M_b}{2}$ and $\frac{N_b}{2} \leq j \leq N_a - \frac{N_b}{2}$, the padding is called valid. Sampling interval of i and j is in CNN called stride, which also is a important parameter in CNN.

Besides, filter should have same dimension as input data, if input data is 3 dimensional, the 3rd dimension of data is called channel, then the filter is also 3 dimensional, when computing the convolution result, convolutional compute is done in first two dimensions then the result of each channel is added, finally output a two dimensional result.

The figure 3.17 shows a 2 dimensional convolution example of input data and single filter with 4×4 and 3×3 dimensions respectively, where the output can be controlled in 3 kinds of shape by choosing different padding methods and stride in both direction is 1. After adding bias and activation function to convolution result, the computed data can be the output of convolutional layer.

pooling layer Pooling layer is commonly follows after convolutional layer, which plays the role of downsampling to the input data so that can avoid overfitting of networks. In this layer, the input data is divided into certain number of rectangle areas(areas can partly overlapped) according to the pooling filter size and stride size, then data in each area are processed according to the pooling strategy. Assume that input data size is (W_i, H_i) , filter size of pooling layer is (F_w, F_h) , stride is s , then the output data size of pooling layer is $(\frac{W_i-F_w}{s} + 1, \frac{H_i-F_h}{s} + 1)$.

There are usually two types of pooling strategy max pooling and average pooling, which output max value in filter area and average value in filter area respectively, figure 3.18 shows examples of max pooling and average pooling, where max pooling filter size is $(2,2)$, stride is 1, average pooling filter size is $(2,2)$, stride is 2.

back propagation in CNN All the backpropagation in ANN is derived by chain rule, in following the backpropagation algorithm of CNN is explained. Assume there is a convolutional layer where input data is A with dimensions $(3,3)$, filter is W with dimensions $(2,2)$. Assume the cross-correlation(as explained above, the convolutional layer in CNN calculate with cross-correlation instead of convolution in fact, since it doesn't influence the parameter updating and network result) is executed by valid padding with 1 stride, then cross-correlation result is B with dimensions $(2,2)$. Let the output value be $C(2,2)$ with equation 3.49, where f is activation function, b is bias on corresponding value.

$$C(i,j) = f(B(i,j) + b(i,j)) \quad (3.49)$$

And the cross-correlation process $A * W = B$ is shown in figure 3.19, where the gradients of

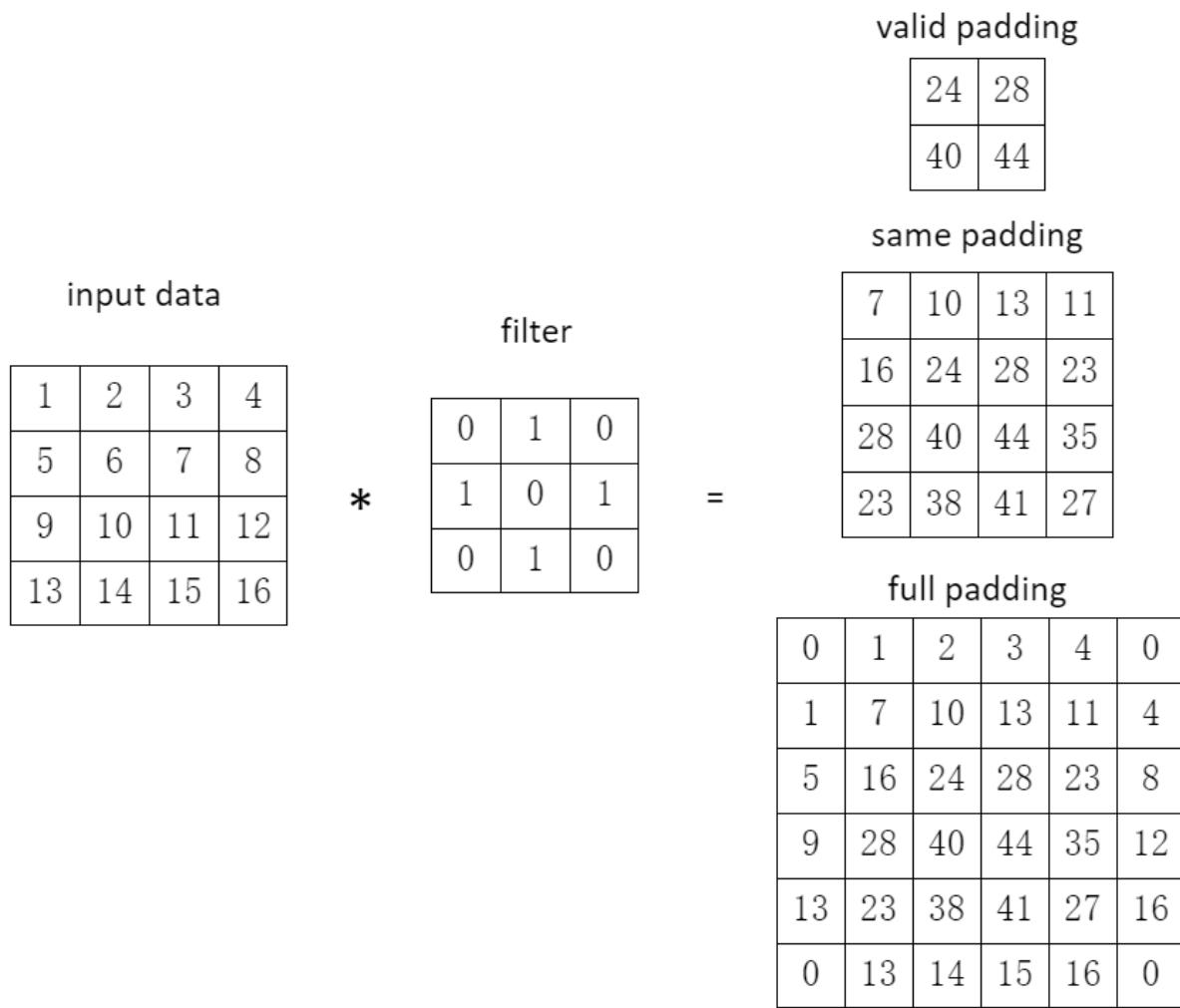


Figure 3.17: an example of 2 dimensional convolution with 3 kinds of padding

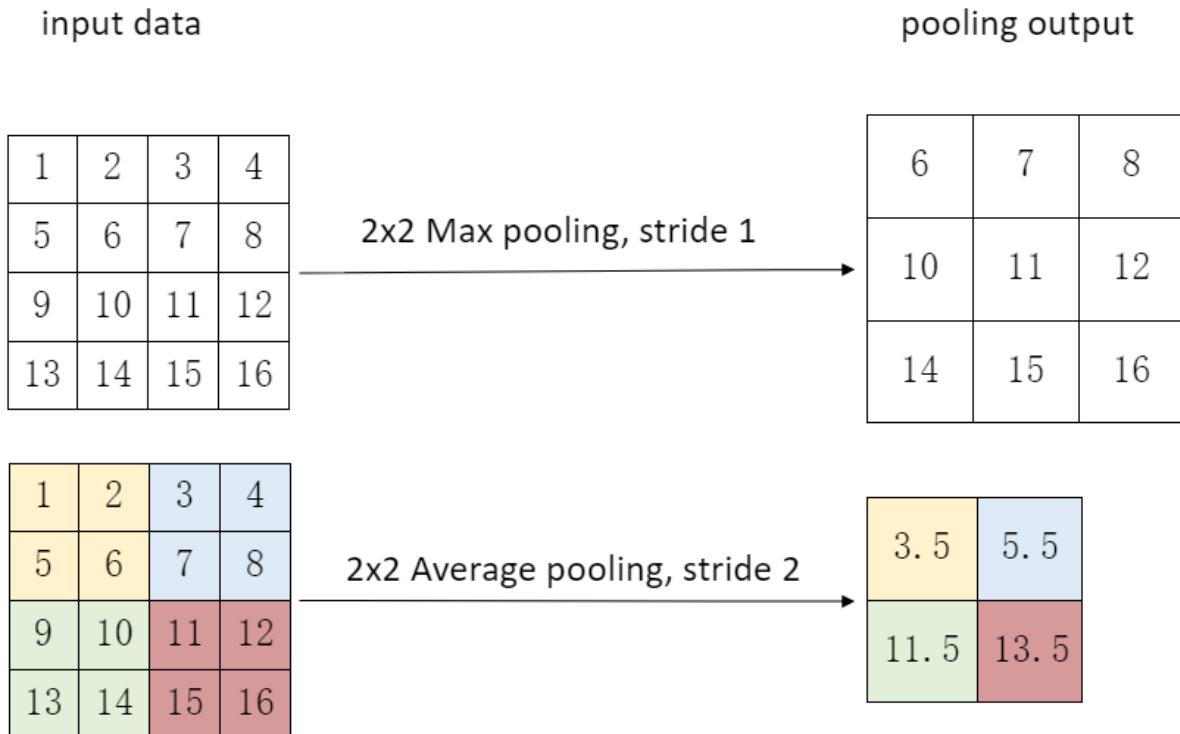


Figure 3.18: an example of 2 dimensional convolution with 3 kinds of padding

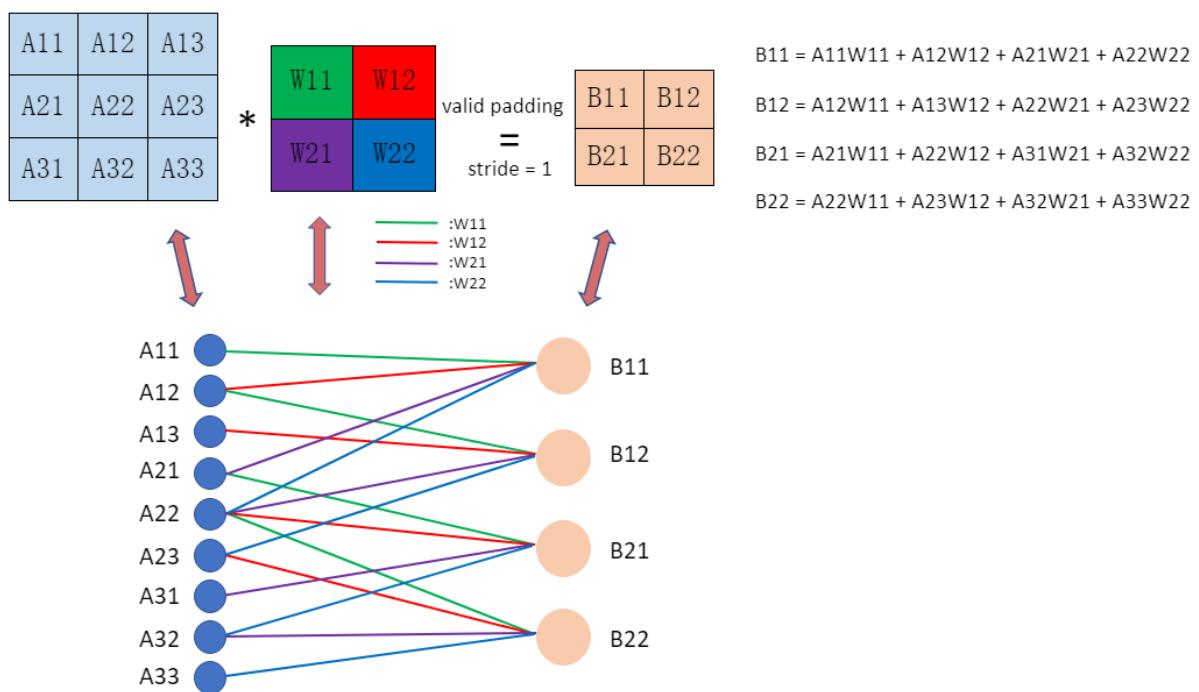


Figure 3.19: an example of cross-correlation process with intuitive graphical explanation

error E on each weight of W are:

$$\begin{aligned}\frac{\partial E}{\partial W_{11}} &= \frac{\partial E}{\partial B_{11}} \frac{\partial B_{11}}{\partial W_{11}} + \frac{\partial E}{\partial B_{12}} \frac{\partial B_{12}}{\partial W_{11}} + \frac{\partial E}{\partial B_{21}} \frac{\partial B_{21}}{\partial W_{11}} + \frac{\partial E}{\partial B_{22}} \frac{\partial B_{22}}{\partial W_{11}} \\ \frac{\partial E}{\partial W_{12}} &= \frac{\partial E}{\partial B_{11}} \frac{\partial B_{11}}{\partial W_{12}} + \frac{\partial E}{\partial B_{12}} \frac{\partial B_{12}}{\partial W_{12}} + \frac{\partial E}{\partial B_{21}} \frac{\partial B_{21}}{\partial W_{12}} + \frac{\partial E}{\partial B_{22}} \frac{\partial B_{22}}{\partial W_{12}} \\ \frac{\partial E}{\partial W_{21}} &= \frac{\partial E}{\partial B_{11}} \frac{\partial B_{11}}{\partial W_{21}} + \frac{\partial E}{\partial B_{12}} \frac{\partial B_{12}}{\partial W_{21}} + \frac{\partial E}{\partial B_{21}} \frac{\partial B_{21}}{\partial W_{21}} + \frac{\partial E}{\partial B_{22}} \frac{\partial B_{22}}{\partial W_{21}} \\ \frac{\partial E}{\partial W_{22}} &= \frac{\partial E}{\partial B_{11}} \frac{\partial B_{11}}{\partial W_{22}} + \frac{\partial E}{\partial B_{12}} \frac{\partial B_{12}}{\partial W_{22}} + \frac{\partial E}{\partial B_{21}} \frac{\partial B_{21}}{\partial W_{22}} + \frac{\partial E}{\partial B_{22}} \frac{\partial B_{22}}{\partial W_{22}}\end{aligned}\tag{3.50}$$

further

$$\begin{aligned}\frac{\partial E}{\partial W_{11}} &= \frac{\partial E}{\partial B_{11}} A_{11} + \frac{\partial E}{\partial B_{12}} A_{12} + \frac{\partial E}{\partial B_{21}} A_{21} + \frac{\partial E}{\partial B_{22}} A_{22} \\ \frac{\partial E}{\partial W_{12}} &= \frac{\partial E}{\partial B_{11}} A_{12} + \frac{\partial E}{\partial B_{12}} A_{13} + \frac{\partial E}{\partial B_{21}} A_{22} + \frac{\partial E}{\partial B_{22}} A_{23} \\ \frac{\partial E}{\partial W_{21}} &= \frac{\partial E}{\partial B_{11}} A_{21} + \frac{\partial E}{\partial B_{12}} A_{22} + \frac{\partial E}{\partial B_{21}} A_{31} + \frac{\partial E}{\partial B_{22}} A_{32} \\ \frac{\partial E}{\partial W_{22}} &= \frac{\partial E}{\partial B_{11}} A_{22} + \frac{\partial E}{\partial B_{12}} A_{23} + \frac{\partial E}{\partial B_{21}} A_{32} + \frac{\partial E}{\partial B_{22}} A_{33}\end{aligned}\tag{3.51}$$

from above equation 3.51 it can be derived that

$$\frac{\partial E}{\partial W} = A * \frac{\partial E}{\partial B} = A * \delta\tag{3.52}$$

where cross-correlation is valid padding with one stride. Then the convolutional layer weight update equation is given by

$$w_{ij}^* = w_{ij} - \eta (A * \delta)_{ij}\tag{3.53}$$

Here, value δ is decided by next layer structure, if it is normal network like in figure 3.16 which called fully connected layer, the value can be calculated with equation 3.44.

In following δ is derived under case that next layer is convolutional layer, then the influence of convolutional layer to the previous layer weight update should be discussed. Since matrix A is output value of previous layer, the gradient $\frac{\partial E}{\partial A}$ should be considered for previous layer. Similarly as in equations 3.50 - 3.52, after analyze $\frac{\partial E}{\partial A}$ using chain rule it can be found that:

$$\frac{\partial E}{\partial A} = \delta * Wr\tag{3.54}$$

where $Wr_{i,j} = W_{-i,-j}$, and cross-correlation is full padding with one stride. Then δ of previous layer with activation function f can be defined as

$$\delta_{ij}^{l-1} = (\delta_l * Wr_l)_{ij} \frac{\partial f(B_{ij}^{l-1})}{\partial B_{ij}^{l-1}}\tag{3.55}$$

which can be used to update previous layer parameters of convolutional layer. Bias update for convolutional layer is same as that of fully connected layer as in equation 3.47.

