



Lightweight Deep Neural Network with Data Redundancy Removal and Regression for DOA Estimation in Sensor Array

Aifei Liu ¹ , Jiapeng Guo ¹, Yauhen Arnatovich ^{1,*} and Zhiling Liu ²

¹ The School of Software, Northwestern Polytechnical University, Xi'an 710072, China; liuaifei@nwpu.edu.cn (A.L.)

² Nanjing Electronic Equipment Institute, Nanjing 210007, China

* Correspondence: yauhen@nwpu.edu.cn

Abstract: In this paper, a lightweight deep neural network (DNN) for direction of arrival (DOA) estimation is proposed, of which the input vector is designed to remove data redundancy as well as remaining DOA information. By exploring the Vandermonde property of the steering vector of a uniform linear array (ULA), the size of the newly designed input vector is greatly reduced. Furthermore, the DOA estimation is designed as a regression problem instead of a classification problem; that is, the lightweight DNN designs the output vector as the estimated DOAs of sources, of which the size is much shorter than that of the spatial spectrum used as the output vector in the conventional DNN. The reductions in the sizes of input and output vectors lead to a reduction in the sizes of hidden layers, achieving lightwightness of the neural network. The analysis illustrates that when the number of sensors is 22, the number of parameters in the lightweight DNN is three orders of magnitude less than that in the conventional DNN. The simulation results demonstrate the lightweight DNN can provide high DOA estimation accuracy with the shortest testing time. It performs better than the conventional DNN. Furthermore, it is superior to traditional solutions such as the multiple signal classification (MUSIC) method and conventional beamforming (CBF) method in harsh conditions like low signal-to-noise ratios (SNRs), closely spaced sources, and few snapshots.

Keywords: DOA estimation; lightweight deep neural network; data redundancy; deep learning; regression



Citation: Liu, A.; Guo, J.; Arnatovich, Y.; Liu, Z. Lightweight Deep Neural Network with Data Redundancy Removal and Regression for DOA Estimation in Sensor Array. *Remote Sens.* **2024**, *16*, 1423. <https://doi.org/10.3390/rs16081423>

Academic Editors: Jiahua Zhu, Xiaotao Huang, Jianguo Liu, Xinbo Li, Gerardo Di Martino, Shengchun Piao, Junyuan Guo and Wei Guo

Received: 21 March 2024

Revised: 14 April 2024

Accepted: 15 April 2024

Published: 17 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Direction of arrival (DOA) estimation is a widely studied topic in the signal processing area, which performs a key role in wireless communications, astronomical observation, and radar applications [1–5]. The conventional beamforming (CBF) method is a classical solution for DOA estimation. However, it suffers from Rayleigh limit. Subsequently, many traditional methods were proposed to meet the accuracy requirement and high resolution of DOA estimation, such as the minimum variance distortionless response (MVDR) beamformer (also referred to as the Capon beamformer) [6], multiple signal classification method (MUSIC) algorithm [7], estimation of signal parameters using rotational invariance techniques (ESPRIT) algorithm [8] and their variants [9–13]. However, the above-mentioned traditional methods require operations such as singular value decomposition and/or the inversion on the array covariance matrix of the received signal and/or spatial spectrum searching. As a result, their computational complexity is high, which makes it difficult for them to meet real-time requirements. Moreover, most of them have large estimation errors under harsh scenarios such as when the DOAs of source signals have small angular intervals or the signal-noise ratio (SNR) is low. To overcome the drawbacks of the traditional solutions, many studies use machine learning methods to solve the problem of DOA estimation, these methods first establish a training dataset with DOA labels, and then utilize existing machine learning techniques such as radial basis function (RBF) [14] and support

vector regression (SVR) [15] to apply the derived mapping to the test data for DOA estimation. These methods require significant effort to learn the mapping during the training stage. However, once the mapping is learned and fixed after the training stage, they directly apply the mapping to process the testing data without labels to obtain DOA estimates. It is noted that the mapping only involves calculations of additions and multiplications, which avoids matrix inverse, decomposition, and spectrum searching. Thus, in the testing stage, they acquire higher computational efficiency compared to traditional methods [16], but they heavily rely on the generalization characteristics of machine learning technology. That is, only when the training data and test data have almost the same distribution, satisfactory test results can be obtained.

In recent years, DOA estimation based on deep learning methods has gained great attention due to its high accuracy and high computational efficiency during the testing phase. In 2015, a single-layer neural network model based on classification was designed to implement DOA estimation [17]. Since then, more and more improved neural networks aiming at solving DOA estimation have been proposed. In 2018, a deep neural network (DNN) was proposed, which contains a multitask auto-encoder and a set of parallel multi-layer classifiers, with the covariance vector of the array output as an input to the DNN, the auto-encoder decomposes the input vectors into sub-regions of space, then the classifiers output the spatial spectrum for DOA estimation [18]. In 2019, a deep convolutional neural network (CNN) was developed for DOA estimation by mapping the initial sparse spatial spectrum obtained from the covariance matrix to the true sparse spatial spectrum [19]. In 2020, a DeepMUSIC method was proposed for DOA estimation, by using multiple CNNs each of which is dedicated to learning the Multiple Signal Classification (MUSIC) spectra of an angular sub-region [20]. In 2021, a CNN with 2D filters was developed for DOA prediction in the low SNR [21], by mapping the 2-D covariance matrix to the spatial spectrum labeled according to the true DOAs of source signals. In 2023, a DNN framework for DOA estimation in a uniform circular array was proposed, using transfer learning and multi-task techniques [22]. The existing results show that deep learning frameworks provide better performance than traditional methods in harsh conditions such as low SNRs and small angle intervals between the DOAs of two source signals.

It is noted that all of the above-mentioned DNN-based DOA estimation methods choose to use the whole array covariance matrix of the received signal or its upper triangular elements or their transformation as the input of the network, which contains lots of redundant information when the array is uniformly linear. In addition, most of them try to match DOA estimation with the classification problem and thus use the spatial spectrum (labeled by the true DOAs of source signals or given by the existing traditional MUSIC method) as their output vector. Therefore, in the existing DNN-based DOA estimation approaches, the data redundancy in the input vector and the large size of the output vector lead to large sizes of hidden layers and make the DNN models complex overall, resulting in low computational efficiency.

There are a few works [23–25] that use neural networks with regression for DOA estimation. In [23], the neural network and a particle swarm optimization (PSO) were combined for DOA estimation, which might be trapped into a minimum solution. In [24], a DNN with regression was developed to estimate the DOA of a single source signal, without considering the situation of multiple source signals. In [25], a DNN with regression was designed for DOA estimation of multiple source signals. However, it does not consider the data redundancy in a uniform linear array (ULA).

In this paper, we consider a ULA, which is the most generally adopted array geometry for DOA estimation due to its regular structure and well-developed techniques according to the Nyquist sampling theorem [26]. By exploring the property of the ULA, a lightweight DNN is proposed by designing an input vector with data redundancy removal and using the regression fashion for DOA estimation. The lightweight DNN significantly reduces the sizes of the input vector, hidden layers, and output vector, which leads to a reduction in the number of trainable parameters of the neural network and computational load. Meanwhile,

the proposed lightweight DNN can preserve DOA estimation accuracy and performs better than the method in [25]. It is noted that by considering that the array signal is different from the image signal and DOA information is hidden in each element of the input vector obtained from the covariance matrix of the array signal, we utilize a fully connected deep neuron network to obtain the mapping from the input vector to the DOAs of source signals.

Throughout this paper, $*$, T , H , and E represent the conjugate, transpose, conjugate transpose, and expectation operations, respectively.

