# Radar's Direction of Arrival Estimation
# 5AUA0 Final report Group24


Ji Jiaqi
1877879
j.ji@student.tue.nl

Yang Bo
1796372
b.yang2@student.tue.nl



## Abstract

*DOA(direction of arrival) estimation is vital for radars to detect surrounding targets in autonomous driving industry. This report proposes a solution for the estimation of number of sources together with their corresponding DOAs. Especially, we focus on the case where the observation has only one snapshot. Neural networks are combined with maximum likelihood estimation (MLE) to obtain a good results and save computation resources as well. The solution is tested under different scenarios and shows good performance for signals with low level of noise.*


## 1. Introduction

In recent years, as autonomous driving technology has developed rapidly, there has been a growing interest in related technologies. One such technology is radar, which plays a crucial role in target recognition for autonomous vehicles. Modern cars have several radars that are used to improve driver experience, for instance, in providing parking assistance. These radar sensors are placed at different positions on the vehicle to gather information about objects around the car. Besides, driven by the progress in deep learning(DL) technique, empowering radar DOA estimation with deep learning has gradually become a research focus. [1]

In our situation, we need the model to make predictions with one snapshot. So that the result can instantly reflect latest changes. For single snapshot, neural network has better performance than classic methods. [2] So in the report, we explored the usage of multilayer perceptrons(MLP) for estimating the number of sources and the direction of arrival(DOA), which refers to the angle from which a signal or wave arrives at a sensor array. In the situation that there are M antennas uniformly separated (ULA), with the distance between two antennas being d, and all N sources are far-field, then the received signal x can be expressed as:

$$x = A(\theta) \cdot s + \eta \tag{1}$$

s is the signal emitted by the sources, $\eta$ is the noise, A($\theta$) is steering matrix, can be expressed as:

$$A(\theta) = \begin{bmatrix} e^{j2\pi\lambda^{-1}d\theta_1} & ... & e^{j2\pi\lambda^{-1}d\theta_N} \\ ... & ... & ... \\ e^{j2\pi\lambda^{-1}M\theta_1} & ... & e^{j2\pi\lambda^{-1}M\theta_N} \end{bmatrix} \tag{2}$$

in which $\lambda$ is the radar's wavelength, d = $\lambda$/2.

Our goal is to propose a solution, such that given received signals with only 1 snapshot, the method can identify the number of sources detected, together with their DOAs. We will also try to make the solution robust against noises and capable of handling more complex situations(more sources). We will test our method with signals having different signal-to-noise ratios(SNR)

We investigate the opportunities of combining neural networks with MLE for DoA estimation. Two MLPs with different structures are utilized for estimating the number of sources and DOA estimation, respectively.

Our results have indicated that machine learning can be an attractive approach for the existing challenges of traditional algorithms, such as multiple signal classification (MUSIC) [3], estimation signal parameter via rotation (ESPRIT) [2], which tend to be computationally expensive and exhibit performance degradation in the presence of noise and reverberation.

## 2. Related work

We gained inspiration from bialer's work. [4] this paper used 8 fully connected layers as the neural network, output the DOA estimation and number of sources separately. On the basis of it, we improved the network structure, and used another MLP to estimate the number of sources; After building the backbone of our architectue, we think perhaps use more complex neural networks and import convolutional layers will make the performance better, and Fang's work [5] used ResNet50 to achieve DOA estimation. When we use ResNet18 to improve our network, we found it hard to train and convergence, the loss didn't decrease anymore after 200 epochs. However, this paper proposed a input

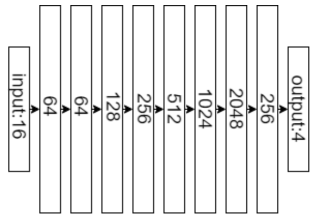



Figure 1. MLP architecture for classification

layer, it used the snapshot to form it's covariance matrixusing its real and imaginary parts as 2 channels. We adopt this method to form our inputs.

# 3. Method

In general, two different Multi-layer Perceptrons(MLP) were used separately for the classification of the number of sources and the estimation of DOAs.

Rogers [6] presented a deep-learning-based approach with 15 layers that can achieve an accuracy of 82 percent for estimating the number of sources. However, in our experiment, simply increasing the depth of the network showed little contribution to the performance in classification.

After several tests in different structures, we decided to use a lighter MLP for estimating the number of sources. Upon passing the inputs to the classifier, based on its output(number of sources) we pass the inputs to different pretrained MLPs with the same architecture to estimate doas.

## 3.1. Classification of Sources

The Classification part is relatively easy compared to the estimation part. It can simply be identified as a classification problem with 4 possible classes. Many networks have been proposed to solve similar classification tasks. Our MLP architecture for this task is shown in 1.

