UNSUPERVISED ADAPTATION OF DEEP NEURAL NETWORKS FOR SOUND SOURCE LOCALIZATION USING ENTROPY MINIMIZATION

Ryu Takeda and Kazunori Komatani

Osaka University, The Institute of Scientific and Industrial Research
8-1, Mihogaoka, Ibaraki, Osaka 567-0047, Japan

ABSTRACT

This paper describes an unsupervised method of adapting deep neural networks (DNNs) for sound source localization (SSL). DNNs-based SSL achieves high localization accuracy for sound data that are similar to training data. However, the accuracy deteriorates if a sound source is at an unknown position in unknown reverberant environments. We solve the problem by using unsupervised adaption of the DNNs' parameters to the observed sound signals. Entropy is used as the objective function and minimized to optimize the parameters on the basis of the gradient method. Adaptation without overfitting is achieved by using 1) a parameter adaptation layer, such as linear transform network, and 2) early stopping of the parameter updates. Experimental results indicated that our method improved localization accuracy by a maximum of 20 points for unknown positions and reverberant data.

Index Terms— Sound source localization, Deep neural networks, Unsupervised adaptation, Machine learning

1. INTRODUCTION

1.1. Background

Sound source localization (SSL) is the most fundamental function for autonomous robots (or systems) [1], because it enables them to **detect sound events** and **determine their locations**. These capabilities are essential for robots to separate and identify sound sources and determine whether they should react to events. Additionally, robots need both localization accuracy and robustness to unknown environments because they must operate under many different environments.

Sound source localization based on deep neural networks (DNNs) [2, 3, 4, 5] is completely based on machine learning, and it is one of the promising SSL methods for robot audition. DNNs directly estimate the posterior probabilities of the "*position labels*," including that of the presence of sound from multi-channel sound signals. Even a non-specialist can tune DNNs with less effort compared with other localization methods [6, 7] because all the parameters are automatically optimized to a target robot configuration, such as the microphone arrangement and the robot's shape. DNNs-based SSLs can also handle non-speech signals and multiple sound sources [8].

The remaining problem of SSL based on DNNs is performance degradation in different environments from that of the training (Fig. 1); here, some kind of adaptation method is required. DNNs tends to give high probability to the "no sound" label for unknown positions and unknown reverberant environments [5], because they do not match data in the training set. There are two possible approaches to dealing with this problem. The first is multi-condition training by generating various sound data on the basis of a generative model. The other is parameter adaptation of DNNs to the unknown data (see the lower part of Fig.1). Even if we prepare a number of possible data patterns for multi-condition training as the former, unknown patterns will inevitably appear in the real environment.

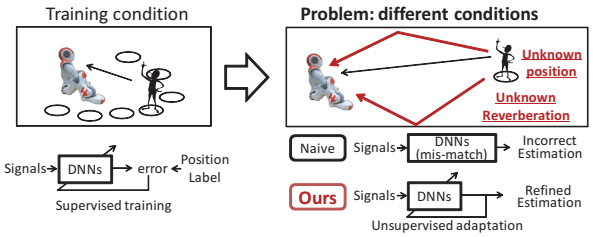

Fig. 1. Problem of SSL based on DNNs and our approach

We propose a segment-wise unsupervised adaptation of DNNs for SSL with less overfitting. Unsupervised adaptation means here that some of the DNNs' parameters are adapted to each *segmented observed signals* without using supervised data in order to improve localization accuracy. Entropy is used as the objective function for unsupervised adaptation because it can be obtained by substituting supervised terms in the cross-entropy cost function with its estimation. We avoid overfitting by using: 1) parameter selection for adaptation (linear input network: LIN [9] with complex-value or the whole fully-connected networks) and 2) early stopping of the parameter update [10]. Without these techniques, the DNNs would output a "no sound" label or false labels in most cases after adaptation. We conducted experiments to assess the performance of the adaptation method for unknown sound locations and unknown reverberant speech after training DNNs *with a massive number of sound position patterns*. Our contributions are 1) the investigation of the effect of complex-domain LIN adaptation and 2) the performance analysis and discussion of DNN-based SSL under various position conditions including different heights and distances.

