TIME-FREQUENCY MASKING STRATEGIES FOR SINGLE-CHANNEL LOW-LATENCY SPEECH ENHANCEMENT USING NEURAL NETWORKS

Mikko Parviainen, Pasi Pertilä, Tuomas Virtanen

Laboratory of Signal Processing
Tampere University of Technology
Tampere, FINLAND

Peter Grosche

Huawei European Research Center
Munich, Germany

ABSTRACT

This paper presents a low-latency neural network based speech enhancement system. Low-latency operation is critical for speech communication applications. The system uses the time-frequency (TF) masking approach to retain speech and remove the non-speech content from the observed signal. The ideal TF mask are obtained by supervised training of neural networks. As the main contribution different neural network models are experimentally compared to investigate computational complexity and speech enhancement performance. The proposed system is trained and tested on noisy speech data where signal-to-noise ratio (SNR) ranges from -5 dB to +5 dB and the results show significant reduction of non-speech content in the resulting signal while still meeting a low-latency operation criterion, which is here considered to be less than 20 ms.

Index Terms— Speech enhancement, neural networks

1. INTRODUCTION

Speech enhancement is important in many audio applications including noise reduction of poor quality recording, background noise suppression of audio over communication channel, and improving speech intelligibility of hearing-aid devices. Many applications require low-latency operation to guarantee good user experience. In the past speech enhancement has been approached by methods like spectral subtraction (e.g. [1, 2, 3]) and Wiener filtering (see e.g. [4]). These rely on estimated noise or statistics that, when misestimated, causes artifacts and leakage of non-speech content to the enhanced signal. Recently, machine learning approaches such as deep neural networks (DNNs) have become popular in speech enhancement and separation [5][6], because they are capable of learning complex non-linear enhancement models.

In the time-frequency masking based speech enhancement the noisy input signal is *masked* so that the target signal, i.e. speech, is retained. Several approaches have been introduced to estimate the mask, including binary mask [7], ratio mask [8], and complex-valued mask [9]. In the neural network based approach the desired mask is *predicted* from the noisy input signal in the framewise processing. For low-latency speech communication operations, the length of the frame has to be short (e.g. $<$ 20 ms) and it is required that the processing occurs within the specified time constraint. The processing time is affected by the chosen length of the synthesis time-window. Furthermore, the complexity of the neural network structure affects the processing time from the input to the output and therefore the prediction time per time frame is an important property.

This research has been supported by the Huawei Innovation Research Program (HIRP).

DNNs such as fully-connected deep neural network (FC-DNN) and recurrent neural network (RNN) are state-of-the-art approaches to obtain time-frequency (TF) mask [10], but their practical computational performance has not been thoroughly investigated for TF masking based speech enhancement, i.e., what is the throughput time of DNN based speech enhancement models. To investigate this matter, this paper focus on the use of FC-DNNs and RNNs for speech enhancement in the low-latency scenario. Different neural network models and masking approaches are compared to investigate their computational complexity and speech enhancement performance.

The rest of the paper is organized as follows. Section 2 presents the proposed enhancement approach. Section 3 describes the method to obtain the enhancement model. Section 4 presents the data used for training and testing. Section 5 presents the evaluation of the enhancement model and Section 6 the achieved results. Section 7 concludes the paper.

2. TIME-FREQUENCY MASKING FOR SIGNAL ENHANCEMENT

The signal model used in this work is

$$x(k) = s(k) + n(k), \tag{1}$$

where $x(k)$ is the observed noisy signal, $s(k)$ is the clean signal, $n(k)$ is the background noise, and k denotes time index. Due to the nature of the TF masking based enhancement, the conversion to the frequency domain is made using short-time Fourier transform (STFT). The conversion is performed over a short block of time domain samples. The length K of this window and the frame-hop determine the minimum algorithmic latency of the system. With frame-hop $K/2$, to synthesize the first enhanced frame there is a delay of $K + K/2$ samples. However, the first $K/2$ samples of the enhanced frame can be synthesized after K input samples. After this initial delay, the following enhanced frames can be produced at one frame-hop interval (e.g. $K/2$) given that processing time of the samples is less or equal to the one frame-hop time.