2. Background

Assume that K -independent far-field source signals $\{s_k(t)\}_{k=1}^K$ with a wavelength λ and DOAs of $\{\theta_k\}_{k=1}^K$ impinge on an M -element uniform linear array (ULA) with an inter-element spacing d . Moreover, it is assumed that the source signals and the array sensors are on the same plane. The received data of the array can be expressed as

$$\mathbf{r}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t), \quad (1)$$

where $\mathbf{n}(t)$ is an additive and zero-mean white Gaussian noise vector, $\mathbf{A} = [\mathbf{a}(\theta_1), \dots, \mathbf{a}(\theta_K)]$, $\mathbf{s}(t) = [s_1(t), \dots, s_K(t)]^T$; In particular, $\mathbf{a}(\theta_k)$ is an M -dimensional steering vector, which is defined as

$$\mathbf{a}(\theta_k) = [1, e^{-j2\pi\frac{d\sin\theta_k}{\lambda}}, \dots, e^{-j2\pi\frac{d\sin\theta_k}{\lambda}(M-1)}]^T. \quad (2)$$

The array covariance matrix \mathbf{R} can be expressed as

$$\mathbf{R} = E[\mathbf{r}(t)\mathbf{r}^H(t)] = \mathbf{A}\mathbf{R}_s\mathbf{A}^H + \sigma_n^2\mathbf{I}_M, \quad (3)$$

where $\mathbf{R}_s = E[\mathbf{s}(t)\mathbf{s}^H(t)]$, σ_n^2 is the noise power, and \mathbf{I}_M is an identity matrix with a size of $M \times M$. In practice, due to the finite snapshots, the covariance matrix \mathbf{R} can be estimated as

$$\hat{\mathbf{R}} = \frac{1}{N} \sum_{t=1}^N \mathbf{r}(t)\mathbf{r}^H(t), \quad (4)$$

where N is the number of snapshots, and $\hat{\bullet}$ means the approximation of the quantity above which it appears.

Equation (4) illustrates that $\hat{\mathbf{R}}$ is a conjugate symmetric matrix. Utilizing this feature, many real-valued deep learning methods use the upper triangular elements as their input vectors [18,20]. Define the vector composed of the off-diagonal upper triangular elements of $\hat{\mathbf{R}}$ by \mathbf{z} , that is

$$\mathbf{z} = [\hat{R}(1,2), \dots, \hat{R}(1,M), \hat{R}(2,3), \dots, \hat{R}(2,M), \dots, \hat{R}(M-1,M)]^T. \quad (5)$$

It is noted that for a real-valued DNN network, the input vector needs to be real-valued. Therefore, by concatenating the real and imaginary parts of \mathbf{z} , we obtain $\tilde{\mathbf{z}}$ below.

$$\tilde{\mathbf{z}} = [Real(\mathbf{z}^T), Imag(\mathbf{z}^T)]^T / \|\mathbf{z}\|_2, \quad (6)$$

where $\|\dots\|_2$ defines L_2 norm. $Real\{\bullet\}$ and $Imag\{\bullet\}$ represent the real and imaginary parts of a complex value, respectively.

In [18], a fully connected DNN method with classification was developed for DOA estimation, and it utilizes the vector $\tilde{\mathbf{z}}$ as its input, named as the conventional DNN in this paper. Note that the input vector $\tilde{\mathbf{z}}$ contains data redundancy and costs the computational load without performance improvement. Moreover, since the conventional DNN is based on classification fashion, its output is equal to $\lceil \frac{\theta_{max} - \theta_{min}}{\eta} \rceil$, where $[\theta_{min}, \theta_{max})$ is the angle-searching range of the sources, and η is the grid; with $\lceil x \rceil$ is equal to the smallest integer not smaller than x . Therefore, the size of its output vector is much larger than the number of DOAs of sources, which further increases the computational load.

In the following, we analyze the data redundancy in the ULA and design a new input vector that removes data redundancy and retains DOA information. In a sequence, the lightweight DNN is proposed by using the newly designed input vector and employing the regression fashion for DOA estimation.

3. Data Redundancy Removal

3.1. Development of Data Redundancy Removal

In this section, we first prove that the conventional input vector $\tilde{\mathbf{z}}$ in Equation (6) contains data redundancy. Afterwards, we propose a new input vector that removes data redundancy and retains DOA information.

According to Equations (2) and (3), the array covariance matrix \mathbf{R} can be expanded as

$$\mathbf{R} = \sum_{k=1}^K \mathbf{a}(\theta_k) \mathbf{a}^H(\theta_k) \sigma_{s_k}^2 + \sigma_n^2 \mathbf{I}_M, \quad (7)$$

where $\sigma_{s_k}^2$ is the power of the k -th source signal.

Define the matrix $\mathbf{B}_k = \mathbf{a}(\theta_k) \mathbf{a}^H(\theta_k)$ and its element at m -th row and l -th column as $B_k(m, l)$. According to Equation (2), we obtain that

$$B_k(m, l) = e^{j\varphi_k(m-l)} \sigma_{s_k}^2, \quad (8)$$

where $\varphi_k = -j2\pi \frac{d \sin \theta_k}{\lambda}$. Therefore, by substituting Equation (8) into Equation (7), we have

$$R(m, l) = \sum_{k=1}^K B_k(m, l) + \text{sgn}(m, l) \sigma_n^2 = \sum_{k=1}^K e^{j\varphi_k(m-l)} \sigma_{s_k}^2 + \text{sgn}(m, l) \sigma_n^2. \quad (9)$$

where $\text{sgn}(m, l) = \begin{cases} 1 & \text{if } m = l \\ 0 & \text{if } m \neq l \end{cases}$. As a result, from Equation (9), we observe Lemma 1 below.

Lemma 1. *When the array is ULA, all the elements along the sub-diagonal, super-diagonal and diagonal lines of the covariance matrix \mathbf{R} are equal.*

Lemma 1 can be illustrated in Equation (10) below.

$$\mathbf{R} = \begin{bmatrix} \rho & \beta & v & \cdots & \epsilon \\ \beta^* & \rho & \beta & \ddots & \vdots \\ v^* & \beta^* & \ddots & \ddots & v \\ \vdots & \ddots & \ddots & \rho & \beta \\ \epsilon^* & \cdots & v^* & \beta^* & \rho \end{bmatrix} \quad (10)$$

where ρ, β, v , and ϵ are elements of the covariance matrix \mathbf{R} .

As shown in Equation (5), the conventional input vector uses all the upper triangular elements of the covariance matrix \mathbf{R} , which contains duplicate information and leads to data redundancy according to Lemma 1.

On the other hand, from Equation (9), it is observed that the elements along the diagonal lines are affected by noise power and source signal power. However, they do not contain information about DOAs of sources. Thus, they shall not be involved in the input vector of the DNN model. In addition, by observing Equation (9), we define

$$\mathbf{z}_1 = [\hat{R}(1, 2), \hat{R}(1, 3), \hat{R}(1, 4), \dots, \hat{R}(1, M)]^T. \quad (11)$$

By considering the above-mentioned observation, Lemma 1, and the conjugate symmetric feature of the covariance matrix \mathbf{R} , we obtain that indeed for a real-valued DNN

model and the expected covariance matrix \mathbf{R} , \mathbf{z}_1 contains all the useful elements relevant to the DOAs of sources and discards duplicate data in the vector \mathbf{z} in Equation (5), leading to the removal of data redundancy.