1.2. Related Work on Adaptation of Deep Neural Networks

Unsupervised adaptation for DNNs-based SSL has not been studied much because the searches of DNNs-based SSL are still rare. Moreover, the task needs incremental/segment-wise unsupervised adaptation similar to blind source separation [11] or blind dereverberation [12] because it must handle *spatial information not the features of source* as in the case in speech recognition. The acoustic environment and source positions also change dynamically. While NNs-based SSL has been widely studied for narrow-band antennas [3], the topic of adaptation has not been treated in this case because the conditions of the antenna usually do not change once it has been

built. The azimuth of sound position is usually estimated.

DNNs used in automatic speech recognition (ASR) have several model adaptation methods that can avoid overfitting mainly for speaker adaptation [13, 14], but these methods all require several utterances, more than five, for adaptation. The unsupervised adaptations are based on a statistical generative model and maximum likelihood estimation, such as constrained maximum likelihood linear regression (CMLLR) [15]. The more popular *semi*-supervised approaches use a LIN or a linear hidden network (LHN) [16]; our adaptation takes advantage of the LIN used in [9] with another objective function. Maximizing a posterior is also used for objective functions to avoid overfitting [17]. These general methods can be used to make further improvements to DNNs-based SSL.

2. DISCRIMINATIVE LOCALIZATION BASED ON DEEP NEURAL NETWORKS

This section is an overview of the SSL based on DNNs we proposed [5]. Note that naïve DNNs fail training when the training data includes various position patterns, such as different heights and distances. In this paper, all the variables in the models are represented in the short-time Fourier transformation (STFT) domain with frame index t and frequency-bin index w [12].

2.1. Input Features of DNNs

The observation model of our approach is used for generating training data of DNNs with measured impulse responses and speech corpora. The arrival process of the sound from M sound sources to N microphones embedded in a robot ($M < N$) is modeled as a linear time-invariant system. The observed signal vector $\boldsymbol{x}_w[t] = [x_{w,1}[t], ..., x_{w,N}[t]]^T$ is represented as

$$\boldsymbol{x}_w[t] = \sum_{m=1}^{M} \boldsymbol{a}_w(\boldsymbol{r}_m) s_{w,m}[t] + \boldsymbol{n}_w[t], \quad (1)$$

where $s_{w,m}[t]$ represents an m-th source sound signal and $\boldsymbol{n}_w = [n_{w,1}[t], ..., n_{w,N}[t]]^T$ is a noise signal vector. The $\boldsymbol{a}_w(\mathbf{r}) = [a_{w,1}(\boldsymbol{r}), ..., a_{w,N}(\boldsymbol{r})]^T$ is an steering vector (SV) from the reference sound position, $\boldsymbol{r}$, to each microphone.

The input feature of DNNs is a set of complex eigenvectors of the correlation matrix of the observed signal vector $\boldsymbol{R}_w = \mathbb{E}[\boldsymbol{x}_w[t]\boldsymbol{x}_w^H[t]]$ at each frequency bin w [18]. The notation $\cdot^H$ denotes the Hermitian transpose and $\mathbb{E}[\cdot]$ means an expectation operator. The eigenvectors and eigenvalues of $\boldsymbol{R}_w$ are obtained by applying eigenvalue decomposition (EVD) and sorted in descending order; $\boldsymbol{E}_w = [\boldsymbol{e}_{w,1}, ..., \boldsymbol{e}_{w,N}] \in \mathbb{C}^{N \times N}$ for the former and $\boldsymbol{\Lambda}_w = \mathrm{diag}[\lambda_{w,1}, ..., \lambda_{w,N}]$ for the latter. Here, $\boldsymbol{e}_{w,i} \in \mathbb{C}^N$ $(i = 1, ..., M)$ corresponds to a basis set of signal space and $\boldsymbol{e}_{w,j} \in \mathbb{C}^N$ $(j = M+1, ..., N)$ corresponds to that of noise space. Finally, the concatenate vector $\boldsymbol{E}$ of eigenvectors in noise space is used as the input feature of DNNs. Here, $\boldsymbol{E} = [\boldsymbol{e}_{w_l,M+1}^T, ..., \boldsymbol{e}_{w_l,N}^T, ..., \boldsymbol{e}_{w_h,M+1}^T, ..., \boldsymbol{e}_{w_h,N}^T]^T$, where w_l and w_h are respectively the lower and upper indices of the frequency bin for localization. This feature is extracted every 110 milliseconds due to the expectation operator (block-wise).

