Autoregressive Parameter Estimation with DNN-based Pre-processing

Cui, Zihao; Bao, Changchun; Nielsen, Jesper Kjær; Christensen, Mads Græsbøll

Published in:
Proceedings of the International Conference on Acousics, Speech, and Signal Processing

DOI (link to publication from Publisher):
10.1109/ICASSP40776.2020.9053755

Publication date:
2020

Document Version
Accepted author manuscript, peer reviewed version

AUTOREGRESSIVE PARAMETER ESTIMATION WITH DNN-BASED PRE-PROCESSING

Zihao Cui,[1] *Changchun Bao,*[1] *Jesper Kjær Nielsen,*[2] *Mads Græsbøll Christensen*[2]

[1] Beijing University of Technology, Faculty of Information Technology, Beijing China
cuizihao@emails.bjut.edu.cn, baochch@bjut.edu.cn
[2] Audio Analysis Lab, CREATE, Aalborg University, Denmark
{jkn,mgc}@create.aau.dk

ABSTRACT

In this paper, a method for estimating the autoregressive parameters from a signal segment is proposed. The method is based on a deep neural network (DNN) in combination with the classical Levinson-Durbin recursion (LDR). The DNN acts as a pre-processor for the LDR and can be trained on different metrics commonly encountered in speech processing using a generalized analysis-by-synthesis (GABS) structure where the LDR acts as the encoder. Unlike end-to-end data-driven approaches, this structure ensures that the DNN is easy to train and initialize since the DNN only has to learn a simple mapping. The results confirm this and show that the proposed method produces an AR-spectrum that efficiently represents the speech spectrum in terms of the Itakura-Saito divergence, Kullback-Leibler divergence, log-spectral distortion, and speech distortion.

***Index Terms*—** Auto-regressive model, Levinson-Durbin recursion, DNN, generalized analysis-by-synthesis.

1. INTRODUCTION

The AR coefficients play an important role in many speech applications such as speech recognition [1], coding [2, 3], and enhancement [4, 5]. Therefore, the estimation of the AR parameters from an observed signal segment has been a classical signal processing problem, and many different estimators have been proposed over many decades. The classical way of estimating the AR-parameters is to solve the Yule-Walker equations [6, 7] which can be performed efficiently using the Levinson-Durbin recursion (LDR) [8, 9]. This works extremely well for unvoiced speech which can be accurately modelled using an autoregressive process. For voiced speech, however, the excitation signal does not resemble a white, Gaussian excitation signal as in the autoregressive process, but is much more accurately modelled by an impulse train [3]. As a consequence of this, many alternative ways of estimating the AR-parameters have been proposed based on the prior knowledge on the power spectral density (PSD) [10, 11, 5] or the excitation signal [3, 12, 13]. For example, El-Jaroudi and Makhoul proposed in [10] the discrete all-pole (DAP) approach in which the AR-parameters are estimated by minimising the Itakura-Saito (IS) divergence for a discrete set of points, leading to better performance for voiced speech. Based on a harmonic residual assumption, Murthi and Rao [11] proposed an AR estimator to match the envelope of the speech spectrum based on the idea of minimum variance distortionless response (MVDR) which gives a robust estimation of the AR-parameters for both voiced and unvoiced signals. The prior information that the excitation signal of voiced speech is similar to a periodic impulse train has also been utilized in sparse linear predictive coding (LPC) [3] in which the AR-parameters are also modelled as being sparse.

In addition to classical signal processing methods, data-driven approaches to estimating the AR-parameters have also been proposed [5, 14]. One recent example of this is the part-defined auto-encoder (PAE) [5] in which the analysis-by-synthesis (ABS) strategy [15] is combined with an auto-encoder [16]. In this approach, a DNN is trained to learn the mapping from the raw data to the reflection coefficients which can easily be translated into the corresponding AR-parameters. While this approach is conceptually simple, the DNN is hard to train and initialize since the mapping from the raw data to the reflection coefficients is complicated and non-linear.