Backpropagation of pooling sample is executed by method called upsampling. In case of max pooling, in feedforward stage, the max value position of input data in each step of max pooling

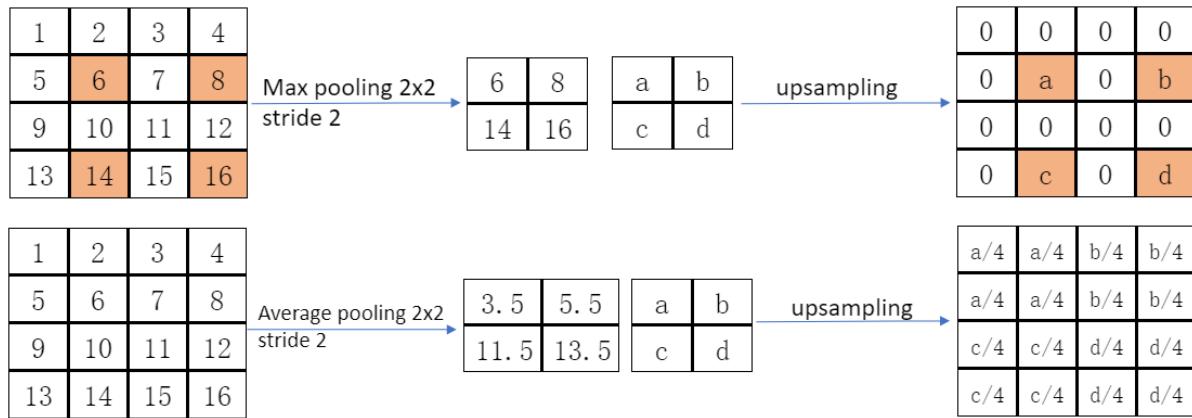


Figure 3.20: an example of pooling layer upsampling

is marked, and in backpropagation stage each value of δ is given to corresponding marked position, other values are set zero. In case of average pooling in backpropagation stage, each value of δ is averaged according to the size of filter and given back to input matrix of layer step by step. The figure 3.20 shows an example of pooling layer upsampling, where matrix $[a,b; c,d]$ is δ of pooling layer. Therefore, to previous layer of pooling layer, δ^{l-1} can be defined as

$$\delta_{ij}^{l-1} = (\text{upsample}(\delta^l))_{ij} \frac{\partial f(I_{ij}^{l-1})}{\partial I_{ij}^{l-1}} \quad (3.56)$$

where activation function of layer $l - 1$ is $f(I_{ij}^{l-1})$. So far, the feedforward and backpropagation principle of convolutional layer and pooling layer are explained.

CNN structure The basic block of CNN is commonly composed of a convolutional layer and a pooling layer. The figure 3.21 shows one example structure of CNN, which contains two convolution network blocks, where subsampling means pooling process. The data which be processed by convolutional layer is called feature map, since the filters of CNN play the role of input data feature capturing.

CNN achieved many good result in many areas especially in image processing, it has some advantages over standard multilayer perceptron. Firstly, from comparing the figures 3.16 and 3.19 can be found that the biggest difference between them is that the weight in CNN are shared between perceptrons, which reduces the number of parameters of network and make the training more efficient, another one difference is that the structure of CNN output data is highly correlated with original input data because the filter in each cross-correlation step focuses on limited data area which brings advantage to CNN since in many cases the spatial feature is also very important. The above statements about advantage of CNN is only the author's personal opinion, in [8] the detailed analysis to CNN are given.

3.7.7 Recurrent neural network

A recurrent neural network(called for short RNN) is a type of multilayer perceptron which is suitable to process sequential information. In standard multilayer perceptron it is assumed that

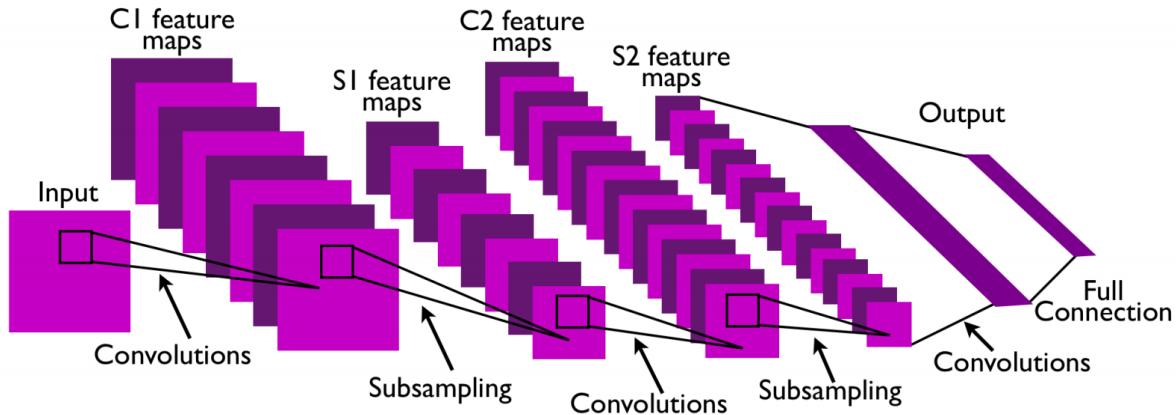


Figure 3.21: structure of a common convolutional networks [9]

the input and output data is independent of each other. But in many practical situations, the relationship between inputs or that between outputs also should be paid attention, for example natural language processing, especially in this article, the data is radar signal from different receiver antenna sensor element which is highly related to each other.

The following figure 3.22 shows the basic concept of recurrent neural network, where x_t is input data in time step t , s_t is hidden state at time step t , o_t is output data at time step t , and U, W, V are weights between input data and hidden state, hidden state at time step $t - 1$ and t , output and hidden state respectively. And the hidden state is calculated by

$$s_t = f_h(Ux_t + Ws_{t-1} + b1) \quad (3.57)$$

where f_h is activation function which is usually \tanh or $ReLU$, $b1$ is bias to hidden state. And the output is

$$o_t = f_o(Vs_t + b2) \quad (3.58)$$

where f_o is activation function, $b2$ is bias to output data.

One RNN network has only one hidden layer, whose output can influence next time step output, which is only and most important difference between standard multilayer perceptron and RNN. There are many variants of RNN, for example Long short-term memory, gated recurrent unit and so on. In this article, the gated recurrent unit(called for short GRU) is used to build up the network model, therefore in following the GRU is discussed.

gated recurrent unit As shown in figure 3.23, the data processing principle of GRU is similar with a standard RNN, one thing remarkable is that the output and hidden state of GRU at same time step are same values.

In following the concrete internal structure of GRU is discussed which combined with data feedforward and backpropagation.

Figure 3.24 presents the internal structure and data feedforward in GRU, where

$$r_t = \text{sigmoid}(W_r * h_{t-1} + U_r * x_t) \quad (3.59)$$

$$z_t = \text{sigmoid}(W_z * h_{t-1} + U_z * x_t) \quad (3.60)$$

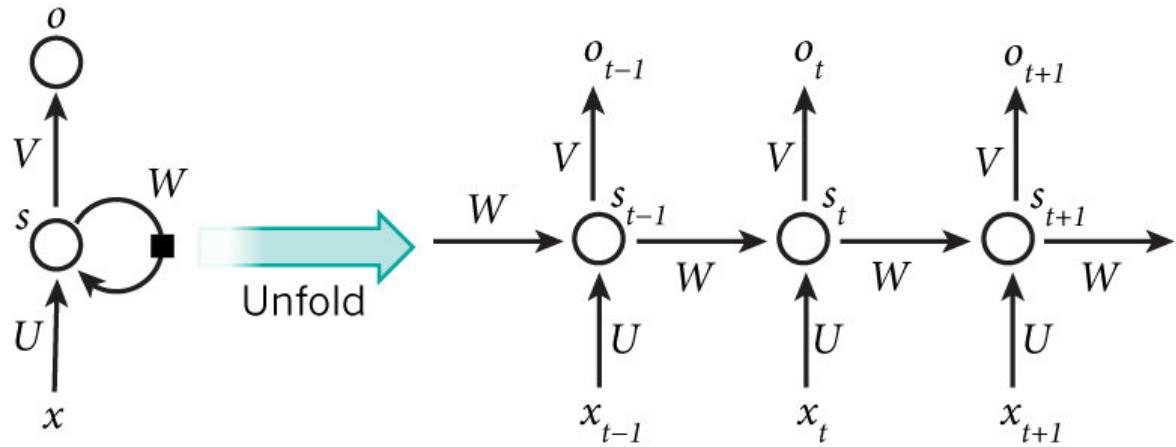


Figure 3.22: A recurrent neural network and the unfolding in time of the computation involved in its forward computation [10]

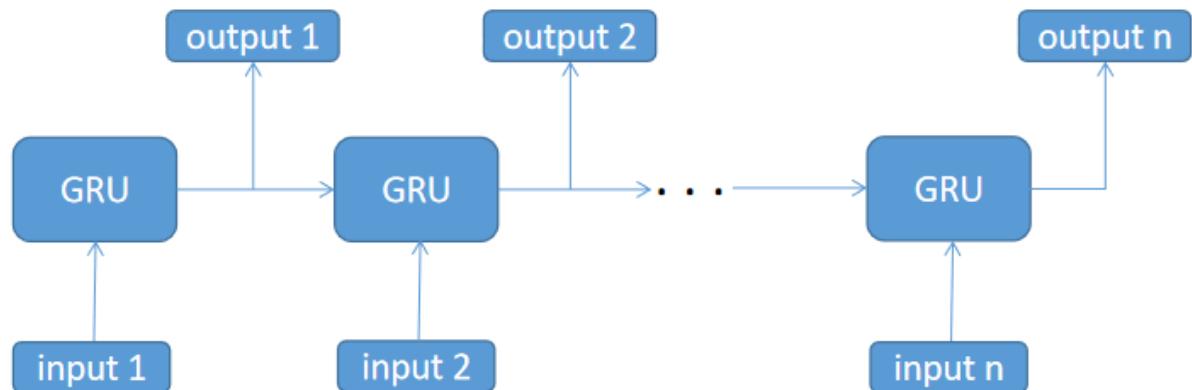


Figure 3.23: GRU feedforward with sequential input data

GRU unit feed forward

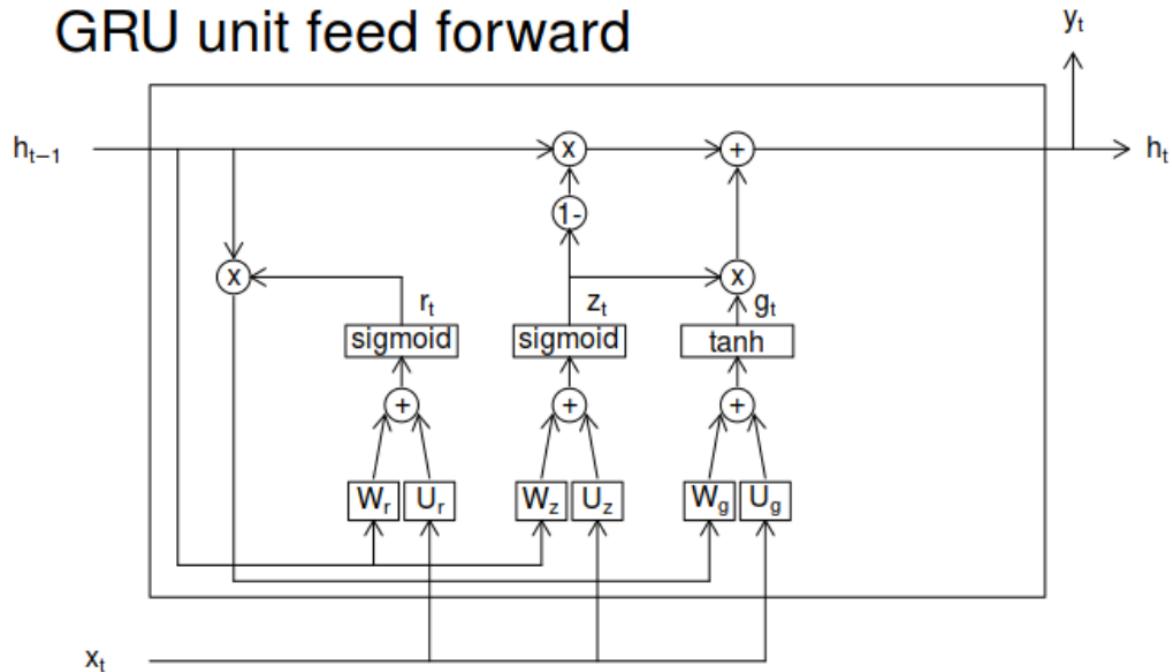


Figure 3.24: internal structure and data feedforward of GRU [11]

$$g_t = \tanh(W_g * (h_{t-1}r_t) + U_g * x_t) \quad (3.61)$$

$$h_t = y_t = h_{t-1}(1 - z_t) + z_t g_t \quad (3.62)$$

Figure 3.25 shows backpropagation in GRU, where

$$\delta 3 = \delta 1 + \delta 2 \quad (3.63)$$

$$\delta 4 = (1 - z_t)\delta 3 \quad (3.64)$$

$$\delta 5 = h_{t-1}\delta 3 \quad (3.65)$$

$$\delta 6 = 1 - \delta 5 \quad (3.66)$$

$$\delta 7 = g_t\delta 3 \quad (3.67)$$

$$\delta 8 = z_t\delta 3 \quad (3.68)$$

$$\delta 9 = \delta 7 + \delta 8 \quad (3.69)$$

$$\delta 10 = \delta 8 \tanh'(g_t) \quad (3.70)$$

$$\delta 11 = \delta 9 \text{sigmoid}'(z_t) \quad (3.71)$$

$$\delta 12 = \delta 10 * W_g^T \quad (3.72)$$

$$\delta 13 = \delta 10 * U_g^T \quad (3.73)$$

$$\delta 14 = \delta 11 * W_z^T \quad (3.74)$$

$$\delta 15 = \delta 11 * U_z^T \quad (3.75)$$

$$\delta 16 = \delta 13 h_{t-1} \quad (3.76)$$

GRU unit back propagation

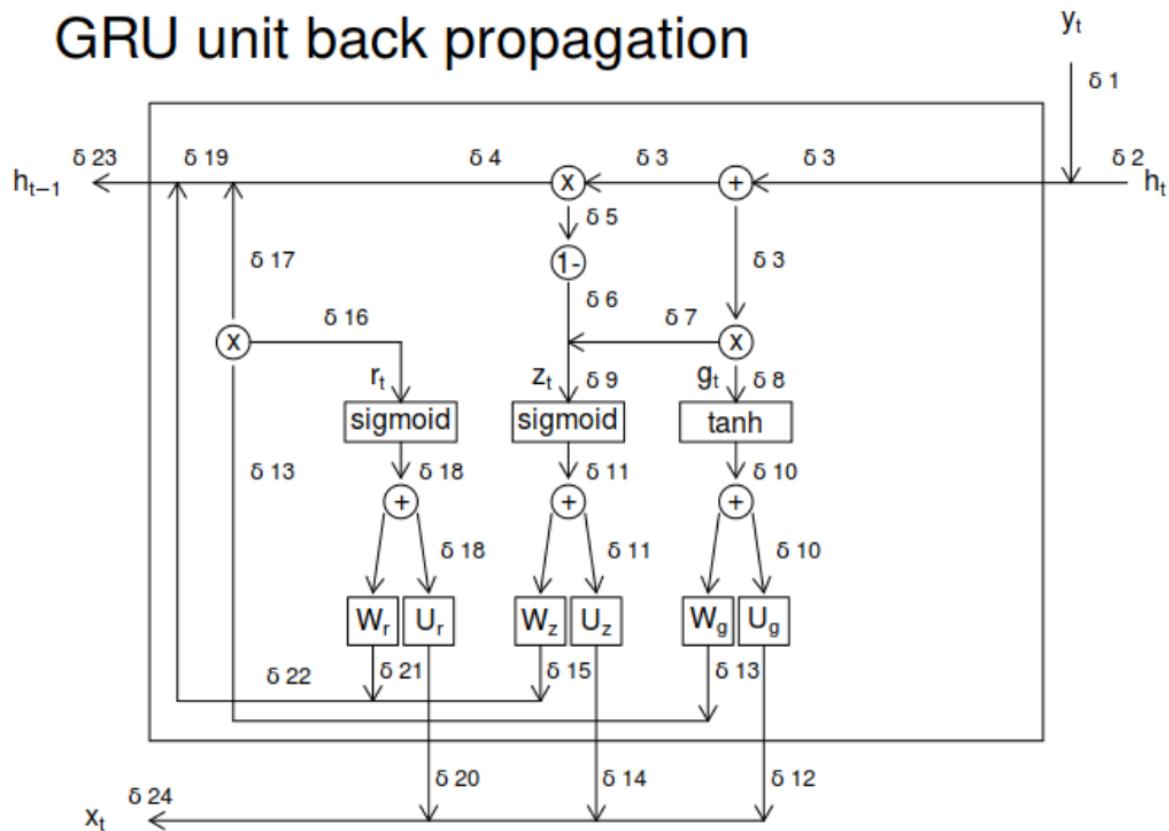


Figure 3.25: internal structure and error backpropagation of GRU [11]

$$\delta_{17} = \delta_{13} r_t \quad (3.77)$$

$$\delta_{18} = \delta_{17} \text{sigmoid}'(r_t) \quad (3.78)$$

$$\delta_{19} = \delta_{17} + \delta(4) \quad (3.79)$$

$$\delta_{20} = \delta_{18} * W_r^T \quad (3.80)$$

$$\delta_{21} = \delta_{18} * U_r^T \quad (3.81)$$

$$\delta_{22} = \delta_{21} + \delta_{15} \quad (3.82)$$

$$\delta_{23} = \delta_{19} + \delta_{22} \quad (3.83)$$

$$\delta_{24} = \delta_{12} + \delta_{14} + \delta_{20} \quad (3.84)$$

Since all δ in GRU are clear, the updates of weights and bias can be calculated using equations 3.46 and 3.46. Because δ_{23} can be calculated, backpropagation compute of previous layer is also possible.