2.2. Posterior Probability Estimation by DNNs

The DNNs estimate the posterior probability $p(z|\boldsymbol{E})$ of discrete variable z with K location symbols from $\boldsymbol{E}$. The discrete symbols (location labels) z are defined by the system developer according to the required resolution of the application. The labels are defined by dividing the space. For example, the label "no sound" represents a no sound source, and the label "0°" represents a sound source located in the range of $[-2.5°, 2.5°]$. The overview of our DNN's configuration is shown in Fig. 2. The main difference from other DNN configurations is that ours directly uses complex features, the phase- and multi-channel structure of audio signals. This configuration can be divided into two phases: 1) the extraction of a *directional image* by using the directional activate functions (DAFs) in the complex domain and 2) the propagation and hierarchical and gradual integration of a directional image in the real domain. The directional image is an activation pattern that differs according to the SVs of sound sources. The DAFs are applied to the input feature $\boldsymbol{x}_{1,w} = [\boldsymbol{e}_{w,M+1}^T, ..., \boldsymbol{e}_{w,N}^T]^T$ at every frequency bin. After that, the output vectors of each layer are gradually integrated by applying affine projection with weight parameters $\boldsymbol{W}_*$ and sigmoid function as usual.

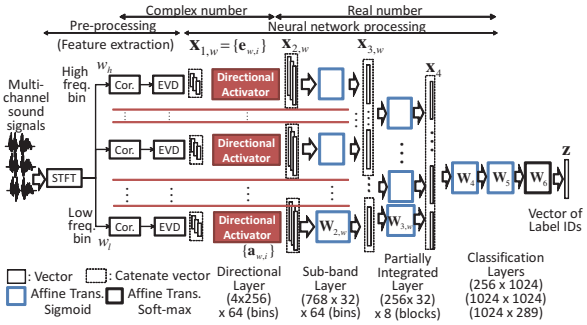

Fig. 2. Configuration of DNNs

The DAFs $\boldsymbol{f}_w(\boldsymbol{x})$ are based on the following inner product that can simultaneously measure intensity and time difference between observation and inner parameters like traditional SSL methods [19].

$$\boldsymbol{f}_w(\boldsymbol{e}_{w,j}) = [f(\boldsymbol{e}_{w,j}; \boldsymbol{a}_{w,1}), ..., f(\boldsymbol{e}_{w,j}; \boldsymbol{a}_{w,N_w})]^T, \text{ and } (2)$$

$$f(\boldsymbol{x}; \boldsymbol{a}) = 1 - |\boldsymbol{a}^H \boldsymbol{x}|/||\boldsymbol{x}||, \quad (3)$$

where $\boldsymbol{a}_{w,j} \in \mathbb{C}^N$ $(j = 1, ..., N_w)$ is a parameter that behaves like an SV, and N_w is the number of parameters at frequency bin w. Note that *DNNs learn these DAFs that work similar to SVs through back propagation*. Therefore, the trained parameters may become compressed expressions of real SVs under appropriate N_w. This parameter is determined experimentally.

2.3. Problem under Unknown Conditions and Our Approach

Our configuration of DNNs for SSL degrades localization accuracy under unknown conditions, and the DNNs will output false labels or the "no sound" label. Since DAFs and fully-connected networks are optimized for training data, they are not suitable for reverberant signals and unknown source positions. Therefore, adaptation methods are required in order to improve the accuracy even in such environments.

Although unsupervised adaptation of parameters is essential for DNN-based SSL, it has a risk of overfitting that degrades localization accuracy. It is important to investigate the performance with standard adaptation methods. We use the entropy as an objective function instead of the cross-entropy because it is the simplest modification that enables unsupervised adaptation. We also exploit two methods to avoid overfitting: 1) parameter selection for adaptation (complex-domain LIN [9] or the whole network) and 2) early stopping of the parameter update [10].