In our task, since we have 8 antennas, one single observation we received should be a vector with a length of 8, and each of the elements should be a complex value. The networks do not process complex values, so that we separated the imaginary parts and the real parts of the observation and reshape them into a new vector with a length of 16. The first 8 contains the real parts and the last 8 elements are corresponded to the imaginary parts.Which can be the input of our structure.

The network is composed by 8 full-connected layers. Each of the layer is followed by a batch-normalization layer and a Relu activation function. The result will be pushed through another full-connected layer to produce the 4 outputs. Softmax function will be applied to these 4 outputs so that each of them can represent the probability of the number of the sources is in between 1 to 4.

We will use cross-entropy as our loss function, which is a classic loss function for classification problems. The formula of this function can be expressed as:

$$H(x) = -\sum_{i=1}^{n} p(x_i) log(p(x_i)) \tag{3}$$

Where $p(x_i)$ is the probability for ith class.

## 3.2. Estimation of DoAs

Like paper [4], usually this problem can be considered as a regression problem: Using networks to predict the DOAs for the corresponding sources. After several experiments, we found that, if we use MLPs to directly predict the DOAs, the results will deteriorate rapidly when the SNR of the observations becomes low. Besides, when the SNR is high, the network will easily over-fit, producing bad results in validation set.

paper [7] gave us some insights about the possibility of considering the estimation of angles as a classification as well. However, in our experiment, our MLP had a poor performance when using cross-entropy loss function to classify the angles.

Alternatively, we decided to use our MLP to predict the probability distribution of the DOAs. This is proved to be a more effective way. One possible explanation for this is that compared to the angles themselves, the probability distribution has more characteristics that can be easier to be learned by the networks. In our case, the doas are between -70 to 70 degrees, with a resolution of 1 degree. For one doa, there are 141 possible values. Thus, when training our networks, we will transform the target doa into a vector filled with zeros with a length of 141, and set the elements in the position corresponding to the angles to 1. This operation is similar to n-hot coding. It is still a regression problem, but now the network will learn the probability distribution and have 141 outputs corresponding to 141 possible angles. Further post-processing is necessary to produce the actual angles.

Our network responsible for this part is shown in 2. This structure is composed with 7 fully-connected layers, and each layer is followed by a batch-norm layer and a Relu activation function, just like the classification part. The difference is, the 141 outputs will go through a Sigmoid activation function to make the values in range (0,1). For that they represent the probabilities for angles.

To make the input have more information to learn, we calculate the covariance matrix of a given observation X. For one obersvation X with one snapshot, complex values and a size of $1 \times 8$, the covariance matrix can be approximated by:

$$R = X^H X \tag{4}$$

Where $X^H$ is the Hermitian transpose of $X$. In our case, the size of the covariance matrix should be $8 \times 8$. We also noted that the covariance matrix is symmetric, so we can only extract the upper triangular part of the matrix, separate

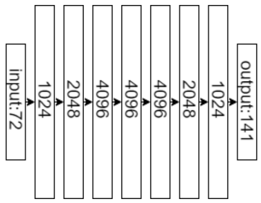



Figure 2. MLP architecture for estimation

the real parts and imaginary parts,(similar to the classification part), construct a new vector with these elements. The size of this new vector should be $1 \times 72$.

We chose root Mean Square Error as our loss function.

#### 3.2.1 Post processing

To evaluate the performance of the angle estimation, we will use root mean square error(RMSE) as our metric. RMSE can be computed by:

$$\text{RMSE} = \frac{1}{L}\sum_{l=1}^{L}(\sqrt{\frac{1}{K}\sum_{k=1}^{K}(\hat{\theta}_{k,l} - \theta_k)^2}) \tag{5}$$

Where K is the number of antennas, L is the batch size, $\hat{\theta}$ is the predicted angle and $\theta$ is the corresponding truth.

Upon acquiring the predicted probability distribution, we can sort these angles with the order of their probabilities. And select the ones with the highest probabilities as our prediction for the angles. The drawback of this method is that, for example, suppose we have 3 sources, and the true angle is [6, 35, 70]. While the angles with the highest 4 probabilities are actually [6, 7, 35, 70]. This can be possible, in which case the probability of 70 is high, but due to insufficient fitting of the probability distribution, the probabilities of 6 and 7 are both higher than that of 70. If we just output the first 3 angles as our prediction, which is [6, 7, 35], then the RMSE for this prediction would be very large.

To compensate for that, instead of directly choosing the first 3 angles as our prediction, we choose the first 6 angles. Then we exploit Maximum Likelihood Estimation(MLE) algorithm to find the best 3 predictions from these 6 angles.

