2018 15th Workshop on Positioning, Navigation and Communications (WPNC)

DOA Estimation of Two Targets with Deep Learning

Yuya Kase[†], Toshihiko Nishimura[†], Takeo Ohgane[†], Yasutaka Ogawa[†],
Daisuke Kitayama[††], and Yoshihisa Kishiyama[††]

[†] Graduate School of Information Science and Technology, Hokkaido University
Kita 14, Nishi 9, Kita-ku, Sapporo, Hokkaido, 060–0814 Japan
Email: {kase, nishim, ohgane, ogawa}@m-icl.ist.hokudai.ac.jp

[††] Research Laboratories, NTT DOCOMO, INC.
Hikarinooka 3-6, Yokosuka, Kanagawa, 239–8536, Japan
Email: {daisuke.kitayama.dy, kishiyama}@nttdocomo.com

***Abstract*—Direction of arrival (DOA) estimation of radio waves is demanded in many situations. In addition to MUSIC and ESPRIT, which are well-known traditional algorithms, compressed sensing has been recently applied to DOA estimation. If a large computational load as seen in some of compressed sensing algorithms is acceptable, it may be possible to apply deep learning to DOA estimation. In this paper, we propose estimating DOAs using deep learning and discuss training data preparation and designing for a specific scenario. The simulation results show reasonably-high estimation accuracy, performance dependency on training data preparation, and effectivity of specialized deep neural network.**

I. Introduction

Direction of arrival (DOA) estimation of radio waves is applicable to many cases including localization of users in mobile communication and a radar mounted on an automobile. As an estimation method using an array antenna, methods exploiting the eigen structure of the correlation matrix of received signals such as multiple signal classification (MUSIC) and estimation of signal parameters via rotation invariance (ESPRIT) techniques are widely known [1].

In recent years, high resolution estimation methods based on compressed sensing have also been proposed [2]. Compressed sensing is generally described as a method of estimating an original signal, which is given by a sparse vector, from fewer observations than dimensions of the signal vector [3]. In an example of DOA estimation using a compressed sensing solver called as half-quadratic regularization (HQR), higher accuracy has been reported compared with MUSIC [4].

Nonlinear algorithms such as compressed sensing require a heavy computational load in general. Actually, HQR needs an iterative calculation of an inverse matrix of which dimension is the same as that of the original signal vector. For another nonlinear algorithm, in recent years, deep learning has been applied to various application fields along with the development of computer technology, and its very high capability in problem solving is now widely-known.

Deep learning is a machine learning technique using a deep neural network (DNN) with multiple layers, and has been extensively studied in the fields of images, sounds, languages, etc., starting with the research by Hinton et al. [5], [6]. Although the DNN requires a large amount of calculations during the learning process, estimation using the learned network can be performed with simple calculation which mainly consists of multiplication of a matrix and a vector. Thus, it can be said that use of the DNN for DOA estimation is an attractive application when a heavy computational load is acceptable.

In this paper, we apply a DNN to DOA estimation and evaluate the estimation performance under a scenario where two equal-power and uncorrelated signals are incident on a uniform linear array. In addition to discussion on types of training data, we consider designing the DNN for a specific scenario involving two close incident waves with close DOAs and a parallel use of multiple DNNs designed for different scenarios. In the rest of paper, the array structure and signal arrival model, the configuration and training process of DNN, and performance evaluation by computer simulation are described.

II. Correlation Matrix of Received Signal

We assume that K plane waves with a wavelength λ and a complex amplitude $s_k(t)$ are incident at an angle θ_k and a time t on an L-element uniform linear array with an element spacing d as shown in Fig. 1. The received signal at the lth antenna can be expressed as

$$x_l(t) = \sum_{k=1}^{K} s_k(t) e^{-j\frac{2\pi}{\lambda}(l-1)d\sin\theta_k} + n_l(t), \tag{1}$$

where $n_l(t)$ is an additive white noise at the lth antenna.

