RECENT IMPROVEMENTS TO NEURAL NETWORK BASED ACOUSTIC MODELING IN THE EML REAL-TIME TRANSCRIPTION PLATFORM

Volker Fischer, Omid Ghahabi, Siegfried Kunzmann[1]

EML European Media Laboratory GmbH, Berliner Straße 45, D-69120 Heidelberg
fischer@eml.org

Abstract: In this paper, we report some recent improvements to DNN/HMM hybrid acoustic modeling for the EML real-time large vocabulary speech recognition system, including the introduction of speaker adaptive long short term memory units (LSTMs) and efficient online decoding with deep bidirectional LSTMs. Based on a thorough latency analysis of our baseline large vocabulary speech recognizer we first abandon multi-pass recognition with fMLLR adapted acoustic features and further simplify decoding by dropping text independent vocal tract length normalization (VTLN) which was identified as a major bottleneck for real time applications. Subsequently, we improve accuracy by a variety of measures that include artificial training data augmentation and the use of additional features derived from an online speaker diarization module currently under development. Moreover, we investigate a hierarchy of feed forward and recurrent neural networks for a further reduction of word error rate. Finally, we demonstrate that established DNN pruning techniques are also applicable to bidirectional LSTMs, resulting in both an appropriate network size and substantial runtime savings. Our experiments are carried out on the publicly available WSJCAM0 corpus. Being simultaneously recorded with both a head-mounted and a desk-mounted microphone it enables us to study the impact of each of the proposed methods also in case of a channel mismatch between training and test data. The methods described in this paper yield an improvement of up to 15 percent relative to the baseline DNN/HMM acoustic model.

1 Introduction

In the past decade, automatic speech recognition has experienced huge gains from the use of deep neural networks (DNNs) for acoustic modeling [1]. Powerful hardware and sophisticated training algorithms enabled the training of deep networks with tens of millions of parameters that show improvements of more than 30% compared to the conventional Gaussian Mixture Model (GMM/HMM) approach [2]. More recently, recurrent network architectures like, for example, the gated recurrent unit (GRU, [3]) and the long short term memory cell (LSTM, [4, 5, 6, 7]) have been introduced to better capture the temporal dynamics of speech. Bidirectional LSTMs [8, 9, 10], cf. Sec. 3, are nowadays state-of-the-art, but production systems with online decoding requirements are frequently still using LSTMs or even plain feed forward neural networks (FFNN) with fMLLR adapted features and multi-pass decoding [11].

The work presented in this paper seeks to drop the latter by utilizing the superior modelling capabilities of recurrent neural networks. After a brief review of our baseline approach in Section 2, we sketch some recent enhancements towards our goals in Section 3 and present some experimental results – for both matched and unmatched test data – in Section 4. Finally, Section 5 concludes the paper with an outlook on further work.

[1] S. Kunzmann is now with Amazon.com, Inc.

2 Baseline System Overview

Our baseline approach to acoustic modeling is based on the RASR/NN toolkit introduced in [12] and is described in some detail in [13]. In our standard training setup, we use rectified nonlinearities [14], supervised discriminative pre-training [2], stochastic gradient descent with L2 regularization and dropout, and a frame-wise cross-entropy training criterion. For most languages our neural networks are trained with 500 - 1500 hours of data. The training data is usually divided into two or three (overlapping) subsets, each used to run one independent training epoch on a copy of the network. Subsequently, the copy yielding the lowest cross-validation error rate is used as seed model for the next training epoch.

Feature extraction computes 16 MFCCs in VTLN feature space [15], the degree of voicing, and an optional pitch value. A gender-informed, semi-randomized version of vocal tract length *perturbation* (VTLP, [16]) is used to augment the training data. Temporal dynamics are captured by the concatenation of 9 consecutive frames and an LDA transformation is used to reduce the feature vector dimension to 45. For online speaker adaptation in a multi-pass decoding scenario we apply an additional per speaker fMLLR transform that is estimated in the conventional GMM/HMM framework [17]. Finally, a context window of 5 to 17 frames is applied which outputs 225 to 765 features that are subjected to global mean and variance normalization and serve as neural network input. Once training has converged, we compute *DNN state priors* [18] for decoding by averaging the activations of the softmax output layer over all training frames.