After the conversion signal model becomes as follows in the STFT domain.

$$x(t, f) = s(t, f) + n(t, f), \tag{2}$$

where $s(t, f)$ is the clean signal, $n(t, f)$ is the background noise, and $x(t, f)$ is the observed noisy signal. t and f are the time frame index and the frequency bin index, respectively. With TF masking, the STFT of the estimated clean signal $\hat{s}(t, f)$ is obtained as

$$\hat{s}(t, f) = x(t, f)m(t, f), \tag{3}$$

where $m(t, f)$ is the TF mask. Finally, the time-domain signal is obtained by applying inverse STFT to $\hat{s}(t, f)$ and using the overlap-add method [11] to process sequential frames with 50 % overlap.

The TF mask can be estimated in many ways. In this work, two mask types are used. The log-amplitude-ratio mask [12] uses the clean signal $s(t, f)$ and the noisy signal $x(t, f)$ to estimate TF mask:

$$m_R(t, f) = \log_{10} \frac{|s(t, f)|}{|x(t, f)|}. \quad (4)$$

The generalized Wiener mask (see e.g. [4]) is defined as

$$m_W(t, f) = \frac{|s(t, f)|^p}{|s(t, f)|^p + |n(t, f)|^p}, \quad (5)$$

where p for the classic Wiener mask is 2. As shown by (5), the generalized Wiener masks need the noise estimate $n(t, f)$.

3. TIME-FREQUENCY MASK ESTIMATION USING NEURAL NETWORK

The chosen approach to predict TF masks is based on *supervised learning* (see, e.g. [13]) and neural networks, i.e., the features of the input signal are shown to the neural network with the target output. Here, the target output is the TF mask, which, when applied, extracts the target signal from the observed signal. During the training, the neural network evolves into a model, which produces the TF masks given the input features.

3.1. DNN models

The network architectures used in this work include FC-DNNs and RNNs. From the RNNs variants, long short-term memory (LSTM) and gated recurrent unit (GRU) are used. FC-DNNs is chosen for its simplicity and therefore it is expected to have low frame processing time. The recurrent structures are chosen for their inherent capability to exploit the temporal structure of the data.

Based on a pilot test, two-layer FC-DNNs, and LSTM and GRU with four to five layers are most promising for this task. Therefore only those architectures are discussed. The number of neurons is chosen also based on the initial results and the most promising networks are presented in this paper. Hyperbolic tangent (tanh) and rectified linear unit (relu) are used as the hidden activation functions. The output layer activation is always linear.

3.2. Training of Neural Network for Mask Prediction

The features are extracted from the input audio and they are fed to the neural network, which predicts the TF mask $\hat{m}(t, f)$. Here, the features are the natural logarithm of the magnitude spectrum and they are standardized by removing the mean and scaling the variance to unity.

Instead of feeding just current frame's features, feature vector concatenation is tested to exploit temporal dependence of the data. The features of the four frames preceding the current frame are stacked into a long feature vector. These augmented feature vectors are used only with FC-DNNs, since RNNs exploit the temporal structure inherently. The sequence length of the RNNs in training phase is 64 frames.

The optimization algorithm of the network was selected empirically by observing the behavior of the training error over the epochs. Adamax [14] performed best for this task, but also Adadelta [15] and Adagrad [16] were used due to better convergence of the training error in some cases or just to see its performance in comparison to Adamax. Mean squared error is used as the loss function training the neural network.

3.3. Mask post-processing

Due to relatively short time windows required for low-latency operation, large changes in sequential mask values may result in distortion (perceived as "roughness") in the masked signal (using longer window length, e.g., 32 ms "roughness" is not present). Here, first-order exponential smoothing was used with 0.8 as the smoothing factor.

4. DATA AND PROCESSING DESCRIPTION

The dataset is derived from the publicly available CHiME3 [17] samples. CHiME3 contains speech data derived from Wall Street Journal (WSJ0) corpus (83 speakers), and background samples, which include cafeteria (CAF), street (STR), pedestrian (PED), and bus (BUS). The noisy samples are obtained by mixing the clean WSJ0 sentences with the background according to (1). The background signal is mixed with signal-to-noise ratio (SNR) ranging from -5 dB to +5 dB. The SNR for each sample is randomly chosen from the range above. The SNR is global value over the sentence. All the available clean speech sentences were used and the noise signal was obtained by extracting a randomly chosen segment from the background recordings. The length of the segment was matched to the length of the clean speech, which is possible because the background recordings in CHiME3 are tens of minutes long whereas the clean speech sentences are few seconds long. Each background segment was used only once. The data is split into isolated training (75 %), validation (18.75 %), and test (6.25 %) set. The isolation means that each speaker can be only in one of the sets.