In this paper, we instead propose to use a DNN as a pre-processor for a classical AR-parameter estimator. Specifically, the pre-processor converts the raw data into autocorrelation values which are then converted to AR-parameter estimates using LDR. The DNN is designed with fixed input and output layers that makes the training and initialization of the DNN much easier than in the PAE method since the DNN is only responsible for learning a simple mapping. The DNN can be optimised for different metrics such as the Itakura-Saito divergence or the log-spectral distortion [17]. Interestingly, the approach also resembles the generalized ABS (GABS) method [18] where the modifier is the DNN-based pre-processor, and the decoder and encoder are the LDR and the computation of the AR-spectrum from the AR-parameters, respectively.

The paper is organized as follows: in Sec. 2 we describe

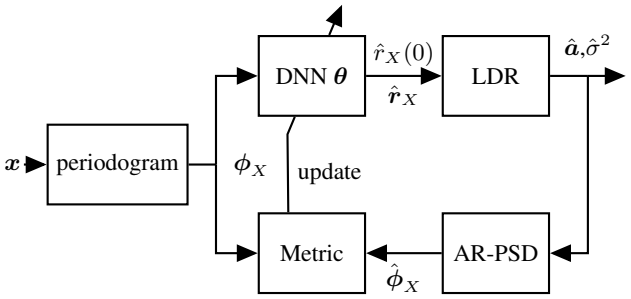

Fig. 1. The training stage of the DNN-based AR estimator.

the classical AR estimator based on the LDR, the PAE, and the proposed AR-estimator with a DNN-based pre-processor. These methods are evaluated and compared in Sec. 3. Finally, the conclusions are given in Sec. 4.

2. AR PARAMETER ESTIMATION

As alluded to in the introduction, we here augment the classical way of estimating AR parameters with a DNN-based pre-processor to obtain an efficient representation of the speech spectrum. Efficient here relates to the metric (e.g., Itakura-Saito or log-spectral distortion) used to measure the distance between the estimated AR-spectrum $\hat{\boldsymbol{\phi}}_X$ and the periodogram $\boldsymbol{\phi}_X$ pertaining to a signal segment $\boldsymbol{x}$. Fig. 1 shows how the DNN-based pre-processor is trained on one or several of these metrics to obtain such an efficient representation. Before we go into more details on how the pre-processor is trained, however, we first describe classical AR parameter estimation.

2.1. Classical AR Parameter Estimation

A p'th order autoregressive (AR) process $x(n)$ is a stationary random signal given by

$$x(n) = -\sum_{i=1}^{p} a_i x(n-i) + e(n) \tag{1}$$

where a_i is the i'th AR parameter and $e(n)$ is a white Gaussian excitation signal with variance σ^2. The power spectral density (PSD) of such an AR-process is given by

$$\hat{\phi}_X(k) = \frac{\sigma^2}{\left|1 + \sum_{i=1}^{p} a_i \mathrm{e}^{-j\omega_k i}\right|^2}\,, \quad k = 0, \ldots, N-1 \tag{2}$$

where $\omega_k = 2\pi k/N$. An estimate of the PSD can be obtained by replacing the true AR parameters with estimated ones. The classical way of estimating these AR parameters from a signal segment

$$\boldsymbol{x} = \begin{bmatrix} x(0) & x(1) & \cdots & x(N-1) \end{bmatrix}^T \tag{3}$$

is to compute the parameters that minimize the power of the excitation signal, i.e.,

$$(\hat{\boldsymbol{a}}, \hat{\sigma}^2) = \underset{\boldsymbol{a},\sigma^2}{\operatorname{argmin}} \sum_{n=0}^{N-1} e^2(n) \tag{4}$$

where

$$\boldsymbol{a} = \begin{bmatrix} a_1 & a_2 & \cdots & a_p \end{bmatrix}^T\,. \tag{5}$$

