DeepMUSIC: Multiple Signal Classification via Deep Learning

Article *in* IEEE Sensors Letters · March 2020

DOI: 10.1109/LSENS.2020.2980384

1 author:

Ahmet M. Elbir
University of Luxembourg
107 PUBLICATIONS **1,704** CITATIONS

DeepMUSIC: Multiple Signal Classification via Deep Learning

Ahmet M. Elbir, *Senior Member, IEEE*

Abstract—This letter introduces a deep learning (DL) framework for the classification of multiple signals in direction finding (DF) scenario via sensor arrays. Previous works in DL context mostly consider a single or two target scenario which is a strong limitation in practice. Hence, in this work, we propose a DL framework called DeepMUSIC for multiple signal classification. We design multiple deep convolutional neural networks (CNNs), each of which is dedicated to a subregion of the angular spectrum. Each CNN learns the MUSIC spectra of the corresponding angular subregion. Hence, it constructs a non-linear relationship between the received sensor data and the angular spectrum. We have shown, through simulations, that the proposed DeepMUSIC framework has superior estimation accuracy and exhibits less computational complexity in comparison with both DL and non-DL based techniques.

Index Terms—Deep learning, Direction finding, DOA estimation, CNN, MUSIC, Deep MUSIC.

I. INTRODUCTION

Direction finding (DF) is a crucial task for direction-of-arrival (DOA) estimation in a variety of fields including, radar, sonar, acoustics and communications [1]. While there are several different approaches in the literature, the MUSIC (MUltiple SIgnal Classification) algorithm [2] is the most popular method for this purpose.

In the literature, most of the algorithms are model-based approaches such that the performance of the DOA estimation algorithms strongly relies on the perfectness of the input data [3]. In order to mitigate this drawback, learning-based and data-driven architectures are proposed so that the non-linear relationship between the input and output data can be learned by neural networks [4]–[6]. Hence, as a class of machine learning, deep learning (DL) has gained much interest recently. DL is capable of uncovering complex relationships in data/signals and, thus, can achieve better performance. While there are several papers to demonstrate the performance of DL in wireless communications [7]–[9], limited number of works are considered in the context of DOA estimation and array signal processing [6], [10].

DOA estimation via DL is considered in [4] where a multilayer perceptron (MLP) architecture is proposed to resolve two target signals. In [5], the authors studies the same problem, also for two signal case, by exploiting the sparsity of the received signal in angular domain and design a deep convolutional neural network (CNN). A single sound target case is assumed in [11] and an MLP architecture is proposed to estimate the target DOA angle for wideband case. Acoustic scenario is also studied in [12] by incorporating long short term memory (LSTM) with CNN for online DOA estimation, which is also performed for a single target case. In a recent work [13], cognitive radar scenario is considered where DL is applied for sparse array selection and DOA estimation for a single target. This approach is extended for two targets in [14] for sparse array design.

A. M. E. is with the Department of Electrical and Electronics Engineering, Duzce University, Duzce, Turkey. e-mail: ahmetelbir@duzce.edu.tr, ahmetmelbir@gmail.com.

A common assumption in above works is that the number of targets is assumed to be small. This is because the complexity of the generation of the training data and the training overhead become more difficult as the number of targets increases. Specifically, the data set length increases on the order of N^K for K being the number of targets and N is the number grid points in the angular spectrum. In order to reduce the complexity, in this work, we proposed a multiple deep network approach for multiple target estimation. In particular, we design multiple CNNs, each of which is dedicated for a subregion of the angular spectrum. Hence, we partition the angular spectrum into non-overlapping subregions, and assume that there is a single target in each subregion. This assumption is relevant [4], [6] and it can be generalized for higher number of targets with close separation by simply increasing the number of deep networks. In order to feed the deep networks, the covariance of the sensor outputs is used as a common input. Then, the output label of each network is the MUSIC spectra of the corresponding angular subregion. Hence, we call the proposed approach DeepMUSIC which yields the MUSIC spectra at the output. The main contributions of the proposed DL approach are as follows. 1) We have introduced a DL approach to estimate multiple target DOAs whereas the previous works can only work for limited number of targets. 2)DeepMUSIC provides less computation time as compared to both DL- and non-DL-based approaches. 2) DeepMUSIC has higher DOA estimation accuracy than the conventional DL-based techniques. It provides asymptotic performance for moderate SNR levels and it performance maxes out in high SNR due to the loss of precision because of the biased nature of DL-based approaches.

