INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Bandwidth Embeddings for Mixed-bandwidth Speech Recognition

Gautam Mantena, Ozlem Kalinli, Ossama Abdel-Hamid, Don McAllaster

Apple Inc., USA

{gmantena,okalinli,oabdelhamid,dmcallaster}@apple.com

Abstract

In this paper, we tackle the problem of handling narrowband and wideband speech by building a single acoustic model (AM), also called mixed bandwidth AM. In the proposed approach, an auxiliary input feature is used to provide the bandwidth information to the model, and bandwidth embeddings are jointly learned as part of acoustic model training. Experimental evaluations show that using bandwidth embeddings helps the model to handle the variability of the narrow and wideband speech, and makes it possible to train a mixed-bandwidth AM. Furthermore, we propose to use parallel convolutional layers to handle the mismatch between the narrow and wideband speech better, where separate convolution layers are used for each type of input speech signal. Our best system achieves 13% relative improvement on narrowband speech, while not degrading on wideband speech.

Index Terms: speech recognition, deep neural networks, bandwidth embeddings, bandwidth aware training.

1. Introduction

Currently, there are many devices and equipment that receive both narrowband and wideband speech for automatic speech recognition (ASR) based applications. In conventional approaches, different acoustic models (AMs) are built to handle narrow and wideband speech separately since their sampling frequencies are different (8 kHz vs 16 kHz). However, it is not very economical or efficient to collect large amounts of training data for each of the tasks. A simple solution is to downsample the wideband speech and treat it similar to that of the narrowband. However, wideband has information that is useful to detect certain phonemes and is lost with downsampling [1, 2]. Moreover, models built on narrowband tends to perform worse on the wideband speech [1, 3]. Hence, it is not a trivial task to build mixed-bandwidth AMs.

In general, bandwidth expansion (BWE) of speech can be used to convert narrowband to wideband speech [4–7]. BWE is a technique used to reconstruct the high frequency components of the narrowband using the correlation that exists between low and high frequency of the speech signal [6]. In [5–7], deep neural network architectures such as feed forward network and a variant of restricted Boltzman machine (RMB) have been used to generate the higher frequency components. In [2], some issues reported for BWE are: (a) BWE is quite complicated and often introduces errors, and (b) in certain cases, the improvements in the recognition are seen only for less amounts of wideband speech ($\leq$ 50 hrs of transcribed data). Thus modeling based approaches have been explored.

In [2, 3], mixed-bandwidth AM training is considered as a missing feature problem. That is, for narrowband speech, the spectral features represent information only from 0-4 kHz and the remaining 4-8 kHz are missing. In [2], 22 and 27 filter bank filters were used for 8 kHz and 16 kHz data. The 22 filter bank features of 8 kHz data correspond to the 0-4 kHz of the 16 kHz data. To make sure all the features have the same dimension, zero padding is applied to the remaining 5 missing dimensions of the 8 kHz data. This approach is a simple and effective method. In [3], training a Gaussian mixture model hidden Markov model (GMM-HMM) using a modified expectation maximization (EM) algorithm was proposed to handle these extended narrowband features. In recent years, deep learning based acoustic modeling have been shown to be successful for many state-of-the-art automatic speech recognition (ASR) systems [8–10]. Use of the extended features in combination with powerful AMs such as deep neural networks (DNNs) has shown to perform well on mixed-bandwidth speech [2, 11]. In [2], it has been shown that: (a) DNNs can learn the variations in the narrow and wideband speech, (b) a single DNN can be used to recognize mixed-bandwith speech, and (c) improved recognition performance can be achieved on wideband speech. It is important to note that, in these techniques, different feature extraction techniques are used for narrow and wideband speech. In this paper, we show that deep neural network based AMs are powerful and can handle such variations in the data automatically with the help of bandwidth embeddings. Another modeling based approach was proposed in [12], where narrowband data was limited and thus transfer learning was used to improve the performance of the system for the narrowband speech. There, a separate model was built for each of the narrow and wideband speech tasks. Our work focuses on building a single model which performs well on both wide and narrowband speech and thus different from the work described in [12].