#### 3.2.2 Maximum Likelihood Estimation

Maximum Likelihood Estimation is not a deep learning algorithm. It just searches in all combinations of angles for the one that minimizes the residual error, which can be mathematically expressed as:

$$\varepsilon(\theta) = x - \widetilde{A}(\theta) \cdot (\widetilde{A}^{+}(\theta) \cdot x) \tag{6}$$

And finding the minimal residual error can be expressed as:

$$\widetilde{\theta} = \underset{\theta}{argmin} \left\|\varepsilon(\theta)\right\|^2 \tag{7}$$

Where $\widetilde{A}^{+}(\theta)$ is the pseudo-inverse matrix of $\widetilde{A}(\theta)$, and $\widetilde{A}(\theta)$ is the estimation of steering matrix $A(\theta)$, which is stated in equation(2). The desired angles $\widetilde{\theta}$ shall be the solution of equation(7).

The drawback of this algorithm is that it will become extremely expensive in computation when trying to estimate 3 or more sources. The reason is that the number of combinations will grow exponentially as the number of sources increases. For instance, the visible region is from -90 to 90 degrees with a resolution of 1. For 3 sources, one source has 181 possible angles. Thus 3 sources will have $181^3$ combinations.

However, we can reduce the number of possible combinations, as stated in 3.2.1, we choose the 6 angles with the highest probabilities, and then we make MLE only search for an optimal solution in these 6 angles. This step can greatly reduce the computation time for MLE and increase our accuracy in estimation.

## 4. Experiments

### 4.1. DataSet

Based on the assumption that the signal is narrow-banded, far-field, and our array is ULA, the observation of the data in this problem can be calculated using a suitable process: Denote M as number of angles, N as data size, means the number of observations to create,L as coordinate vector of the antennas and P as the number of antennas. A is a $N \times M$ matrix filled with First we calculate the 3-D matrix: x, whose size should be $N \times M \times P$

$$x[:,:,i] = Ae^{\frac{1j(2\pi fcL[i]dsin(\frac{\pi a}{180}))}{C_0}} \cdot e^{\frac{1j\cdot 2\pi r}{C_0}} \tag{8}$$

Where $r$ is a random number for creating variations, $a$ is the DoA corresponding to the snapshots,$fc$ is the frequency of the signal and $c_0$ is speed of propagation of the wave. Then we perform the summation operation alone the 2nd dimension of this matrix and get the snapshot, whose size is $N \times P$.

By adjusting the angles and the data size, we can create sufficient data for the purpose of training and evaluation. Similarly, white Gaussian noise can be added to the snapshots with variable SNRs (signal-to- noise ratio) so that the data can be more comprehensive.

### 4.2. Experiment

The DOAs were set to integer numbers ranging from -70 to 70 degrees. For classification, We mixed data from 7 SNRs (0,5,10,15,20,25,30) and 4 sources, with 409600 samples in each case, and regenerate 40960*28 samples as test set. While for DOA estimation part, for each number of sources(1 to 4), we generated 409600 samples, including 7 SNR scenarios. And when evaluating, we used test sets we generated and Monte-Carlo trial to see its performance.

# 5. Results and Discussion

## 5.1. Classification

Our MLP's performance over the test set stated in 4.2 is included in 3. It shows that the correct rate of the classification increases rapidly when SNR is between 0 and 10, and grows slower as the SNR keeps increasing. The accuracy eventually reaches a stable state (close to 100%) for SNRs in 20dB and higher. We can conclude that the noise can be a vital factor that influences the performance. Although the training data is comprehensive and huge, the network's ability to deal with noise is still limited. It is possible that fine-tuning parameters and modifying the structure can improve the result. But we think it would be wise to introduce some techniques to pre-process and denoise the data, and then feed it to the network. For example, some denoising algorithms like ISTA, might be possible to be modified to apply in our case.

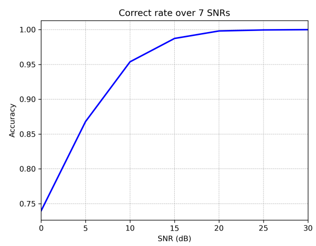
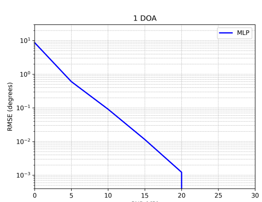