It is worth noticing that in practice, due to the limit of the number of snapshots, the elements along any off-diagonal line of the estimated covariance matrix $\hat{\mathbf{R}}$ are not exactly equal. On the other hand, as shown in Equation (9), the elements along each super-diagonal line contain the same information about DOAs. Thus, we propose to take the average of all the elements along each super-diagonal line of the estimated covariance matrix $\hat{\mathbf{R}}$ to obtain a new vector without data redundancy, denoted as \mathbf{z}_{sum} . Define the i -th element of \mathbf{z}_{sum} as $z_{sum}(i)$, we have

$$z_{sum}(i) = \frac{1}{M-i} \sum_{m=1}^{M-i} \hat{R}(m, m+i), i = 1, 2, \dots, M-1. \quad (12)$$

Therefore, according to Equations (11) and (12), we can construct two vectors (that is, $\tilde{\mathbf{z}}_1$ and $\tilde{\mathbf{z}}_{sum}$) as shown in Equations (13) and (14).

$$\tilde{\mathbf{z}}_1 = [\text{Real}(\mathbf{z}_1^T), \text{Imag}(\mathbf{z}_1^T)]^T / \|\mathbf{z}_1\|_2, \quad (13)$$

$$\tilde{\mathbf{z}}_{sum} = [\text{Real}(\mathbf{z}_{sum}^T), \text{Imag}(\mathbf{z}_{sum}^T)]^T / \|\mathbf{z}_{sum}\|_2. \quad (14)$$

It is noted that both $\tilde{\mathbf{z}}_1$ and $\tilde{\mathbf{z}}_{sum}$ remove data redundancy and can be used as the input vector of the DNN network theoretically. However, due to the limit of the number of snapshots in practice, the lightweight deep neural network (DNN) proposed in the following does not converge when the vector $\tilde{\mathbf{z}}_1$ is used as the input vector of the DNN. Therefore, we choose the vector $\tilde{\mathbf{z}}_{sum}$ as the input of the proposed DNN in the following, which ensures convergence. On the other hand, it is noted that the conventional input vector $\tilde{\mathbf{z}}$ using upper triangular elements as shown in Equation (5) has a dimension of $M(M-1)$. In contrast, the new input vector $\tilde{\mathbf{z}}_{sum}$ has a dimension of $2(M-1)$. Therefore, the new input vector reduces the dimension to $M/2$ times that of the conventional input vector. This implies that the nodes in the following hidden layers can be correspondingly reduced, which contributes to forming a lightweight DNN.

As a sequence, the data redundancy removal developed for the ULA above can be applied to the matrix $\hat{\mathbf{R}}$ to obtain the input vector without data redundancy (i.e., $\tilde{\mathbf{z}}_{sum}$).

3.2. Analysis of Data Redundancy Removal

According to Equations (3) and (4), $\hat{\mathbf{R}}$ is the maximum-likelihood estimate of the expected \mathbf{R} , and thus the estimation error $\Delta\mathbf{R}$ always exists [27]; that is,

$$\Delta\mathbf{R} = \hat{\mathbf{R}} - \mathbf{R}. \quad (15)$$

In addition, the proposed lightweight DNN is based on the $\tilde{\mathbf{z}}_{sum}$ in Equation (14). In contrast, the method in [25] uses the vector composed of the off-diagonal upper triangular elements; that is, $\tilde{\mathbf{z}}$ in Equation (6). Both $\tilde{\mathbf{z}}_{sum}$ and $\tilde{\mathbf{z}}$ are based on the estimated covariance matrix $\hat{\mathbf{R}}$. Consequently, these elements are also subject to estimation inaccuracies, which subsequently precipitate errors in DOA estimation. It is expected that a larger estimation error of $\tilde{\mathbf{z}}_{sum}$ or $\tilde{\mathbf{z}}$ leads to a higher DOA estimation error. We define the estimation error of $\tilde{\mathbf{z}}_{sum}$ by $\Delta\tilde{\mathbf{z}}_{sum}$, which is given as

$$\Delta\tilde{\mathbf{z}}_{sum} = \tilde{\mathbf{z}}_{sum} - \tilde{\mathbf{z}}_{sum}^{exp}, \quad (16)$$

where the elements of $\tilde{\mathbf{z}}_{sum}^{exp}$ are obtained by replacing the estimated covariance matrix $\hat{\mathbf{R}}$ with the expected one \mathbf{R} in Equation (12).

Similarly, we define the estimation error of $\tilde{\mathbf{z}}$ by $\Delta\tilde{\mathbf{z}}$, which is written as

$$\Delta\tilde{\mathbf{z}} = \tilde{\mathbf{z}} - \tilde{\mathbf{z}}^{exp}, \tag{17}$$

where the elements of $\tilde{\mathbf{z}}^{exp}$ are obtained by replacing the estimated covariance matrix $\hat{\mathbf{R}}$ with the expected one \mathbf{R} in Equation (5). Define the 2-norm of the vectors $\Delta\tilde{\mathbf{z}}_{sum}$ and $\Delta\tilde{\mathbf{z}}$ as $\|\Delta\tilde{\mathbf{z}}_{sum}\|_2$ and $\|\Delta\tilde{\mathbf{z}}\|_2$, respectively.

In the following, a comparative analysis of the numerical outcomes for $\|\Delta\tilde{\mathbf{z}}_{sum}\|_2$ and $\|\Delta\tilde{\mathbf{z}}\|_2$ is presented. Assuming that the ULA consists of 22 elements with an inter-element spacing equal to $\frac{\lambda}{2}$. Supposing that there are two source signals with the same SNR impinge onto the array with DOAs of $\theta_1 = -40.55^\circ$ and $\theta_2 = -36.3^\circ$, respectively. The number of snapshots equals 400. The number of trials is 200.

When SNR = -10 dB, we obtain $\|\Delta\tilde{\mathbf{z}}_{sum}\|_2 = 0.19$, and $\|\Delta\tilde{\mathbf{z}}\|_2 = 0.37$. When SNR = 5 dB, we obtain $\|\Delta\tilde{\mathbf{z}}_{sum}\|_2 = 0.06$, and $\|\Delta\tilde{\mathbf{z}}\|_2 = 0.09$. Overall, $\|\Delta\tilde{\mathbf{z}}_{sum}\|_2 < \|\Delta\tilde{\mathbf{z}}\|_2$. This fact leads to better performance of the proposed lightweight DNN with its input as $\tilde{\mathbf{z}}_{sum}$, as comparisons of the method in [25] with its input as $\tilde{\mathbf{z}}$. This fact matches with numerical results in Section 5.

Furthermore, from the analysis above, we obtain that $\|\Delta\tilde{\mathbf{z}}_{sum}\|_2$ decreases as the SNR increases, which implies that the performance of the lightweight DNN with $\tilde{\mathbf{z}}_{sum}$ gets better as the SNR increases.

4. Lightweight DNN for DOA Estimation

In this section, we propose a lightweight DNN for DOA estimation, which is illustrated in Figure 1. As shown in Figure 1, the proposed lightweight DNN model utilizes the newly developed input vector $\tilde{\mathbf{z}}_{sum}$ as its input vector. Furthermore, different from the conventional DNN model with classification [18–21], the new DNN model is a regression model and has an output vector with a dimension equal to the number of sources, which approaches to the vector of true DOAs of sources in a regression fashion. It is noted that by considering the DOAs of sources are continuous values, the DNN model with regression can match the task of DOA estimation naturally. It is noted that in practice, prior to DOA estimation, the estimation of the number of sources can be accomplished by the classical methods such as the Minimum Description Length (MDL) and the Akaike Information Criterion (AIC) methods [28]. In addition, by considering that the array signal is different from the image signal and DOA information is hidden in each element of the input vector which is obtained from the covariance matrix of the array signal, we select a fully connected deep neuron network to extract the mapping from the input vector to the DOAs of source signals.

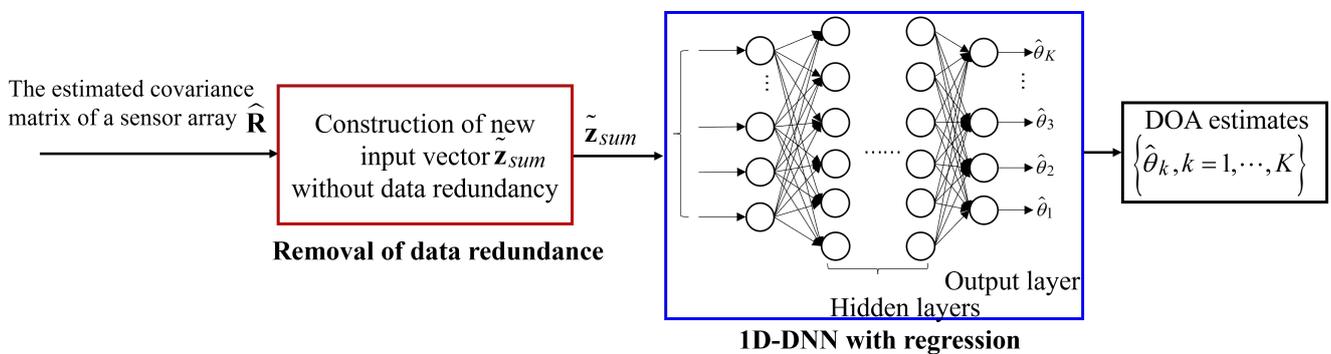


Figure 1. Proposed lightweight DNN for DOA estimation in a ULA array.