In this paper, we focus on a modeling approach for mixed-bandwidth speech recognition. AMs often tend to perform poorly on unseen data such as new speaker, different noise conditions, etc. To overcome these problems, techniques such as speaker or noise aware training have been explored [13–17]. In these techniques, auxiliary information such as speaker codes [13, 14], i-vectors [15], and bottleneck (BN) vectors [16,17] are explicitly provided as input to the model. In [18,19], speaker and noise embedding vectors are obtained by training another neural network classifier. Here, we propose to use an auxiliary input feature to the model indicating the bandwidth of the input speech, and bandwidth embeddings are jointly learned as part of the acoustic model training. This work is similar to the approach described in [13, 14] where speaker representations (also referred to as speaker codes) are learned during model training. To the best of our knowledge, there is no prior work to use embeddings for mixed-bandwith ASR. The major contributions of this work are as follows:

- Use of embeddings to learn representations for narrow and wideband speech. We show that these features derived from the embedding layer can be used to capture the variations in the data and thus help us to build a mixed-bandwidth speech AM. The embedding vectors for narrowband and wideband speech are learned during the model training; hence easy to use.

http://dx.doi.org/10.21437/Interspeech.2019-2589

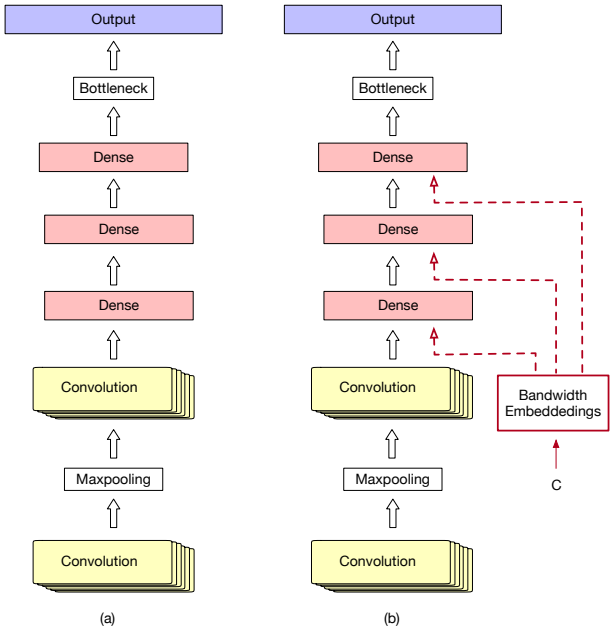

Figure 1: *(a) Baseline AM architecture containing two layers of convolution layers, 3 layers of fully connected layers, a liner bottleneck layer and then followed by an output layer, (b) Bandwidth embeddings connected to the dense layers of the baseline architecture, where c represents the type of the speech signal*

- We extend the use of bandwidth embeddings to a new model architecture which uses parallel convolutional layers to process narrow and wideband speech separately.
- Experimental results show that we can build a single mixed-bandwidth model, and achieve a relative improvement of 13% on upsampled narrowband speech in word error rate (WER) over the baseline system, while not degrading performance on the wideband speech.

The outline of the paper is as follows: Section 2 describes the approach to learn bandwidth embeddings. In Section 3, we describe the use of parallel convolutional layers for processing narrow and wideband speech separately. Section 4 describes the database used and followed by detailed evaluations in Section 5. Conclusions are provided in Section 6.

2. Bandwidth Embeddings and AM Training

In this paper, we explore modeling approaches and show that variations in the narrow and wideband speech can be learned and handled via embeddings. Fig. 1(a) shows the architecture of the baseline AM used in this paper. The model consists of convolutional and dense layers. Convolutional layers are used to reduce the spectral variations in the features and have shown to perform well for speech recognition [20,21]. Fig. 1(b), shows the corresponding proposed architecture of the AM which uses an embedding layer connected to all dense layers to handle narrowband and wideband speech jointly. Let weights and bias parameters of a dense layer, l, are represented by W_l and $\mathbf{b}_l$ respectively. The output of the dense layer is given as:

$$\mathbf{o}_l = f(W_l \mathbf{o}_{l-1} + \mathbf{b}_l), \quad (1)$$

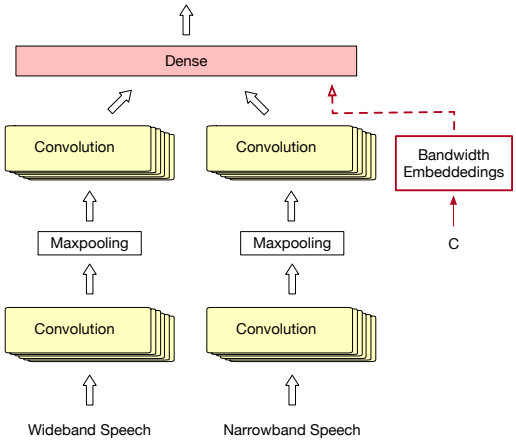

Figure 2: *Parallel convolutional layers for narrow and wideband speech.*

where $f(\cdot)$ is a non-linear activation function. Let $\mathbf{e}^c$ be an n dimensional embedding vector. c is a binary flag distinguishing narrow and wideband data. That is, $c = 0$ represents wideband and $c = 1$ represents narrowband speech. After incorporating the embedding vector, the equation for $\mathbf{o}_l$ is given as follows:

$$\begin{aligned} \mathbf{o}_l &= f(W_l \mathbf{o}_{l-1} + V_l \mathbf{e}^c + \mathbf{b}_l) \\ &= f(W_l \mathbf{o}_{l-1} + \hat{\mathbf{b}}_l), \end{aligned} \quad (2)$$

where $\hat{\mathbf{b}}_l = V_l \mathbf{e}^c + \mathbf{b}_l$. V_l is the weight matrix connecting the embedding vector $\mathbf{e}^c$ to the dense layer l. In this paper, the bandwidth embeddings is connected to the first dense layer ($l = 3$) after two convolutional layers. $V_l \mathbf{e}^c$ is referred to as a bias correction term and thus $\hat{\mathbf{b}}_l$ can be referred to as corrected bias. This correction helps the model to differentiate and better process the narrow and wideband data. $\mathbf{e}^c$ ($c \in \{0, 1\}$) is an n dimensional embedding vector and randomly initialized. During training, they are treated as model parameters and are updated during back-propogation. During decoding, the model uses the embedding vector based on the type of input speech signal and is provided by c.

3. Parallel Convolutional Layers

For speech processing, convolutional layers can be considered as a powerful feature processing units. As mentioned earlier, for narrowband speech, the spectral features represent information only from 0-4 kHz and the remaining 4-8 kHz are missing. Hence, use of convolutional layers on features without any prior processing might not be ideal. A simple approach is to upsample the narrowband speech and use the same convolutions for all types of input signals. From a modeling perspective, as an alternative to upsampling, we can use different convolutional layers (as shown in Fig. 2) for narrow and wideband speech. Filters from these convolutional layers do not share the parameters. We refer to this architecture as parallel convolutional layers. Note that we use shared parameters for the weights connecting the convolutional layers and the dense layer. In Sections 4 and 5, we provide a detailed description of the database used and experimental evaluations to show that embeddings and parallel convolutions can be used to build a single mixed-bandwidth AM which performs well for both the tasks.

4. Database

To evaluate our proposed techniques, we use 3400 and 600 hours of wideband (WB) and narrowband (NB) training data. Evaluations are performed on 54 and 4 hours of wideband and narrowband test sets. We report word error rate (WER) to compare the performance of different models. A more detailed description of the data is available in Table 1.

Table 1: *Statistics of narrow and wideband training and testing data. Note that k used in the numericals represent 1000 units*

Data	Sampling Rate (Hz)	# Utts.	Hours
Training Data			
WB	16k	2023k	3400
NB	8k	321k	600
Evaluation Tests			
WB	16k	50k	54.2
NB	8k	3.7k	4.2

From Table 1, it can be seen that the training data for the wideband speech is much larger than that of the narrowband. In general, such scenarios are common as one often do not have enough training data for each of the task, and wideband speech is more commonly used by many devices including personal assistants these days. The challenge is to exploit such mixed data for training purposes. In Section 5, we provide evaluations and show that the DNNs are powerful and can learn the variations of the narrow and wideband data; thus, avoiding the need for explicit model training for each task.