By assuming that the observed signal is 0 outside the observation window, i.e., that $x(n) = 0$ for $n < 0 \vee n \geq N$, minimising (4) leads to the estimate

$$\hat{\boldsymbol{a}} = -\hat{\boldsymbol{R}}_X^{-1} \hat{\boldsymbol{r}}_X \tag{6}$$

where $\hat{\boldsymbol{R}}_X$ and $\hat{\boldsymbol{r}}_X$ are the estimated covariance matrix and vector given by

$$\hat{\boldsymbol{r}}_X = \begin{bmatrix} \hat{r}_X(1) & \cdots & \hat{r}_X(p) \end{bmatrix}^T \tag{7}$$

$$\hat{\boldsymbol{R}}_X = \begin{bmatrix} \hat{r}_X(0) & \cdots & \hat{r}_X(p-1) \\ \vdots & \ddots & \vdots \\ \hat{r}_X(p-1) & \cdots & \hat{r}_X(0) \end{bmatrix} \tag{8}$$

$$\hat{r}_X(m) = \frac{1}{N} \sum_{n=0}^{N-1-m} x(n+m)x(n)\,, \tag{9}$$

respectively. The estimate of the excitation variance σ^2 is given by

$$\hat{\sigma}^2 = \hat{r}_X(0) + \hat{\boldsymbol{r}}_X^T \hat{\boldsymbol{a}}\,. \tag{10}$$

Since $\hat{\boldsymbol{R}}_X$ is a Toeplitz matrix, the estimate of $\hat{\boldsymbol{a}}$ can be computed very efficiently using the Levinson-Durbin recursion (LDR) [19]. Moreover, the estimate of $\hat{\boldsymbol{a}}$ will guarantee that the all-pole filter model in (1) is stable.

2.2. Part-defined Auto-encoder (PAE)

While estimating the AR spectrum as described above works well for unvoiced speech, it does not work well for voiced speech. The primary reason for this is that the excitation signal for voiced speech is an impulse train instead of a white Gaussian signal so obtaining the AR parameter estimates by minimising the two-norm typically results in an AR spectrum with a too sharp contour [3]. Consequently, many methods have been proposed to solve this problem such as [10, 11, 3]. An alternative to these classical methods is to use a purely data-driven approach for computing the AR-spectrum. Although originally introduced for speech enhancement, the recently proposed PAE in [5] can easily be modified to be an example of such a data-driven approach. For the modified PAE, the functions by training a DNN convert the log-periodogram of a signal segment $\boldsymbol{x}$ into a set of p reflection coefficients (to ensure stability). Then the reflection coefficients can easily be translated into an AR-spectrum. In [5], the DNN was trained to minimize the log-spectral distortion, but we here use other metrics as described in Table 1.

Table 1. Metrics used in the training and test

Metric	Formula
LSD	$\frac{1}{K}\sum_{k=1}^{K}\left[\log_{10}\frac{\phi_X(k)}{\hat{\phi}_X(k)}\right]^2$
IS	$\frac{1}{K}\sum_{k=1}^{K}\left[\frac{\phi_X(k)}{\hat{\phi}_X(k)}-\ln\frac{\phi_X(k)}{\hat{\phi}_X(k)}-1\right]$
KL	$\frac{1}{K}\sum_{k=1}^{K}\left[\phi_X(k)\ln\frac{\phi_X(k)}{\hat{\phi}_X(k)}-\phi_X(k)+\hat{\phi}_X(k)\right]$
β	$\frac{1}{K\beta(\beta-1)}\sum_{k=1}^{K}\left[\phi_X^{\beta}(k)+(\beta-1)\hat{\phi}_X^{\beta}(k)-\beta\phi_X^{\beta}(k)\hat{\phi}_X^{\beta-1}(k)\right]$
SD	$\frac{1}{K}\sum_{k=1}^{K}\left[\phi_X(k)-\hat{\phi}_X(k)\right]^2$

2.3. AR Parameter Estimation with pre-processing

The main problem with PAE is that the DNN has to learn the complicated and non-linear mapping from the log-periodogram to the reflection coefficients. This requires a lot of training data and a good initialization. To make this mapping much simpler, a better approach might be to combine the classical way of computing the AR parameters with a DNN-based pre-processor as illustrated in Fig. 1. This structure is very similar to the generalized analysis-by-synthesis (GABS) structure [18], and the main idea is to modify the signal so that the classical LDR produces an AR-spectrum minimising the distance to the periodogram in terms of one of the metrics listed in Table 1. Specifically, we perform the pre-processing with a DNN with the structure

$$\boldsymbol{\varphi}_X = \log_{10}(\boldsymbol{\phi}_X) \tag{11}$$

$$\boldsymbol{\varphi}_Y = \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{\varphi}_X) \tag{12}$$