3. UNSUPERVISED ADAPTATION OF PARAMETERS

This section explains how to achieve unsupervised adaptation of DNNs. We assume that the unsupervised adaptation is applied to each segment of the segmented observed signals.

3.1. Entropy Minimization

We use entropy as the objective function J for unsupervised adaptation as a first step because we cannot use supervised data. The cross-entropy, which needs true posterior probabilities, is usually used for discriminative training of DNNs. For our unsupervised training, the "true" probabilities are substituted with their DNNs estimation, and the cross-entropy becomes the "self"-entropy. We assume that the initial estimation are correct to some degree before adaptation, and false estimations are modified through adaptation by using these partial correct estimations.

The definition of entropy J and its derivative is as follows

$$J(\boldsymbol{\Theta}) = \mathbb{E}[-\textstyle\sum_i p_i \log p_i], \quad \dfrac{\partial J}{\partial p_i}(\boldsymbol{\Theta}) = \mathbb{E}[-\log p_i - 1], \qquad (4)$$

where Θ represents all the parameters of the DNNs and p_i represents the estimated location probability from the i-th output node, z_i, of the DNNs in Fig. 2. The expectation means averaging over the samples used for adaptation. By using the chain rule and taking the partial derivatives of the output parameters of each layer, we can calculate the gradient of the target parameters $\boldsymbol{\theta} \in \Theta$ for the update.

3.2. Parameter Selection for Adaptation

Selecting parameters $\boldsymbol{\theta}$ for adaptation is important because effective constraints may avoid overfitting in adaptation. Without any constraints, parameters become optimized for trivial solutions; for example, DNNs might always output the "no sound" label.

We have two realizations for parameter selection because DNNs work as a feature extractors and classifiers: 1) a feature transformation network (linear input network [9]) and 2) a whole classification network (fully-connected network). As for the former, we set new small layers after feature extraction, e.g., after EVD in Fig. 2. Each eigenvector is transformed and the parameters are updated as follows

$$\hat{\boldsymbol{e}}_{w,i} = \boldsymbol{V}_w \boldsymbol{e}_{w,i} + \boldsymbol{b}_w, \ \ (i = M+1, ..., N), \qquad (5)$$

$$\boldsymbol{V}_w \leftarrow \boldsymbol{V}_w - \alpha \textstyle\sum_i \boldsymbol{\delta}_{w,i} \boldsymbol{e}_{w,i}^H \qquad (6)$$

where $\boldsymbol{V}_w \in \mathbb{C}^{N \times N}$ and $\boldsymbol{b}_w \in \mathbb{C}^N$ represent a complex matrix and bias vector, respectively. $\boldsymbol{\delta}_{w,i}$ is a propagated error vector corresponding to each input vector. The initial value of $\boldsymbol{V}_w$ is an identity matrix, and the bias $\boldsymbol{b}_w$ is not used in this paper. This transformation is expected to modify the error of the extracted eigenvectors caused by reverberation. As the latter, the $\boldsymbol{W}_*$ in Fig. 2 are updated. The whole update of the parameters of the classification layers has a high risk of overfitting the observed signals and of failure to adapt. Therefore, we must investigate the actual influence of these methods through experiments.

3.3. Early Stopping

We use early stopping of the parameter update [10] to avoid overfitting of the parameters to the observed sound signals. The iteration of the parameter update stops at a fixed number of iterations before convergence. In the experiments, we check the localization performance when we use different learning rates α. Such techniques are usually used in the *training phase* of neural networks to avoid overfitting, and they are expected to have the same effect on adaptation. Although the determination of the number of iterations and the learning rates are important problem, we reveal the effect of this method on SSL in this paper.