Our decoder itself is a state of the art large vocabulary continuous speech recognizer [19, 20], which has been highly optimized for commercial use, but does retain the flexibility of a research decoder. Recent enhancements that are not in the scope of this paper include online versions of voice activity detection [21], speaker diarization, and language identification; see, however, Sec. 3.4 for the use of speaker diarization features as input to neural network based acoustic modeling.

We found *text independent* VTLN [15] working best, if a decent amount (more than 1.5 – 2 seconds) of audio is available for the GMM-based computation of a speaker's warping factor during runtime. Since improvements quickly vanish if only a limited amount of audio can be buffered in order to fulfill the latency requirements of many of our target applications, we decided to no longer use VTLN, but rely on plain MFCCs instead.

3 Recent Improvements

3.1 Recurrent Neural Networks

For the training of recurrent neural networks, we switched from RASR/NN to RETURNN [22], a configurable neural network training toolkit based on *Theano* [23] that provides a Python interface for the seamless access to acoustic feature vectors and Viterbi alignments computed with our RASR based acoustic modeling and decoding environment. With the main focus on the implementation of different recurrent cells (GRU, LSTM, etc.), the toolkit also offers several well-known optimization methods like, for example, ADADELTA [24] and ADAM [25], and additional features such as multi-GPU training. Moving from ADADELTA to ADAM, which is also the recommended optimization method in [8], gave us remarkable performance gains in terms of both training time and accuracy across several languages and for all recurrent and non-recurrent network topologies we considered so far.

The tight integration of RASR into RETURNN can also be utilized during runtime, if RASR created feature vectors are passed to Theano for neural network forwarding, and the resulting scores are returned to RASR's Viterbi search for decoding. However, since this is not feasible

in a highly scalable production environment, we implemented LSTMs in our commercially used variant of RASR. So far, our online production decoder only uses the „Vanilla“ long short term memory cell (cf. [7]) *without peepholes* that constitutes the building block for the hidden network layers. The forward equations are repeated here for the sake of the discussion that follows:

$$\boldsymbol{z_t} = \boldsymbol{\varphi}(\boldsymbol{W_z x_t} + \boldsymbol{R_z y_{t-1}} + \boldsymbol{b_z}) \quad (1)$$

$$\boldsymbol{i_t} = \boldsymbol{\sigma}(\boldsymbol{W_i x_t} + \boldsymbol{R_i y_{t-1}} + \boldsymbol{b_i}) \quad (2)$$

$$\boldsymbol{f_t} = \boldsymbol{\sigma}(\boldsymbol{W_f x_t} + \boldsymbol{R_f y_{t-1}} + \boldsymbol{b_f}) \quad (3)$$

$$\boldsymbol{o_t} = \boldsymbol{\sigma}(\boldsymbol{W_o x_t} + \boldsymbol{R_o y_{t-1}} + \boldsymbol{b_o}) \quad (4)$$

$$\boldsymbol{c_t} = \boldsymbol{c_{t-1}} \odot \boldsymbol{f_t} + \boldsymbol{z_t} \odot \boldsymbol{i_t} \quad (5)$$

$$\boldsymbol{y_t} = \boldsymbol{\varphi}(\boldsymbol{c_t}) \odot \boldsymbol{o_t} \quad (6)$$

Here, x_t is the layer input at time t, c_t is the internal state at time t, and y_t is the output at time t; W_z, W_i, W_f, W_o, are the forward weight matrices for the block input and the three gates (**i**nput, **o**utput, **f**orget), R_z, R_i, R_o, and R_f are the respective recurrent weight matrices, and b_z, b_i, b_o, and b_f are bias vectors. φ(.) denotes the input and output block nonlinearity (here: $\varphi(.) = tanh()$), $\sigma(.)$ is the logistic sigmoid function, and ⊙ denotes element-wise vector multiplication.