Defining an $L \times K$ matrix $\boldsymbol{A}$ of which the (l,k)th entry is

$$a_{l,k} = e^{-j\frac{2\pi}{\lambda}(l-1)d\sin\theta_k}, \tag{2}$$

we can rewrite (1) as

$$\begin{aligned} \boldsymbol{x}(t) &= [x_1(t),\, x_2(t), \ldots,\, x_L(t)]^T \\ &= \boldsymbol{A}\boldsymbol{s}(t) + \boldsymbol{n}(t), \end{aligned} \tag{3}$$

978-1-5386-6436-0/18/$31.00 ©2018 IEEE

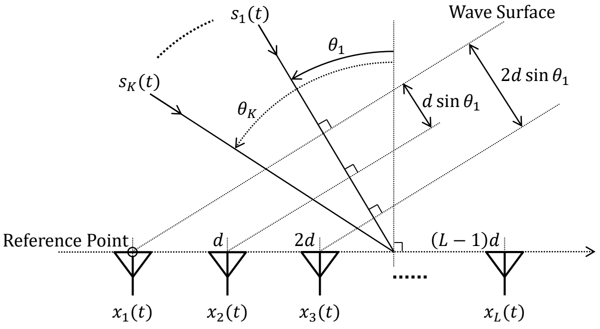

Fig. 1. An L-element uniform linear array and incident waves.

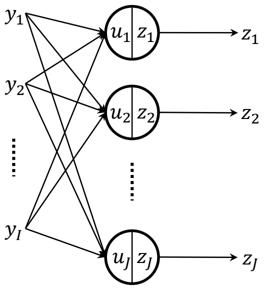

Fig. 2. A factor graph for a single layer neural network.

where $[\cdot]^T$ denotes the transpose and

$$\boldsymbol{s}(t) = [s_1(t),\, s_2(t), \ldots,\, s_K(t)]^T \tag{4}$$

$$\boldsymbol{n}(t) = [n_1(t),\, n_2(t), \ldots,\, n_L(t)]^T. \tag{5}$$

The $L \times L$ correlation matrix of received signal vector $\boldsymbol{x}(t)$ is expressed as

$$\boldsymbol{R}_{xx} = E\big[\boldsymbol{x}(t)\,\boldsymbol{x}^H(t)\big] = \boldsymbol{A}\boldsymbol{S}\boldsymbol{A}^H + \boldsymbol{R}_N, \tag{6}$$

where $E[\cdot]$ and $[\cdot]^H$ denote the ensemble average and the conjugate transpose, respectively. $\boldsymbol{S}$ and $\boldsymbol{R}_N$ are the $K \times K$ signal and noise correlation matrices, respectively, and are given by

$$\boldsymbol{S} = E\big[\boldsymbol{s}(t)\,\boldsymbol{s}^H(t)\big] \tag{7}$$

$$\boldsymbol{R}_N = E\big[\boldsymbol{n}(t)\,\boldsymbol{n}^H(t)\big]. \tag{8}$$

We assume that all the noise components are mutually uncorrelated and have the same power. Thus, we have

$$\boldsymbol{R}_N = \sigma^2 \boldsymbol{I}, \tag{9}$$

where $\boldsymbol{I}$ is the identity matrix and σ^2 is the noise power. Then, (6) can be rewritten as

$$\boldsymbol{R}_{xx} = \boldsymbol{A}\boldsymbol{S}\boldsymbol{A}^H + \sigma^2 \boldsymbol{I}. \tag{10}$$

Note that this receive correlation matrix is a Hermitian matrix. As will be described later, the correlation matrix is used as an input of our DNN.

III. DOA Estimation with Deep Learning

A. Formulation of Deep Neural Network

In general, a single layer neural network of J units with I inputs and J outputs can be illustrated as Fig. 2. The output of the jth unit z_j can be expressed as

$$z_j = f(u_j) \tag{11}$$

$$u_j = \sum_{i=1}^{I} w_{j,i} y_i + b_j, \tag{12}$$

where f, y_i, $w_{j,i}$, and b_j are all real-valued and represent an activation function, the ith input, the weight for the jth unit multiplied by y_i, and a constant bias, respectively. We can modify (12) by assuming the 0th input $y_0 = 1$ and the corresponding weight $w_{j,0} = b_j$ as

$$u_j = \sum_{i=0}^{I} w_{j,i} y_i. \tag{13}$$

These relationships can be expressed in a vector-matrix form as

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_J \end{bmatrix} = \begin{bmatrix} f(u_1) \\ f(u_2) \\ \vdots \\ f(u_J) \end{bmatrix} \tag{14}$$