Table 1. Parameters of experiment

Number of sources	0 or 1 at each block
Training source	48 males, 48 females speech
Test source	2 males and 2 females (**speaker open**)
Transfer function	Anechoic (for training)
	Reverberant (RT_{20} 800 [ms]) (for test)
Position patterns	5760 = 360 (direction)
for training data	$\times$ 4 (distance) $\times$ 4 (height)
Position labels	289 = 72 directions $\times$ 4 + no-sound
DNN Input / Output dim.	768 ($\{\boldsymbol{e}_{i,w}\}_{i=2,3,4,w=21,...,84}$) / 289
DNN Middle Layer	Shown in Fig. 2
Learning rate adaptation	AdaGrad [20]

4. EXPERIMENTS

The experiments were designed to assess the effectiveness of our unsupervised adaptation method. The effect of two different adaptations (LIT or whole fully-connected networks) and early stopping were also investigated. The parameters were adapted to each utterance speech signal.

4.1. Experimental Setup

Recording conditions: All speech data were generated using impulse responses recorded in an anechoic and a reverberant room. The size of the reverberant room was 7.83 [m]$\times$5.87 [m]$\times$2.57 [m] (depth x width x height), and its reverberant time was about RT_{20} 800 [ms]. Four-channel impulse responses were recorded at 16 kHz by using microphones embedded in a humanoid NAO robot [21]. For the impulse responses of the training set, the resolution of the directional angle (azimuth) was 1° (360 directions), and the number of combinations of distance and height were 16, as shown in Fig. 3. For the impulse responses of the test set, eight angles $\{0, 45, 90, 135, 180, 215, 270, 315\}$ in degrees were measured at 100 and 200 [cm] distance and at 125 [cm] height. Note that the positions for the test set were not included in the training set to evaluate DNNs' performance in the severest case where both position and reverberant are unknown.

Feature extraction: The STFT parameters were set to be the same for all experiments: the size of the Hamming window was 512 points (32 [ms]), and the shift size was 160 points (10 [ms]). The block size for calculating $\boldsymbol{R}_w$ was 11 (110 [ms]). The bandwidth used for features was set to $[656 - 2625]$ [Hz], and 64 frequency bins were used for SSL. These configurations are listed in Table 1.

Data for training and test set: The speech data for training came from 48 male and 48 female speakers using the Acoustical Society of Japan-Japanese Newspaper Article Sentences (ASJ-JNAS) corpora[1], and one hour of data was used. The data for the test came from two male and two female speakers and were different from the training data in the same corpora. There was an average of seven utterances per speaker, and the content was phonetically balanced sentences. The training and test sets were generated using four-channel impulse responses of each environment. Gaussian noise of 20 dB was added to the speech signals of the training set, and 40 dB was added to the test set. The total number of labels was 289 and the resolution

[1] http://research.nii.ac.jp/src/JNAS.html

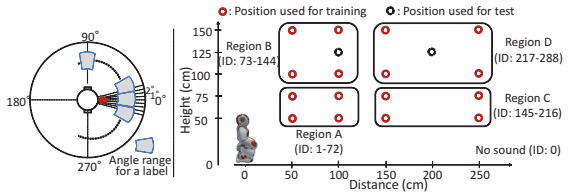

Fig. 3. Azimuth resolution and patterns of loud speaker locations

in azimuth for localization was 5°. The label ID "0" represents "no sound source", and the others represent the source locations, i.e., IDs 1-72 for the azimuth in the region A, IDs 73-144, 145-216 and 217−288 for the azimuth in regions B, C and D in Fig.3. The correct labels were added on the basis of voice activity of clean speech signals block-by-block (every 110 [ms]).

Configuration of DNNs: There were four dimensions and 256 directional activators in each w-th sub-band in the DNNs. There were eight blocks in the partially-integrated layer. The network sizes of the sub-band, partially-integrated and classification layers corresponded to 256×768, 32×256, 32×256, 256×1024, 1024×1024 and 1024×289. There were a total of 768 dimensions of features for the DNN input. The output dimensions were 289 to classify all labels. All weight parameters were initialized by using a Gaussian distribution $N(0, 0.025)$. The cross-entropy was used as the objective function for training, and we stopped training after only two epochs with 90% block-level accuracy for the training set. The unsupervised adaptation was applied to each utterance signal, and the number of iterations for early stopping was set to 200 empirically. We checked the performance of several learning rates.

Evaluation criteria: We calculated the correctness of the test set classification at the block level. **Note that this criterion is not based on the geometrical distance**. The total number of blocks for the test data per position was 1,000, and the ratio of no-sound blocks of the test speech signals was 8.6%.