As shown in Figure 1, the proposed lightweight DNN is a fully connected network with regression and contains an input layer, several hidden layers with activation functions, and an output layer. Furthermore, each node of each layer in the network is connected to each node of the adjacent forward layer. The input data flows into the input layer,

passes through the hidden layers, and turns into the output of the network, which gives DOA estimates. The detailed structure of the proposed lightweight DNN model and its construction of the training data set are given as follows. It is noted that both the proposed lightweight DNN method and the method in [25] use regression for DOA estimation. The difference between the proposed lightweight DNN method and the one in [25] is that the proposed lightweight DNN method removes the data redundancy and significantly reduces the trainable parameters, by analyzing the property of the covariance matrix of a ULA and the parameters of the network.

4.1. Detailed Structure of Lightweight DNN

The computations of hidden layers are feedforward as

$$\mathbf{h}_l = g_l(\mathbf{W}_{l,l-1}\mathbf{h}_{l-1} + \mathbf{b}_l), l = 1, 2, \dots, L - 1, \quad (18)$$

where L is the total number of the layers except for the input layer; \mathbf{h}_l represent the output vector of the l -th layer; $\mathbf{W}_{l,l-1}$ is the weight matrix between the $(l-1)$ -th layer and l -th layer; \mathbf{b}_l is the bias vector of the l -th layer; g_l is the activation function of the l -th layer. The activation function is set as $g_l(\bullet) = \tanh(\bullet)$, which is expressed as

$$\tanh(\alpha) = \frac{e^\alpha - e^{-\alpha}}{e^\alpha + e^{-\alpha}}, \quad (19)$$

where α is a real value. The output vector of the output layer is given as

$$\mathbf{h}_L = \mathbf{W}_{L,L-1}\mathbf{h}_{L-1} + \mathbf{b}_L. \quad (20)$$

In the training phase, the proposed DNN is performed in a supervised manner with the training data-label set, and the parameters of the DNN are adjusted to make the output vector \mathbf{h}_L approach to the label, which is composed of the DOAs of source signals. We define the number of input vectors by I . Then, the training data set can be expressed as $\Gamma = \{\mathbf{x}_{(1)}, \dots, \mathbf{x}_{(I)}\}$ with its label set $\Psi = \{\bar{\theta}_{(1)}, \dots, \bar{\theta}_{(I)}\}$, $\mathbf{x}_{(i)}$ and $\bar{\theta}_{(i)}$ are the i -th input vector and its label, respectively. $\mathbf{x}_{(i)}$ is equal to $\tilde{\mathbf{z}}_{sum}$ generated in the i -th numerical experiment. $\bar{\theta}_{(i)}$ is a K -dimensional vector composed of the true DOAs of sources in the i -th numerical experiment.

The set of all the trainable parameters in the lightweight DNN model can be collectively referred to as Ω . The update of Ω follows back-propagation towards minimizing the Mean Square Error(MSE) loss function as follows.

$$\hat{\Omega} = \underset{\Omega}{\operatorname{argmin}} \frac{1}{IK} \sum_{i=1}^I \|\mathbf{h}_{L,(i)} - \bar{\theta}_{(i)}\|_2^2, \quad (21)$$

where $\|\bullet\|_2$ represents 2-norm, which measures the distance between the output vector of the network and the corresponding label, $\mathbf{h}_{L,(i)}$ represents the output vector of the network corresponding to the i -th input vector. In the testing phase, the output vector of the output layer gives the estimated values of the DOAs of source signals explicitly.

For the lightweight DNN model, we define the size of the input vector by $\tilde{J} = 2(M - 1)$. Note that with more layers and larger sizes of layers, the expressivity power of the network is increased during the training stage. However, the network tends to overfit the training data. As a result, in the testing stage, the performance is obviously degraded due to the lack of generalization. Furthermore, referring to [18], for the balance between the expressivity power with deeper network and aggravation with more network parameters, we set the number of hidden layers to be 2 and their sizes are equal to $\lfloor \frac{2}{3}\tilde{J} \rfloor$ and $\lfloor \frac{4}{9}\tilde{J} \rfloor$, respectively, where $\lfloor x \rfloor$ is equal to the largest integer not larger than x .

4.2. Construction of Training Data Set

Assuming that the searching angle range of the source signals is from θ_{min} to θ_{max} , the angular interval between two source signals in this range is defined as Δ , which is sampled from a set of $[\Delta_{min}, \Delta_{min} + \Delta_d, \Delta_{min} + 2\Delta_d, \dots, \Delta_{max}]$, where Δ_{min} , Δ_{max} , and Δ_d are the minimum angle interval between the DOAs of two source signals, the maximum angle interval, and an angle increment, respectively. In this way, any two source signals in this range that are spatially close to each other and those with large spacing can all be included in the training data set. Since the elements used as the input vector from the covariance matrix are not affected by the order of DOAs of source signals, with the DOA of the first source signal is sampled with a grid η from θ_{min} to $\theta_{max} - (K - 1)\Delta$, and the DOA of the k -th source signal is $\theta_1 + (k - 1)\Delta, k = 2, \dots, K$. Furthermore, in order to adapt to the performance fluctuations in low SNRs, input vectors with multiple SNRs lower than 0dB are trained at the same time, making the lightweight DNN better adapted to unknown low and high SNRs during the testing phase.

4.3. Analysis of Number of Trainable Parameters

In this section, we present a comparative analysis of the proposed lightweight DNN, against the method in [25], the conventional DNN in [18], deep convolution network (DCN) in [19], and DeepMUSIC in [20], focusing on the number of trainable parameters. For the conventional DNN model in [18], we follow the setting in [18]. That is, for the autoencoder, we denote the size of each of the input and output layers as $J = M(M - 1)$, define the number of each encoder and decoder has one hidden layer with a size of $\lfloor \frac{J}{2} \rfloor$, and define the number of spatial subregion as p . As a sequence, we obtain that for each of the multilayer classifiers after the autoencoder, the sizes of two hidden layers are equal to $\lfloor \frac{2}{3}J \rfloor$ and $\lfloor \frac{4}{9}J \rfloor$, respectively. In addition, the size of output layer (denoted as γ) for each multilayer classifier is equal to

$$\gamma = \lceil \frac{\theta_{max} - \theta_{min}}{\eta p} \rceil. \tag{22}$$

Correspondingly, according to the analysis in Section 3.1 for the proposed lightweight DNN, we have $\tilde{J} = \frac{2}{M}J$. By following the above-mentioned definitions and the structure of the lightweight DNN, conventional DNN, and method in [25], we can obtain the number of parameters in the three fully connected DNN models, as shown in Table 1.

Table 1. Analysis of number of trainable parameters in fully-connected DNN methods.