II. ARRAY SIGNAL MODEL

Consider K far-field signals impinging on an M-element uniform linear array (ULA). Then, the output of the antenna array in the baseband can be given by

$$\mathbf{y}(t_i) = \sum_{k=1}^{K} \mathbf{a}(\theta_k) s_k(t_i) + \mathbf{n}(t_i), \quad i = 1, \ldots, T, \tag{1}$$

where T is the number of data snapshots and $s_k(t_i) \in \mathbb{C}$ is the signal emitted from the k-th target which is located with the DOA angle θ_k with respect to the antenna array. $\mathbf{a}(\theta_k) \in \mathbb{C}^M$ denotes the array steering vector whose m-th element is given by

$$a_m(\theta_k) = \exp\{-j \frac{2\pi \bar{d}(m-1)}{\lambda} \sin(\theta_k)\}, \tag{2}$$

where $\lambda = \frac{c_0}{f_c}$ is the wavelength for f_c being the carrier frequency and c_0 is the speed of light. $\mathbf{n}(t_i) \sim \mathcal{N}(\mathbf{0}_M, \sigma_n^2 \mathbf{I}_M)$ is zero-mean spatially and temporarily white additive Gaussian noise vector which corrupts the emitted signal with variance σ_n^2. Using the array output in (1) the covariance matrix of the received signal can be written as

$$\mathbf{R}_{\mathrm{y}} = \mathbb{E}\{\mathbf{y}\mathbf{y}^{\mathsf{H}}\} = \mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^{\mathsf{H}} + \sigma_n^2 \mathbf{I}_M, \tag{3}$$

where $\mathbb{E}\{\cdot\}$ denotes the expectation operation, $\boldsymbol{\Gamma} = \mathrm{diag}\{\sigma_1^2, \sigma_2^2, \ldots, \sigma_K^2\}$ is a $K \times K$ matrix whose diagonal entries are the signal variances and $\mathbf{A}$ is the array steering matrix

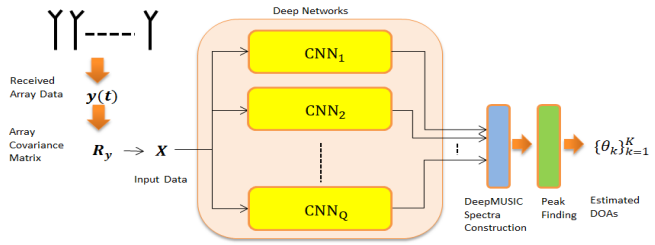

Fig. 1. The overall DeepMUSIC framework for DOA estimation.

defined as $\mathbf{A} = [\mathbf{a}(\theta_1), \mathbf{a}(\theta_2), \ldots, \mathbf{a}(\theta_K)] \in \mathbb{C}^{M\times K}$[1]. Through eigendecomposition, we can rewrite (3) as $\mathbf{R}_y = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^H$, where $\mathbf{\Lambda}$ is a diagonal matrix composed of the eigenvalues of $\mathbf{R}_y$ in descending order as $\mathbf{\Lambda} = \text{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_M\}$ and $\mathbf{U} = [\mathbf{U}_S\ \mathbf{U}_N]$ is an $M \times M$ eigenvector matrix whose first K column vectors correspond to the signal subspace by $\mathbf{U}_S$ and the remaining $M-K$ column vectors are the noise subspace eigenvectors as $\mathbf{U}_N \in \mathbb{C}^{M\times M-K}$. Using the orthogonality of signal and noise subspaces (i.e., $\mathbf{U}_S \perp \mathbf{U}_N$), and the fact that the columns of $\mathbf{U}_S$ and $\mathbf{A}$ span the same space, we have $||\mathbf{U}_N^H\mathbf{A}||_{\mathcal{F}} = 0$ where $\mathcal{F}$ denotes the Frobenius norm [2]. Now, we can write the MUSIC spectra $P(\theta)$ as

$$P(\theta) = \frac{1}{\mathbf{a}^H(\theta)\mathbf{U}_N\mathbf{U}_N^H\mathbf{a}(\theta)}, \tag{4}$$

whose largest K peaks correspond to the target DOA angles $\{\theta_k\}_{k=1}^K$. To obtain (4) in practice, we use the sample covariance matrix $\hat{\mathbf{R}}_y$ since $\mathbf{R}_y$ is not available. As a result, $\hat{\mathbf{R}}_y = \frac{1}{T}\sum_{i=1}^T \mathbf{y}(t_i)\mathbf{y}^H(t_i)$ is the input to the deep network.