$$\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_J \end{bmatrix} = \begin{bmatrix} w_{1,0} & w_{1,1} & \cdots & w_{1,I} \\ w_{2,0} & w_{2,1} & \cdots & w_{j,I} \\ \vdots & \vdots & & \vdots \\ w_{J,0} & w_{J,1} & \cdots & w_{J,I} \end{bmatrix} \begin{bmatrix} 1 \\ y_1 \\ \vdots \\ y_I \end{bmatrix}, \tag{15}$$

and simplified as

$$\boldsymbol{z} = f(\boldsymbol{u}) \tag{16}$$

$$\boldsymbol{u} = \boldsymbol{W} \begin{bmatrix} 1 \\ \boldsymbol{y} \end{bmatrix}. \tag{17}$$

DNN has a multilayer structure. Fig. 3 shows a DNN example where M single layer neural networks are stacked. The output of the mth layer can be expressed as

$$\boldsymbol{z}^{(m)} = f\left(\boldsymbol{u}^{(m)}\right), \tag{18}$$

where

$$\boldsymbol{u}^{(m)} = \boldsymbol{W}^{(m)} \begin{bmatrix} 1 \\ \boldsymbol{z}^{(m-1)} \end{bmatrix}, \tag{19}$$

and $\boldsymbol{z}^{(0)} = \boldsymbol{y}$.

B. DOA Estimation

In the paper, we use the $L \times L$ correlation matrix $\boldsymbol{R}_{xx}$ as an input of a DNN for DOA estimation. Since $\boldsymbol{R}_{xx}$ is a Hermitian matrix, the upper and lower triangular parts have the same information. Thus, we use only the lower triangular part. However, it is needed to decompose the complex value of

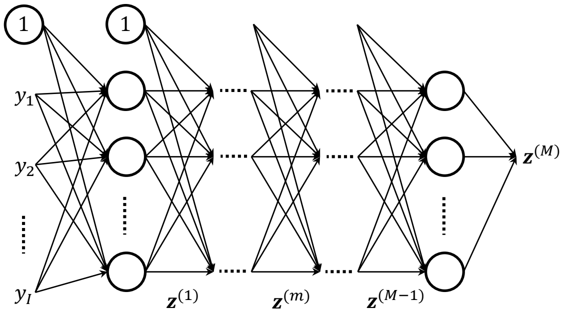

Fig. 3. A factor graph expression of a deep neural network.

each entry of $\boldsymbol{R}_{xx}$ into two real values except for the diagonal elements. Then, the input vector of the DNN is written as

$$\boldsymbol{y} = \Big[r_{1,1},\, r_{2,2}, \ldots,\, r_{L,L},\, \Re(r_{2,1}),\, \Im(r_{2,1}), \\ \Re(r_{3,1}),\, \Im(r_{3,1}),\, \Re(r_{3,2}), \ldots,\, \Im(r_{L,L-1}) \Big]^T, \qquad (20)$$

where $\Re(\cdot)$ and $\Im(\cdot)$ denote the real and imaginary parts, respectively. The dimension of the column vector $\boldsymbol{y}$, i.e., the number of inputs becomes L^2.

On the other hand, the number of output units depends on a required angle resolution and search range. In the paper, we relate each output to a certain unique angle within the search range. When the angle resolution is $\Delta\theta$ and the search range is between θ_{min} and θ_{max}, the number of output units becomes $(\theta_{\text{max}} - \theta_{\text{min}})/\Delta\theta + 1$. Each output value is defined to express the probability [%] that a plane wave is incident at the corresponding angle. In the training phase, each output value is set as

$$z_j^{(M)} = \begin{cases} 100 & \text{when a plane wave is incident} \\ 0 & \text{otherwise} \end{cases}. \qquad (21)$$

IV. Evaluation of Estimation Accuracy

A. Simulation Conditions

We numerically evaluated the DOA estimation performance of the DNN. The followings are parameters used here. The number of targets is two. It was assumed that both signals from the targets had equal power and were uncorrelated. Also, the narrowband assumption was used for (1) to hold. The center frequency was set to 2 GHz. The DOA of each plane wave ranged from −60° to 60°. A total of 100,000 random tests were evaluated. Although each DOA was set to a random integer angle per test, we coordinated two DOAs not to be coincident. The number of array elements was five, and the element spacing was half-wavelength of 2 GHz. We used 100 snapshots of received signals to calculate the correlation matrix in each test.