4.2. Results and Discussion

Table 2 summarizes the block-level correctness of each position and each DNN for reverberant speech data. The *w/o adapt.* entries denote the results of pure DNNs without adaptation. *LIT* and *Whole* denote the results of adaptation with a linear input transform and whole parameters of fully-connected networks, respectively. *w/ ES* and *w/o ES* correspond to results with or without early stopping. Note that the location patterns of 200 [cm] distance and 125 [cm] height were not included in the training set.

Fist, we discuss the performance of DNN-based SSL without adaptation. The performance without adaptation ranged from 13.4% to 60.0% for 100 [cm] distance and from 7.0% to 52.9% for 200 [cm] distance. We can see that the performance also depends on the azimuth of the source's location, and some *blind spots* exist. DNNs failed localization at several positions, such as 135° azimuth and 100 cm distance. The reason the performance of the test set with the 200 [cm] distance was worse than that of the 100 [cm] distance is that corresponding patterns were not in the training set. This phenomenon is partly due to bias through training and partly due to reverberation. Tuning of the number of nodes in the DNNs will equalize this non-uniform performances. The average correctness of the same-position test set in an anechoic environment was about 60.0%.

Next, we discuss the impact of the parameter selection (LIT or Whole) and early stopping (ES). The correctness of LIT and Whole with ES were better than those of them without ES, where overfitting occurs and the performance severely degrades. The improvement from *w/o adapt* is a maximum of about 20 points at 315°, while some results become worse than those of *w/o adapt*. Some results at 0° and 315° were almost the same performance with those in an anechoic environment. Since the performances of the LIT and Whole conditions with ES are almost the same, it is not clear which method is superior. For further improvement, we should design better objective function and efficient update of parameters without overfitting.

Table 2. Block-level correctness for reverberant speech data (%).

dist.	azimuth	w/o adapt (baseline)	LIT w/o ES	LIT w/ ES	Whole w/o ES	Whole w/ ES
100 cm	0°	60.0	3.4	**67.8**	12.6	67.4
	45°	13.4	4.4	15.1	1.8	**15.7**
	90°	23.4	6.2	**29.9**	6.1	25.8
	135°	14.8	7.5	**15.4**	7.5	10.5
	180°	**26.9**	8.5	14.5	11.2	22.7
	225°	**15.6**	8.6	13.7	8.7	13.2
	270°	38.9	4.8	**41.1**	26.7	39.0
	315°	38.6	8.1	59.0	22.5	**59.1**
	Avg	29.0	6.4	**32.1**	12.1	31.7
200 cm	0°	52.9	26.8	**66.0**	35.4	64.4
	45°	45.8	5.5	49.8	27.1	**53.2**
	90°	**15.8**	7.8	11.4	8.6	13.7
	135°	7.0	8.6	6.3	**8.9**	6.2
	180°	**9.3**	8.6	8.6	9.0	8.8
	225°	21.3	8.6	**26.7**	12.3	23.7
	270°	20.6	6.8	**22.6**	9.2	22.5
	315°	**12.4**	8.0	9.0	8.1	8.7
	Avg	23.1	10.1	25.1	14.8	**25.2**

4.3. Remaining Issues

We should solve the following problems that still remain: 1) optimization of the DNN structure for SSL, 2) DNN training with data augmentation, and 3) development of more efficient parameter update scheme.

The former two ideas are essential for improving the baseline performance. Since the localization accuracy currently depends on the azimuth of the sound location, such non-uniform performance has to be equalized. This phenomenon is similar to speaker adaptation in ASR (position corresponds to speaker).

The last may be realized by combining SSL with blind reverberation or conventional SSL methods, not only using adaptation scheme of the DNNs. For example, the LIT with an orthogonal constraint will match the property of the eigenvectors. Since the results for LIT in our experiment also showed that *back propagation worked even at LIT in the STFT domain*, the reverberation filter in the STFT domain [12] can be optimized in terms of localization. Exploiting conventional SSL is a reasonable way to improve accuracy because NNs and a statistical model are used in the language processing to improve performance [22].