$$\hat{r}_X(m) = \frac{1}{N}\sum_{k=0}^{N-1} 10^{\varphi_Y(k)} \mathrm{e}^{j\omega_k m} \tag{13}$$

for $m = 0, 1, \ldots, p$ where $\boldsymbol{\theta}$ contains the DNN parameters and

$$\boldsymbol{\phi}_X = \begin{bmatrix}\phi_X(0) & \cdots & \phi_X(\lfloor N/2 \rfloor)\end{bmatrix}^T \tag{14}$$

$$\boldsymbol{\varphi}_Y = \begin{bmatrix}\varphi_Y(0) & \cdots & \varphi_Y(\lfloor N/2 \rfloor)\end{bmatrix}^T \tag{15}$$

$$\varphi_Y(k) = \varphi_Y(N-k)\,. \tag{16}$$

It is important to note that the fixed input and output layers of the DNN are selected so that the mapping $\boldsymbol{f}_{\boldsymbol{\theta}}(\cdot)$ is extremely simple. For example, setting it to unity so that $\boldsymbol{\varphi}_Y = \boldsymbol{\varphi}_X$ asymptotically gives the same estimates of the AR parameters as the classical AR parameter estimator described in Sec. 2.1. This also suggests that setting the mapping to unity is a good initialization for training the parameters of the mapping $\boldsymbol{f}_{\boldsymbol{\theta}}(\cdot)$. Finally, we also remark that $\boldsymbol{f}_{\boldsymbol{\theta}}(\cdot)$ operates on the log-spectrum since this has proven to work better than operating directly on the spectra [20].

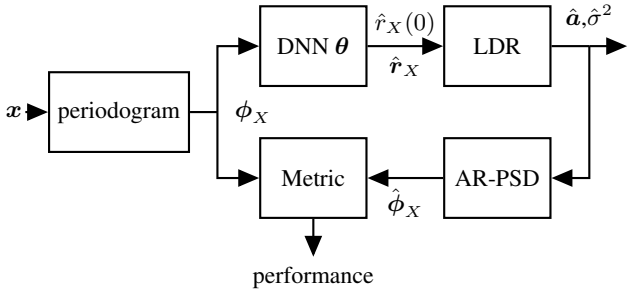

Fig. 2. The test stage of the DNN-based AR estimator.

3. SYSTEM EVALUATION AND COMPARISON

3.1. Data set and the proposed DNN structure

All experiments were conducted on speech data from the TIMIT corpus [21] which we down-sampled to 8 kHz. These signals were segmented into $N = 256$ samples long segments with 50% overlap and windowed by a sine window [22]. 600 randomly selected utterances were used for the training of the DNNs, and another 40 randomly selected utterances were selected for testing the data-driven methods as well as the traditional LDR and the DAP approach [10]. For both the PAE and the DNN-based pre-processor, the DNNs were trained by optimising different metrics. Specificially, DNNs were trained for the LSD, IS divergence, KL divergence, and beta divergence with $\beta = 0.5$. These metrics are listed in Table 1.

The performance was evaluated as illustrated in Fig. 2. For the LSD, the IS divergence, and the KL-divergence, the training and performance metrics were the same. For the last case, however, the training was performed using the beta divergence with $\beta = 0.5$ and the performance evaluated using the speech distortion (SD) metric.

An LPC order of $p = 12$ was used for both training and testing. The DAP and LDR were run on MATLAB 2018b while the other methods were run in Pytorch [23]. All the DNN-based methods were trained by Adam [24]. The mapping $\boldsymbol{f}_{\boldsymbol{\theta}}(\cdot)$ in the DNN-based pre-processor had three hidden fully connected layers, each consisting of 2048 units and each having a rectified linear activation unit (Relu) [25]. For a fair comparison, the PAE method had the same three hidden layers which were proceded by a fourth hidden fully connected layer responsible for mapping $\boldsymbol{\varphi}_Y$ into the reflection coefficients. This fourth layer had 129 units and produces the reflection coefficients using a tanh activation function.