According to the number of antennas, 25 inputs are connected to the DNN. We assumed that required angle resolution was 1° and that the search range of DOA was between −60° and 60°. Thus, 121 output units are prepared to assign each output to one of integer angles from −60° to 60°. The number of intermediate layers is four where each layer has 150 units. The activation function of all units is set to a ramp function shown in Fig. 4, and we have

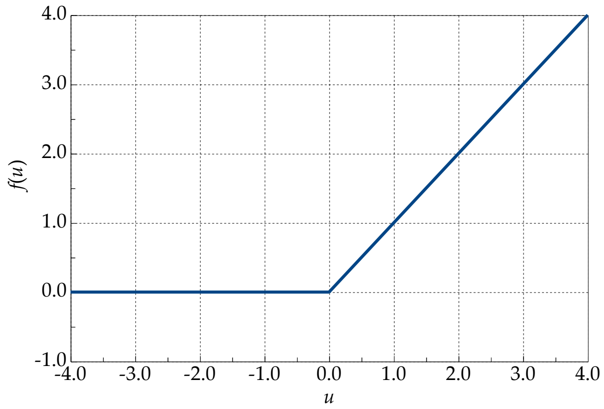

Fig. 4. A ramp function.

$$f(u) = \max(u, 0) = \begin{cases} u & \text{for } u \geq 0 \\ 0 & \text{elsewhere} \end{cases}, \qquad (22)$$

except for the output layer where the identity function is used. Online learning based on back propagation was applied to the DNN using 3,000,000 training data where each training data sample was generated by setting two different random integer DOAs within the search range and white Gaussian noise. Adaptive moment estimation [7] was used to determine the learning rate.

The estimation performance is evaluated using three measures: estimation success rate, root mean square error (RMSE), and absolute value of error. The estimation success is defined as the case where both estimated DOAs are equal to the true value. RMSE is expressed as

$$\text{RMSE} = \sqrt{\frac{1}{KN} \sum_{k=1}^{K} \sum_{n=1}^{N} \left(\hat{\theta}_k^{(n)} - \theta_k^{(n)} \right)^2}, \qquad (23)$$

where $[\cdot]^{(n)}$ and N denotes the nth test and the number of tests, respectively.

B. Dependency on Training Data

When generating the training data, the SNR was changed in pre-determined patterns:

(i) constant 0 dB (“0 dB”)
(ii) constant 30 dB (“30 dB”)
(iii) increased linearly from 0 dB to 30 dB (“increase”)
(iv) increased in 5 dB step from 0 dB to 30 dB (“stepwise”)
(v) decreased linearly from 30 dB to 0 dB (“decrease”)
(vi) random in the range between 0–30 dB (“random”)

Each SNR transition is illustrated in Fig. 5.

Estimation success rates and RMSEs of DNNs trained according to the SNR patterns (i) to (vi) are shown in Figs. 6 and 7. Note that the abscissa represents SNR not for the

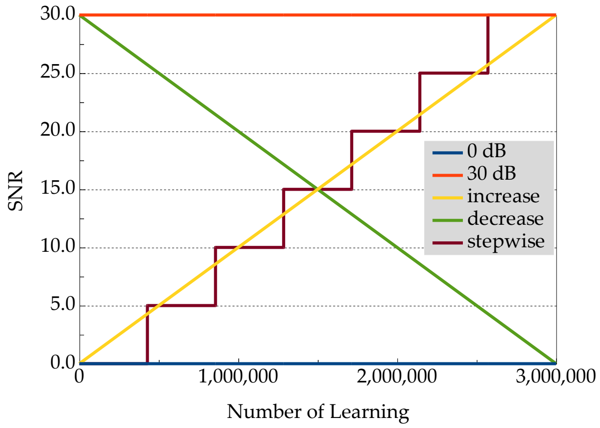

Fig. 5. SNR transition in each training pattern. (The random case is not shown.)