In this work we can formulate the problem as estimating the target DOA angles $\{\theta_k\}_{k=1}^K$ when the array output $\{\mathbf{y}(t_i)\}_{i=1}^T$ is given. For this purpose we introduce a DL framework as shown in Fig. 1 which is fed by the array covariance matrix $\mathbf{R}_y$ and it gives the MUSIC spectra $P(\theta)$ at the output.

III. DOA ESTIMATION VIA DEEP LEARNING

The proposed DeepMUSIC framework accepts the array covariance matrix as input and yields the MUSIC spectra at the output. In the following, we first design the labels and the input of the proposed deep networks, then discuss the network architectures and the training.

In the proposed DL framework shown in Fig 1, we design Q ($\geq K$) deep networks, each of which is dedicated to a subregion of the angular spectrum. Partitioning the angular spectrum allows us to estimate the multiple target locations more effectively. The use of a single deep network is computationally prohibitive due to the fact that the training data must contain all candidate multiple target locations, whose complexity increase on the order of N^K for N DOA grid points. DOA estimation via partitioned angular spectrum is more efficient such that a reasonable assumption is made such that there is a single target present in each angular subregion [4].

Let $\Theta = \{\Theta^{\text{start}}, \ldots, \Theta^{\text{final}}\}$ be the set of DOA angles where the MUSIC cost function in (4) is evaluated for the starting and final DOA angles Θ^{start} and Θ^{final} respectively. Then, we denote the MUSIC spectra in (4) by $\mathbf{p} \in \mathbb{R}^N$ as $\mathbf{p} = [P(\Theta^{\text{start}}), \ldots, P(\Theta^{\text{final}})]^T$. To obtain the MUSIC spectra for each subregion, we partition $\mathbf{p}$ and Θ into Q non-overlapping subregions such that $\Theta = \cup_{q=1}^Q \Theta_q$, where each angular set is defined by

$$\Theta_q = \{\Theta_q^{\text{start}}, \Theta_q^{\text{start}} + \gamma, \Theta_q^{\text{start}} + 2\gamma, \ldots, \Theta_q^{\text{final}} - \gamma\}, \tag{5}$$

where $\gamma = \frac{|\Theta^{\text{start}} - \Theta^{\text{final}}|}{N}$ is the angular resolution and $\Theta_q^{\text{final}} = \Theta_{q+1}^{\text{start}}$. Hence, the number of elements in Θ_q is $|\Theta_q| = L$ where $L = N/Q$ which is assumed to be an integer number without loss of generality. We can also rewrite $\mathbf{p}$ as

$$\mathbf{p} = [\mathbf{p}_1^T, \mathbf{p}_2^T, \ldots, \mathbf{p}_Q^T]^T. \tag{6}$$

In particular, the MUSIC spectra for the q-th subregion is represented by an $L \times 1$ real-valued vector as

$$\mathbf{p}_q = [P(\Theta_q^{\text{start}}), P(\Theta_q^{\text{start}} + \gamma), P(\Theta_q^{\text{start}} + 2\gamma), \ldots, P(\Theta_q^{\text{final}} - \gamma)]^T. \tag{7}$$