4 2D DOA Estimation using MUSIC Algorithm

To verify the performance of ANN model in 2D DOA estimation, in this article the performance of neural network model is compared with that of MUSIC algorithm. Therefore before training the ANN model, the music algorithm is implemented in this chapter.

4.1 FMCW mm Wave Radar Simulator establishing in MATLAB

First of all, a radar simulator should be established to generate simulated data for MUSIC algorithm and ANN training. In this article, the simulator is established using phased array system toolbox of a software called MATLAB, which is developed by MathWorks.

It is assumed that the transmitter antenna element and center element of receiver antenna are located at origin of 3 dimensional coordinate system, an 1 transmitter 3×3 rectangular receiver array antenna is used to generate radar data, the corresponding receiver array geometry and size is shown in figure 4.1, where element spacing is half wavelength of 77GHz wave with light speed. Noise figure of radar receiver is set to 4.5dB.

The noise figure is defined as

$$F = SNR_i - SNR_o \quad (4.1)$$

where SNR_i and SNR_o is input and output SNR respectively. SNR is abbreviation of signal noise ratio, which defined as

$$SNR = \frac{P_{signal}}{P_{noise}} = \frac{A_{signal}}{A_{noise}} \quad (4.2)$$

where P_{signal} , P_{noise} , A_{signal} , A_{noise} are power of signal, power of noise, amplitude of signal, amplitude of noise respectively, if use dB in calculation it can be alternatively written as

$$SNR = 10 \log_{10}(P_{signal}) - 10 \log_{10}(P_{noise}) \quad (4.3)$$

The transmitted FMCW signal with 1GHz bandwidth $288\mu\text{s}$ sweep time is presented in figure 4.2.

Other required parameters of radar and targets are presented in chapter 2. After establishing the radar system and target model, radar signal data of targets in different range, velocity and 2D DOA angle can be simulated. Since the MUSIC algorithm requests covariance matrix of target signal, in order to compare the performance of ANN model with that of MUSIC algorithm, in this article all the target signal data are used in form of covariance matrix.

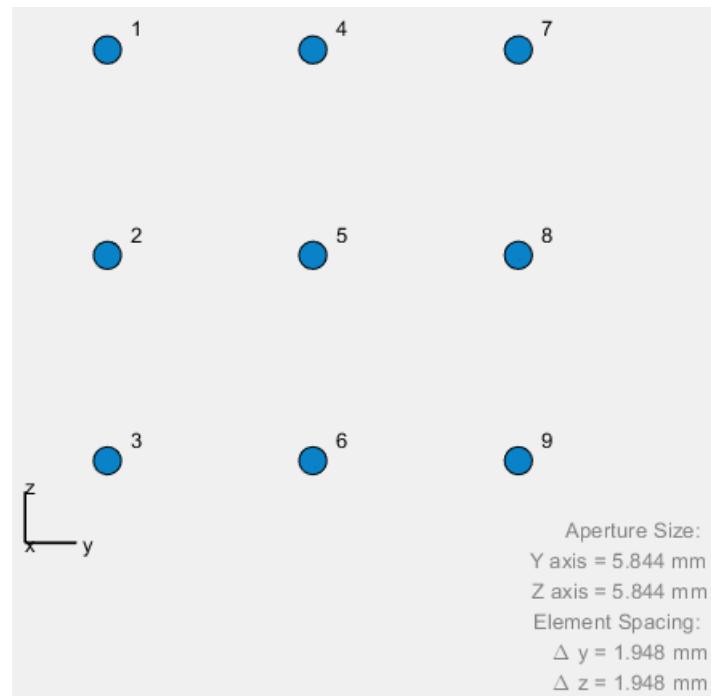
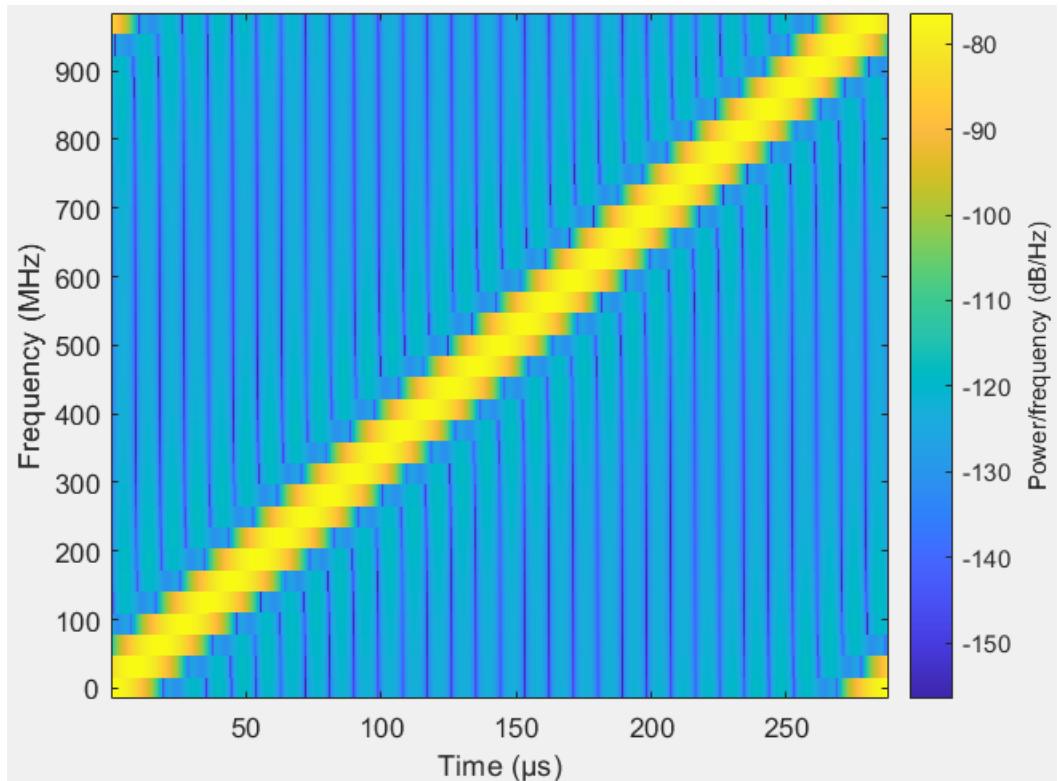


Figure 4.1: radar receiver array geometry and size

Figure 4.2: Transmitted FMCW signal, whose bandwidth is 1GHz, sweep time is $288\mu\text{s}$

4.2 Determining the maximum detectable targets number using MUSIC algorithm under given radar configuration

The maximum detectable number of targets using single chirp radar signal is influenced by radar configuration, noise, measuring error and pre-processing method. That means under determined simulation environment, the maximum number of detectable targets are also determined. In terms of neural network training, the quality of training data is one the most important factor which influences the model performance, and the high relationship of data and its label guarantees the quality of data. Therefore, in this section, the maximum detectable number of target using MUSIC algorithm is explored under last section given radar configuration and target parameters.

The determining process is as follows. At first step, 10000 radar signal samples of one target with random 2D DOA angles are generated. Then each sample is used to detect 2D DOA of targets using MUSIC algorithm. Next, the mean absolute error of each estimation result against true target DOA is calculated. If the mean absolute error is acceptable, then increase target number and repeat the process until the mean absolute error is not acceptable. Then the maximum target number with acceptable mean absolute error is the maximum detectable targets number.

Azimuth angle mean absolute error from each sample signal can be computed by following equation:

$$MAE_{azi} = \frac{\sum_{i=1}^m |y_{az} - x_{az}|}{m} \quad (4.4)$$

where m , y_{az} and x_{az} are true number of targets, true azimuth angle of targets and using MUSIC algorithm estimated azimuth angle of target respectively. Similarly, the elevation angle mean absolute error from each sample signal is given by

$$MAE_{el} = \frac{\sum_{i=1}^m |y_{el} - x_{el}|}{m} \quad (4.5)$$

In the calculation of mean absolute error, if the MUSIC algorithm fails to find out existence of target, mean absolute error of the corresponding target is set to $(61^\circ, 41^\circ)$, which is maximum value of error. The matching of true angle value and detected angle value is calculated using Euclidean distance. Besides, the resolution of target angle detection is set to 1° .

After calculate mean absolute error of each sample, the mean value of mean absolute error of 10000 samples are derived in order to summarize the quality of radar signal. The mean value of mean absolute error is shown in figure 4.3. Considering the resolution is 1° , MUSIC algorithm shows best performance when targets number is under 3, azimuth and elevation errors of 3 targets detection result are 1.78° and 1.20° respectively which is still acceptable, when targets number is more than 3, MUSIC algorithm can't guarantee a good performance in 2D DOA angle estimation. Then the conclusion can be derived that the maximum detectable number of targets when using 1 chirp signal of corresponding radar configuration and SNR can be seen as 3.

In following sections, ANN are developed for 2D DOA estimation of 1 to 3 targets, since under chapter 2 given simulation environment, simulated radar signal can provide good training data quality only when the target number is under 4.

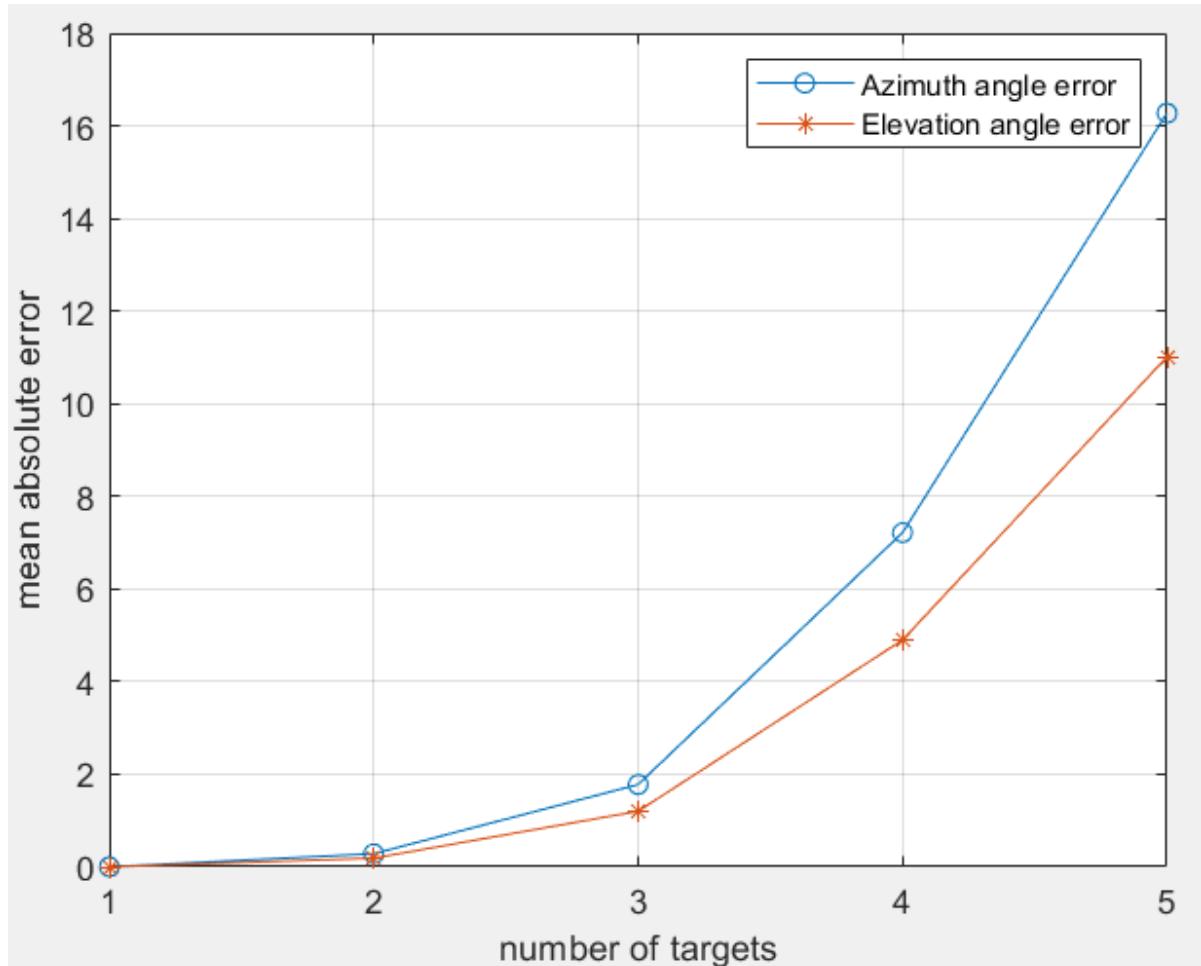


Figure 4.3: Mean absolute error of MUSIC algorithm detected 2D DOA target angles, where sample number of each stage is 10000

5 Data Pre-processing for Artificial Neural Network

In this section, it is introduced that how the data generated and pre-processed for training data and test data for the ANN.

5.1 Convert complex data into real data

The radar signal is collected in complex form, whose real part represents signal amplitude while the imaginary part stands for signal phase information. In order to compare the performance of ANN with that of MUSIC algorithm, the signal should be transformed into covariance matrix form, which is still in complex form. Assume that the radar signal in covariance matrix is \mathbf{X} , where matrix element $x_{a,b}$ is complex value. Since the number of receiver antenna element is 9, $a,b \in 1,2,3,\dots,9$.

As the next step, covariance matrix \mathbf{X} should be convert into real matrix, because common neural network has no ability to calculate complex value. There are many difference way to convert complex matrix to real matrix, in this article, a new matrix \mathbf{Y} with 9 rows and 18 columns is established, when assume that $y_{i,j}$ is elements of \mathbf{Y} , the relationship of $x_{a,b}$ and $y_{i,j}$ can be defined as

$$real(x_{a,b}) = y_{a,2b-1} \quad (5.1)$$

$$imag(x_{a,b}) = y_{a,2b} \quad (5.2)$$

where $real(x_{a,b})$ is real part of $x_{a,b}$, $imag(x_{a,b})$ is imaginary part of $x_{a,b}$. An schematic diagram explains the rule with converting 2×2 complex matrix into real matrix, which shown in figure 5.1.

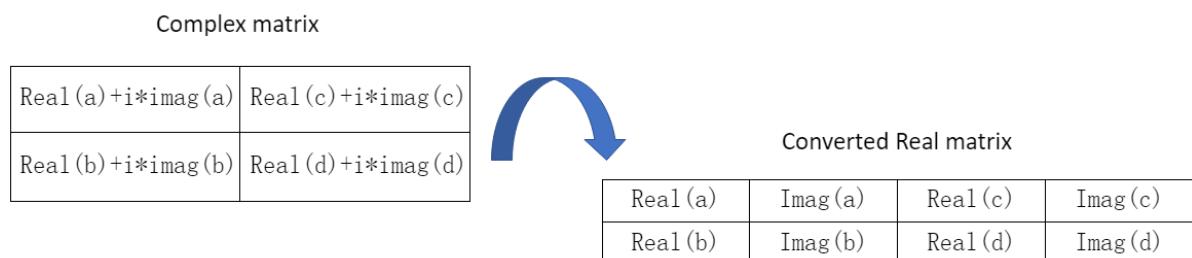


Figure 5.1: this figure explains how a complex matrix is convert into real matrix in this article

5.2 Data normalization

In machine learning area, data normalization is commonly used to increase model performance. The main purpose of data normalization is to map all the data into a limited range for example (0,1) or (-1,1) so that to avoid influence of data anomaly.

In this article, the converted matrix \mathbf{Y} is normalized into range of (0,1) using min-max normalization which defined by

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (5.3)$$

where x' is normalized element of sample data, $\min(x)$ is minimum element value of sample data, $\max(x)$ is maximum element value of sample data.

A comparison experiment is executed to verify the necessity to normalize input data. 7503 samples for one target 2D DOA estimation are generated without normalization while the other 7503 samples are not normalized, when training the ANN, 99% of data is used to train the network, 1% is used to validate the training result. The experiment result is shown in figure 5.2, where the epoch means the process that whole training data is trained for one time. It can be seen in the figure (b) of 5.2, that if the input data is normalized, the validation loss drop down to almost 0 which means the model is able to estimate target 2D DOA angles. In other side, in figure (b) of 5.2, the validation loss keeps high value along the training process while train loss is very low, which means overfitting and failure of model training. This experiment verified the necessity of normalization of input data.