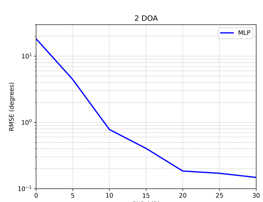

Figure 3. Classification performance

## 5.2. Estimation of DOA

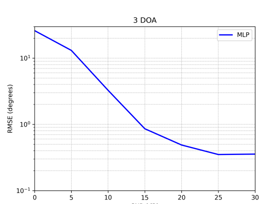
4 shows the performance of our MLP on the test set stated in 4.2 Similarly in classification case, noise can still be a vital factor for estimation performance. RMSE is relatively high when SNR is close to 0, indicating large estimating errors. The performance increases rapidly as SNR goes up, and reaches a stable state when the SNR is close to 30. The number of sources to estimate is another factor affecting performance. Generally, the MLP has better performance when estimating fewer sources. Besides, when SNR is high, our network shows decent accuracy with RMSE close to 0. This is a good result, contributed by both the post processing method in 3.2.1 for excluding false predictions and predicition in probability distributions, when can fit the target better, compared to predicting angles directly.



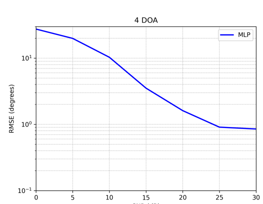

(a) 1 DOA




(b) 2 DOA



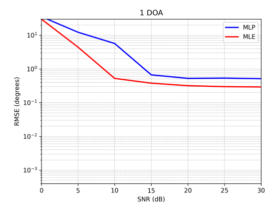

(c) 3 DOA



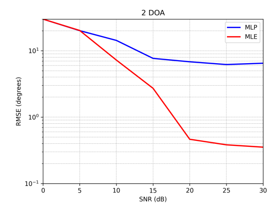

(d) 4 DOA

Figure 4. DOA estimation on test set.

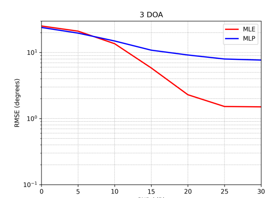
We also performed Monte-Carlo trials with our trained MLP under similar situations, The performance of MLE was also recoded. The result is shown in 5 Our network shows similar performance to MLE when SNR is low. And is outperformed by MLE when SNR goes up. But our network still shows advantages in computation time. The overall performance is much worse than that in 4, because Monte-Carlo trial introduces non-integer numbers as DOAs, while our network is trained on integer angles. One possible solution is to expand our training data with DOAs composed by non-integer numbers.



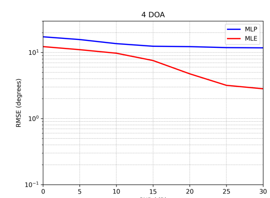

(a) 1 DOA




(b) 2 DOA




(c) 3 DOA




(d) 4 DOA

Figure 5. DOA estimation Monte-Carlo trial.

# 6. Conclusion

In this report, we proposed two MLPs to perform estimation the number of sources and DOAs separately. When estimating the DOAs, we combined our network with MLE to achieve better results. We can say that the network has a decent performance for integer DOAs and relatively low SNRs. Compared to regular MLE, our solution is much faster and maintains a decent accuracy. In this case, the problem is solved. However, for complex situations, where SNR is low and DOAs have non-integer values, the performance of our networks needs to be improved.

## References


[1] Jonas Fuchs, Robert Weigel, and Markus Gardill. Single-snapshot direction-of-arrival estimation of multiple targets using a multi-layer perceptron. In *2019 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, pages 1–4. IEEE, 2019. 1

[2] Richard Roy and Thomas Kailath. Esprit-estimation of signal parameters via rotational invariance techniques. *IEEE Transactions on acoustics, speech, and signal processing*, 37(7):984–995, 1989. 1

[3] Njeri P Waweru, Dominic Bernard Onyango Konditi, and Philip Kibet Langat. Performance analysis of music, root-music and esprit doa estimation algorithm. *International Journal of Electronics and Communication Engineering*, 8(1):209–216, 2014. 1

[4] Oded Bialer, Noa Garnett, and Tom Tirer. Performance advantages of deep neural networks for angle of arrival estimation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3907–3911. IEEE, 2019. 1, 2

[5] Wenwei Fang, Dingke Yu, Xin Wang, Yuzhang Xi, Zhihui Cao, Chunyi Song, and Zhiwei Xu. A deep learning based mutual coupling correction and doa estimation algorithm. In *2021 13th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–5. IEEE, 2021. 1

[6] John Rogers, John E Ball, and Ali C Gurbuz. Estimating the number of sources via deep learning. In *2019 IEEE Radar Conference (RadarConf)*, pages 1–5. IEEE, 2019. 2

[7] Marcio L. Lima de Oliveira and Marco J. G. Bekooij. Resnet applied for a single-snapshot doa estimation. In *2022 IEEE Radar Conference (RadarConf22)*, pages 1–6, 2022. 2