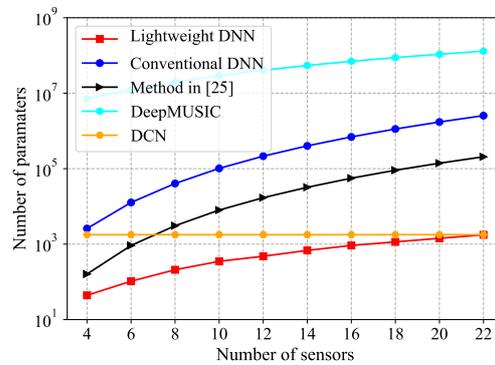
Number of Parameters	Autoencoder	Hidden Layer 1	Hidden Layer 2	Output Layer
Lightweight DNN	N.A.	$(\tilde{J} + 1) \times \lfloor \frac{2}{3}\tilde{J} \rfloor$	$(\lfloor \frac{2}{3}\tilde{J} \rfloor + 1) \times \lfloor \frac{4}{9}\tilde{J} \rfloor$	$(\lfloor \frac{4}{9}\tilde{J} \rfloor + 1) \times K$
Method in [25]	N.A.	$(J + 1) \times \lfloor \frac{2}{3}J \rfloor$	$(\lfloor \frac{2}{3}J \rfloor + 1) \times \lfloor \frac{4}{9}J \rfloor$	$(\lfloor \frac{4}{9}J \rfloor + 1) \times K$
Conventional DNN	$(J + 1) \times \lfloor \frac{J}{2} \rfloor + (\lfloor \frac{J}{2} \rfloor + 1) \times J \times p$	$(J + 1) \times \lfloor \frac{2}{3}J \rfloor \times p$	$(\lfloor \frac{2}{3}J \rfloor + 1) \times \lfloor \frac{4}{9}J \rfloor \times p$	$(\lfloor \frac{4}{9}J \rfloor + 1) \times \gamma \times p$

Table 2 shows the number of trainable parameters in DeepMUSIC and DCN by following the parameter settings in [19,20], which are mainly from the convolution layers and dense layers. For DeepMUSIC, C_{in1} represents the number of input channels, K_{s1} is the kernel size of the first two convolution layers, and K_{s2} is the kernel size of convolution layer 3 and convolution layer 4. N_f is the number of filters. C_{out1} and C_{out2} represent the sizes of the first and second dense layers, respectively. For DCN, K_{s3}, K_{s4}, K_{s5} and K_{s6} represent the kernel size of the first till fourth convolution layers, of which the number of filters are N_{f1}, N_{f2}, N_{f3} , and N_{f4} , respectively. There is no dense layer.

Table 2. Analysis of number of trainable parameters in CNN-based methods.

Number of Parameters	Convolution Layer 1	Convolution Layer 2	Convolution Layer 3	Convolution Layer 4	Dense Layer 1	Dense Layer 2
DeepMUSIC	$K_{s1}^2 \times C_{in1} \times N_f$	$K_{s1}^2 \times N_f \times N_f$	$K_{s2}^2 \times N_f \times N_f$	$K_{s2}^2 \times N_f \times N_f$	$M^2 \times N_f \times C_{out1}$	$C_{out1} \times C_{out2}$
DCN	$K_{s3} \times C_{in2} \times N_{f1}$	$K_{s4} \times N_{f1} \times N_{f2}$	$K_{s5} \times N_{f2} \times N_{f3}$	$K_{s6} \times N_{f3} \times N_{f4}$	N.A.	N.A.

When $\theta_{min} = -60^\circ, \theta_{max} = 60^\circ, p = 6, \eta = 1^\circ, K = 2, C_{in1} = 3, C_{in2} = 2, K_{s1} = 5, K_{s2} = 3, K_{s3} = 25, K_{s4} = 15, K_{s5} = 5, K_{s6} = 3, N_f = 256, N_{f1} = 12, N_{f2} = 6, N_{f3} = 3, N_{f4} = 1, C_{out1} = 1024, C_{out2} = 120$, the total parameters of the above-mentioned five deep learning methods versus the number of sensors are shown in Figure 2. From Figure 2, we can see that the number of trainable parameters in the lightweight DNN is significantly reduced compared to those of the conventional DNN, method in [25], and DeepMUSIC. In particular, when the number of sensors is 22, the number of parameters in the lightweight DNN is three orders, two orders, and five orders of magnitude less than that in the conventional DNN, the method in [25], and DeepMUSIC, respectively. This fact contributes to fitting the DNN-based DOA estimation into the embedded system. In addition, the lightweight DNN method has fewer parameters than the DCN method when the number of sensors is less than 22. The DCN method remains constant regardless of the number of sensors. This is because the input of the DCN method is the spatial spectrum proxy, which has a fixed length equal to $\lceil \frac{\theta_{max} - \theta_{min}}{\eta} \rceil$. On the other hand, the inputs of other methods are all explicitly relevant to the dimension of the array covariance matrix. Thus, their parameters are related to the number of sensors.

**Figure 2.** Trainable parameters in the DNN models versus number of sensors.

4.4. Analysis of Computational Complexity

Analogous to the approach detailed in [11], we quantify the primary computational complexity through the calculation of real-valued multiplications, as given in Table 3. In this table, L pertaining to the DCN denotes the length of the input vector, which is set as 120. In addition, we define

$$\tilde{\gamma} = \lceil \frac{\theta_{max} - \theta_{min}}{\eta} \rceil. \quad (23)$$

Note that when $\eta = 0.1$ and $M = 22$, we have $\tilde{\gamma} \gg J > \tilde{J} > M > K$ [11] and $(M - 1)(M - K) \approx (M + 1)M \approx J$. According to the settings in Section 4.3, it is found from Table 3 that the computational complexity of the CBF, MUSIC, DeepMUSIC, and DCN methods is significantly higher than that of the fully-connected DNN-based methods, which corresponds to the testing time in Table 4 below.

Table 3. Analysis of primary computational complexity.

Algorithms	Primary Computational Complexity
Lightweight DNN	$\mathcal{O}[\tilde{J} \times \lfloor \frac{2}{3}\tilde{J} \rfloor + \lfloor \frac{2}{3}\tilde{J} \rfloor \times \lfloor \frac{4}{9}\tilde{J} \rfloor + \lfloor \frac{4}{9}\tilde{J} \rfloor \times K]$
Method in [25]	$\mathcal{O}[J \times \lfloor \frac{2}{3}J \rfloor + \lfloor \frac{2}{3}J \rfloor \times \lfloor \frac{4}{9}J \rfloor + \lfloor \frac{4}{9}J \rfloor \times K]$
Conventional DNN	$\mathcal{O}[J \times \lfloor \frac{1}{2} \rfloor + \lfloor \frac{1}{2} \rfloor \times J \times p + J \times \lfloor \frac{2}{3}J \rfloor \times p + \lfloor \frac{2}{3}J \rfloor \times \lfloor \frac{4}{9}J \rfloor \times p + \lfloor \frac{4}{9}J \rfloor \times \gamma \times p]$
MUSIC	$4 \times \mathcal{O}[(M+1)(M-K)\tilde{\gamma} + M^2K]$
CBF	$4 \times \mathcal{O}[(M+1)M\tilde{\gamma}]$
DeepMUSIC	$\mathcal{O}[K_{s1}^2 \times C_{in1} \times N_f \times M^2 + K_{s1}^2 \times N_f \times N_f \times M^2 + K_{s2}^2 \times N_f \times N_f \times M^2 \times 2 + M^2 \times N_f \times C_{out1} + C_{out1} \times C_{out2}]$
DCN	$\mathcal{O}[K_{s3} \times C_{in2} \times N_{f1} \times L + K_{s4} \times N_{f1} \times N_{f2} \times L + K_{s5} \times N_{f2} \times N_{f3} \times L + K_{s6} \times N_{f3} \times N_{f4} \times L]$

Table 4. Averaged testing time for one trial.

Method	Light Weight DNN	Method in [25]	Conventional DNN	MUSIC with Grid 1°	CBF with Grid 1°	MUSIC with Grid 0.1°	CBF with Grid 0.1°	Deep MUSIC	DCN
Testing time/ms	0.9	1.1	3.3	3.7	2.3	22.9	17.9	26.3	20.4

5. Results

In this section, by conducting simulation experiments, the proposed lightweight DNN is compared with the conventional DNN [18], the method in [25], DeepMUSIC in [20] and DCN in [19] in terms of testing time and the root-mean-square-error (RMSE) of DOA estimation. In addition, the traditional spectrum-based methods such as MUSIC and CBF are also included for comparisons. Furthermore, the Cramér–Rao Bound (CRB) of DOA estimation [2] is given as a lower bound. The DNN models are implemented using TensorFlow as the backend. In the testing stage, for a fair comparison of testing time, all the above-mentioned methods are executed on the Intel(R) Core(TM) i7-8750H CPU at 2.20 GH.