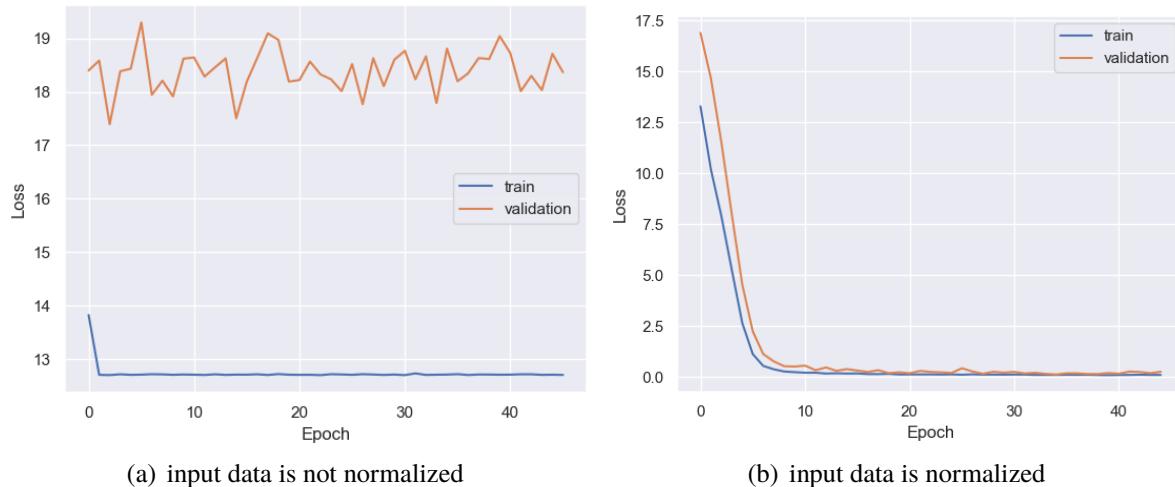


Figure 5.2: comparison of learning curve when input data is normalized and not normalized

5.3 Label design

Label shows the corresponding target angles of input data. In this article, two forms of label are considered.

value label The first form of label direct uses the true 2D DOA angles of targets, which in this article called value label. Since the target azimuth and elevation angle ranges are limited in range of -30° to 30° and -20° to 20° respectively, there are two choices when choosing value label, first one is using the ranges of [0,61] and [0,41], second one is using the ranges of [-30,30] and [-20,20] which correspond all the possible angle value with 1° resolution.

An experiment is executed in order to compare the two kinds of value label, 12505 radar signal samples with one target are generated, 99% of data is used to train the network, 1% is used to validate the training result. at the first training the first kind of value label is used while the second training uses second kind of value label, the experiment result is shown in figure 5.3, where figure (a) shows the training result with label ranges [0,61] and [0,41], figure (b) shows the that with label ranges [-30,30] and [-20,20]. It can be seen from the experiment result that the first kind value label shows a faster convergence speed in its learning curve. But at the end of training, in terms of angle estimation accuracy, the two kinds value label has no much difference, therefore, it can be said that the second kind value label has a slight advantage over the first kind value label.

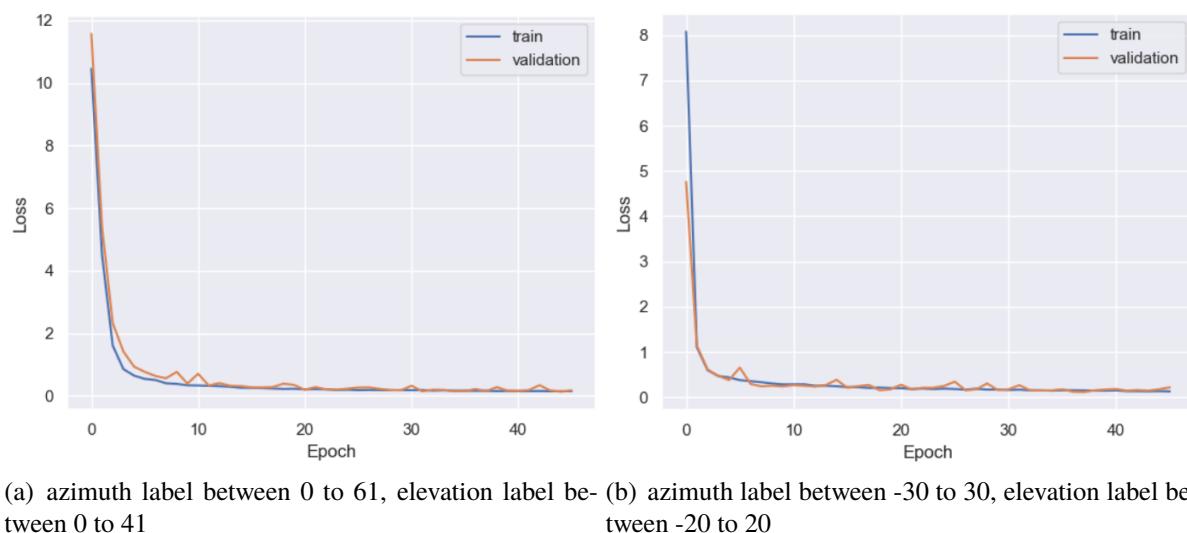


Figure 5.3: comparison of learning curve between different value label

one hot label The second form label uses the form of one-hot-label, which is widely used in area of machine learning. One hot label is a representation of categorical variables as binary vectors, which is suitable for multiclass classification.

For example, there are pictures of 3 kinds of animals dog cat and pig should be classified using neural network model. There can be basically two kinds of label plans as shown in figure 5.4. As the basic label plan, value label still can be chosen which represents label information with integer values, all the pictures of dog are labeled as 1 while cat picture and pig picture are labeled 2 and 3 respectively. As to one hot label, the value label is converted into a binary vector where all values are zero except the index of label integer which is marked with 1, as shown in figure 5.4, one hot label of dog is (1,0,0) is converted from integer 1, while that of cat (0,1,0) converted from integer 2 and pig's one hot value (0,0,1) from integer 3.



Figure 5.4: one hot label plan under classification mission with 3 classes

In terms of 2D DOA estimation data label, the one hot label can be converted from value label whose ranges are [0,61] and [0,41]. The one hot label in this case is a 61×41 dimensional binary matrix whose all elements are zero except the index of targets angles which are marked with 1. For example, if there are two targets with 2D DOA angles (20,10) and (-5,0), the corresponding value label is 2×2 dimensional integer matrix [51,26; 31,21], then in its one hot label matrix, elements with index [51,31] and [26,21] are marked with 1 while the other elements are marked with 0.

One hot label in the 2D DOA estimation problem has both advantages and disadvantages. The first advantage is that the label dimension doesn't change when targets number is different, theoretically one hot label covers all situations in 2D DOA angle estimation. But the one hot label is too big which requires total 2501 output values in model output layer, which brings huge compute pressure. On the other hand, since the loss is calculated with 2500 outputs while only 1-3 of them corresponds to the target angles, the model performance can't be guaranteed compared to that with value label, which can be seen as an extreme multilabel classification problem.

As a conclusion it can be said that value label has a better performance than one hot label on current 2D DOA estimation problem, but the disadvantage of value label is that the label dimension is fixed according to number of targets, that means if model uses value label, the number of targets should be known in advance. The one hot label shouldn't change whatever the number of targets is, that means one hot label can be used when number of targets is unknown, if the extreme multilabel classification problem is solved, the one hot label has better potential than value label.

6 2D DOA Estimation using Artificial Neural Network

In this chapter, 2D DOA estimation is implemented based on different type ANN models. After introducing the result, comparisons of different ANN models and MUSIC algorithm are presented.

6.1 ANN based 2D DOA angle estimation using data with value label

In this section, 2D DOA estimations using 3 kinds of ANN are implemented. And the performances of different models and MUSIC algorithm are compared with each other. Adam[12] optimizer is chosen for all the models. Mean absolute loss is used in model training. Input data is 9×18 dimensional matrix as described in section 5. In terms of label, the form of value labels [0,61],[0,41] for azimuth angle label and elevation angle are used.

6.1.1 Single target 2D DOA angle estimation

In single target 2D DOA angle estimation model training, in order to verify the generalization ability of models, relative small training data set which contains randomly generated 1000 samples is used, and 7503 samples are used as validation data set while other 12505 samples are used as test data set.

fully connected neural network model The structure and learning curve of fully connected neural network based single target 2D DOA estimation model are shown in figure 6.1 and 6.2 respectively. It can be seen in the figure 6.1 that the model is composed of input layer, one hidden layer with 128 perceptrons, two output layers both with one perceptron, where one perceptron outputs azimuth angle estimation value while the other one outputs elevation angle estimation value. The model has total 21122 trainable parameters.

In learning curve figure 6.2, the value of loss is mean value of azimuth angle error and elevation angle error, which is same in other learning curve figures. The training and validation loss approaches value of zero as increasing number of epoch, the azimuth angle mean absolute error and elevation angle mean absolute error of test data are 0.29° and 0.08° respectively, which verifies the good performance of fully connected neural network model in single target 2D DOA estimation.

CNN model The structure and learning curve of CNN based single target 2D DOA estimation model are shown in figure 6.3 and 6.4 respectively. As can be seen in the figure 6.3, model

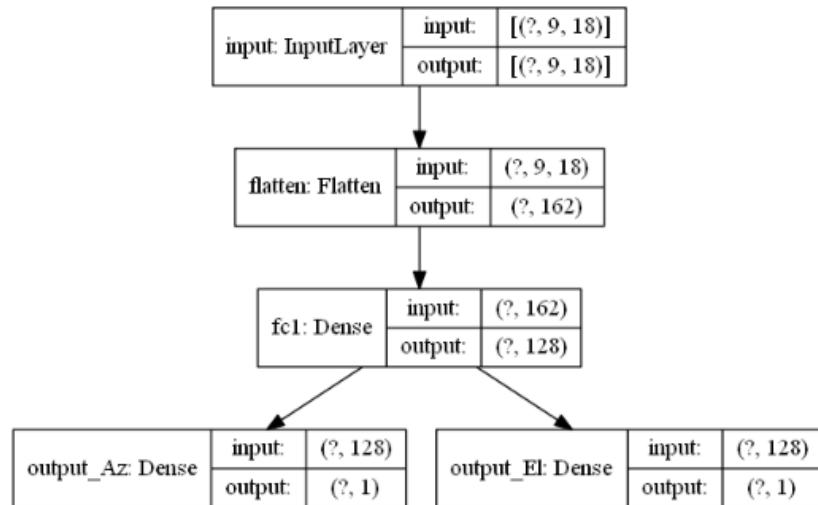


Figure 6.1: structure of fully connected neural network based single target 2D DOA estimation model

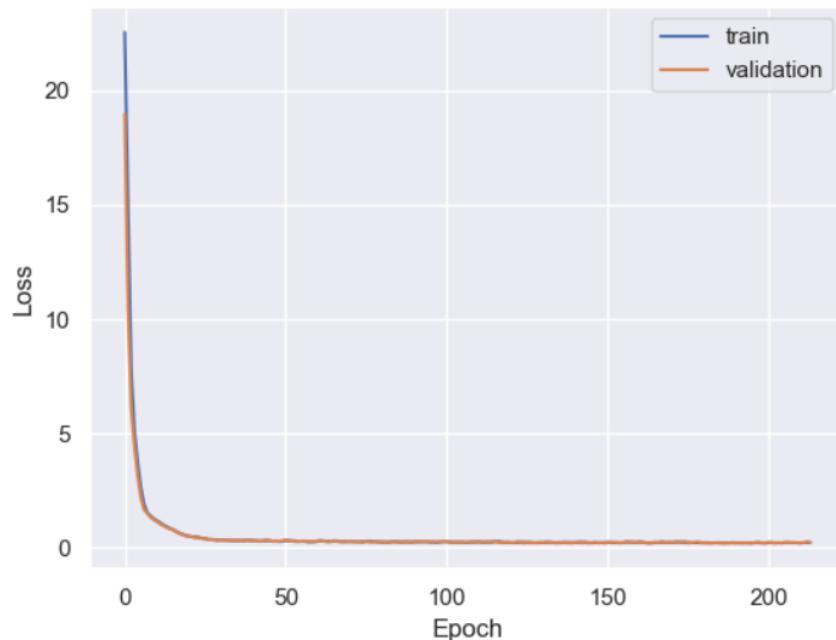


Figure 6.2: learning curve of fully connected neural network based single target 2D DOA estimation model

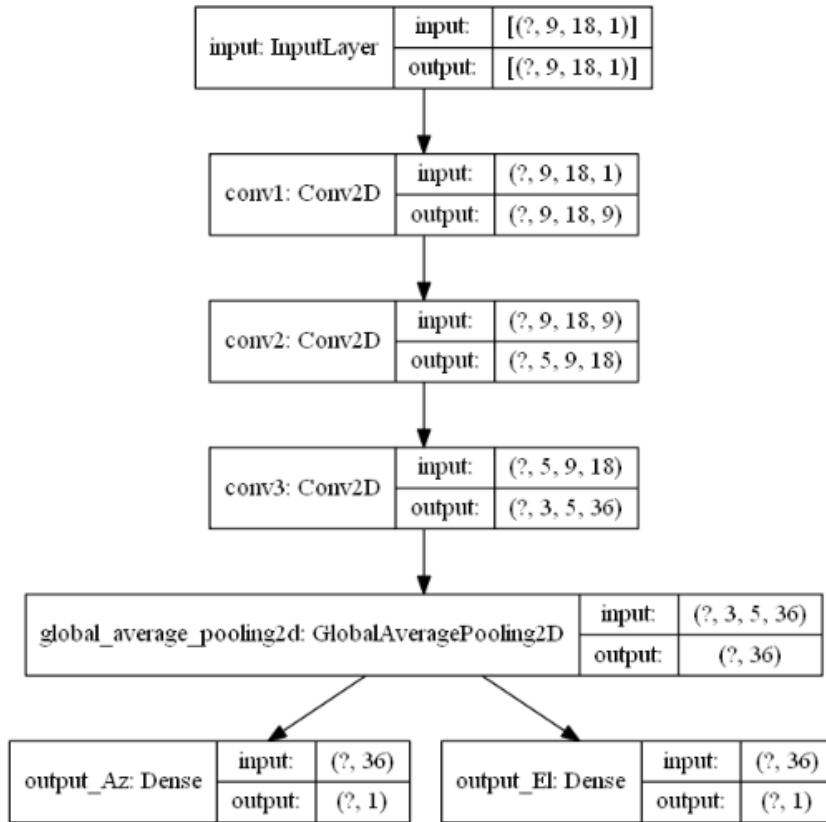


Figure 6.3: structure of fully connected neural network based single target 2D DOA estimation model

has 4 hidden layers, whose number of filters, filter size and strides of first 3 2D convolutional layers are (9,18,36), 3×3 and (1,2,2). Last layer of hidden layers is global average pooling, which outputs average value of each input data channel of layer. Model has 2 output layers both with single perceptron, where one perceptron outputs azimuth angle estimation value while the other one outputs elevation angle estimation value. The model has total 7508 trainable parameters.

Besides, azimuth angle mean absolute error and elevation angle mean absolute error of test data are 0.23° and 0.14° respectively, which verifies the good performance of CNN model in single target 2D DOA estimation.

GRU model The structure and learning curve of GRU based single target 2D DOA estimation model are shown in figure 6.5 and 6.5 respectively. It is shown in figure 6.5 that the model has one hidden layer with 64 GRU and two output layer with single perceptron, where one perceptron outputs azimuth angle estimation value while the other one outputs elevation angle estimation value. The model has total 16258 trainable parameters.

The azimuth angle mean absolute error and elevation angle mean absolute error of test data are 0.05° and 0.07° respectively, which means GRU based neural network model can work in single target 2D DOA estimation.