For each metric, DNNs were trained with different initial conditions, drop-out rates, learning rates, and biases. For the method referred to as DNNs, the initial condition was the

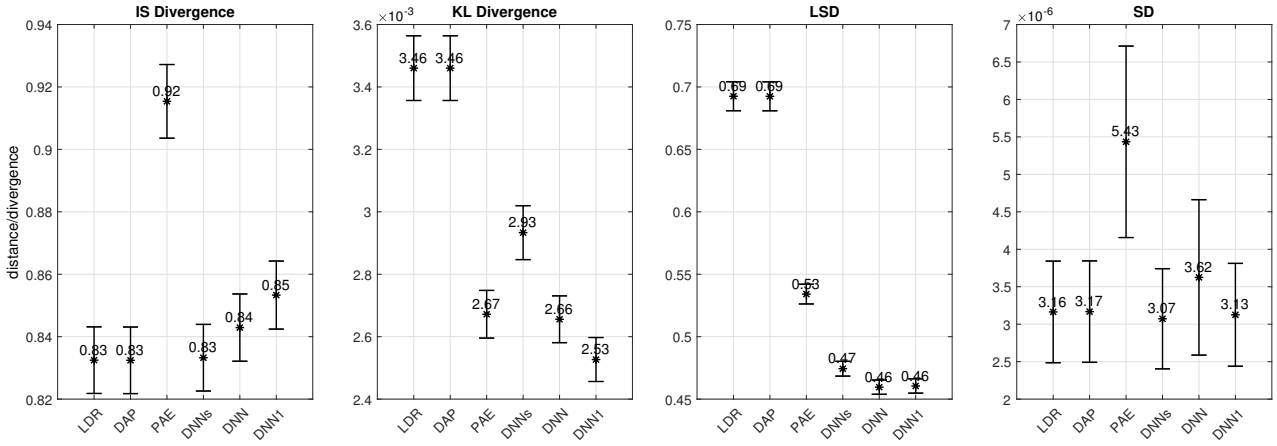

Fig. 3. The performance of different AR estimators illustrated by the mean and 95% confidence values. For the IS divergence, KL divergence, and the LSD, the training and testing was evaluated on the same metric. In the last case, the beta-divergence with $\beta = 0.5$ was used for training and the speech distortion for testing.

identity matrices for the weights, the drop-out rate was zero, the learning rate was $2 \cdot 10^{-7}$, the three hidden layers had the bias parameters 15, 0, and 0, respectively, and the output layer had a bias parameter of -15. For the methods referred to as DNN and DNN_1, the default in Pytorch was used for the initialization of the weights and the bias parameters, the learning rate was $2 \cdot 10^{-5}$, and the drop-out rates were 0.01 and 0.2, respectively.

3.2. Results

Fig. 3 shows the mean performance of different methods with 95 % confidence intervals. As described earlier, the training and test metrics were the same, except for the last case were the beta divergence with $\beta = 0.5$ was used for training and the speech distortion for testing. For the case of the IS-divergence, only the PAE method is significantly worse than the remaining methods which more or less have the same performance. Since DAP and asymptotically LDR minimize the IS divergence, it is hardly surprising that no other method performs better than these methods. The DNN-based methods seem to have the same or a slightly worse performance which is encouraging since this suggests that over-fitting has been avoided in the training. On the other hand, the over-fitting might explain why the PAE has a significantly worse performance than the rest of the methods. For both the KL-divergence and the LSD, all data-driven methods significantly outperform the classical LDR and DAP approaches. This shows that the idea of implementing a pre-processor can potentially lead to improvements when other performance metrics than the IS distortion is used. In the last case where different metrics are used for training and testing, all methods seem to have the same performance, except for the PAE which seems to be slightly worse.

4. CONCLUSIONS

In this paper, a classical AR estimator combined with a DNN-based pre-processor is proposed and compared to a completely data-driven AR estimator called PAE as well as the classical AR estimators LDR and DAP. The main motivation for using the proposed method is that the DNN-based pre-processor is much easier to initialise and train than PAE in which the DNN has to learn the complicated mapping from the raw data to the reflection coefficients. The results supported that including such a DNN-based pre-processor in a classical AR estimator is more robust to over-fitting and initialization than PAE and can potentially lead to performance improvements for other metrics than the Itakura-Saito distortion which the classical AR estimators LDR and DAP are (asymptotically) optimised for.

5. ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Grant No.61831019, No.61471014 and No. 61231015) and in part by the China Scholarship Council.

6. REFERENCES

[1] D. Roe, "Speech recognition with a noise-adapting codebook," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.* IEEE, 1987, vol. 12, pp. 1139–1142.

[2] M. Schroeder and B. Atal, "Code-excited linear prediction(CELP): High-quality speech at very low bit rates,"

in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Tampa, FL, USA, 1985, vol. 10, pp. 937–940, IEEE.

[3] D. Giacobello, M. G. Christensen, M. N. Murthi, S. H. Jensen, and M. Moonen, "Sparse Linear Prediction and Its Applications to Speech Processing," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 20, no. 5, pp. 1644–1657, July 2012.

[4] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 2, pp. 443–445, Apr. 1985.

[5] Z. Cui and C. Bao, "Linear Prediction-based Part-defined Auto-encoder Used for Speech Enhancement," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2019, pp. 6880–6884.

[6] G. U. Yule, "On a method of investigating periodicities disturbed series, with special reference to wolfer's sunspot numbers," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 226, no. 636-646, pp. 267–298, 1927.

[7] G. T. Walker, "On periodicity in series of related terms," *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, vol. 131, no. 818, pp. 518–532, 1931.

[8] N. Levinson, "The wiener (root mean square) error criterion in filter design and prediction," *Journal of Mathematics and Physics*, vol. 25, no. 1-4, pp. 261–278, 1946.

[9] James D., "The fitting of time-series models," *Revue de l'Institut International de Statistique*, pp. 233–244, 1960.

[10] A. El-Jaroudi and J. Makhoul, "Discrete all-pole modeling," *IEEE Trans. Signal Process.*, vol. 39, no. 2, pp. 411–423, Feb. 1991.

[11] M. N. Murthi and B. D. Rao, "All-pole modeling of speech based on the minimum variance distortionless response spectrum," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 3, pp. 221–239, May 2000.

[12] L. Shi, J. R. Jensen, and M. G. Christensen, "Least 1-norm pole-zero modeling with sparse deconvolution for speech analysis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.* IEEE, 2017, pp. 731–735.

[13] L. Shi, J. K. Nielsen, J. R. Jensen, and M. G. Christensen, "A variational EM method for pole-zero modeling of speech with mixed block sparse and Gaussian excitation," in *Proc. European Signal Processing Conf.*, Aug 2017, pp. 1784–1788.

[14] Y. Yang and C. Bao, "Rs-cae-based ar-wiener filtering and harmonic recovery for speech enhancement," *IEEE/ACM Trans. Audio, Speech, and Lang. Process.*, vol. 27, no. 11, pp. 1752–1762, 2019.

[15] C. G. Bell, H. Fujisaki, J. M. Heinz, K. N. Stevens, and A. S. House, "Reduction of Speech Spectra by Analysis-by-Synthesis Techniques," *J. Acoust. Soc. Am.*, vol. 33, no. 12, pp. 1725–1736, Dec. 1961.

[16] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[17] A. Gray and J. Markel, "Distance measures for speech processing," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 24, no. 5, pp. 380–391, Oct. 1976.

[18] W. B. Kleijn, R. P. Ramachandran, and P. Kroon, "Generalized analysis-by-synthesis coding and its application to pitch prediction," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 1992, vol. 1, pp. 337–340 vol.1.

[19] G. Cybenko, "The numerical stability of the Levinson-Durbin algorithm for Toeplitz systems of equations," *SIAM Journal on Scientific and Statistical Computing*, vol. 1, no. 3, pp. 303–319, 1980.

[20] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM Trans. Audio, Speech, and Lang. Process.*, vol. 22, no. 12, pp. 1849–1858, 2014.

[21] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," *NASA STI/Recon Technical Report N*, vol. 93, Feb. 1993.

[22] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec. 2015, pp. 504–511.

[23] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic Differentiation in PyTorch," in *NIPS-W*, 2017.

[24] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Dec. 2014.

[25] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 315–323.