During training, once $\mathbf{p}$ is obtained, we assign the MUSIC spectra of each subregion $\{\mathbf{p}_q\}_{q=1}^Q$ as the labels of each deep network. To construct the input data, we use the real, imaginary and the angular values of the covariance matrix $\mathbf{R}_y$. Let $\mathbf{X}$ be an $M \times M \times 3$ real-valued matrix, then the (i,j)-th entry of the first and the second "channels" of $\mathbf{X}$ are given by $[[\mathbf{X}]_{:,:,1}]_{i,j} = \text{Re}\{[\mathbf{R}_y]_{i,j}\}$ and $[[\mathbf{X}]_{:,:,2}]_{i,j} = \text{Im}\{[\mathbf{R}_y]_{i,j}\}$ respectively. Similarly, the (i,j)-th entry of the third "channel" of $\mathbf{X}$ is defined as $[[\mathbf{X}]_{:,:,3}]_{i,j} = \angle\{[\mathbf{R}_y]_{i,j}\}$ where $\angle\{\cdot\}$ returns the angle information of a complex quantity. While other input structures are possible such as the real and imaginary parts of the upper triangle of the covariance matrix [4], we observed that the above approach provides better feature extraction performance inherit in the input as well as achieving satisfactory mapping accuracy [9], [13].

We design Q identical CNN structures to estimate the target DOA angles. We demonstrate the network architecture of the proposed CNN structure in Fig. 2. Each CNN is composed of 17 layers including input and output layers. The overall deep network structure for the q-th subregion can be represented by a non-linear mapping function $\mathbf{\Sigma}_q(\mathbf{X}) : \mathbb{R}^{M\times M\times 3} \rightarrow \mathbb{R}^L$. In particular, we have

$$\mathbf{\Sigma}_q(\mathbf{X}) = f^{(17)}\big(f^{(16)}(\cdots f^{(1)}(\mathbf{X})\cdots)\big) = \mathbf{p}_q, \tag{8}$$

where $f^{(14)}(\cdot)$ denotes a fully connected layer which maps an arbitrary input $\bar{\mathbf{x}} \in \mathbb{R}^{C_x}$ to the output $\bar{\mathbf{y}} \in \mathbb{R}^{C_y}$ by using the weights $\bar{\mathbf{W}} \in \mathbb{R}^{C_x\times C_y}$. Then the c_y-th element of the output of the layer can be given by the inner product

$$\bar{\mathbf{y}}_{c_y} = \langle \bar{\mathbf{W}}_{c_y}, \bar{\mathbf{x}} \rangle = \sum_i [\bar{\mathbf{W}}]_{c_y,i}^T \bar{\mathbf{x}}_i, \tag{9}$$

for $c_y = 1, \ldots, C_y$ and $\bar{\mathbf{W}}_{c_y}$ is the c_y-th column vector of $\bar{\mathbf{W}}$ and $C_x = C_y = 1024$ is selected for $f^{(14)}(\cdot)$.

In (8), $\{f^{(i)}(\cdot)\}_{i=\{2,5,8,11\}}$ represent the convolutional layers. The arithmetic operation of a single filter of a convolutional layer can be defined for an arbitrary input $\bar{\mathbf{X}} \in \mathbb{R}^{d_x\times d_x\times V_x}$ and output $\bar{\mathbf{Y}} \in \mathbb{R}^{d_y\times d_y\times V_y}$ as

$$\bar{\mathbf{Y}}_{p_y,v_y} = \sum_{p_k,p_x} \langle \bar{\mathbf{W}}_{v_y,p_k}, \bar{\mathbf{X}}_{p_x} \rangle, \tag{10}$$

where $d_x \times d_y$ is the size of the convolutional kernel, and $V_x \times V_y$ are the size of the response of a convolutional layer. $\bar{\mathbf{W}}_{v_y,v_k} \in \mathbb{R}^{V_x}$ denotes the weights of the v_y-th convolutional kernel, and $\bar{\mathbf{X}}_{p_x} \in \mathbb{R}^{V_x}$

[1]While (3) requires the uncorrelated signal assumption, the proposed DL approach can also work well for correlated/coherent signals since the non-linear mapping provided by DeepMUSIC does not rely on the statistical properties of the signal [13]. To generate the output label for a coherent scenario, the MUSIC spectra can be obtained by employing spatial smoothing algorithm.

Algorithm 1 Training data generation for DeepMUSIC.

Input: J_α, J_β, T, M, Q, K, $\text{SNR}_{\text{TRAIN}}$.
Output: Training data sets $\{\mathcal{D}_q\}_{q=1}^{Q}$.