5. Evaluations

As shown in Fig. 1, we use a 7 layer deep neural network with 2 convolutional layers, 4 dense layers and followed by an output layer. We use SELU (scaled exponential linear units) [22] as activations for all the hidden layers except for the bottleneck layer. The bottleneck layer is a dense layer with linear activations and is often used to reduce the model size [21]. We use softmax function as the activations for the output layer. The convolutions used in the first and the second layers consists of 128 filters with kernel sizes of 9×9 and 3×4 respectively. The maxpooling functionality does not have any trainable parameters and hence not considered as layer. The kernel size and the strides used in maxpooling are 1×3 and 1×3 respectively. The dense, bottleneck and the output layer consists of 1024, 512 and approximately 8000 units. The input to the network are 40 dimensional log mel filter bank features with left and right context of 10. The model is trained using cross-entropy loss function. We use this architecture in all the evaluations performed in Sections 5.1-5.4

5.1. Baseline AMs

In this section, we present baseline experiments and their results. For training, we use a combination of wideband and narrowband speech shown in Table 1. We built three different AMs: (a) model AM1 built using only wideband speech (WB), (b) AM2 built using only narrowband speech (NB), (c) AM3 built using mixed-bandwidth speech (WB + NB), and (d) AM4 is built using mixed bandwidth data where NB speech is upsampled to 16 kHz. Note that during testing: (a) NB test data is upsampled to 16 kHz when AM1 and AM4 models are used, and (b) WB test data is downsampled to 8 kHz when AM2 model is used. We use sox for resampling the speech data [23].

Table 2: *Evaluating AMs trained using a combination of narrow and wideband data. The word error rates (WER) reported reflect the baseline performance of the ASR systems.*

Model	Training Data	WER (%) WB	WER (%) NB
AM1	WB	13.1	23.8[1]
AM2	NB	22.4[2]	21.0
AM3	WB + NB	13.6	26.2
AM4	WB + NB[1]	13.4	20.9 [1]

From Table 2, it can be seen that: (a) for WB test set, AM1 performs better than AM2 and AM3, (b) for NB test set, AM2 performs better than AM1 and AM3, and similar to the performance of AM4. This is because, DNNs tend to perform well in matched conditions. On the other hand, DNNs tend to perform well with increasing amounts of training data. However, in this case, AM3 and AM4 do not reflect such improvement. This is because the spectrum of narrow and wideband speech is different and hence training a model by mixing such data is not trivial. In Section 5.2, we show that using bandwidth embeddings we can exploit the use of both narrow and wideband speech for training a single model.

5.2. Experiments with Bandwidth Embeddings

In this section, we build AMs using an embedding layer connected to the first dense layer as shown in Fig. 1(b). Two sets of embedding vectors representing the narrow and wideband speech are learned as part of the AM training and hence simple to use.

Table 3: *Baseline system performances vs AM trained with embeddings.*

Model	WER (%) WB	WER (%) NB
AM1	13.1	23.8[1]
AM2	22.4[2]	21.0
AM3	13.6	26.2
+ Embeddings	**12.9**	**20.2**
AM4	13.4	20.9[1]
+ Embeddings	**13.0**	**18.2**[1]

In Table 3, we present results where AM3 and AM4 are trained together with the proposed bandwidth embeddings. It can be seen that bandwidth embeddings help to improve the performance of AM3 model with relative improvement of 5% and 23% in word error rate for WB and NB test sets. Also, AM3 + embeddings performs similar to that of AM1 for the WB test set, and performs better than AM2 for NB test set since it can leverage WB data and use more data for training compared to AM2. In Table 3, we see that AM4 + embeddings performs better than all the other systems with a relative improvement in

[1]Data is upsampled to 16 kHz

[2]Data is downsampled to 8 kHz

word error rate of 13% as compared to AM2 for the narrowband speech. This is because, to build AM4 models we use the same sampling rate for both types of speech input since narrowband speech is upsampled. Without upsampling of the narrowband speech (e.g. in AM3), the spectral features corresponding to 0-4 kHz do not overlap to that of the features extracted for wideband speech. These results indicate that a single AM can be used for mixed-bandwidth speech recognition. For an analysis, we evaluated AM3 model training by varying the embedding vector size from 32 to 256, and the results show that 128 was the best performing embedding size. Hence, we use 128 dimensional embedding vectors for evaluations in all experiments.