3.2 Online decoding with bidirectional LSTMs

Our BLSTMs use the topology suggested in [9], i.e. we do a combination of the forward and backward direction after each layer, cf. Figure 1. We use the same number of memory cells for each direction (usually 512) and thereby double the number of hidden layer parameters when compared to a unidirectional LSTM. However, in Sec. 3.5 we discuss that node pruning can be used to find a network topology that retains the improved accuracy of the BLSTM with a number of parameters less than for the unidirectional LSTM.

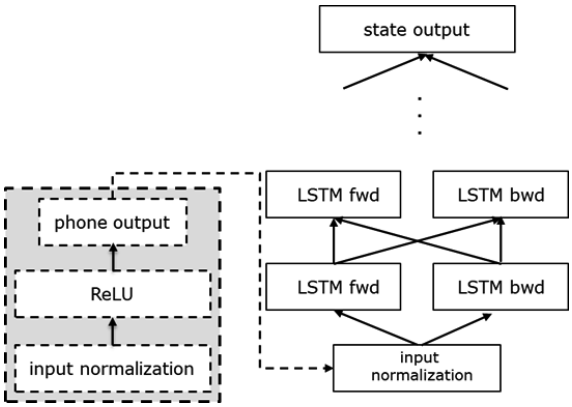

Figure 1 - Bidirectional LSTM topology (solid lines; cf. [9]). Forward (fwd) layers process feature vectors $\{x_1, x_2, \ldots x_T\}$ from left to right $(t = 1, 2, \ldots T)$, while for backward (bwd) layers the order is from right to left ($t = T, T-1, \ldots 1$). Dashed boxes show a single layer feed forward network for phoneme recognition that provides optional input features (cf. Sec. 3.4).

Bidirectional LSTMs frequently yield lower word error rates than unidirectional LSTMs [4, 8], but are not straightforward to use in a production environment that has to carefully balance accuracy versus latency: Whereas there are no latency issues with unidirectional LSTMs that process sequences of feature vectors in forward direction, i.e. for $t = 1, \ldots, T$, BLSTMs make

use of future information by processing sequences also in backward direction, i.e. for $t = T, ..., 1$. Buffering the entire sequence of T feature vectors is clearly infeasible in online recognition, and therefore a limited look-ahead has to be used for the backward directed computation of acoustic model state posteriors. In [9] the BLSTM is evaluated on (overlapping) sliding windows with up to 100 frames and the (weighted) average network output over several windows is used as state posterior. Our approach is similar, but uses just one window and no overlap. In backward direction we reset the cell's internal state c_t and output y_t to zero for each window, whereas in forward direction we do so only at utterance end.

3.3 Data augmentation with WSOLA

Data augmentation via perturbation of the speaking rate has been introduced in [26] and uses an instance of the WSOLA algorithm (waveform similarity overlap and add; cf. [27]). After removing VTLN/VTL*P* from the acoustic frontend we used WSOLA with two different speed perturbations (90%, 110%) as proposed in [26].

3.4 Feature augmentation

More recently, we have started to use additional input features for (B)LSTM training, namely the output of a phoneme recognition network, and features used by an online speaker diarization module currently under development. Both approaches are orthogonal, but have not been combined yet.

3.4.1 Phoneme recognition features

This approach is motivated by the use of two hierarchical feature extraction networks in our previous approach to TANDEM acoustic modeling [28, 11], and a combination of a Time Delay Neural Networks (TDNN) and a LSTMs has been used more recently also in [29]. Our approach uses a shallow network with a single, non-recurrent, rectifying hidden layer and a softmax output layer for phoneme recognition, cf. Figure 1. Phoneme training targets are obtained via coarsening from the same context dependent state alignment that is used for (B)LSTM training, and so far we also use the same input features, i.e. LDA transformed MFCCs, but with a context window. The investigation of alternative features and alignments is subject to future work.