5. CONCLUSION

We tackled the problems of SLL based on DNNs for unknown source locations and unknown reverberant environments. The problem is solved by unsupervised adaptation using both parameter selection for adaptation and early stopping of the parameter update. Our experiments revealed that unsupervised adaptation improves localization accuracy of DNN-based SSL.

Acknowledgement This work was partly supported by JSPS KAKENHI Grant Numbers JP15K16051 and JP16H02869.

6. REFERENCES

[1] K. Nakadai, T. Lourens, H. G. Okuno, and H. Kitano, "Active audition for humanoid," in *Proc. of 17 th National Conf. on Artificial Intelligence*, 2000, pp. 832–839.

[2] W.-H. Yang and K.-K. Chan P.-R. Chang, "Complex-valued neural-network for direction-of-arrival estimation," *Electronics Letters*, vol. 30, no. 7, pp. 574–575, 1994.

[3] K.-L. Du, A.K.Y. Lai, K.K.M. Cheng, and M.N.S. Swamy, "Neural methods for antenna array signal processing: a review," *Signal Processing*, vol. 82, no. 4, pp. 547–561, 2002.

[4] H. Tsuzuki, M. Kugler, S. Kuroyanagi, and A. Iwata, "An approach for sound source localization by complex-valued neural network," *IEICE Trans. on Information and Systems*, vol. 96, no. 10, pp. 2257–2265, 2013.

[5] R. Takeda and K. Komatani, "Sound source localization based on deep neural networks with directional activate function exploiting phase information," in *Proc. of International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 405–409.

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

[7] T. Mizumoto, K. Nakadai, T. Yoshida, R. Takeda, T. Otsuka, T. Takahashi, and H. G. Okuno, "Design and implementation of selectable sound separation on the texai telepresence system using HARK," in *Proc. of ICRA*, 2011, pp. 2130–2137.

[8] R. Takeda and K. Komatani, "Discriminative multiple sound source localization based on deep neural networks using independent location model," in *Proc. of IEEE Workshop on Spoken Language Technology*, 2016, pp. 603–609.

[9] V. Abrash, H. Franco, A. Sankar, and M. Cohen, "Connectionist speaker normalization and adaptation," in *Proc. of Eurospeech*, 1995, pp. 2183–2186.

[10] C.M. Bishop, *Pattern Recognition and Machine Learning*, Shpringer, 2006.

[11] H. Saruwatari, T. Kawamura, T. Nishikawa, A. Lee , and K. Shikano, "Blind source separation based on a fast-convergence algorithm combining ICA and beamforming," *IEEE Trans. on Speech and Audio Processing*, vol. 14, no. 2, pp. 666–678, 2006.

[12] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. Juang, "Blind speech dereverberation with multi-channel linear prediction based on short time fourier transform representation," in *Proc. of International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 85–88.

[13] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Proc. of International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 7893–7897.

[14] T. Yoshioka, A. Ragni, and M. JF Gales, "Investigation of unsupervised adaptation of DNN acoustic models with filter bank input," in *Proc. of International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 6344–6348.

[15] S. P Rath, D. Povey, and K. Veselỳ, "Improved feature processing for deep neural networks.," in *Proc. of Interspeech*, 2013, pp. 109–113.

[16] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. D. Mori, "Linear hidden transformations for adaptation of hybrid ann/hmm models," *Speech Communication*, vol. 49, no. 10, pp. 827–835, 2007.

[17] Z. Huang, S. M. Siniscalchito, I-F. Chen, W. Jiadong, and C.-H. Lee, "Maximum a posteriori adaptation of network parameters in deep models," in *Proc. of Interspeech*, 2015, pp. 1076–1080.

[18] D. Torrieri and K. Bakhru, "Simplification of the MUSIC algorithm using a neural network," in *Proc. of MILCOM*, 1996, vol. 3, pp. 873–876.

[19] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. on Anttenas and Propagation*, vol. AP-32, no. 3, pp. 276–280, 1986.

[20] J. Duchi, Elad. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *The Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[21] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. O. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, "Mechatronic design of Nao humanoid," in *Proc. of International Conference on Robotics and Automation*, 2009, pp. 769–774.

[22] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. of Interspeech*, 2010, pp. 1045–1048.