5.3. Experiments with Parallel Convolutional Layers

Table 4: *Evaluations performed using parallel convolutions on AM3 and AM4 models.*

Model	Training Data	WER (%) WB	NB
AM3	WB + NB	13.6	26.2
+ Embeddings		12.9	20.2
+ Parallel Conv.		13.4	19.9
+ Embeddings & Parallel Conv.		**13.0**	**19.6**
AM4	WB + NB[1]	13.4	20.9
+ Embeddings		13.0	**18.2**
+ Parallel Conv.		**12.7**	21.0
+ Embeddings & Parallel Conv.		13.2	19.9

In Table 4, we present results using parallel convolutional layers for AM3 and AM4 setups. It can be seen that using an embedding layer or parallel convolutional layers is improving the performance on either WB or NB or both the test sets. To further improve the system, we explore the use of combining bandwidth embeddings with parallel convolution layers. For the AM3 setup, AM3 + embeddings & parallel convolutional layers performs the best, and compared to AM3 baseline, it provides 4% and 25% relative improvement for WB and NB test sets, respectively. Use of parallel convolutional layers for AM3 model training has increased the model size approximately by 200k parameters, which is 1% increase in model parameter size. For the AM4 setup, AM4 + embedding layer performs the best for the NB test set providing 13% relative improvement over the AM4 baseline. Also, experimental results indicate that AM4 setup does not benefit much from the parallel convolution layers for handling the NB test set. This may be due to the fact that since the narrowband data was upsampled to 16 kHz in AM4 training, there is no mismatch in the filter banks used for narrowband and wideband speech; hence there is no need to use separate convolution layers for wideband and narrowband speech data. Whereas in AM3 training, parallel convolution layers help more since narrowband speech is not upsampled, and hence filterbanks used for the wideband and narrowband speech are different. In other words, the parallel convolution layers help to reduce the mismatch between features for the AM3 setup. Note that, upsampling of narrowband speech does not provide any new information for the 4k-8k Hz bands. Hence, we believe that AM4 + paralel convolutions is performing well on WB test set by separating convolutional layers for narrowband and wideband speech and possibly reducing the noise that comes from higher frequency of upsampled narrowband speech.

5.4. Result Summary

In Table 5, we summarize all the evaluations performed from Sections 5.1 to 5.3. AM1 and AM2 are baseline systems which are trained on wideband or narrowband speech respectively. AM3 and AM4 models are trained in combination of bandwidth embeddings and parallel convolutional layers. These models primarily differ in the sampling rate of the train and test data. That is, for AM4 models, narrowband speech was upsampled to 16 kHz, whereas raw narrowband speech was used without any pre-processing in AM3.

Table 5: *A summary of the performance of different models where: (a) AM1 and AM2 models are the baseline systems, (b) AM3 models are built using a combination of WB and NB data, and (c) AM4 models are built using WB and upsampled NB data.*

Model	Training Data	WER (%) WB	NB
AM1	WB	13.1	23.8[1]
AM2	NB	22.4[2]	21.0
AM3	WB + NB	13.6	26.2
+ Embeddings & Parallel Conv.		**13.0**	**19.6**
AM4	WB + NB[1]	13.4	20.9
+ Embeddings		**13.0**	**18.2**

In Table 5, it can be seen that the use of both embeddings and parallel convolutional layers gives the best performance for the model AM3. Compared to AM2 baseline system, we see a relative improvement of 6% in WER for the NB test set, while matching AM1 performance on the WB test set. In AM3, due to the mismatch in the filter bank features, it seems helpful to have separate convolutional layers for narrow and wideband speech. For AM4, compared with the AM2 model, the use of bandwidth embeddings gives a relative improvement of 13% in WER for the NB test set, while matching AM1 performance on the WB test set. AM4 uses upsampled narrowband speech and thus using bandwidth embeddings only seems sufficient.