The sampling frequency of the database and processing is 16000 Hz. Since low latency operation is required, the window length is 16 ms (256 samples) and sequential time frames overlap by 50 % (128 frame hop). Each frame is windowed using a squared-root of Hann window function to guarantee perfect reconstruction using overlap-add. The length of STFT is 256 samples of which 129 first values are retained per frame.

5. EVALUATION

The proposed system is implemented in Python v. 2.7 utilizing Keras v. 2.0.8 deep learning library [18]. The speech enhancement performance of the proposed method is evaluated using Source to Distortion Ratio (SDR) [19] from the synthesized audio. The SDR scores are obtained using mir evaluation toolbox [20]. First, SDR score is calculated between the unprocessed and the reference resulting in SDR_{noisy}. Next, SDR score is calculated between the file processed by the enhancement model and the reference file (clean signal) resulting in SDR_{enh}. The ultimate score of a given file is obtained as the difference between values: $\text{SDR}_{\Delta} = \text{SDR}_{enh} - \text{SDR}_{noisy}$. The mean SDR_{Δ} over all the files in the test set is the performance score of a given model.

In order to estimate the enhancement model's potential use in a real-time system, the processing time of one frame-hop is calculated by calling the `time` method of the Python module `time` [21] when the processing of a frame starts. This time is stored and after the processing is done `time` method is called again and the one frame hop is the average difference of these times calculated over test audio file. The nominal one frame-hop processing time is obtained using a MacBook Pro Core i5 2.9 GHz 13-Inch (Late 2016) 16 GB memory. The frame hop processing time includes data read from

memory, noisy data conversion to the frequency domain, feature extraction, the processing time of the model (`predict` function of the Keras [18] model running Tensorflow (v. 1.3) [22] backend was used), conversion to the time domain signal, and write to the memory.

6. RESULTS

In this section the speech enhancement results of the proposed approach for speech enhancement are presented with a specific emphasis on the prediction speed. Different network architectures and their training schemes are compared in Section 6.1 and 6.2. The performance of the different masking approaches is presented in Section 6.3.

6.1. Comparison of Network Architectures

This section presents the comparison between different neural network architectures and the number of layers and nodes per layer. This comparison is relative, i.e., all networks are trained with the same features, mask type (logarithm-ratio-mask (4)), and mask post-processing scheme and only the network and its training scheme is changing.

Figures 1, 2, and 3 present the performance in terms of SDR_Δ as a function of one frame-hop prediction time (in milliseconds) for FC-DNN, LSTM, and GRU networks, respectively. Since, one frame-hop is 8 ms (16 ms window length and 50 % overlap), the networks exceeding this limit (illustrated with the dashed black line) would not meet the real-time operation with the comparable computational power. However, networks achieving prediction times $\approx$ 10 ms are still possible candidates since the prediction times were run on the general purpose laptop running regular operating system processes during the prediction time test. Thus, with more computational resources and more optimized implementation of the models, the points in the figures would move to the left while the line would remain in the same position.

In each figure, the performance of a given network is presented by a unique color/marker (within the figure). The network architecture is described in the legend of each figure; in Figure 1 the best performing network achieves SDR_Δ of 5.0 dB with approximately 2.1 ms one frame-hop prediction time and it is marked with a ▲ symbol. The network is a two-layer FC-DNN ("Dense" is a regular layer of a feed forward neural network) and it consists of 1000 units in each layer with relu as the hidden activation function. This network is trained using 0.25 dropout rate, batch size is 10, optimizer is Adagrad [16], feature vector length (FVL) is 645, and the number of epochs is 150.

In Figure 2 best performing LSTM while meeting the low-latency criterion is presented with a ◀ symbol and it achieves SDR_Δ of 5.3 dB with approximately 6.0 ms one frame-hop prediction speed. This network consists of four layers with 256 units per layer and has tanh as the hidden activation. The network is trained without dropout, batch size is 10, optimizer is Adamax, FVL is 129, and number of epochs is 150.