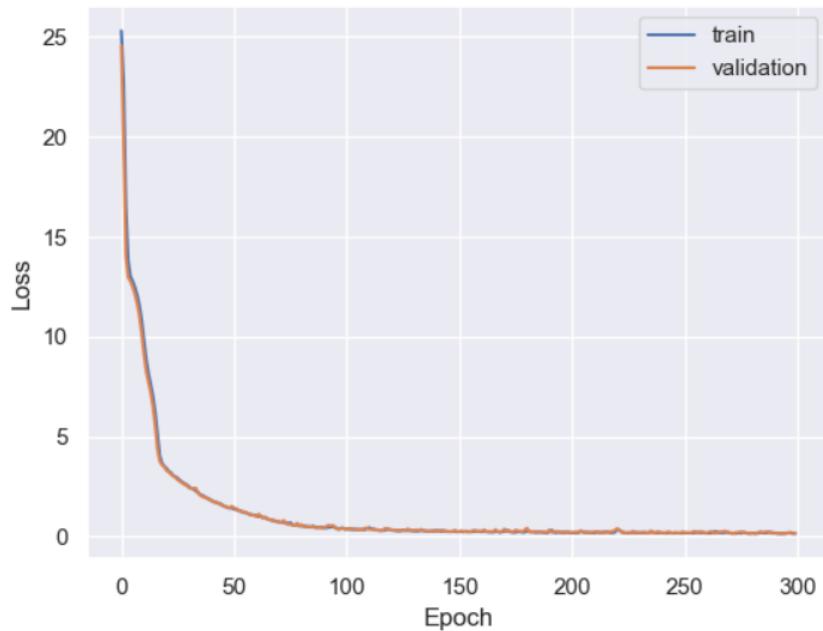


Figure 6.4: learning curve of CNN based single target 2D DOA estimation model

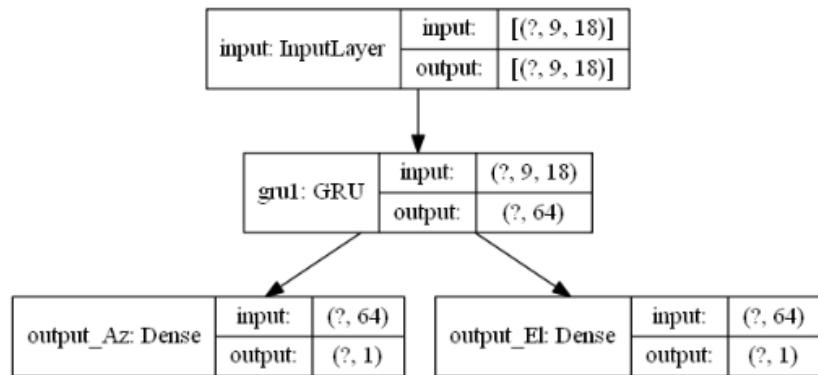


Figure 6.5: structure of fully connected neural network based single target 2D DOA estimation model

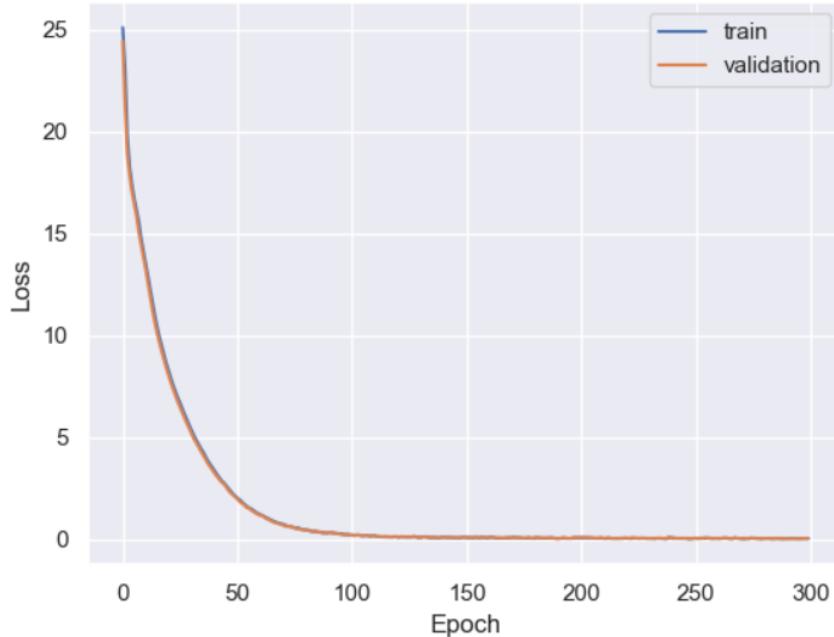


Figure 6.6: learning curve of CNN based single target 2D DOA estimation model

comparison of different model The figure (a) of 6.7 shows the comparison of different model performance on test data, considering the angle resolution is 1° , it can be said that the three kinds of ANN models all meet the performance requirement on single target 2D DOA estimation. GRU based model shows the best performance among all the models which above given.

The number of trainable parameters represents the required computing resources when training and using the model, considering the number of trainable parameters which shown in figure (b) of 6.7, fully connected neural network based model has the worst performance with most number of parameters.

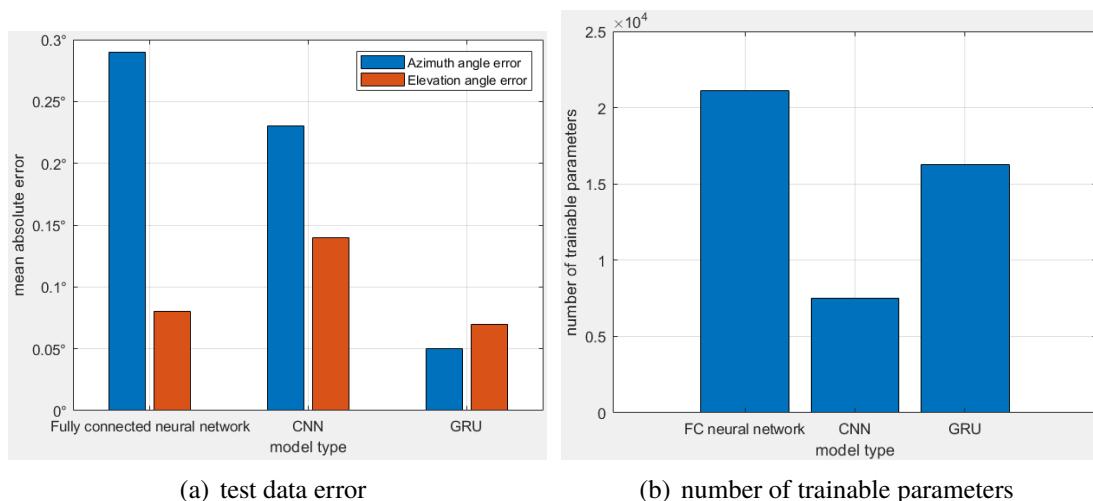


Figure 6.7: comparison of different single target 2D DOA estimation models

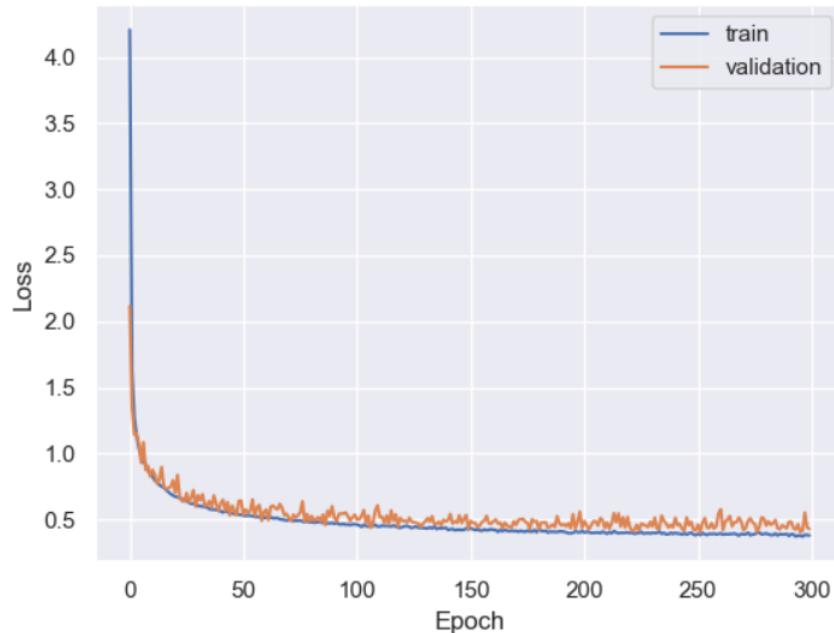


Figure 6.8: learning curve of CNN based double target 2D DOA estimation model

6.1.2 Double targets 2D DOA angle estimation

In double target 2D DOA angle estimation model training, 1×10^5 samples are used as training data set, while other two data set with 2500 samples are used in validation and testing. In following sections, neural networks only with fully connected layer is not considered because of the unsatisfactory performance.

CNN model The structure and learning curve of CNN based double target 2D DOA estimation model are shown in figure 6.9 and 6.8 respectively. As can be seen in the figure 6.9, model has 7 hidden layers, whose number of filters, filter size and strides of first 6 2D convolutional layers are (9,18,36,72,144,288), 3×3 and (1,2,2,2,2,2). Last layer of hidden layers is global average pooling. Model has 2 output layers both with double perceptrons, where each layer outputs azimuth angle estimation values while the other outputs elevation angle estimation values. The model has total 498982 trainable parameters.

Besides, azimuth angle mean absolute error and elevation angle mean absolute error of test data are 0.43° and 0.45° respectively.

GRU model The structure and learning curve of GRU based double target 2D DOA estimation model are shown in figure 6.11 and 6.10 respectively. Model is composed of input layer, one hidden layer with 128 GRU, 2 output layers both with double perceptrons, where each layer outputs azimuth angle estimation values while the other outputs elevation angle estimation values. The model has total 57348 trainable parameters.

The azimuth angle mean absolute error and elevation angle mean absolute error of test data are 0.22° and 0.29° respectively.

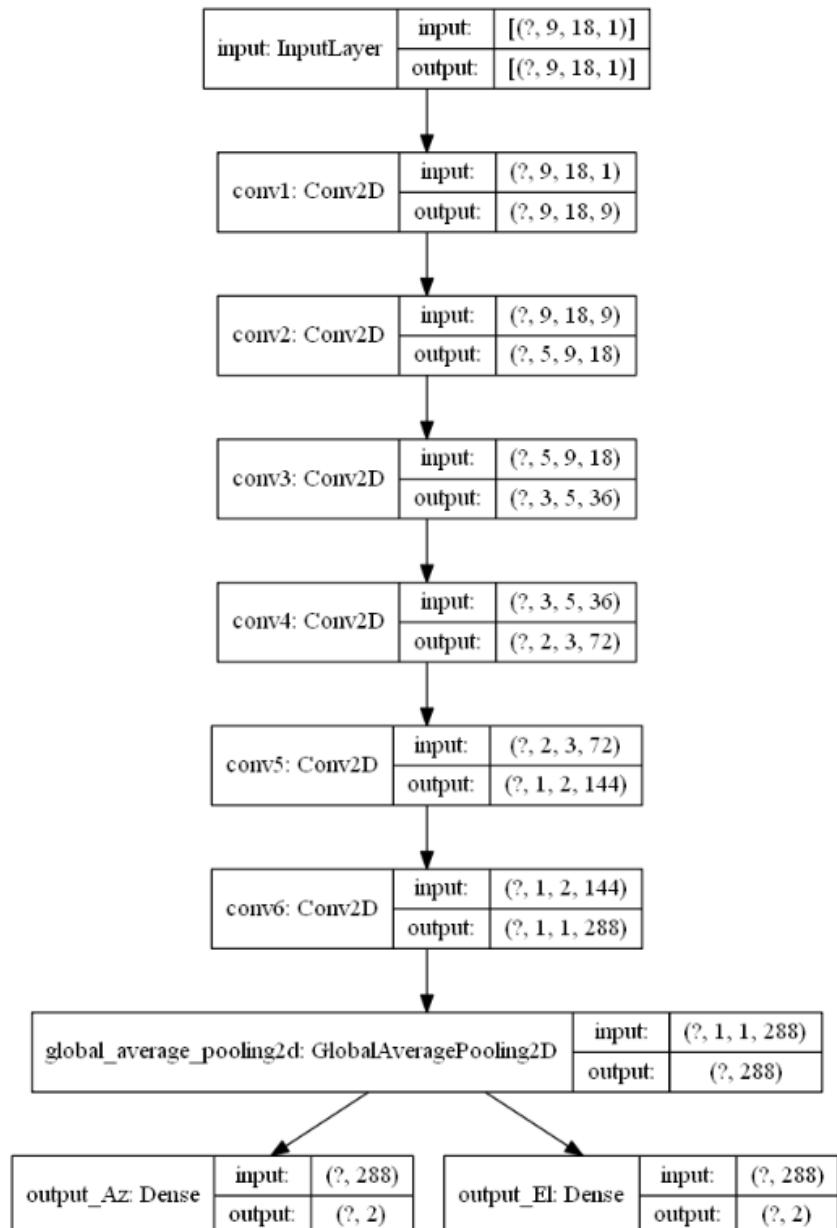


Figure 6.9: structure of CNN based double target 2D DOA estimation model

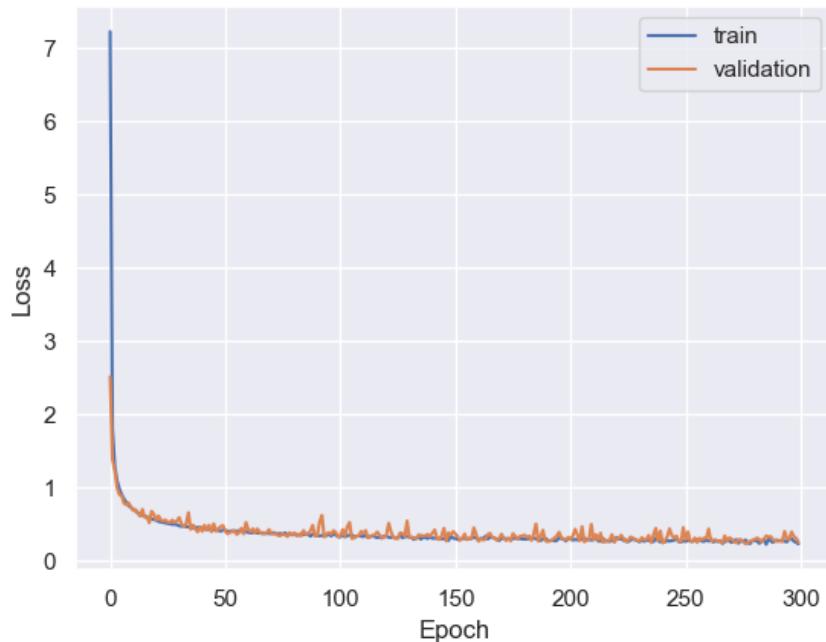


Figure 6.10: learning curve of GRU based double target 2D DOA estimation model

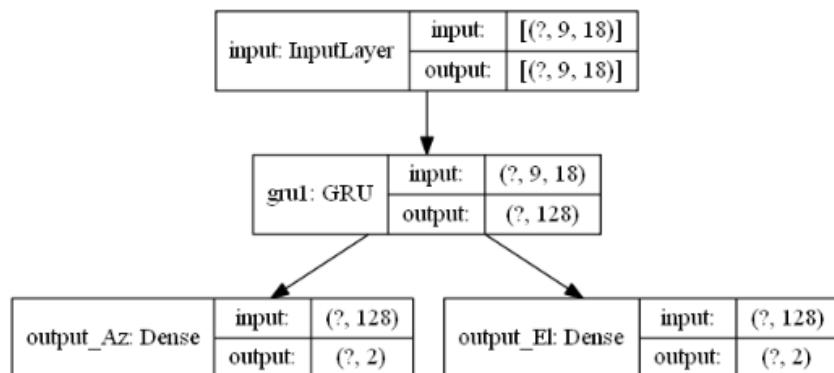


Figure 6.11: structure of GRU based double target 2D DOA estimation model

comparison of different model The figure (a) in 6.12 shows the comparison of CNN and GRU model performance on test data, considering the angle resolution is 1° , it can be said that the both of ANN models meet the performance requirement on double target 2D DOA estimation. Figure (b) in 6.12 presents number of trainable parameters in each model.

If the attentions are paid to both model performance on DOA estimation and model trainable parameters number, it can be found in figure 6.12 that GRU based model has better performance with less trainable parameters.

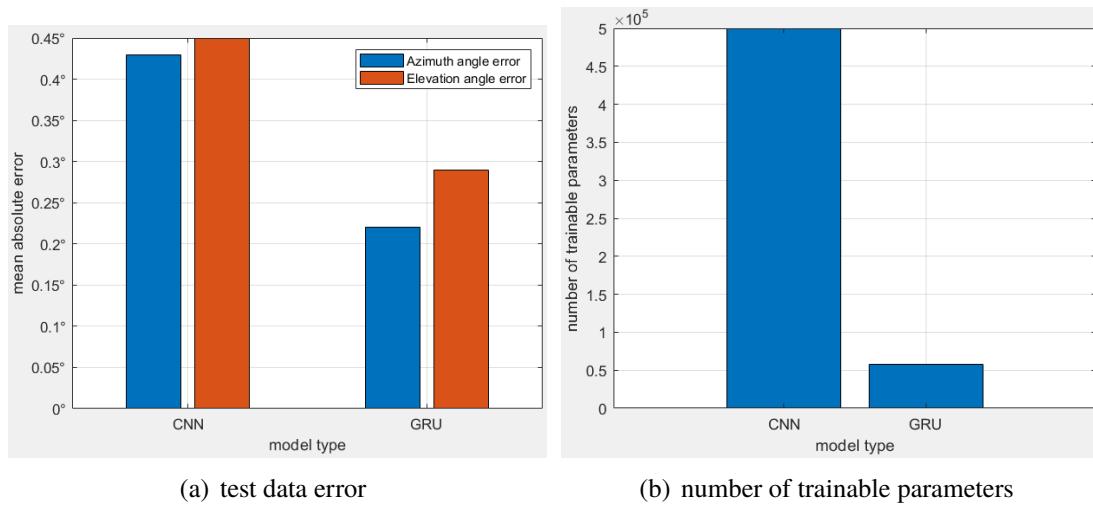


Figure 6.12: comparison of different double targets 2D DOA estimation models

6.1.3 Three targets 2D DOA angle estimation

In three target 2D DOA angle estimation model training, two training data sets with 5×10^5 and 1.45×10^6 samples are used in order to compare the performance of model under different number of training samples. Other two data set with 2500 samples are used in validation and testing.