Generate J_α DOA angle sets $\{\theta_k^{(\alpha)}\}_{k=1}^{K}$ such that $\theta_k^{(\alpha)} \in [\Theta_k^{\text{start}}, \Theta_k^{\text{final}}]$ for $\alpha = 1, \ldots, J_\alpha$.
Initialize with $\mu = 1$ while the dataset length is $J = J_\alpha J_\beta$.
for $1 \le \alpha \le J_\alpha$ **do**
    Construct $\mathbf{A}^{(\alpha)} = [\mathbf{a}(\theta_1^{(\alpha)}), \ldots, \mathbf{a}(\theta_K^{(\alpha)})]$.
    Construct $\tilde{\mathbf{R}}_{\text{y}}^{(\alpha)} = \mathbf{A}^{(\alpha)} \mathbf{\Gamma}^{(\alpha)} \mathbf{A}^{(\alpha)}$.
    Using $\tilde{\mathbf{R}}_{\text{y}}^{(\alpha)}$, obtain noise subspace $\mathbf{U}_{\text{N}}^{(\alpha)}$.
    Compute $P^{(\alpha)}(\theta)$ in (4) for $\theta \in [\Theta^{\text{start}}, \Theta^{\text{final}}]$ and construct $\mathbf{p}^{(\alpha)}$ and the partitioned spectra $\{\mathbf{p}_q^{(\alpha)}\}_{q=1}^{Q}$.
    for $1 \le \beta \le J_\beta$ **do**
        Generate $s_k^{(\alpha,\beta)}(t_i) \sim \mathcal{CN}(\mathbf{0}_K, \mathbf{I}_K)$ for T snapshots.
        Generate noisy array output $\mathbf{y}^{(\alpha,\beta)}(t_i) = \sum_{k=1}^{K} \mathbf{a}(\theta_k^{(\alpha)}) s_k^{(\alpha,\beta)}(t_i) + \mathbf{n}^{(\alpha,\beta)}(t_i)$, where $\mathbf{n}^{(\alpha,\beta)} \sim \mathcal{CN}(\mathbf{0}_M, \sigma^2_{\text{TRAIN}} \mathbf{I}_M)$.
        Construct sample covariance matrix $\mathbf{R}_{\text{y}}^{(\alpha,\beta)} = \frac{1}{T} \sum_{i=1}^{T} \mathbf{y}^{(\alpha,\beta)}(t_i) \mathbf{y}^{(\alpha,\beta)\text{H}}(t_i)$.
        Form the input data $\mathbf{X}^{(\mu)}$ as
        $[[\mathbf{X}^{(\mu)}]_{:,:,1}]_{i,j} = \text{Re}\{[\mathbf{R}_{\text{y}}^{(\alpha,\beta)}]_{i,j}\}$.
        $[[\mathbf{X}^{(\mu)}]_{:,:,2}]_{i,j} = \text{Im}\{[\mathbf{R}_{\text{y}}^{(\alpha,\beta)}]_{i,j}\}$.
        $[[\mathbf{X}^{(\mu)}]_{:,:,3}]_{i,j} = \angle\{[\mathbf{R}_{\text{y}}^{(\alpha,\beta)}]_{i,j}\}$.
        Form the output of the q-th CNN as $\mathbf{z}_q^{(\mu)} = \mathbf{p}_q^{(\alpha)}$.
        Design input-output pair for q-th CNN as $(\mathbf{X}^{(\mu)}, \mathbf{z}_q^{(\mu)})$.
        $\mu = \mu + 1$.
    end for β,
end for α,
Training data for the q-th CNN is obtained from the collection of the input-output pairs as $\mathcal{D}_q = \big((\mathbf{X}^{(1)}, \mathbf{z}_q^{(1)}), (\mathbf{X}^{(2)}, \mathbf{z}_q^{(2)}), \ldots, (\mathbf{X}^{(J)}, \mathbf{z}_q^{(J)})\big)$.

is the input feature map at spatial position p_x. Hence we define p_x and p_k as the 2-D spatial positions in the feature maps and convolutional kernels, respectively [10], [15]. In the proposed architecture, we use 256 filters, the first two of which are of size 5×5 and the remaining two have 3×3 filters.