In Figure 3 best performing low-latency GRU achieves SDR_Δ of 5.5 dB with approximately 7.8 ms one frame-hop prediction speed and it is marked with a $\times$ symbol. The network consists of five layers with 256 units per layer and has relu as the hidden activation. The network is trained with dropout rate 0.25, batch size is 400, optimizer is Adamax, FVL is 129, and number of epochs is 250.

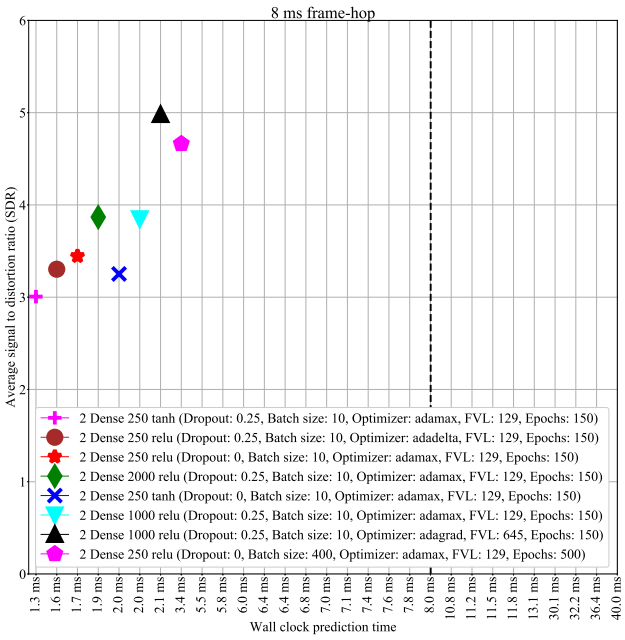

Fig. 1: FC-DNN performance as a function of one frame-hop prediction time.

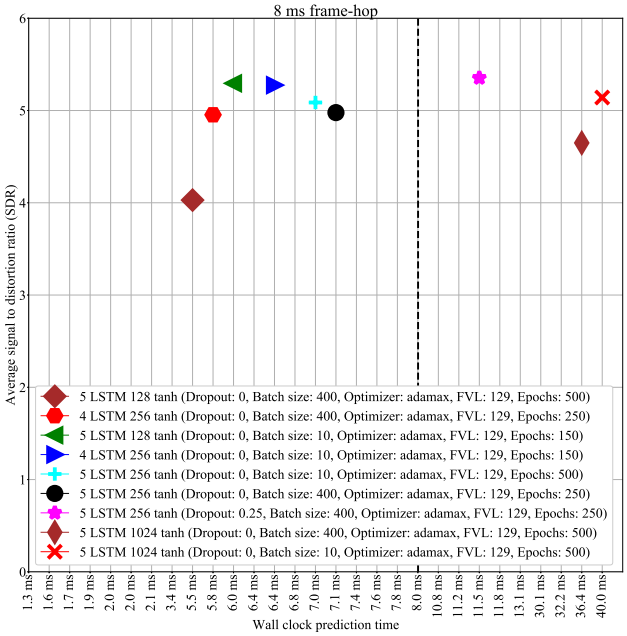

Fig. 2: LSTM performance as a function of one frame-hop prediction time.

6.2. Detailed Analysis of the Results

As noted above, the enhancement performance of LSTM and GRU networks is better than that of FC-DNN networks. However, FC-DNNs in general have lower one frame-hop processing time than the recurrent structures. Thus, while sacrificing the en-

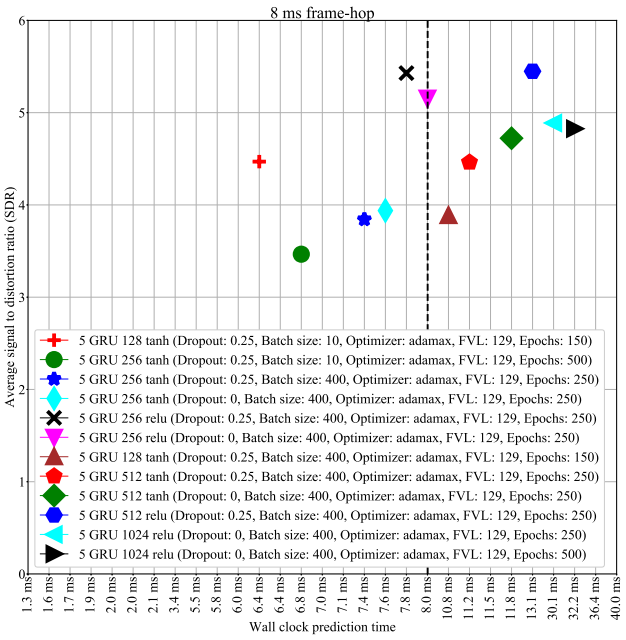

Fig. 3: GRU performance as a function of one frame-hop prediction time.