CNN model The structure and learning curve of CNN based double target 2D DOA estimation model under different training sets are shown in figure 6.14 and 6.13 respectively. As can be seen in the figure 6.14, model has 10 hidden layers with 9 convolutional layers and 1 global average pooling layer. The strides and padding of 9 convolutional layers are 1 and valid padding respectively, filter size of first 6 convolutional layers are 3×3 while that of last 3 convolutional layers are 2×2 .

2 output layers both with three perceptrons, where each layer outputs azimuth angle estimation values while the other outputs elevation angle estimation values. The model has total 3544503 trainable parameters.

Figure 6.13 shows the influence of sample number to the model training result, azimuth angle mean absolute error and elevation angle mean absolute error of test data with 1×10^5 samples are 1.46° and 1.80° respectively, while under 1.45×10^6 train samples they are 1.40° and 1.66° .

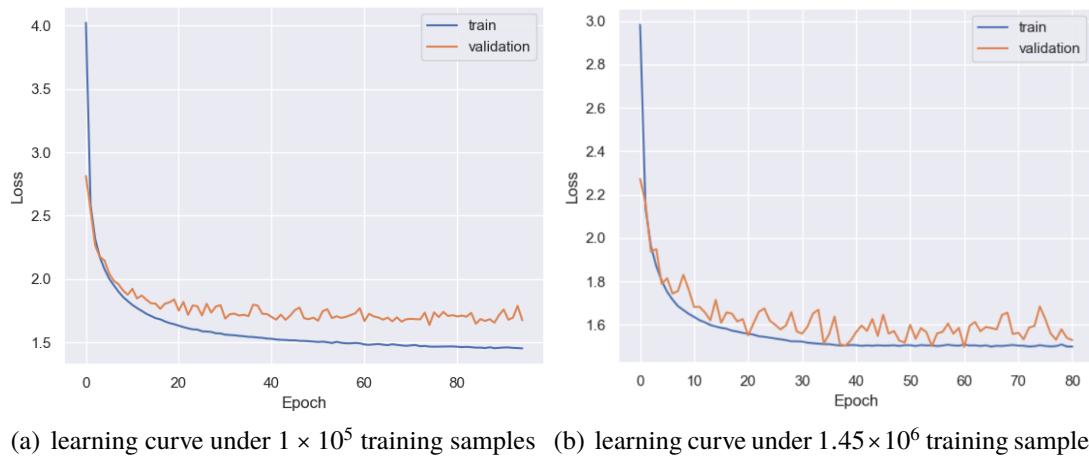


Figure 6.13: comparison of CNN based three target 2D DOA estimation model performance on different training samples

GRU model The structure and learning curve of GRU based double target 2D DOA estimation model under different training sets are shown in figure 6.16 and 6.15 respectively. As can be seen in the figure 6.16, model has 1 hidden layer with 256 GRUs, 2 output layers both with three perceptrons, where each layer outputs azimuth angle estimation values while the other outputs elevation angle estimation values. The model has total 213510 trainable parameters.

Figure 6.15 shows the influence of sample number to the model training result, azimuth angle mean absolute error and elevation angle mean absolute error of test data with 1×10^5 samples are 1.10° and 1.25° respectively, while under 1.45×10^6 train samples they are 0.67° and 0.75° .

comparison of different model The figure (a) in 6.17 shows the comparison of CNN and GRU model performance on test data with 1.45×10^6 samples. Considering the angle resolution is 1° , the two models can't provide a reliable 2D DOA estimation results since the mean values are larger than 0.5° , but when compared with performance of MUSIC algorithm, GRU based model has better performance than MUSIC.

Figure (b) in 6.17 presents number of trainable parameters in each model. If the attentions are paid to both of performance on DOA estimation and trainable parameters number, it can be found in figure 6.12 that GRU based model has better performance with less trainable parameters.

Besides, figure 6.13 and 6.15 shows that training sample data size is also an important factor which influences model performance. Normally, big training data set can guarantee a better performance of model.

6.1.4 Random number targets 2D DOA angle estimation

In this section, an additional ANN model for detecting number of targets is introduced, then it can be combined with 2D DOA estimation models to implement random targets 2D DOA angle estimation. As for 2D DOA estimation models, after comparing the model performances, in last previous sections developed GRU based models for 1,2,3 targets 2D DOA estimation are chosen.

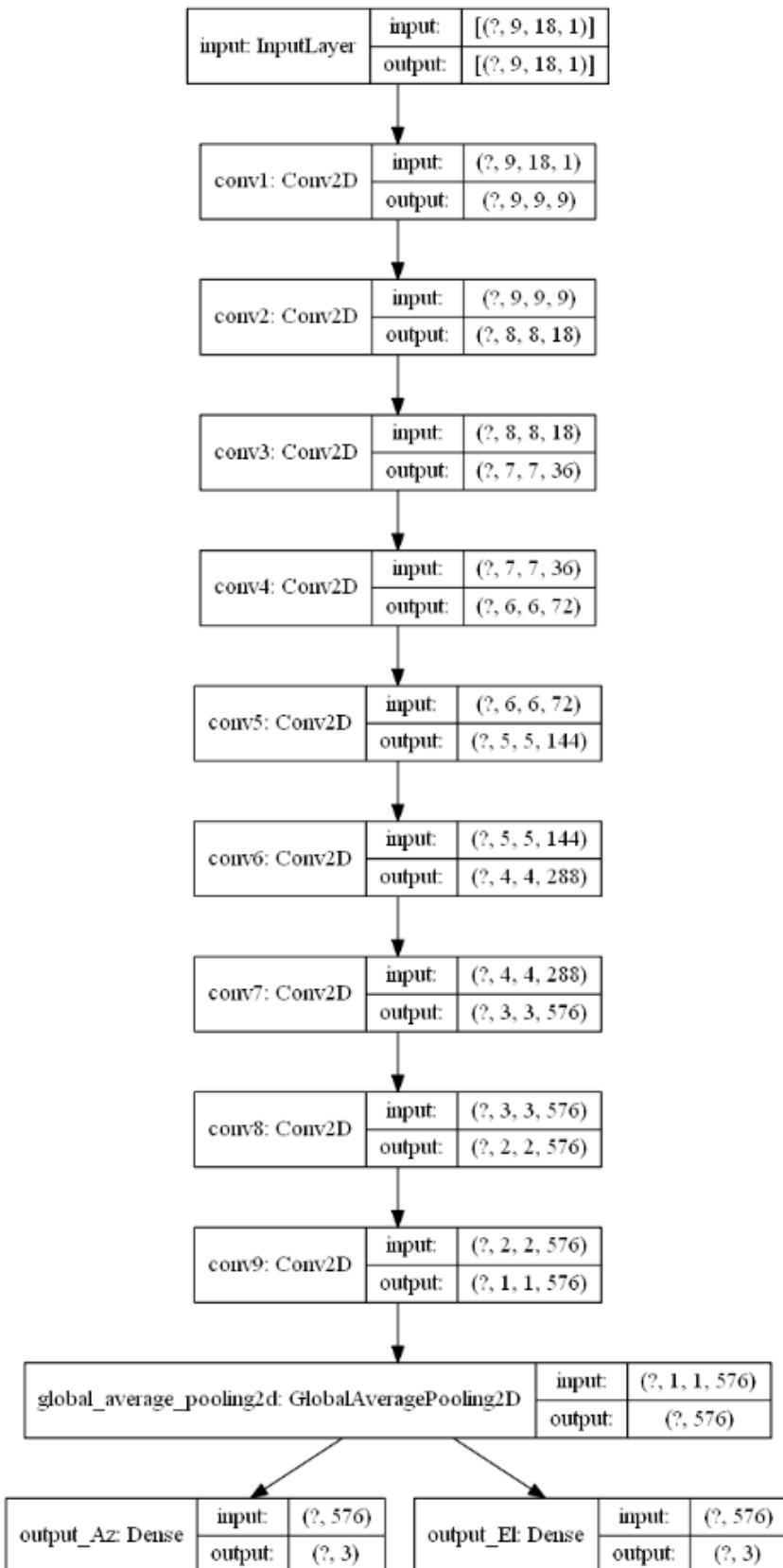


Figure 6.14: structure of CNN based three targets 2D DOA estimation model

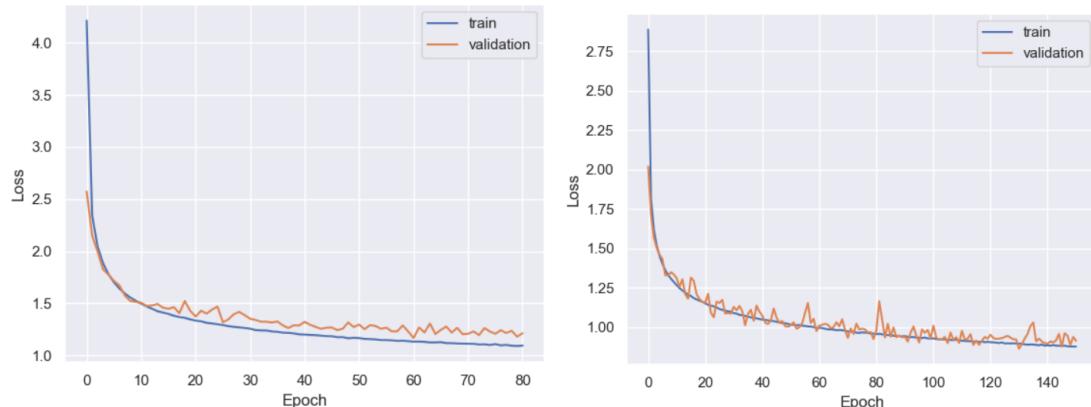
(a) learning curve under 1×10^5 training samples (b) learning curve under 1.45×10^6 training samples

Figure 6.15: comparison of GRU based three target 2D DOA estimation model performance on different training samples

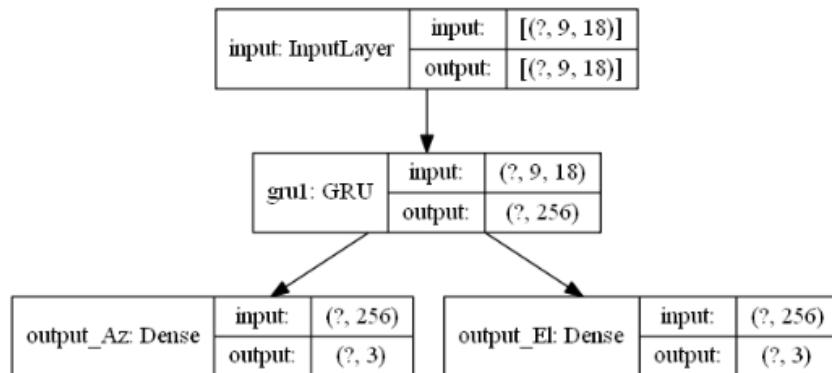


Figure 6.16: structure of GRU based three targets 2D DOA estimation model

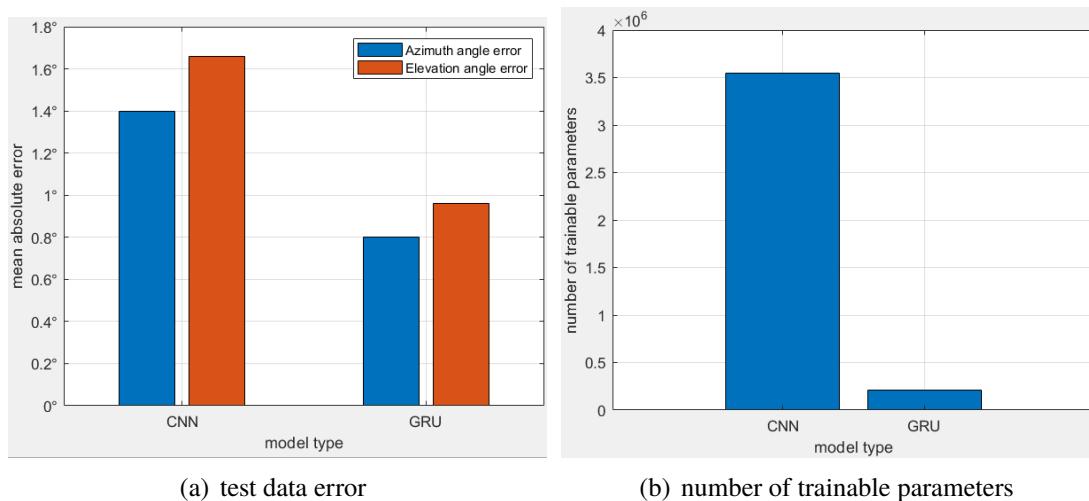


Figure 6.17: comparison of different three targets 2D DOA estimation models

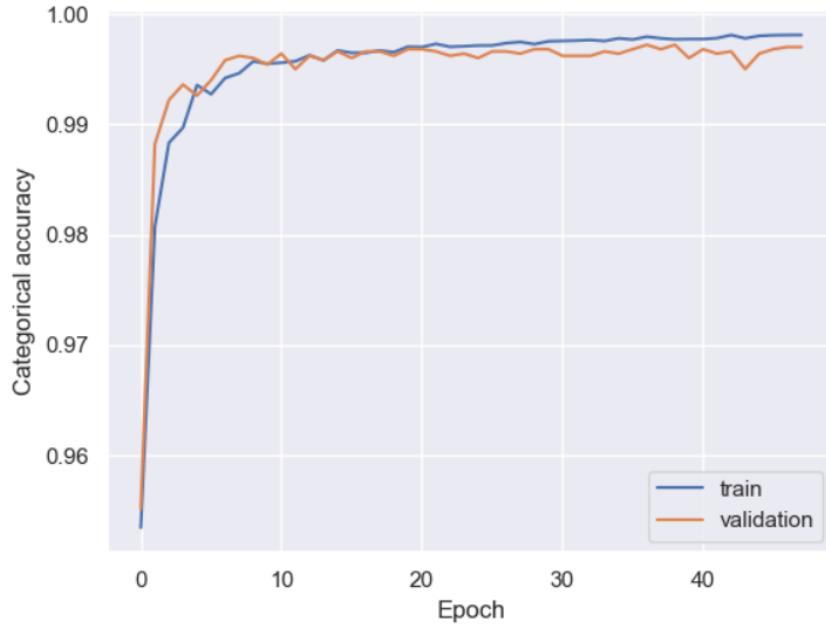


Figure 6.18: learning curve for target number detection model D_{12} in forms of epoch vs. categorical accuracy

1-2 random number targets 2D DOA angle estimation We assume that model D_{12} detects number of targets when radar data with 1-2 random targets is given , name of targets 2D DOA estimation model is M_i , where $i \in 1,2$ is detectable targets number of model.

Input data of model D_{12} is 9×18 dimensional matrix as shown in section 5, while the label is in form of one-hot-label which is also introduced in section 5, here it is assumed that one-hot-labels are $[1,0]$ and $[0,1]$ which correspond single target and double targets respectively. The input layer and hidden layers of model D_{12} have same structure as the model which is shown in figure 6.14. D_{12} has one output layer with softmax activation function which outputs 1×2 dimensional value as the one-hot-label of input data. Trainable parameter number of model D_{12} is 3542195. 1.2×10^5 samples are used for model D_{12} training, while two data sets with 5000 and 9718 samples are used for validation and test respectively. Figure 6.18 shows the learning curve of model D_{12} in forms of epoch vs. categorical accuracy, where categorical accuracy is defined by

$$CA = \frac{N_{correct}}{N_{total}} \quad (6.1)$$

where $N_{correct}$ and N_{total} mean number of correct predictions and total number of predictions respectively. In this case, if index of maximum value in outputs is equal to index of value 1 in label, the prediction is seen as correct. The categorical accuracy under 9718 random generated samples is 99.69%.

Then the process for 1-2 random target 2D DOA angle estimation is shown in figure 6.19. As the first step, the radar signal sample is inputted into model D_{12} , according to the outputs of model D_{12} , the number of targets are determined. Afterwards, an 2D DOA estimation model is chosen based on the target number determination result, model M_1 and M_2 are corresponding models for targets number 1 and 2 respectively. As final step, the radar signal sample is inputted to chosen model to output the 2D DOA angles of targets.