In (8), $\{f^{(i)}(\cdot)\}_{i=\{3,6,9,12\}}$ are the normalization layers and $\{f^{(i)}(\cdot)\}_{i=\{4,7,10,13\}}$ are the rectified linear unit (ReLU) layers which are defined as $\text{ReLU}(x) = \max(0, x)$. $f^{(15)}(\cdot)$ is a dropout layer and $f^{(16)}(\cdot)$ is a softmax layer defined for an arbitrary input $\bar{\mathbf{x}} \in \mathbb{R}^D$ as $\text{softmax}(\bar{x}_i) = \frac{\exp\{\bar{x}_i\}}{\sum_{i=1}^{D} \exp\{\bar{x}_i\}}$. Finally, the output layer $f^{(17)}(\cdot)$ is a regression layer of size $L \times 1$. The current network parameters are obtained from a hyperparameter tuning process providing the best performance for the considered scenario [9], [13].

The proposed deep networks are realized and trained in MATLAB on a PC with a single GPU and a 768-core processor. We used the stochastic gradient descent algorithm with momentum 0.9 and updated the network parameters with learning rate 0.01 and mini-batch size of 128 samples. Then, we reduced the learning rate by the factor of 0.5 after each 10 epochs. We also applied a stopping criteria in training so that the training ceases when the validation accuracy does not improve in three consecutive epochs. Algorithm 1 summarizes the training data generation.

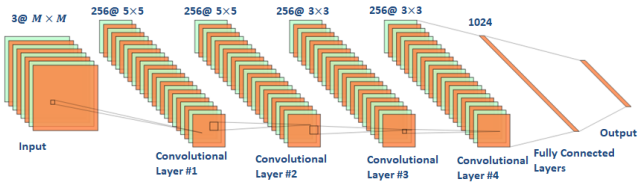

Fig. 2. Deep network architecture for the proposed algorithm. Each convolutional layer block also includes normalization and ReLU layers.

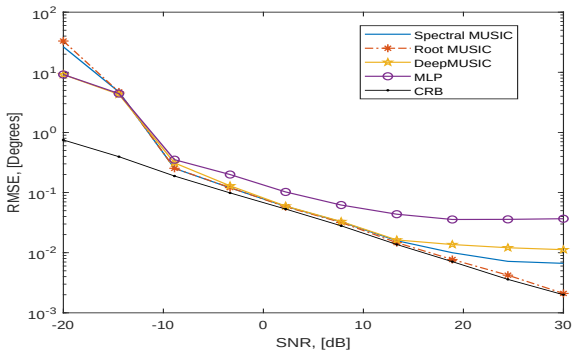

Fig. 3. DOA estimation performance versus SNR for $K = 2$.

IV. NUMERICAL SIMULATIONS

In this section, we present the performance of the proposed DeepMUSIC algorithm in comparison with the MLP structure in [4], the MUSIC algorithm [2] and the Cramer-Rao lower Bound (CRB) [16]. In the training stage, we use the angular spectrum as $[\Theta^{\text{start}}, \Theta^{\text{final}}] = [-60°, 60°]$ [4] with $N = 2^{12}$ grid points. We select $K = 5$ and the target DOAs are randomly located in $Q = 8$ subregions. In particular, J_α DOA angle sets $\{\theta_k^{(\alpha)}\}_{k=1}^{K}$ are realized for $\alpha = 1 \ldots, J_\alpha$ and the DOA angles are selected as the angles drawn uniform randomly from Θ. Then, for each realization, noisy array outputs $\mathbf{y}^{(\alpha,\beta)}(t_i)$ are generated for $\beta = 1, \ldots, J_\beta$. We select $J_\alpha = J_\beta = 100$ and $T = 500$ in out simulations. When generating the data, we use four different SNR levels, i.e., $\text{SNR}_{\text{TRAIN}} = \{15, 20, 25, 30\}$ dB where $\text{SNR}_{\text{TRAIN}} = 10 \log_{10}(\sigma_{\text{S}}^2 / \sigma_{\text{TRAIN}}^2)$ and $\sigma_{\text{S}}^2 = 1$. Hence, the total data set length is $J = 4 J_\alpha J_\beta = 40000$. Further, 80% and 20% of all generated data are chosen for training and validation datasets, respectively. For the prediction process, we select the DOA angles as the floating angles generated uniform randomly in the subregions defined above. Then, $J_T = 100$ Monte Carlo experiments are conducted to obtain the statistical performance of the proposed DeepMUSIC framework. Throughout the simulations, we use a ULA with $M = 16$ antennas with $\lambda = \bar{d}/2$, $T = 100$.