5.1. Simulation Settings

Assuming that the ULA consists of 22 elements with an inter-element spacing equal to $\frac{\lambda}{2}$. Supposing that there are two source signals impinging onto the array, of which the DOA range is from $\theta_{min} = -60^\circ$ to $\theta_{max} = 60^\circ$. The angular interval between the DOAs of two source signals is from $\Delta_{min} = 2^\circ$ to $\Delta_{max} = 40^\circ$, with $\Delta_d = 2^\circ$ and $\eta = 1^\circ$. The SNRs for different source signals are equal and SNR_k is defined as the power ratio of the k -th source signal to noise in dB, which is given below.

$$SNR_k = 10 \log_{10} \frac{\sigma_{s_k}^2}{\sigma_n^2}. \quad (24)$$

For the DNN models, in the training phase, the snapshots are set as 400. In addition, using input vectors from multiple SNRs of $\{-13 \text{ dB}, -10 \text{ dB}, -5 \text{ dB}, 0 \text{ dB}\}$ to train the network simultaneously. Moreover, 10 groups of covariance vectors are collected for each direction setting with random noise. Therefore, $(118 + 116 + \dots + 80) \times 4 \times 10 = 79,200$ input vectors are collected in the training dataset in total. The learning rate is $\mu = 0.001$ and the mini-batch size is 32, the order of training data is shuffled in each epoch.

5.2. MSE Loss during Training and Validation

In this section, as given in Figure 3, we provide the training and validation MSE loss of the proposed lightweight DNN versus the number of epochs by randomly dividing

the training data into 80% for training and 20% for validation. From Figure 3, we observe that the training loss and validation loss gradually reduce when the number of epochs increases and converges at about 400 epochs. Furthermore, they are close to each other. Therefore, we conclude that the proposed lightweight DNN with the input vector after the removal of data redundancy can accomplish the task of DOA estimation well. The detailed performance of DOA estimation in the testing stage is given as follows.

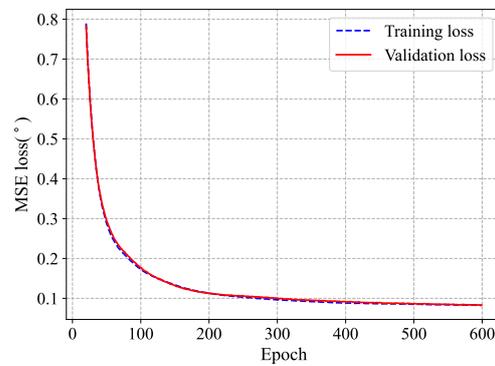


Figure 3. MSE loss versus epoch given by the proposed lightweight DNN.

5.3. RMSE versus SNRs and Testing Time

In the testing phase, in order to verify the generalization of the DNN models, the noises and source signals are different from those that appeared in the training phase. In addition, the DOAs of sources are set to be non-integer (that is, off-grid), which does not appear in the training stage. The DOAs of source signals are $(\theta_1, \theta_2) = (-40.55^\circ, -36.3^\circ)$. The RMSE is used to measure the testing performance of different methods, defined as

$$RMSE = \sqrt{\frac{1}{GK} \sum_{g=1}^G \sum_{k=1}^K |\hat{\theta}_{k,g} - \theta_k|^2}, \quad (25)$$

where G is the number of Monte Carlo simulation experiments, which is set as 200. $K = 2$. $\hat{\theta}_{k,g}$ represents the DOA estimation value of the k -th source signal in the g -th experiment. In this part, SNR is taken from -16 dB to 10 dB with an interval of 2 dB and the number of snapshots is 400. The RMSE of DOAs estimated by the above-mentioned methods under different SNRs is given in Figure 4. Table 4 shows the averaged testing time for one trial. Figure 4 illustrates that the proposed lightweight DNN performs better than the method in [25], conventional DNN and the MUSIC and CBF methods with a grid of 1° . Its superiority is obvious when the SNR is lower than -8 dB. Moreover, the time spent by the lightweight DNN is about four times less than that spent by the MUSIC method with a grid of 1° . On the other hand, the proposed lightweight DNN has estimation accuracy lower than the MUSIC method with a grid of 0.1° . This is because the DNN-based approach yields biased estimators [20]. In contrast, the MUSIC method provides unbiased estimation when the source signals are uncorrelated and the number of arrays and snapshots is large [29,30]. It is noted that in Figure 4, the CBF method always fails because it suffers from the Rayleigh limit. In addition, the MUSIC method with a grid of 0.1° performs closely to the CRB when the SNR is larger than -8 dB. On the other hand, its performance deviates from the CRB when the SNR is larger than 5 dB. This phenomenon is caused by the limit of the searching grid in the MUSIC Method. Furthermore, as illustrated in Figure 4, the performance of the DeepMUSIC method is similar to that of the MUSIC method with a grid of 1° . This is because the label of the DeepMUSIC is the spatial spectrum of the MUSIC method and the grid in the DeepMUSIC method is equal to 1° to be consistent with the grid for other DNN methods. In addition, the DCN method performs better than the other methods except the lightweight DNN method, in most cases. In terms of testing time as given in Table 4, both DeepMUSIC and DCN methods cost much more than the lightweight DNN method.

It is noted that the higher estimation accuracy of the MUSIC method with a grid of 0.1° costs more spectrum searching load and the time it takes is about 25 times more than that by the lightweight DNN, as shown in Table 4. In addition, it is observed the time spent by the lightweight DNN is about 3 times less than that by the conventional DNN.

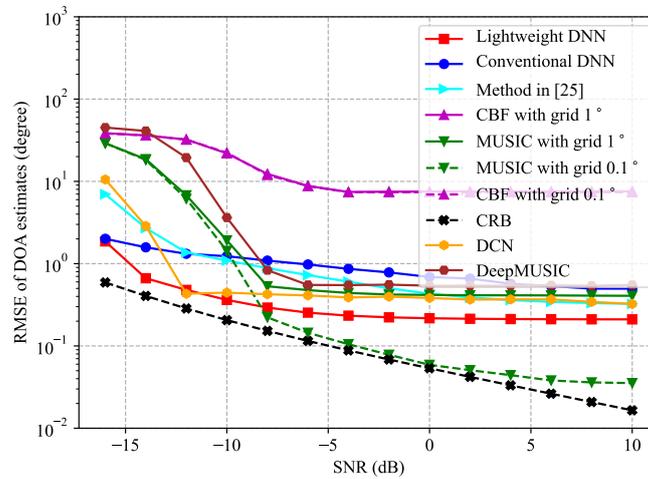


Figure 4. RMSE versus SNR when $(\theta_1, \theta_2) = (-40.55^\circ, -36.3^\circ)$.

5.4. RMSE versus DOA Separations

In this part, the RMSE of different methods is shown with the variation of intervals between the DOAs of two source signals. The DOA of the first source signal is set to be -40.55° and the DOA of the second source signal is equal to $-40.55^\circ + \tilde{\Delta}$, where $\tilde{\Delta}$ is taken from the set of $\{2.25^\circ, 4.25^\circ, 8.25^\circ, \dots, 32.25^\circ, 36.25^\circ\}$ in sequence. The number of snapshots is 400. When SNR is -2 dB, the RMSE of DOA estimated by the above-mentioned methods under different DOA separations is shown in Figure 5. From Figure 5, it can be seen that the lightweight DNN performs better than the MUSIC method and CBF method when their searching grid is set to be 1° . Furthermore, it is always superior to the method in [25], conventional DNN method, DeepMUSIC method, and DCN method. Similar to the Figure 4, the MUSIC method with a grid of 0.1° approaches the CRB in most cases. However, it is noted that in a very small DOA separation such as 2.25° , even the MUSIC method with a grid of 0.1° fails. In contrast, the lightweight DNN performs well. In addition, it is shown that the CBF method with a grid of 0.1° gradually approaches the CRB when the DOA separation increases.