3.4.2 Speaker diarization features

Speaker adapation of DNN/HMM hybrid acoustic models can be achieved by either using speaker adapted input features that are usually computed via fMLLR in a conventional Gaussian framework [30], or by augmenting the training feature vectors with a speaker characteristic. In [31], i-vectors are appended to plain MFCC feature vectors, and it is demonstrated that the approach yields result as good as those obtained with VTLN and speaker adapted input features.

Our approach does not compute i-vectors, but seeks to re-use features designed for online speaker diarization. For that purpose, we compute LDA-projected MAP-adapted supervectors for each speaker. 16 MFCC and their first order delta features are computed and a universal background model (UBM) with 64 Gaussians is trained. For each training speaker we run maximum a-posteriori adaptation of the UBM and create a speaker dependent supervector of dimension 2048 out of the Gaussians' means. An LDA transform trained with background data from approx. 3000 speakers is used to reduce the supervector to typically 50 – 200 dimensions. Finally, the projected supervector is appended to each of the respective speakers' base feature vector (cf. Sec. 2).

3.5 Node Pruning

Deep neural networks with millions of parameters are still a challenge to deploy in online automatic speech recognition systems. For non-recurrent networks weight matrix factorization via singular value decomposition (SVD) [32], weight truncation [33], node pruning or combinations thereof [34] have been proposed to reduce the computational cost without significant loss of accuracy. Apart from [35], where SVD is used at an early stage of layer-wise network construction, all methods usually require some additional training for the fine-tuning of the pruned network.

Node pruning based on the output norm [34] measures the importance $I(i, l)$ of unit i in layer l by the average L1 norm of outgoing links:

$$\boldsymbol{I(i,l)} = \frac{1}{N_{l+1}} \sum_{j=1}^{N_{l+1}} |\boldsymbol{w_{ij}^{l+1}}| \tag{7}$$

For LSTM networks the output y^l of layer l is connected to the 4 forward matrices W_z^{l+1}, W_i^{l+1}, W_f^{l+1}, and W_o^{l+1} of layer $l+1$ (cf. Eq. (1)-(4)), which are stacked vertically before Eq. (7) is evaluated for each output unit i. For BLSTMs the summation is carried out over the stacked forward matrices for both the forward and backward direction of layer $l+1$. Finally, all hidden units are sorted according to their importance I, and units with low values are removed along with all relevant links, which in case of LSTM cells includes links to all internal gates and recurrent matrices as well.

We applied output norm based node pruning to all but the first hidden layer and ended up with pruned networks that show the (expected) bottleneck shape for both directions. Moreover, it turned out that the achieved reduction is roughly the same for both directions.

In the original work [34] it is argued that information about the importance of a node is more reliable if pruning is applied to the fully trained network. Consequently, training time increases because fine-tuning of the reduced network is required to retain accuracy. In our work we seek to keep training time constant by applying node pruning right after the full network has been constructed via layer-wise pre-training. In doing so the information used for pruning may be less reliable, but more training iterations are performed on the reduced network.

4 Experiments

For our experiments we used the WSJCAM0 database [36], which was recorded simultaneously with both a close-talk and a desktop microphone. For training we used approx. 125 hours of close-talk data, while tests were carried out with 2.5 hours of close-talk data and 2 hours of (non-matching) desktop data.

All neural network acoustic models use an output layer with 4500 context-dependent triphone states and were trained using ADAM with model averaging after 2 epochs as optimization method. Starting with an initial learning rate of 10^{-3} we ran 20 training epochs and selected the epoch with lowest cross-validation error rate for decoding. Testing was done with a Kneser-Ney smoothed 4-gram language model (created from the training scripts) with an 11.100 words vocabulary and a total of approx. 255.000 n-grams. Search parameters were tuned only once (for the BLSTM acoustic model with 512 cells in each of the 5 hidden layers) to obtain the same processing speed as for the feed forward baseline network.