In Fig. 3 and Fig. 4 we present the DOA estimation performance of the algorithms for $K = 2$ and $K = 6$ respectively. When $K = 2$, we see that DeepMUSIC outperforms MLP and provides very close performance to the spectral and Root-MUSIC algorithms respectively. When $K = 6$, it can be seen that DeepMUSIC performs better than the MUSIC algorithms in low SNR regimes and closely follows the performance of the MUSIC algorithms as SNR increases. The performance of DeepMUSIC can be attributed to the use of convolutional layers which extract the hidden features in the input data whereas the MLP architecture consists of only fully connected layers which do not provide effective feature extraction. Furthermore,

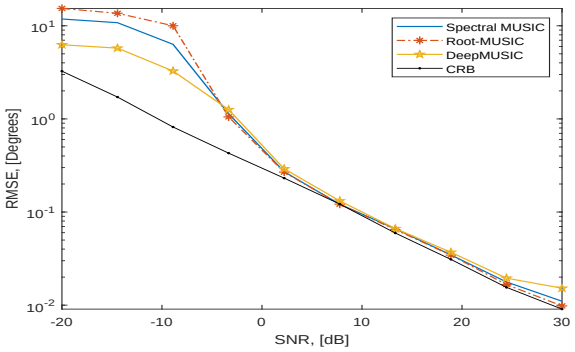

Fig. 4. DOA estimation performance versus SNR for $K = 6$.

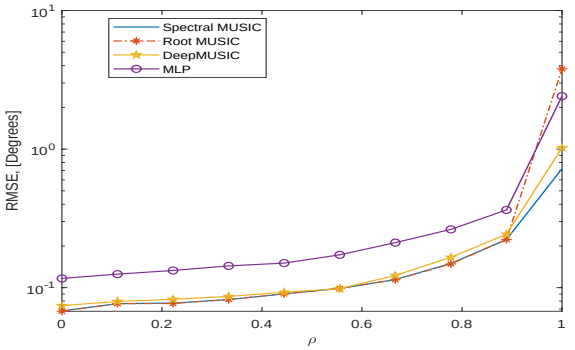

Fig. 5. DOA estimation performance versus correlation coefficient.

MLP only works for two signals which brings a strong limitation for practicality. In contrast, the proposed DeepMUSIC framework can handle multiple target scenario and it exhibits a reasonable performance.

We can also see from Fig. 3-4 that, for high SNR regimes, i.e., SNR$\geq$ 20 dB, the performance of the DL-based methods and the spectral MUSIC algorithm maxes out and does not improve. One of the main reasons of this is due to reaching the angular resolution limit[2] $\gamma = |-60-60|/2^{12} \approx 0.02$, which limits the performance of spectral approaches except Root-MUSIC. Moreover, the performance loss for DeepMUSIC and MLP is due to the loss of precision in the deep networks. This is because, being biased estimators, deep networks do not provide unlimited accuracy. This problem can be mitigated by increasing the number of units in various network layers. Unfortunately, it may lead to the network memorizing the training data and perform poorly when the test data are different than the ones in training. To balance this trade-off, we used noisy data-sets with several $\text{SNR}_{\text{TRAIN}}$ levels during training so that the network attains reasonable tolerance to corrupted/imperfect inputs. While similar performance degradation is also observed in [5], [6], no justification is provided for this issue.

In Fig. 5, the DOA estimation performance is presented with the same simulation settings when there are $K = 2$ correlated target signals as $\mathbf{\Gamma} = \begin{bmatrix} \sigma_1^2 & \rho \\ \rho & \sigma_2^2 \end{bmatrix}$ where ρ is the correlation coefficient. We can see that DeepMUSIC closely follows the MUSIC algorithm and provides less RMSE when fully correlated case (i.e., $\rho = 1$).