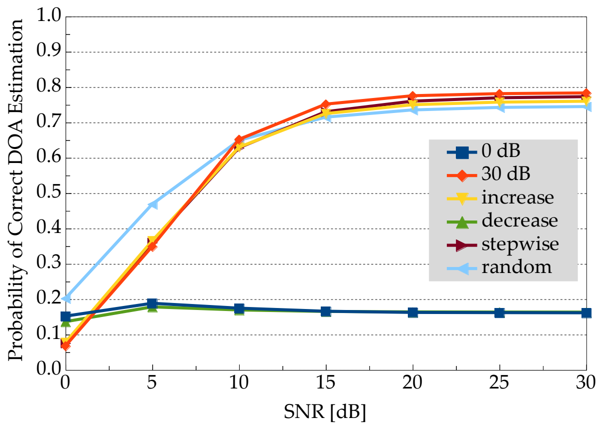

Fig. 6. Success rate performance for each SNR transition pattern.

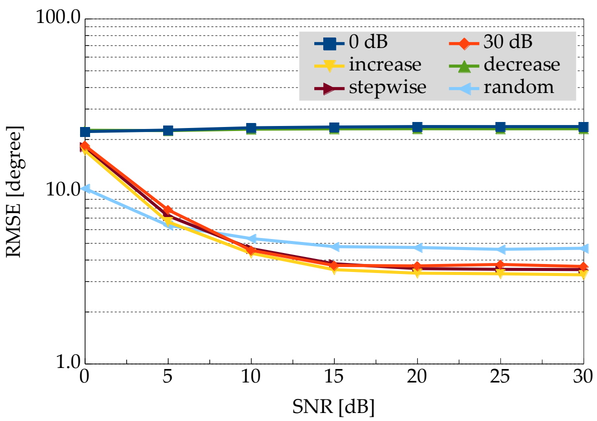

Fig. 7. RMSE for each SNR transition pattern.

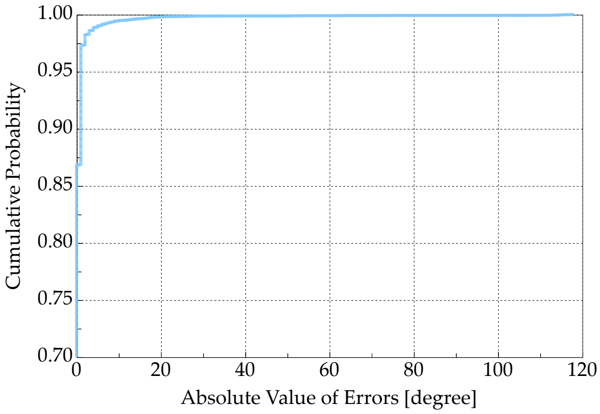

Fig. 8. Cumulative probability of absolute value of errors.

training phase but for the estimation one. These figures clearly indicate that the DNNs trained using high SNR data in the later phase of training provide better performance. Conversely, the DNNs trained using low SNR data in the later phase does not work well. It is supposed that featureless correlation matrices generated by the low SNR data confuse the learning process. The DNN trained using random SNR data achieves a slightly lower estimation accuracy at the SNR of 30 dB. However, the estimation accuracy at the SNR of less than 10 dB is higher than that of the others. It can be said that training with random SNR data produces the DNN suitable for a wide SNR range.

Fig. 8 shows the cumulative probability of the absolute value of errors at the SNR of 30 dB for the case using the DNN trained with random SNR data. Let us recall that the estimation status is flagged as “success” only when both DOAs are correctly estimated. Specifically, Fig. 6 shows that about 25% of tests (100,000 in total) fails in correct estimation. However, Fig. 8 indicates that zero error occurs in about 86% of waves (200,000 in total) and that the probability that the error falls within 1° exceeds 97%. The online learning with 3,000,000 training data may be regarded as a very complicated process. However, it should be emphasized that the obtained DNN achieves high accuracy as in Fig. 8 with a reasonably low complexity in the estimation phase.

C. DNN for Close DOA Scenario

Here, a DNN designed for a special case is discussed. Specifically, we consider the close DOA case of $\theta_1-\theta_2=\pm 1^\circ$. For both training data and test data, the restriction of 1° difference was applied. The estimation success rate using this DNN is shown in Fig. 9. As a reference, the success rate for the DNN trained using random SNR data without restriction on DOAs is also shown. In general, it is very difficult to estimate DOAs of which difference is 1°. Actually, the success rate of the DNN trained using random DOA data is less than 20%. In contrast, the DNN trained using special training data with the restriction $\theta_1-\theta_2=\pm 1^\circ$ achieves the success rate of about 99% at the SNR of 30 dB. The fact that we can design the DNN for specific scenarios is unique to machine-learning-based estimation and shows potential capability of complementary use of such DNNs with other techniques.