Table 1 reports results for both baseline feed forward neural network and uni- and bidirectional LSTMs. Removing VTLN from the acoustic frontend causes an increased word error rate for the test set recorded with the desktop microphone, but – as expected – is easily compensated by the LSTM models. The BLSTM acoustic model works best with a look-ahead of 128 frames (16 % relative improvement over the baseline), but all subsequent experiments

reported below were carried out with a look-ahead of 64 frames in order to keep the decoder's initial latency below one second. The observed degradation for the decoding of desktop data with the unidirectional LSTM acoustic model requires further investigation; further experiments indicate that a larger feature context can further reduce the word error rate for unidirectional LSTMs.

Table 1 - BLSTM decoding results with different look-ahead (la) for matched (close-talk) and unmatched (desktop) test data. Forward networks use a context of 11 frames, whereas LSTM and BLSTM do not.

				% WER		
Network type	Features	Comment		close-talk	desktop	total
FFNN, 5x512	MFCC+VTLP	baseline	ctx=11	20,9	24,5	22,5
	MFCC			20,9	25,2	22,8
LSTM, 5x512	MFCC		ctx= 1	17,2	27,8	21,9
BLSTM, 5x512	MFCC	la= 32		17,3	24,8	20,6
BLSTM, 5x512		la= 64		16,8	23,3	19,7
BLSTM, 5x512		la=128		16,3	22,2	18,9
BLSTM, 5x512		la=256		16,7	22,3	19,2

Table 2 shows results for decoding with the original BLSTM acoustic model and with two variants that apply node pruning after pre-training to reduce the number of hidden units by either 50 or 66 percent. Originally developed for forward DNNs in [34] it turns out that node pruning is applicable to BLSTMs as well and helps to find an appropriate model size.

Table 2 - BLSTM decoding results with and without node pruning after layer-wise pre-training.

			% WER		
Network type	Features	pruning	close-talk	desktop	total
BLSTM, 5x512	MFCC	none	16,8	23,3	19,7
		50%	16,5	22,8	19,3
		66%	16,7	22,9	19,4

Finally, Table 3 gives results for the data and feature vector augmentation techniques sketched in Sections 3.3 and 3.4.

Table 3 – Decoding results with and without additional features (phn: phoneme recognition features, cf. Sec. 3.4.1; spk: speaker diarization features, cf. Sec. 3.4.2).

			% WER		
Network type	Features	Comment	close-talk	desktop	total
BLSTM, 5x512	MFCC	baseline	16,8	23,3	19,7
	MFCC	wsola:0.9, 1.1	16,2	22,9	19,4
	+ phn		16,5	22,7	19,2
	+ 50 spk		16,7	23,3	19,6
FFNN, 5x512	MFCC	baseline	20,9	25,2	22,8
	+ 50 spk		19,8	24,6	21,9
	+200 spk		19,8	24,7	22,0

The use of WSOLA with two different speed perturbations (0.9 and 1.1) and the cascaded network approach depicted in Figure 1 both yield small gains (2.5 % rel.). Whereas for the decoding with FFNN acoustic models the use of 50 additional diarization features yields a relative gain of 3.9 percent, they do not yet improve decoding with BLSTM acoustic models. We plan to revisit this topic once the development of our online speaker diarization module is completed.

5 Conclusion and Outlook

In this paper we described some recent progress on neural network based acoustic modeling in the EML real time transcription platform that yield improvements over our DNN baseline of 22% for matched test data and 10% relative for unmatched test data, respectively. Future work will address training recipes for larger and more heterogeneous training data and further improvement of word error rates for neural network training with speaker diarization features.

6 References

[1] G. E. Hinton, L. Deng, D. Yu et al.: *Deep Neural Networks for acoustic modeling in speech recognition. The shared views of four research groups.* In: *IEEE Signal Processing Magazine*, Vol. 29, No. 6, pp. 82 - 97, 2012.

[2] F. Seide, G. Li, D. Yu: *Conversational Speech Transcription Using Context-Dependent Deep Neural Networks.* In: *Proc. of Interspeech 2011*. Florence, Italy, 2011.