Table 1: Comparison of masking strategies without and with post-processing (indicated with pp). The log-amplitude-ratio mask and the generalized Wiener masks are denoted as m_R and m_W, respectively.

SDR_{noisy}	m_R	m_W $(p=1)$	m_W $(p=2)$	m_R (pp)	m_W (pp) $(p=1)$	m_W (pp) $(p=2)$
-4.5	7.1	8.5	**8.9**	6.3	7.5	8.1
-3.5	6.6	7.8	**8.1**	5.7	6.8	7.2
-2.5	6.4	**7.9**	7.8	5.6	6.8	6.9
-1.5	6.1	**7.5**	7.3	5.3	6.4	6.4
-0.5	6.0	**7.4**	7.1	5.3	6.3	6.2
0.5	5.4	**6.5**	5.9	4.7	5.5	5.1
1.5	5.0	**5.9**	5.2	4.2	4.9	4.3
2.5	4.8	**5.6**	4.8	4.3	4.6	3.9
3.5	4.4	**4.6**	3.7	3.7	3.6	2.8
4.5	3.8	**4.0**	3.0	3.2	2.9	2.0
Average	5.5	**6.6**	6.2	4.8	5.5	5.3

hancement performance, FC-DNN may be good option in a very low-latency operation mode and in applications with scarce computational resources.

The FC-DNNs presented in this paper contain only two layers since for the data used in this work deeper and wider (more neurons per layer) networks overfit; already three-layer networks performed worse in our experiments (results not included in this paper) than any of tested two layer networks for this data. The enhancement performance of FC-DNN can be improved by concatenating sequential feature vectors. The FC-DNN network presented by the black ▲ marker in Figure 1 uses five concatenated feature vectors and it achieves $\text{SDR}_{\Delta} \approx 5.0$ dB (cf. $\text{SDR}_{\Delta} \approx 3.9$ dB without concatenation). Furthermore, the concatenation did not slow down the prediction speed. This result indicates that temporal dependence of data can be exploited in this way in the case of DNNs. However, training with the temporally augmented feature vectors may require changes; in this case the optimizer had to be changed to Adagrad [16] from Adamax due to very slow decrease of the training error with the latter. Furthermore, training with more epochs increases the performance of the FC-DNN, e.g., see model denoted by magenta pentagon.

LSTMs and GRUs networks included in this work can be 4 – 5 layers without overfitting issues, but as the model complexity increases with more layers and neurons and there is no significant observed improvement after certain point and the training parameters are more important to obtain good enhancement performance. Furthermore, the increase of the model complexity results in longer prediction time.

The smoothing of the predicted mask (Section 6.1) was introduced to reduce the perceived "roughness" of the time-frequency masked signal. This approach decreases the SDR_{Δ}, but some listeners preferred the sound quality obtained with the smoothed mask to the unsmoothed mask. For comparison to the results presented above, the traditional spectral subtraction achieves $\text{SDR}_{\Delta} \approx 1.0$ dB (the method implemented in [23] was used).

6.3. The Effect of Time-frequency Mask Type to Performance

This section presents the enhancement results obtained with different masking strategies. The results are obtained with a GRU network: five layers, 128 units per layer, training phase sequence length 64, hyperbolic tangent hidden activation, and linear output activation. For this model the one frame-hop prediction time is approximately 6.5 ms run on the hardware presented in Section 5. Table 1 presents the results with and without mask post-processing (see Section 3.3). The table details by how SDR_{Δ} changes with a given SNR (i.e. SDR_{noisy}) of the unprocessed sound sample as well as the average over all SNRs. E.g., in Table 1, using the log-magnitude-ratio mask m_R (4) without post-processing for the unprocessed samples with $\text{SDR}_{noisy} \approx -4.5$ dB, the average $\text{SDR}_{\Delta} \approx 7.1$ dB.