6. Conclusions

In this paper, we have shown that bandwidth embeddings can be used to build a single model for mixed-bandwidth AM. Further more, we also used different convolutional layers (referred to as parallel convolutional layers) to handle the mismatch between the narrow and wideband speech. Experimental results show that models built using these approaches tend to perform well on narrowband speech without any loss in performance on wideband speech. For the model trained on wideband and upsampled narrowband speech, using bandwidth embeddings provides a relative improvement of 13% in WER on the narrowband test set while maintaining performance for the wideband speech test. We also showed that using bandwidth embeddings and parallel convolutional layers for 8 kHz and 16 kHz input speech signal has resulted in a relative improvement of 6% in WER on the narrowband test set without requiring upsampling of narrowband speech.

7. References

[1] P. J. Moreno and R. M. Stern, "Sources of degradation of speech recognition in the telephone network," in *Proc. of ICASSP*, vol. i, April 1994, pp. I/109–I/112 vol.1.

[2] J. Li, D. Yu, J. Huan, and Y. Gong, "Improving wideband speech recognition using mixed-bandwidth training data in CD-DNN-HMM," in *Proc. of SLT*, Dec 2012, pp. 131–136.

[3] M. L. Seltzer and A. Acero, "Training wideband acoustic models using mixed-bandwidth training data for speech recognition," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 235–245, Jan 2007.

[4] M. L. Seltzer, A. Acero, and J. Droppo, "Robust bandwidth extension of noise-corrupted narrowband speech," in *Proc. of INTERSPEECH*, Sept. 2005.

[5] K. Li and C. Lee, "A deep neural network approach to speech bandwidth expansion," in *Proc. of ICASSP*, April 2015, pp. 4395–4399.

[6] Y. Wang, S. Zhao, W. Liu, M. Li, and J. Kuang, "Speech bandwidth expansion based on deep neural networks," in *Proc. of INTERSPEECH*, Sept. 2015, pp. 2593–2597.

[7] K. Li, Z. Huang, Y. Xu, and C.-H. Lee, "DNN-based speech bandwidth expansion and its application to adding high-frequency missing features for automatic speech recognition of narrowband speech," in *Proc. of INTERSPEECH*, Sept. 2015, pp. 2578–2582.

[8] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, Nov. 2012.

[9] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, Jan. 2012.

[10] D. Yu and L. Deng, *Automatic Speech Recognition - A Deep Learning Approach*. Springer, October 2014.

[11] Z. You and B. Xu, "Improving wideband acoustic models using mixed-bandwidth training data via DNN adaptation," in *Proc. of INTERSPEECH*, Sept. 2014.

[12] X. Zhuang, A. Ghoshal, A. Rosti, M. Paulik, and D. Liu, "Improving DNN bluetooth narrowband acoustic models by cross-bandwidth and cross-lingual initialization," in *Proc. of INTERSPEECH*, Sept. 2017, pp. 2148–2152.

[13] O. Abdel-Hamid and H. Jiang, "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code," in *Proc. of ICASSP*, May 2013, pp. 7942–7946.

[14] S. Xue, O. Abdel-Hamid, H. Jiang, and L. Dai, "Direct adaptation of hybrid DNN/HMM model for fast speaker adaptation in LVCSR based on speaker code," in *Proc. of ICASSP*, May 2014, pp. 6339–6343.

[15] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Proc. of ASRU*, Dec. 2013, pp. 55–59.

[16] H. Huang and K. C. Sim, "An investigation of augmenting speaker representations to improve speaker normalisation for DNN-based speech recognition," in *Proc. of ICASSP*, Apr. 2015, pp. 4610–4613.

[17] S. Kundu, G. Mantena, Y. Qian, T. Tan, M. Delcroix, and K. C. Sim, "Joint acoustic factor learning for robust deep neural network based automatic speech recognition," in *Proc. of ICASSP*, March 2016, pp. 5025–5029.

[18] S. Kim, B. Raj, and I. Lane, "Environmental noise embeddings for robust speech recognition," *CoRR*, vol. abs/1601.02553, 2016. [Online]. Available: http://arxiv.org/abs/1601.02553

[19] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. INTERSPEECH*, Sept. 2017, pp. 999–1003.

[20] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533–1545, Oct 2014.

[21] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets." in *Proc. of ICASSP*, May 2013, pp. 6655–6659.

[22] G. Klambauer, T. U. A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Proc. of NIPS*, 2017, pp. 971–980.

[23] "SoX - Sound eXchange," http://sox.sourceforge.net.