[2]The resolution limit γ can also be viewed as the angular search step size of the MUSIC algorithm [2].

We also compare the computation complexity of the DOA estimation algorithms for the same settings and $K = 2$. We observe that DeepMUSIC and MLP take about 0.0020 s and 0.0110 s whereas spectral MUSIC and Root-MUSIC need 0.0300 s and 0.0040 s to obtain the MUSIC spectra. These results show that DeepMUSIC has the lowest computation time as compared to the competing algorithms. While DeepMUSIC has multiple networks as compared to MLP, it provides less computation time due to the fact that 1) DeepMUSIC has many convolutional layers rather than fully connected layers which involve higher complexity [17] and 2) the multiple networks in DeepMUSIC can be trained and run with parallel processing so that the computational complexity is reduced. Similar observations are also reported in [5].

V. SUMMARY

In this letter, we introduced a DL framework called DeepMUSIC for DOA estimation. The major advantage of the proposed approach is that it can work for multiple targets in comparison with the previous works. Furthermore, DeepMUSIC provides less computational complexity as compared to the conventional techniques.

REFERENCES

[1] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *Signal Processing Magazine, IEEE*, vol. 13, no. 4, pp. 67–94, Jul 1996.

[2] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280, 1986.

[3] B. Friedlander and A. Weiss, "Direction finding in the presence of mutual coupling," *IEEE Trans. Antennas Propag.*, vol. 39, no. 3, pp. 273–284, 1991.

[4] Z. Liu, C. Zhang, and P. S. Yu, "Direction-of-arrival estimation based on deep neural networks with robustness to array imperfections," *IEEE Trans. Antennas Propag.*, vol. 66, no. 12, pp. 7315–7327, Dec 2018.

[5] L. Wu, Z. Liu, and Z. Huang, "Deep Convolution Network for Direction of Arrival Estimation With Sparse Prior," *IEEE Signal Process. Lett.*, vol. 26, no. 11, pp. 1688–1692, Nov 2019.

[6] D. Hu, Y. Zhang, L. He, and J. Wu, "Low-Complexity Deep-Learning-Based DOA Estimation for Hybrid Massive MIMO Systems with Uniform Circular Arrays," *IEEE Wireless Commun. Lett.*, pp. 1–1, 2019.

[7] H. Huang, Y. Peng, J. Yang, W. Xia, and G. Gui, "Fast beamforming design via deep learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 1065–1069, Jan 2020.

[8] Y. Wang, M. Liu, J. Yang, and G. Gui, "Data-Driven Deep Learning for Automatic Modulation Recognition in Cognitive Radios," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4074–4077, April 2019.

[9] A. M. Elbir and K. V. Mishra, "Joint Antenna Selection and Hybrid Beamformer Design using Unquantized and Quantized Deep Learning Networks," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2019.

[10] A. Massa, D. Marcantonio, X. Chen, M. Li, and M. Salucci, "DNNs as Applied to Electromagnetics, Antennas, and Propagation-A Review," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2225–2229, Nov 2019.

[11] L. Wu and Z. Huang, "Coherent svr learning for wideband direction-of-arrival estimation," *IEEE Signal Process. Lett.*, vol. 26, no. 4, pp. 642–646, April 2019.

[12] Q. Li, X. Zhang, and H. Li, "Online direction of arrival estimation based on deep learning," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 2616–2620.

[13] A. M. Elbir, K. V. Mishra, and Y. C. Eldar, "Cognitive radar antenna selection via deep learning," *IET Radar, Sonar & Navigation*, vol. 13, pp. 871–880, 2019.

[14] A. M. Elbir, S. Mulleti, R. Cohen, R. Fu, and Y. C. Eldar, "Deep-sparse array cognitive radar," in *IEEE International Conference on Sampling Theory and Applications*, 2019, pp. 1–5.

[15] J. Cheng, J. Wu, C. Leng, Y. Wang, and Q. Hu, "Quantized CNN: A unified approach to accelerate and compress convolutional networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4730–4743, 2018.

[16] P. Stoica and A. Nehorai, "MUSIC, maximum likelihood, and Cramér-Rao bound: Further results and comparisons," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 12, pp. 2140–2150, 1990.

[17] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.