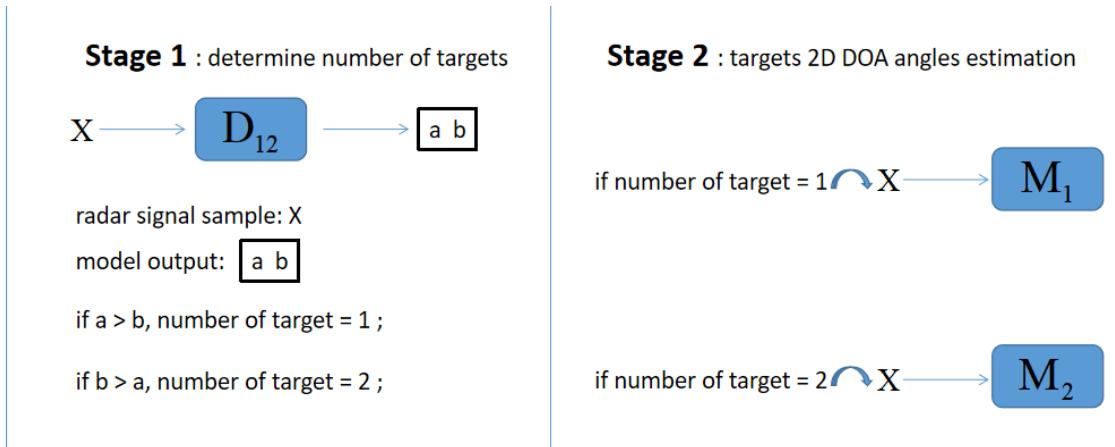


Figure 6.19: 1-2 random target 2D DOA angle estimation process

1-3 random number targets 2D DOA angle estimation We assume that model D_{123} detects number of targets when radar data with 1-3 random target is given , name of targets 2D DOA estimation model is M_i , where $i \in 1,2,3$ is detectable targets number of model.

Input data of model D_{123} is 9×18 dimensional matrix as shown in section 5, while the label is in form of one-hot-label which is also introduced in section 5, here it is assumed that one-hot-labels are [1,0,0], [0,1,0] and [0,0,1] which correspond single target, double targets and three targets respectively. D_{123} has same input layer and hidden layers structures as the model D_{12} . D_{123} has one output layer with softmax activation function which outputs 1×3 dimensional value as the one-hot-label of input data. Trainable parameter number of model D_{12} is 3542772.

3×10^5 samples are used for model D_{123} training, while other two data sets both with 1000 samples are used for validation and test. Figure 6.20 shows the learning curve in forms of epoch vs. categorical accuracy. The categorical accuracy of test data set is 96.7%.

Then the process for 1-3 random target 2D DOA angle estimation is shown in figure 6.21. As the first step, the radar signal sample is inputted into model D_{123} , the number of targets are determined based on the outputs of model D_{123} , . Afterwards, 2D DOA estimation model is chosen according to the number of targets, model M_1 , M_2 and M_3 are corresponding models for targets number 1, 2 and 3 respectively. As final step, the radar signal sample is inputted to chosen model to output the 2D DOA angles of targets.

comparison with MUSIC algorithm In order to verify the performance of ANN based random number target 2D DOA estimation, three test data set with 2500 samples for 1,2 and 3 number of targets are established respectively and 2D DOA estimation was executed using MUSIC algorithm and ANN process.

As can be found in the previous subsections 6.1.4 and 6.1.4, the performance of ANN approach is decided by two parts: accuracy of number of targets detection and accuracy of targets 2D DOA angle detection. Therefore, the comparison of MUSIC algorithm and ANN based approach is presented in these two aspect.

In 1 target 2D DOA estimation, ANN based 1-2 random target 2D DOA estimation process, 1-3 random target 2D DOA estimation process and MUSIC algorithm all have perfect performance, there are no target number misdetection, neither angle estimation error.

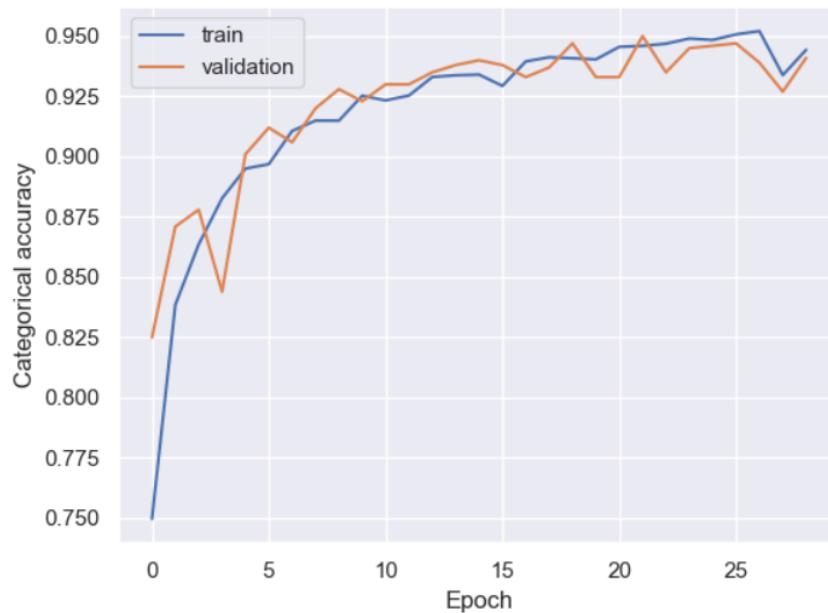


Figure 6.20: learning curve for target number detection model in forms of epoch vs. categorical accuracy

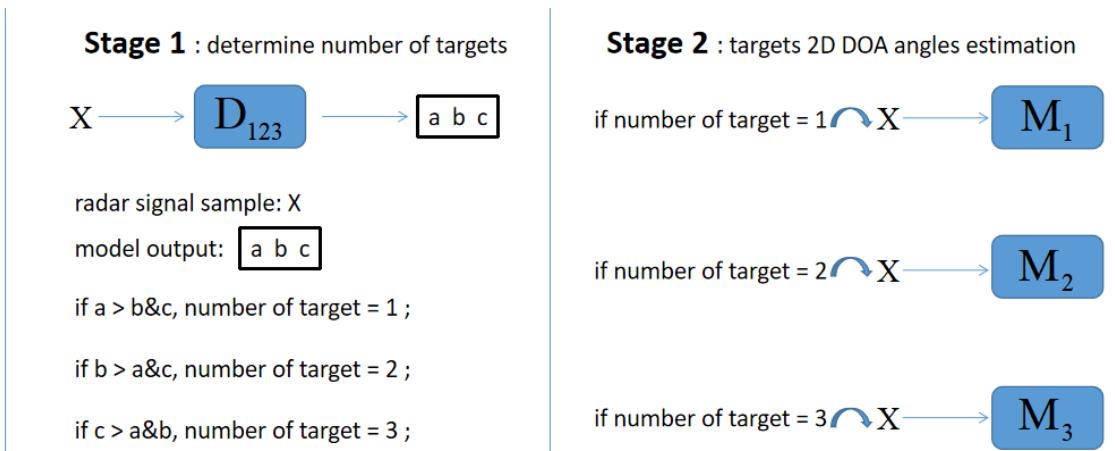


Figure 6.21: 1-3 random target 2D DOA angle estimation process

Figure 6.22 shows the comparison of three kinds of approach in 2 targets 2D DOA estimation, figure (a) shows the angle estimation mean absolute error when number of targets is correctly detected, figure (b) presents target number detection accuracy.

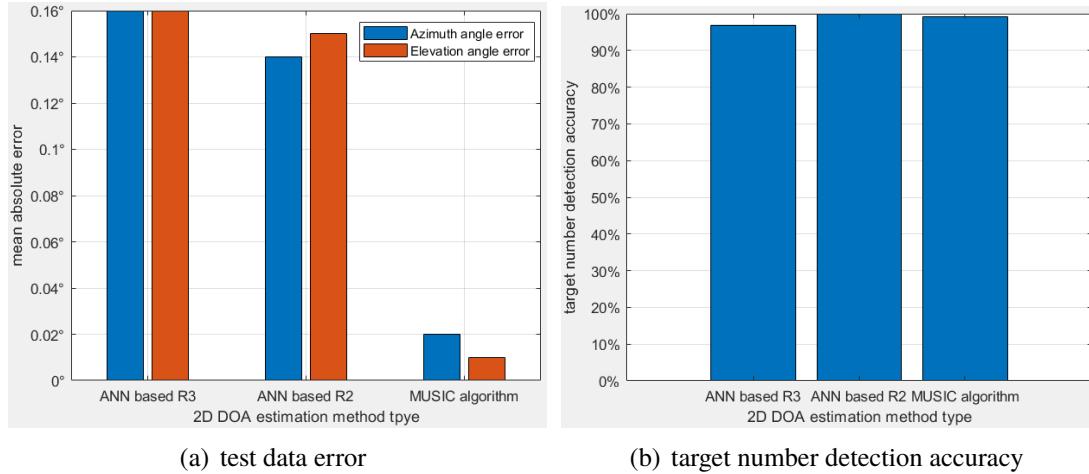


Figure 6.22: comparison of ANN approach and MUSIC algorithm on two targets 2D DOA estimation, where ANN based R3 and ANN based R2 are ANN based 1-3 random target 2D DOA estimation process and 1-2 random target 2D DOA estimation process respectively

Figure 6.23 shows the comparison of ANN based 1-3 random target 2D DOA estimation process and MUSIC algorithm in 3 targets 2D DOA estimation, figure (a) shows the angle estimation mean absolute error when number of targets is correctly detected, figure (b) presents target number detection accuracy.

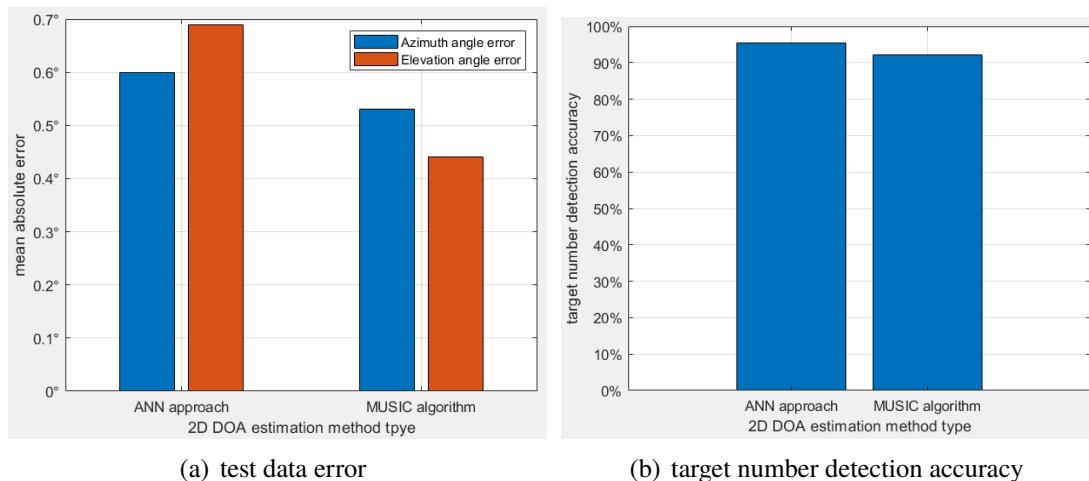


Figure 6.23: comparison of ANN approach and MUSIC algorithm on three targets 2D DOA estimation

In 1-2 random target situation, ANN based process shows a good performance in both of target number detection and angle estimation, target number detection accuracy is higher than

MUSIC algorithm with [100%, 99.84%] when that of MUSIC algorithm is [100%, 99.24%], considering the resolution of angle estimation is 1° , angle estimation mean absolute errors also stay at acceptable level.

When estimating 2D DOA angles of 1-3 random number targets , the performance is mainly limited by the situation when targets number is 3. It can be seen in the figure 6.23, both ANN model and MUSIC algorithm can't guarantee a good performance on both targets number determination and targets angle estimation, the reason can be attributed to the problem of maximum detectable targets which introduced in section 4.2, under given configuration and parameters the radar signal can't always provide enough information to drive all correct DOA angles of targets when targets number is too big.

6.2 ANN based 2D DOA angle estimation using data with one hot label

In this section, an 2D DOA angle estimation model using data with one hot label is introduced. Adam[12] optimizer is chosen for the model. Input data is 18×9 dimensional matrix which is transposed form of data that used in previous section. In terms of label, in section 5.3 introduced one hot label is used. Since one hot label can express all the situations with random number of targets, the model is directly established for random number target 2D DOA estimation. In this section, according to the result of section 4.2, model is trained for 1-3 random number target 2D DOA estimation. 247559 random generated samples are used for model training.

Figure 6.24 shows the structure of model, whose first hidden layer is one 1-dimensional convolutional layer[13], whose filter size is 2, number of filter is 128, stride is 1. Second hidden layer is GRU layer with 383 GRUs, while third hidden layer is one dropout layer[14] with dropout rate of 0.5. Model has one output layer with 2501 perceptrons which activated by sigmoid function. The model has total 1552253 trainable parameters.

A loss called focal loss[15] is used in model training in this section, which defined by

$$FL = -\alpha(1 - p_t)^\gamma \log(p_t) \quad (6.2)$$

where p_t is defined as

$$p_t = \begin{cases} p & , \quad y = 1 \\ 1 - p & , \quad y = 0 \end{cases} \quad (6.3)$$

where y is label value. It can be said that p_t represents the current accuracy of corresponding output. Parameters α and γ are set as 0.25 and 2 respectively. Since the focal loss value can't represent the accuracy of training result, the learning curve is not presented here.

comparison of ANN approach and MUSIC algorithm To verify the performance of ANN model, radar signal is generated randomly for 1-3 targets and processed by ANN model and MUSIC algorithm at the same time, figure 6.23 shows the 2D DOA estimation result of ANN model and MUSIC algorithm for the randomly generated samples. From figure 6.23 it can be known that the model has the basic ability to estimate 2D DOA angles of random number target.

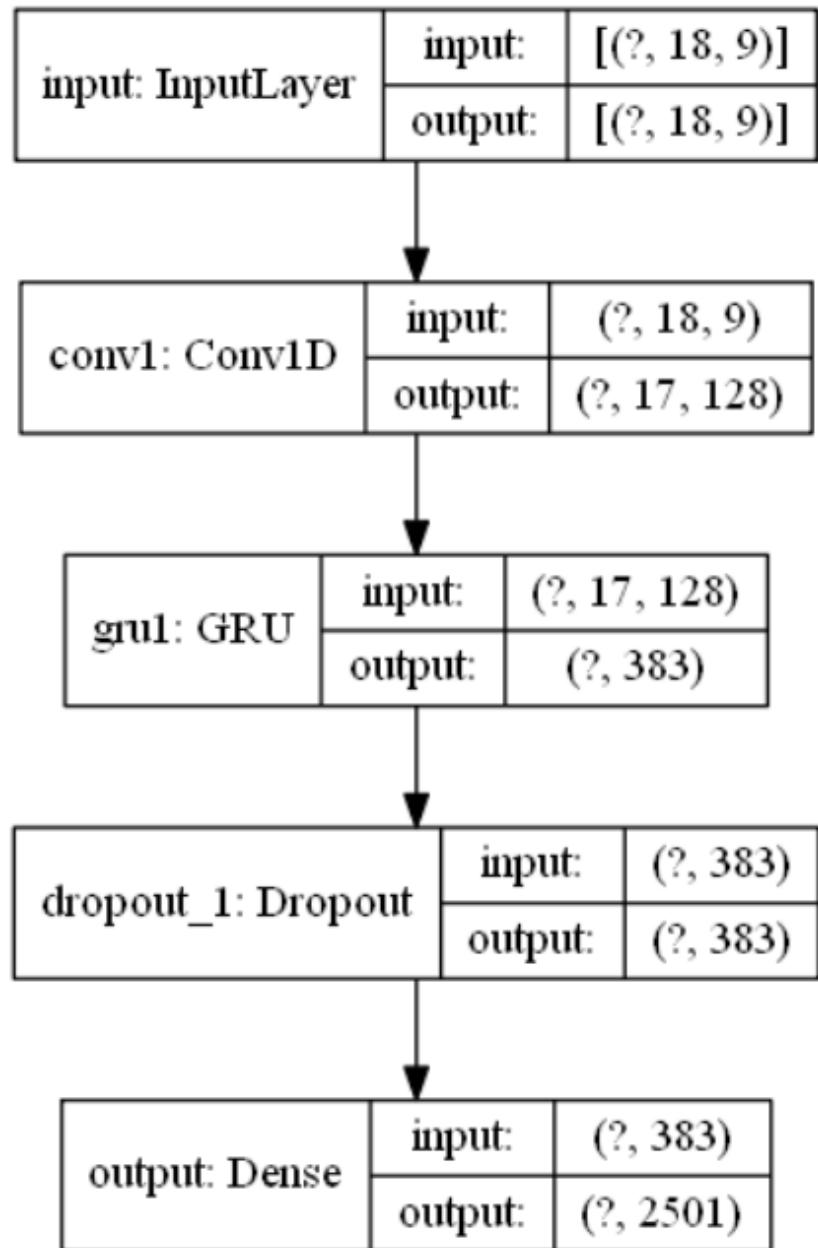


Figure 6.24: structure of random number target 2D DOA estimation model in section 6.2