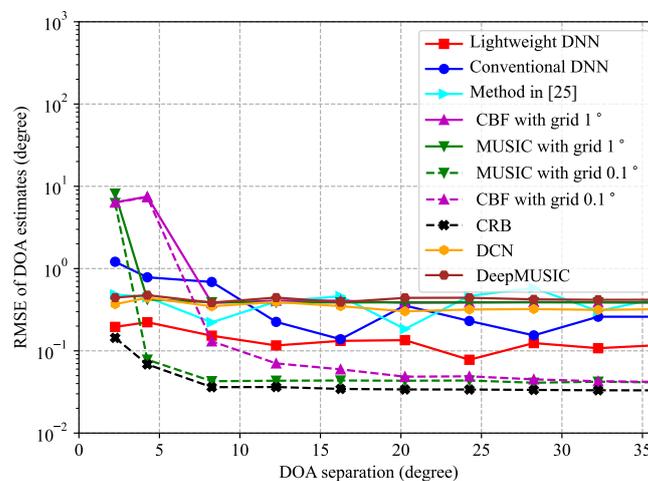


Figure 5. RMSE versus DOA separation when SNR = -2 dB.

5.5. RMSE versus Number of Snapshots

In this part, the number of snapshots is selected from the set of {30, 50, 100, 200, 300, 400, 500, 600, 700, 900, 1200, 1500, 1800, 2000}, the DOAs of the two source signals are $(\theta_1, \theta_2) = (-40.55^\circ, -36.3^\circ)$. Figure 6 shows the RMSE of all methods against the number of snapshots when the SNR is equal to -2 dB. From Figure 6, it is found that except for the CBF method, the other methods perform better when the number of snapshots increases. In addition, it is observed that the DNN models trained in the scenario of 400 snapshots are applicable to the scenarios of more snapshots and fewer snapshots. Furthermore, the lightweight DNN behaves significantly better than the conventional DNN when the number of snapshots is less than 900. As the number of snapshots increases, the estimation accuracy of lightweight DNN is still slightly higher than that of the conventional DNN, method in [25], DeepMUSIC method, and DCN method. It is worth noting that the lightweight DNN is always superior to the MUSIC method with a grid of 1° and it performs better than the MUSIC method with 0.1° when the number of snapshots is less than 100.

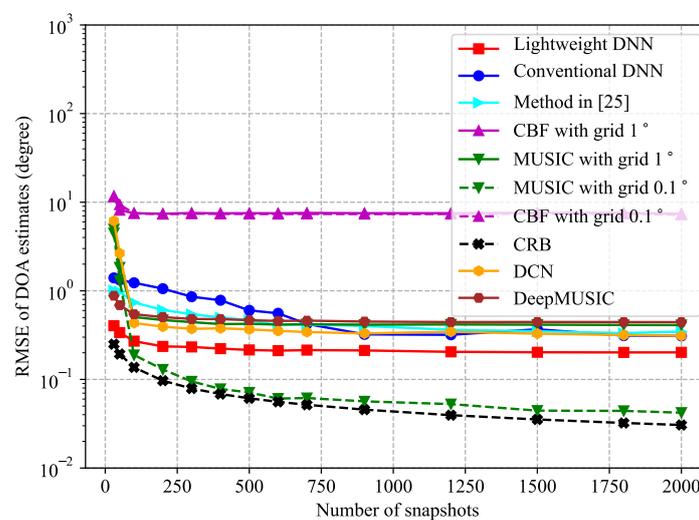


Figure 6. RMSE versus number of snapshots when SNR = -2 dB.

5.6. RMSE versus Power Ratio of Two Source Signals

The DOAs of the two closely spaced source signals are $(\theta_1, \theta_2) = (-40.55^\circ, -38.3^\circ)$. The number of snapshots for both source signals is 400. The SNR for the first source signal is fixed as -2 dB. Figure 7 demonstrates the RMSE versus the power ratio of the second source signal to the first source signal. From Figure 7, we observe that the RMSE of the lightweight DNN method increases from 0.2° to about 1.5° when the power ratio of the second source signal to the first one increases from 1 to 14. Similarly, the RMSE of the conventional DNN, the method in [25], DeepMUSIC, and DCN methods increase slightly with the increment of the power ratio. As shown in Figure 7, the MUSIC and CBF methods always fail because the DOAs of source signals are very close. It is noted that the CRB reduces a bit when the power ratio increases. This is because the power of the second source signal is increased with the increment of the power ratio. However, the CRB is limited by the closely spaced source signals.

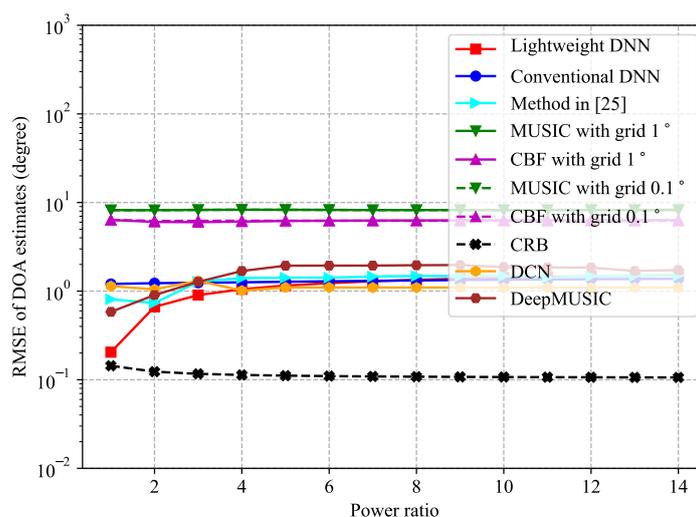


Figure 7. RMSE versus power ratio of second source signal to the first one when $(\theta_1, \theta_2) = (-40.55^\circ, -38.3^\circ)$.

6. Discussion

From the analysis above, it can be seen that the number of total parameters in the lightweight DNN model is significantly reduced compared to those DNN models that use the upper triangular elements of the covariance matrix as input. In particular, when the number of sensors is 22, it is 2 and 3 orders of magnitude less than that in the conventional DNN model and the method in [25], respectively. This fact makes the proposed lightweight DNN suitable for real-time embedded applications. Furthermore, it is noted that the lightweight DNN can preserve high accuracy of DOA estimation and perform better than the conventional DNN and method in [25]. In addition, it provides higher estimation accuracy and costs less trainable parameters and computational load than CNN-based methods such as DeepMUSIC and DCN. Also, it is illustrated that the lightweight DNN performs better than the spatial spectrum-based methods such as MUSIC and CBF method under harsh conditions such as low SNRs and/or closely spaced source signals and/or few snapshots. Moreover, its testing time is obviously shorter than that of the MUSIC and CBF method, due to the avoidance of spectrum searching and matrix decomposition. On the other hand, under good conditions such as high SNRs, the estimation accuracy of the lightweight DNN is lower than the MUSIC method with a grid of 0.1° . This is because the DNN-based methods are biased estimators while the MUSIC method can provide unbiased DOA estimation. It is noted that the MUSIC method with a grid of 0.1° provides higher estimation accuracy with a cost of a testing time of about 25 times more than that of the lightweight DNN. On the other hand, as shown in simulation results, the lightweight DNN can achieve high estimation accuracy such as 0.2° when the SNR is not extremely low (not lower than -6 dB) and the number of snapshots is not very small (not smaller than 100).