The results in Table 1 show that the Wiener mask with $p = 1$ performs best for this type of data achieving $\text{SDR}_{\Delta} \approx 6.6$ dB. The Wiener filter with $p = 2$ achieved $\text{SDR}_{\Delta} \approx 6.2$ dB and may be subjectively less preferable than the Wiener mask with $p = 1$ and log-magnitude-ratio mask ($\text{SDR}_{\Delta} \approx 5.6$ dB). Furthermore, mask post-processing has a negative effect on the enhancement performance in terms of SDR_{Δ}. This result is expectable due to the chosen mask post-processing approach. However, some listeners may prefer the sound samples obtained using the post-processing approach.

7. CONCLUSIONS

This paper investigated a neural network based single-channel speech enhancement aimed at low-latency applications. The regular feed-forward neural networks and recurrent neural networks were tested on a modern laptop. The best performing real-time capable network architecture was gated recurrent unit achieving approximately 6.6 dB improvement in Source to Distortion Ratio (SDR). Long Short-Term Memory architecture achieved similar performance. Fully-connected deep neural network FC-DNN achieved moderate performance compared to the recurrent structures, but their computational economy was better than that of the tested recurrent networks making FC-DNNs a viable solution for applications with less computational resources.

8. REFERENCES

[1] Michael Berouti, Rainer Schwartz, and John Makhoul, "Enhancement of speech corrupted by acoustic noise," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Apr 1979, vol. 4, pp. 208–211.

[2] Rainer Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 5, pp. 504–512, 2001.

[3] Timo Gerkmann and Richard C Hendriks, "Unbiased MMSE-based noise power estimation with low complexity and low tracking delay," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1383 – 1393, 2012.

[4] Eric J. Diethorn, *Subband Noise Reduction Methods for Speech Enhancement*, pp. 91–115, Springer US, Boston, MA, 2004.

[5] Xu Li, Junfeng Li, and Yonghong Yan, "Ideal ratio mask estimation using deep neural networks for monaural speech segregation in noisy reverberant conditions," *Proc. Interspeech 2017*, pp. 1203–1207, 2017.

[6] Gaurav Naithani, Giambattista Parascandolo, Thomas Barker, Niels Henrik Pontoppidan, and Tuomas Virtanen, "Low-latency sound source separation using deep neural networks," in *IEEE Global Conference on Signal and Information Processing*. 2016, pp. 272–276, IEEE.

[7] Yuxuan Wang and DeLiang Wang, "Towards scaling up classification-based speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1381–1390, 2013.

[8] Arun Narayanan and DeLiang Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 7092–7096.

[9] Donald S Williamson, Yuxuan Wang, and DeLiang Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 3, pp. 483–492, 2016.

[10] Felix Weninger, Hakan Erdogan, Shinji Watanabe, Emmanuel Vincent, Jonathan Le Roux, John R Hershey, and Björn Schuller, "Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR," in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2015, pp. 91–99.

[11] "Overlap-Add Synthesis, `https://ccrma.stanford.edu/~jos/parshl/Overlap_Add_Synthesis.html`," Retrieved on April 12, 2018.

[12] Pasi Pertilä, *Parametric Time-Frequency Domain Spatial Audio*, chapter Microphone-Array-Based Speech Enhancement Using Neural Networks, John Wiley & Sons, 2017.

[13] Simon S. Haykin, *Neural Networks: A Comprehensive Foundation*, International edition. Prentice Hall, 1999.

[14] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[15] Matthew D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[16] John Duchi, Elad Hazan, and Yoram Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121 – 2159, 2011.

[17] Jon Barker, Ricard Marxer, Emmanuel Vincent, and Shinji Watanabe, "The third chime speech separation and recognition challenge: Dataset, task and baselines," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 504–511.

[18] "Keras: The Python Deep Learning library, `https://keras.io/`," Retrieved on April 4, 2018.

[19] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[20] Colin Raffel, Brian Mcfee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel P. W. Ellis, "mir_eval: a transparent implementation of common mir metrics," in *In Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR*, 2014.

[21] "`time` – Time access and conversions, `https://docs.python.org/2/library/time.html`," Retrieved on April 12, 2018.

[22] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.

[23] "VOICEBOX: Speech Processing Toolbox for MATLAB, `http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html`," Retrieved on April 6, 2018.