[3] K. Cho, B. v. Merrienboer. C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio: *Learning phrase representations using RNN encoder–decoder for statistical machine translation.* In: *Proc. of Conf. on Empirical Methods in Natural Language Processing.* Doha, Qatar, 2014.

[4] A. Graves and J. Schmidhuber: *Framewise phoneme classification with bidirectional LSTM and other neural network architectures.* In: *Neural Networks*. Vol. 18, No. 5, 2005.

[5] A. Graves, A. Mahamed, and G. Hinton: *Speech recognition with deep recurrent neural networks LSTM.* In: *Proc. of the 2013 IEEE International Conference on Acoustics, Speech, and Signal Processing.* Vancouver, Canada, 2013.

[6] H. Sak, A. Senior, and F. Beaufays: *Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition.* In: *Proc. of Interspeech 2014*. Singapore, 2014.

[7] K. Greff, R. L. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber*: LSTM: A Search Space Odyssey.* In*: IEEE Transactions on Neural Networks and Learning Systems.* Vol. 28, No. 10, Oct. 2017.

[8] A. Zeyer, P. Doetsch, P. Voigtlaender, R. Schlüter, and H. Ney: *A Comprehensive Study of Deep Bidirectional LSTM RNNs for Acoustic Modeling in Speech Recognition*. In: *Proc. of the 2017 IEEE International Conference on Acoustics, Speech, and Signal Processing.* New Orleans, LA, USA, 2017.

[9] A. Zeyer, R. Schlüter, and H. Ney: *Towards online-recognition with deep bidirectional LSTM acoustic models.* In: *Proc. of Interspeech 2016.* San Francisco, CA, USA, 2016.

[10] A. Graves, N. Jaitly, and A. Mahamed: *Hybrid speech recognition with deep bidirectional LSTM.* In: *Proc. of the 2013 IEEE International Conference on Acoustics, Speech, and Signal Processing.* Vancouver, Canada, 2013.

[11] V. Fischer: *Recent improvements to neural network based acoustic modeling in the EML Transcription Platform.* In: *Proc. der Jahrestagung der Deutschen Arbeitsgemeinschaft für Akustik (DAGA*). Aachen, Germany, 2016.

[12] S. Wiesler, A. Richard, P. Golik, R. Schlüter, and H. Ney: *RASR/NN: The RWTH neural network toolkit for speech recognition*. In: *Proc. of the 2014 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Florence, Italy, 2014.

[13] V. Fischer, S. Kunzmann: *Rank based Decoding for Improved DNN/HMM Hybrid Acoustic Models in the EML Transcription Platform.* In: *Proc. of the 12th ITG Symposium on Speech Communication.* Paderborn, Germany, 2016.

[14] M. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, G. Hinton: *On Rectified Linear Units for Speech Processing.* In *Proc. of the 2013 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*. Vancouver, Canada, 2013.

[15] Welling, S. Kanthak, H. Ney: *Improved Methods for Vocal Tract Normalization*. In: *Proc. of the 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Phoenix, AZ., USA, 1999.

[16] N. Jaitly, G. Hinton: *Vocal Tract Length Perturbation (VTLP) improves Speech Recognition.* In: *Proceedings of the 30th International Conference on Machine Learning.* Atlanta, Georgia, 2013.

[17] G. Stemmer, F. Brugnara, and D. Giuliani: *Adaptive Training Using Simple Target Models.* In: *Proc. of the 2005 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*. Philadelphia, PA., USA, 2005.

[18] V. Manohar, D. Povey, S. Khudanpur: *Semi-supervised Maximum Mutual Information Training of Deep Neural Network Acoustic Models.* In: *Proc. of Interspeech 2015.* Dresden, Germany, 2015.

[19] D. Rybach, C. Gollan, G. Heigold, B. Hoffmeister, J. Lööf, R. Schlüter, H. Ney: *The RWTH Aachen University Open Source Speech Recognition System.* In: *Proc. of Interspeech 2009.* Brighton, UK, 2009.