7. Conclusions

In order to make the DNN-based DOA estimation approaches real-time and less costly, we proposed a lightweight DNN model for a ULA. Compared to the conventional DNN model, the proposed lightweight DNN model has two improvements. Firstly, its input vector is designed by using the knowledge of ULA (that is, the steering vector of the ULA has the property of Vandermonde) to implement the removal of data redundancy as well as retain the DOA information. Therefore, the input vector is $\frac{M}{2}$ times less than the conventional DNN model, which contributes to reducing the sizes of the following hidden layers. Secondly, the output vector of the lightweight DNN model is designed in a regression fashion instead of classification, which has a size equal to the number of sources. Overall, the number of total parameters in the lightweight DNN model is significantly

reduced compared to that in the existing DNN models. Furthermore, the lightweight DNN model performs better than the existing DNN models because the lightweight DNN model explores the characteristics of the signal received by the ULA for designing its input.

Author Contributions: Conceptualization and methodology, A.L., J.G. and Y.A.; software, validation, and visualization, J.G. and A.L.; formal analysis and investigation, J.G.; resources and data curation, A.L. and Z.L.; writing—original draft preparation and writing—review and editing, A.L., J.G. and Y.A.; supervision, A.L. and Z.L.; project administration, A.L.; funding acquisition, Y.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by Natural Science Basic Research Program of Shaanxi Province, China (Program No. 2023-JC-YB-573), and in part by Basic Research Program of Taicang City, China, in 2023 (Program No. TC2023JC05), and in part by the Fundamental Research Funds for the Central Universities under Grants D5000210641 and G2022WD01022.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Godara, L.C. Application of antenna arrays to mobile communications. II. Beam-forming and direction-of-arrival considerations. *Proc. IEEE* **1997**, *85*, 1195–1245. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Van Trees, H.L. *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*; John Wiley & Sons: Hoboken, NJ, USA, 2002.
- Pesavento, M.; Trinh-Hoang, M.; Viberg, M. Three more decades in array signal processing research: An optimization and structure exploitation perspective. *IEEE Signal Process. Mag.* **2023**, *40*, 92–106. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Xiao, G.; Quan, Y.; Sun, Z.; Wen, B.; Liao, G. Design of a Digital Array Signal Processing System with Full Array Element. *Remote Sens.* **2023**, *15*, 4043. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Wen, F.; Shi, J.; Wang, X.; Wang, L. Angle estimation for MIMO radar in the presence of gain-phase errors with one instrumental Tx/Rx sensor: A theoretical and numerical study. *Remote Sens.* **2021**, *13*, 2964. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Capon, J. High-resolution frequency-wavenumber spectrum analysis. *Proc. IEEE* **1969**, *57*, 1408–1418. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Schmidt, R. Multiple emitter location and signal parameter estimation. *IEEE Trans. Antennas Propag.* **1986**, *34*, 276–280. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Roy, R.; Kailath, T. ESPRIT-estimation of signal parameters via rotational invariance techniques. *IEEE Trans. Acoust. Speech Signal Process.* **1986**, *37*, 984–995. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Haardt, M.; Nosske, J.A. Unitary ESPRIT: How to obtain increased estimation accuracy with a reduced computational burden. *IEEE Trans. Signal Process.* **1995**, *43*, 1232–1242. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Viberg, M.; Ottersten, B. Sensor array processing based on subspace fitting. *IEEE Trans. Signal Process.* **1991**, *39*, 1110–1121. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Yan, F.G.; Jin, M.; Liu, S.; Qiao, X.L. Real-valued MUSIC for efficient direction estimation with arbitrary array geometries. *IEEE Trans. Signal Process.* **2014**, *62*, 1548–1560. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Liu, A.; Liao, G. An eigenvector based method for estimating DOA and sensor gain-phase errors. *Digit. Signal Process.* **2018**, *79*, 116–124. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Tayem, N.; Kwon, H.M. Conjugate esprit (c-sprit). *IEEE Trans. Trans. Antennas Propag.* **2004**, *52*, 2618–2624. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Pastorino, M.; Randazzo, A. A smart antenna system for direction of arrival estimation based on a support vector regression. *IEEE Trans. Antennas Propag.* **2005**, *53*, 2161–2168. [\[CrossRef\]](#) [\[CrossRef\]](#)
- El Zooghby, A.H.; Christodoulou, C.G.; Georgiopoulos, M. A neural network-based smart antenna for multiple source tracking. *IEEE Trans. Antennas Propag.* **2000**, *48*, 768–776. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Randazzo, A.; Abou-Khousa, M.A.; Pastorino, M.; Zoughi, R. Direction of arrival estimation based on support vector regression: Experimental validation and comparison with MUSIC. *IEEE Antennas Wirel. Propag. Lett.* **2007**, *6*, 379–382. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Xiao, X.; Zhao, S.; Zhong, X.; Jones, D.L.; Chng, E.S.; Li, H. A learning-based approach to direction of arrival estimation in noisy and reverberant environments. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, Australia, 6 August 2015; pp. 2814–2818.
- Liu, Z.M.; Zhang, C.; Philip, S.Y. Direction-of-arrival estimation based on deep neural networks with robustness to array imperfections. *IEEE Trans. Antennas Propag.* **2016**, *66*, 7315–7327. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Wu, L.; Liu, Z.M.; Huang, Z.T. Deep convolution network for direction of arrival estimation with sparse prior. *IEEE Signal Process. Lett.* **2019**, *26*, 1688–1692. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Elbir, A.M. DeepMUSIC: Multiple signal classification via deep learning. *IEEE Sens. Lett.* **2020**, *4*, 1–4. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Papageorgiou, G.K.; Sellathurai, M.; Eldar, Y.C. Deep networks for direction-of-arrival estimation in low SNR. *IEEE Trans. Signal Process.* **2021**, *69*, 3714–3729. [\[CrossRef\]](#) [\[CrossRef\]](#)

22. Labbaf, N.; Oskouei, H.D.; Abedi, M.R. Robust DoA estimation in a uniform circular array antenna with errors and unknown parameters using Deep Learning. *IEEE Trans. Green Commun. Netw.* **2023**, *7*, 2143–2152. [[CrossRef](#)] [[CrossRef](#)]
23. Ping, Z. DOA estimation method based on neural network. In Proceedings of the 2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), Krakow, Poland, 4–6 November 2015; pp. 828–831.
24. Agatonovic, M.; Stankovic, Z.; Milovanovic, I.; Doncov, N.; Sit, L.; Zwick, T.; Milovanovic, B. Efficient neural network approach for 2D DOA estimation based on antenna array measurements. *Prog. Electromagn. Res.* **2013**, *137*, 741–758. [[CrossRef](#)] [[CrossRef](#)]
25. Xiong, Y.; Liu, A.; Gao, X.; Yauhen, A. DOA Estimation Using Deep Neural Network with Regression. In Proceedings of the 2022 5th International Conference on Information Communication and Signal Processing (ICICSP), Shenzhen, China, 26–28 November 2022; pp. 1–5.
26. Zhou, C.; Gu, Y.; Fan, X.; Shi, Z.; Mao, G.; Zhang, Y.D. Direction-of-arrival estimation for coprime array via virtual array interpolation. *IEEE Trans. Signal Process.* **2018**, *66*, 5956–5971. [[CrossRef](#)] [[CrossRef](#)]
27. Reed, I.S.; Mallett, J.D.; Brennan, L.E. Rapid convergence rate in adaptive arrays. *IEEE Trans. Aerosp. Electron. Syst.* **1974**, *10*, 853–863. [[CrossRef](#)] [[CrossRef](#)]
28. Wax, M.; Kailath, T. Detection of signals by information theoretic criteria. *IEEE Trans. Acoust. Speech Signal Process.* **1985**, *33*, 387–392. [[CrossRef](#)] [[CrossRef](#)]
29. Stoica, P.; Nehorai, A. MUSIC, maximum likelihood, and Cramer-Rao bound. *IEEE Trans. Acoust. Speech Signal Process.* **1989**, *37*, 720–741. [[CrossRef](#)] [[CrossRef](#)]
30. Li, F.; Liu, H.; Vaccaro, R.J. Performance analysis for DOA estimation algorithms: Unification, simplification, and observations. *IEEE Trans. Aerosp. Electron. Syst.* **1993**, *29*, 1170–1184. [[CrossRef](#)] [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.