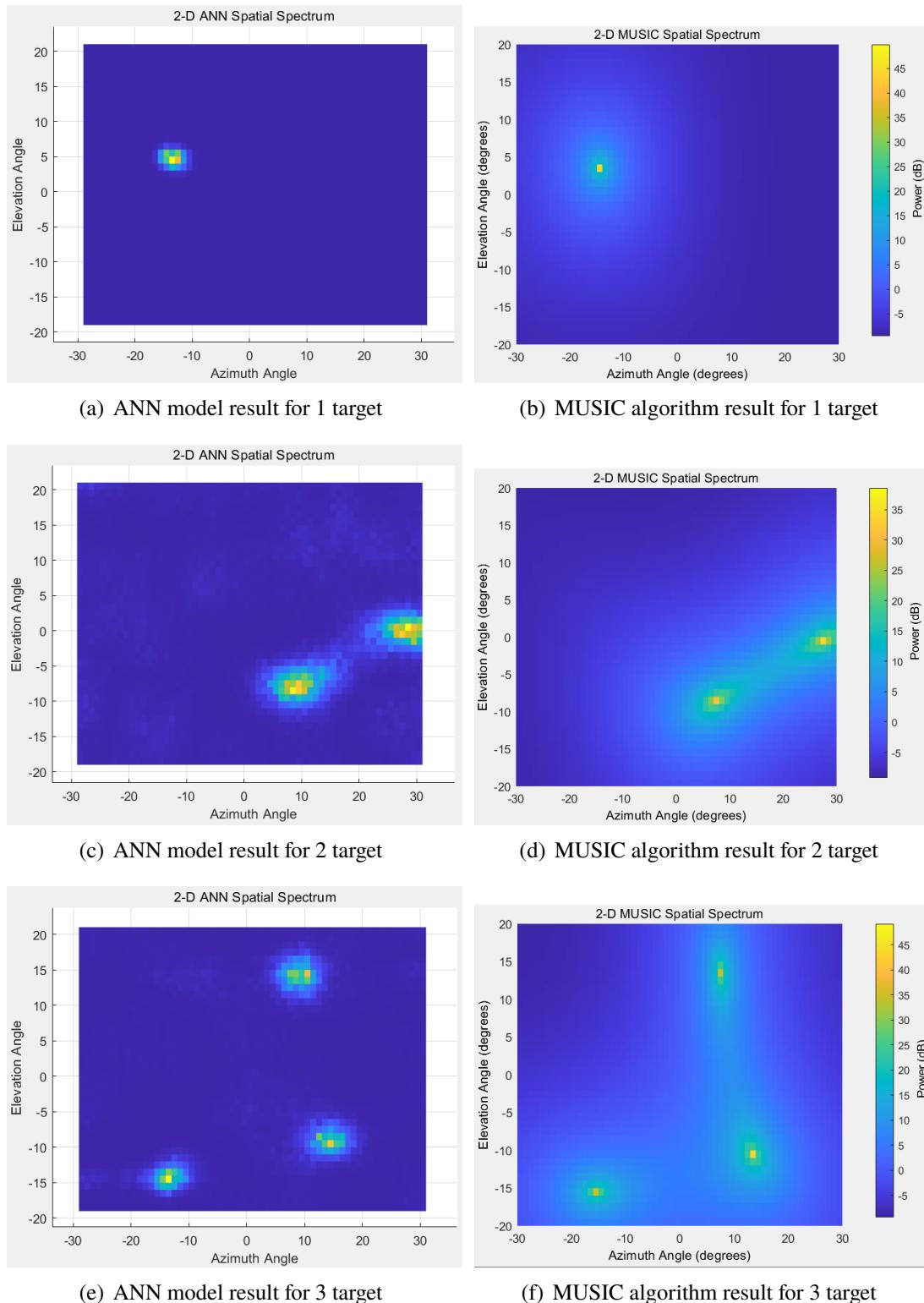


Figure 6.25: comparison of one hot labeled data based ANN approach and MUSIC algorithm on 1-3 random targets 2D DOA estimation

comparison of ANN approaches with one hot labeled data and value labeled data

The accuracy of angle estimation is low than that of ANN model with value number, especially when targets are closely located with each other.

But the advantages of this approach are also obvious.

Firstly, this approach only use one ANN model with 1552253 parameters to implement 2D DOA estimation while method of previous section need to combine 4 ANN models with at least 3542772(for target number determination) + 16258(single target 2D DOA angle estimation) parameters when estimation 1-3 random target DOA angles, which means one hot label based approach can be much faster than value label based approach when implement DOA estimation.

Secondly, when considering expanding the detectable number of target, in value label based approach not only need to reconstruction target number detection model but also have to train and add a new 2D DOA estimation model for new larger number of target. But one hot label based approach only need to add the data samples in the old training data and retrain the model.

Thirdly, in principle of MUSIC algorithm it can be found that the 2D DOA principle doesn't change as the number of target changes, in this aspect it can be said that one hot label based approach used data more efficiently, since the value label based approach trains data separately according to the number of targets while the another one trains all data together.

It can be said that value label based ANN approach has better performance at present, but one hot label based approach has better potential in 2D DOA angle estimation.

7 Summary and Outlook

As the first step of the work, a phased array system toolbox of MATLAB based radar simulator is established to simulate radar data for ANN training and testing. Afterwards, different type ANN based 2D DOA angle estimation models are implemented under given radar configuration and simulation environment using simulated radar signal samples. As final step, 2D DOA angle estimation accuracy of ANN approaches are compared with that of traditional high-performance algorithm MUSIC so that to verify performances of the ANN approaches. In this article, two kinds of ANN approaches are presented and is proved to have at least similar performance as MUSIC algorithm.

The first work of article is presented in section 4. In first part of section, radar simulator and MUSIC algorithm estimator is established under given simulation parameters and radar configuration using phased array system toolbox of MATLAB. In the second part, the maximum detectable number of target when using single chirp radar signal under given simulation environment is explored using statistical method and MUSIC algorithm. Since the MUSIC algorithm shows acceptable performance only when target number is under 4, it is decided to implement 1-3 random target 2D DOA estimator using ANN approach under given simulation environment in the next works.

As to specific simulation environment, a 77GHz FMCW radar signal signal with 1GHz bandwidth is used, where chirp duration is $288\mu\text{s}$, sample rate is 256/s. Half wavelength spacing 3x3 rectangular receiver array antenna and 1 transmitter antenna are used to as signal receiver and transmitter where noise figure is set to 4.5dB and wave speed is light speed. Azimuth and elevation angle detect range is $[-30^\circ, 30^\circ]$ and $[-20^\circ, 20^\circ]$ respectively with resolution of 1° . Target range is limited in 2-5m, RCS of targets is 0.04m^2 .

The second work in the article is 2D DOA estimation using ANN. In this article, there are two kinds of ANN approaches are presented, they are value labeled data based ANN approach and one hot labeled data based ANN approach.

In value labeled data based ANN approach, the 2D DOA angle estimation process is composed of two part. The first part is target number detecting, radar signal sample should inputted to target number detection model to find out the number of targets. The second part is target 2D DOA angle estimation, 2D DOA angle estimation model is chosen according to the target number determination result.

There are two kinds of value labeled data based ANN approaches are developed, they are 1-2 random target 2D DOA estimation process and 1-3 random target 2D DOA estimation process. In 1-2 random target 2D DOA estimation process, both the value labeled data based ANN approach and MUSIC algorithm show nearly perfect performance with little differences in accuracy in target number detection and angle estimation, in target number detection ANN approach shows better performance while in angle estimation MUSIC algorithm has better accuracy.



Figure 7.1: depth image generated from optical image [16]

In terms of 1-3 random target 2D DOA estimation process, in single target detection it also shows perfect performance in target number detection and target DOA angle estimation, in double targets number detection shows worse performance than both 1-2 target number detection process and MUSIC algorithm, but shows better performance in 3 targets number detection than MUSIC algorithm. As to angle estimation accuracy, it shows similar performance as 1-2 random target 2D DOA estimation process in 2 target angle 2D DOA estimation, and similar performance as MUSIC algorithm in 3 target angle 2D DOA estimation.

As for one hot labeled data based ANN approach, it shows similar performance as MUSIC algorithm as shown in figure 6.25. One obvious advantage of this approach over value labeled data based ANN approach is that it is composed of only one ANN model with much less parameters which guarantee faster processing speed than another one approach. Another advantage is that when considering expanding the detectable number of target, this kind of approach only need to add the new data samples in the old training data and retrain the model while the another approach not only need to reconstruction target number detection model but also have to train and add a new 2D DOA estimation model for new larger number of target.

The potential of ANN approach in radar target 2D DOA angle estimation is proved by the work in this article, the 2D DOA estimation performance of ANN approaches is similar with that of MUSIC algorithm, which is proved to have outstanding performance among all the high-resolution traditional algorithms.

As for further work which can be done based on result of this article, radar object classification or depth image generation can be considered. In this article, all the targets are seen as point with specific RCS, but in the practice, all the objects has their geometry shape which can reflect different radar signal with specific features back to radar receive sensors, by capturing the radar signal features of different objects, object classification using radar can be possible.

Since radar signal can provide range information and 2D DOA angle information of targets, it can be tried to build up the depth image of radar targets using ANN approach. At present, one hot labeled data based ANN approach provides image which can represent possibility of target existence on every pixel. In the practice, targets exists on every pixel of image but with different range which is similar with image of camera. When it is assumed that there are targets with different range on every pixel in 2D DOA estimation radar image as showed in figure 6.25, and the 2D DOA estimation and range estimation is done at the same time for all the pixel on image, if let the color of radar image show the distance of target, the depth image can be established which is looked like in figure 7.1.

Bibliography

- [1] A. J. Barabell, “Performance Comparison of Superresolution Array Processing Algorithms. Revised”, *Massachusetts Inst of Tech Lexington Lincoln Lab.*, p. 55, 1998 (cit. on p. 1).
- [2] B. M. Marija Agatonovic, “High resolution two-dimensional DOA estimation using artificial neural networks”, in *2012 6th European Conference on Antennas and Propagation (EUCAP)*, IEEE, Jun. 2012 (cit. on p. 2).
- [3] N. J. G. Fonseca, “On the Design of a Compact Neural Network-Based DOA Estimation System”, in *IEEE Transactions on Antennas and Propagation*, IEEE, Dec. 2009 (cit. on p. 2).
- [4] M. Agatonovic, “Hybrid ANN model for accurate 2D DOA estimation of a radiating source”, in *TELSIKS*, IEEE, Jan. 2014 (cit. on p. 2).
- [5] B. R. Mahafza, “Introduction to radar analysis”. CRC Press, May 1998 (cit. on p. 13).
- [6] M. D. Zeiler, “On rectified linear units for speech processing”, in *International Conference on Acoustics, Speech and Signal Processing*, IEEE, May 2013 (cit. on p. 19).
- [7] D. M. Hawkins, “The Problem of Overfitting”, *Journal of Chemical Information and Computer Sciences*, vol. 44, pp. 1–2, Dec. 2003 (cit. on p. 22).
- [8] I. Hadji and R. P. Wildes, “What Do We Understand About Convolutional Networks?”, cite arxiv:1803.08834, 2018 (cit. on p. 27).
- [9] Y. L. ; K. K. ; C. Farabet, “Convolutional networks and applications in vision”, in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, IEEE, Aug. 2010 (cit. on p. 28).
- [10] S. Rajaram, P. Gupta, B. Andrassy, and T. Runkler, “Neural Architectures for Relation Extraction Within and Across Sentence Boundaries in Natural Language Text”, Apr. 2018 (cit. on p. 29).
- [11] “GRU units”, https://cran.r-project.org/web/packages/rnn/vignettes/GRU_units.html, Accessed: 2010-09-30 (cit. on pp. 30, 31).
- [12] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization”, *CoRR*, vol. abs/1412.6980, 2015 (cit. on pp. 41, 57).
- [13] Y. Zhang, Y. Miyamori, S. Mikami, and T. Saito, “Vibration-based structural state identification by a 1-dimensional convolutional neural network”, *Comput. Aided Civ. Infrastructure Eng.*, vol. 34, pp. 822–839, 2019 (cit. on p. 57).
- [14] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting”, *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014 (cit. on p. 57).
- [15] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, “Focal Loss for Dense Object Detection”, *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017 (cit. on p. 57).
- [16] K. Lasinger, R. Ranftl, K. Schindler, and V. Koltun, “Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer”, *ArXiv*, vol. abs/1907.01341, 2019 (cit. on p. 62).

List of Figures

1.1	autonomous driving sensors	1
2.1	environment perception in autonomous driving	3
3.1	FMCW signal in terms of amplitude-time	5
3.2	FMCW signal in terms of frequency-time	6
3.3	radar signal block diagram	6
3.4	Transmitted Tx signal and received Rx signal from 3 targets in different range	8
3.5	IF signal for 3 targets in different range	8
3.6	2 continuous chirp signal and Rx signal from 3 targets in different range	9
3.7	Phase changing of 3 target signals over chirps	10
3.8	Radar signal in matrix form	12
3.9	Radar signal after FFT	12
3.10	range doppler map of 3 targets	13
3.11	single target signals arrive on different Rx antenna elements	14
3.12	propagation range differences Δr of single target signals with DOA angle α for different Rx antenna elements with element spacing d	14
3.13	signal phase shift of K th target in azimuth and elevation domain, where ak and bk correspond elevation and azimuth angle respectively	15
3.14	structure of an perceptron	18
3.15	activation functions with input value range $[-10,10]$	18
3.16	structure of a multilayer perceptron	20
3.17	an example of 2 dimensional convolution with 3 kinds of padding	24
3.18	an example of 2 dimensional convolution with 3 kinds of padding	25
3.19	an example of cross-correlation process with intuitive graphical explanation	25
3.20	an example of pooling layer upsampling	27
3.21	structure of a common convolutional networks [9]	28
3.22	A recurrent neural network and the unfolding in time of the computation involved in its forward computation [10]	29
3.23	GRU feedforward with sequential input data	29
3.24	internal structure and data feedforward of GRU [11]	30
3.25	internal structure and error backpropagation of GRU [11]	31
4.1	radar receiver array geometry and size	34
4.2	Transmitted FMCW signal, whose bandwidth is 1GHz, sweep time is $288\mu s$	34
4.3	Mean absolute error of MUSIC algorithm detected 2D DOA target angles, where sample number of each stage is 10000	36
5.1	this figure explains how a complex matrix is convert into real matrix in this article	37
5.2	comparison of learning curve when input data is normalized and not normalized	38

5.3	comparison of learning curve between different value label	39
5.4	one hot label plan under classification mission with 3 classes	40
6.1	structure of fully connected neural network based single target 2D DOA estimation model	42
6.2	learning curve of fully connected neural network based single target 2D DOA estimation model	42
6.3	structure of fully connected neural network based single target 2D DOA estimation model	43
6.4	learning curve of CNN based single target 2D DOA estimation model	44
6.5	structure of fully connected neural network based single target 2D DOA estimation model	44
6.6	learning curve of CNN based single target 2D DOA estimation model	45
6.7	comparison of different single target 2D DOA estimation models	45
6.8	learning curve of CNN based double target 2D DOA estimation model	46
6.9	structure of CNN based double target 2D DOA estimation model	47
6.10	learning curve of GRU based double target 2D DOA estimation model	48
6.11	structure of GRU based double target 2D DOA estimation model	48
6.12	comparison of different double targets 2D DOA estimation models	49
6.13	comparison of CNN based three target 2D DOA estimation model performance on different training samples	50
6.14	structure of CNN based three targets 2D DOA estimation model	51
6.15	comparison of GRU based three target 2D DOA estimation model performance on different training samples	52
6.16	structure of GRU based three targets 2D DOA estimation model	52
6.17	comparison of different three targets 2D DOA estimation models	52
6.18	learning curve for target number detection model D_{12} in forms of epoch vs. categorical accuracy	53
6.19	1-2 random target 2D DOA angle estimation process	54
6.20	learning curve for target number detection model in forms of epoch vs. categorical accuracy	55
6.21	1-3 random target 2D DOA angle estimation process	55
6.22	comparison of ANN approach and MUSIC algorithm on two targets 2D DOA estimation, where ANN based R3 and ANN based R2 are ANN based 1-3 random target 2D DOA estimation process and 1-2 random target 2D DOA estimation process respectively	56
6.23	comparison of ANN approach and MUSIC algorithm on three targets 2D DOA estimation	56
6.24	structure of random number target 2D DOA estimation model in section 6.2	58
6.25	comparison of one hot labeled data based ANN approach and MUSIC algorithm on 1-3 random targets 2D DOA estimation	59
7.1	depth image generated from optical image [16]	62