D. Parallel Use of General-Purpose and Specialized DNNs

As described above, the specialized DNN for the close DOA scenario (1° DOA difference) can be used with other

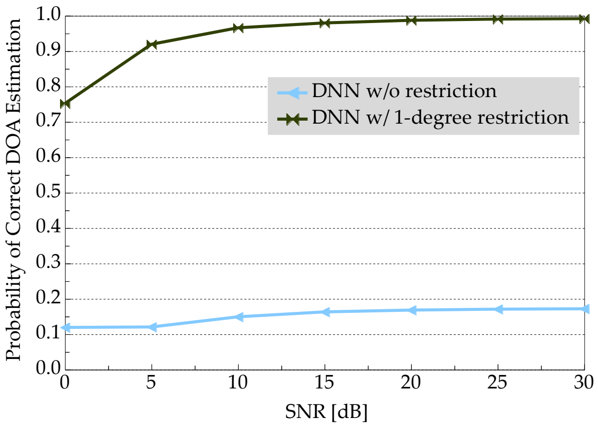

Fig. 9. Success rate performance when $\theta_1 - \theta_2 = \pm 1^\circ$.

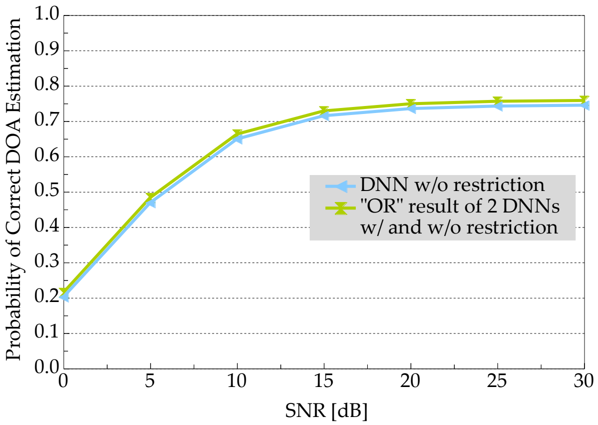

Fig. 10. Success rate performance for parallel use of DNNs.

DOA estimators. Here, we use two DNNs trained using random DOA data and special training data with the restriction $\theta_1 - \theta_2 = \pm 1^\circ$ in parallel. We simply assumed that the estimation status was flagged as "success" when at least one of two DNNs estimated correct DOAs, i.e. the "OR" result of two DNNs' success flags, although the method to select the proper DNN was not yet developed.

Fig. 10 shows the success rates for the parallel use case and for the single use case of the "general-purpose" DNN trained with random DOA data. Since the probability that 1° DOA difference occurs is about 1.7%, preparation for such a case may not be so important. However, the parallel use certainly improves the success rate by about 1.7% compared with the single use of the general-purpose DNN. This property implies the possibility of performance improvement by combination of specially-trained DNNs.

V. Conclusions

In this paper, we have proposed an estimation method using deep learning and examined its basic performance in the case where two narrowband signals are incident on a linear array. The simulation results have shown reasonably-high estimation accuracy of the DNN. In addition, it has been indicated that the DNN designed for a specific scenario achieves very high success rate in the same scenario. An integrated use of such specialized DNNs is an urgent and interesting topic for us.

References

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

[2] A Massa, P. Rocca, and G. Oliveri, "Compressive sensing in electromagnetics — a review," *IEEE Antennas Propag. Mag.*, vol. 57, no. 1, pp. 224–238, Feb. 2015.

[3] K. Hayashi, M. Nagahara, and T. Tanaka, "A user's guide to compressed sensing for communications systems," *IEICE Trans. Commun.*, vol. E96-B, no. 3, pp. 685–712, Mar. 2013.

[4] T. Terada, T. Nishimura, Y. Ogawa, T. Ohgane, and H. Yamada, "DOA estimation for multi-band signal sources using compressed sensing techniques with Khatri-Rao processing," *IEICE Trans. Commun.*, vol. E97-B, no. 10, pp. 2110–2117, Oct. 2014.

[5] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1544, July 2006.

[6] G. E. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, July 2006.

[7] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980v9, Jan. 2017.