[20] D. Nolden, H. Ney, R. Schlüter: *Time Conditioned Search in Automatic Speech Recognition Reconsidered.* In: *Proc. of Interspeech 2010*. Makuhari, Chiba, Japan, 2010.

[21] O. Ghahabi, W. Zhou, V. Fischer: *A Robust Voice Activity Detection for Real-Time Automatic Speech Recognition.* In: *Proc. of ESSV 2018,* Ulm 2018.

[22] P. Doetsch, A. Zeyer, P. Voigtlaender, I. Kulikov, R. Schlüter, and H. Ney: *RETURNN: the RWTH extensible training framework for universal recurrent neural networks.* In: *Proc. of the 2017 IEEE International Conference on Acoustics, Speech, and Signal Processing.* New Orleans, LA, USA, 2017.

[23] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardain, J. Turian, D. Warde-Farley, and Y. Bengio: *Theano: a CPU and GPU math expression Compiler*. In: *Proceedings of the Python for Scientific Computing Conference (SciPy)*. Austin, TX, USA, 2010.

[24] M. Zeiler: *ADADELTA: an adaptive learning rate method*. *arXiv preprint arXiv: 1212.5701*, 2012.

[25] D. Kingma and J. Ba: *Adam: A method for stochastic optimization.* In: *Proc. of the 3rd International Conference on Learning Representations.* San Diego, CA, 2015.

[26] V. Ko, V. Peddinti, D. Povey, and S. Khudanpur*: Audio augmentation for speech recognition.* In: *Proc. of Interspeech 2015*, Dresden, Germany, 2015.

[27] W. Verhelst and M. Roelands: *An Overlap-Add technique based on waveform similarity (WSOLA) for high quality time-scale modifications of speech.* In: *Proc. of the 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Minneapolis, MN, USA, 1993.

[28] C. Plahl, R. Schlüter, H. Ney: *Hierarchical Bottle Neck Features for LVCSR.* In: *Proc. of Interspeech 2010*, Makuhari, Japan, 2010, 1197–1200.

[29] V. Peddinti, G. Chen, V. Manohar, T. Ko, D. Povey, S. Khudanpur: *The JHU ASPIRE Ssystem: Robust LVCSR with TDNNS, iVector adaptation and RNN-LMs.* In: *Proc. of the 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Scottsdale, AZ, USA, 2015.

[30] S. Parthasarathi, B. Hoffmeister, S. Matsoukas, A. Mandal, N. Strom, and S. Garimella: *fMLLR Based Feature-Space Speaker Adaptation of DNN Acoustic Models*. In: *Proc. of Interspeech 2015*, Dresden, Germany, 2015.

[31] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny: *Speaker adaptation of neural network acoustic models using i-vectors.* In*: Proc. of the 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Olomouc, Czech Republic, 2013.

[32] J. Xue, J. Li, and Y. Gong: *Restructuring of deep neural network acoustic models with singular value decomposition.* In: *Proc. of Interspeech 2013*, Lyon, France, 2013.

[33] D. Yu, F. Seide, G. Li, and L. Deng: *Exploiting sparseness in deep neural networks for large vocabulary speech recognition.* In: *Proc. of the 2012 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Kyoto, Japan, 2012.

[34] T. He, Y. Fan, Y. Qian, T. Tan, and K. Yu: *Reshaping deep neural network for fast decoding by node-pruning.* In: *Proc. of the 2014 IEEE International Conference on Acoustics, Speech, and Signal Processing.* Florence, Italy, 2014.

[35] K. Kilgour, I. Tseyzer, T. Nguyen, S. Stueker, and A. Waibel: *Growing a Deep Neural Network Acoustic Model with Singular Value Decomposition.* In: *Proc. of the 12th ITG Symposium on* Speech Communication. Paderborn, Germany, 2016.

[36] T. Robinson, J. Fransen, D. Pye, J. Foote, S. Renals, P. Woodland, and S. Young: *WSJCAM0 Cambridge Read News LDC95S24. Web Download.* Linguistic Data Consortium, Philadelphia, 1995.