Interspeech 2018
2-6 September 2018, Hyderabad

An investigation of mixup training strategies for acoustic models in ASR

Ivan Medennikov[1,2], *Yuri Khokhlov*[1], *Aleksei Romanenko*[2,3], *Dmitry Popov*[3], *Natalia Tomashenko*[2,4], *Ivan Sorokin*[3], *Alexander Zatvornitskiy*[1,2,3]

[1] STC-innovations Ltd, St. Petersburg, Russia
[2] ITMO University, St. Petersburg, Russia
[3] Speech Technology Center Ltd, St. Petersburg, Russia
[4] LIUM, University of Le Mans, France

{medennikov,khokhlov,romanenko,popov-d,sorokin,zatvornitskiy}@speechpro.com,
natalia.tomashenko@univ-lemans.fr

Abstract

Mixup is a recently proposed technique that creates virtual training examples by combining existing ones. It has been successfully used in various machine learning tasks. This paper focuses on applying mixup to automatic speech recognition (ASR). More specifically, several strategies for acoustic model training are investigated, including both conventional cross-entropy and novel lattice-free MMI models. Considering mixup as a method of data augmentation as well as regularization, we compare it with widely used speed perturbation and dropout techniques. Experiments on Switchboard-1, AMI and TED-LIUM datasets shows consistent improvement of word error rate up to 13% relative. Moreover, mixup is found to be particularly effective on test data mismatched to the training data.
Index Terms: speech recognition, mixup, acoustic model training, data augmentation, regularization, lattice-free MMI

1. Introduction

Nowadays, acoustic models based on neural networks play a dominant role in automatic speech recognition (ASR) [1–3]. These networks are usually trained by minimizing average loss function, such as Cross-Entropy (CE), over the training data. This leads to the problem: the network tries to memorize instead of generalize from the data [4]. As a consequence, prediction accuracy decreases drastically on test data which are outside of the training distribution. There are many techniques for improving the generalization ability, such as data augmentation (e.g. speed and volume perturbation [5], vocal tract length perturbation [6]), and regularization (e.g. dropout training [7]).

One of these techniques is so-called mixup, recently proposed in [4]. This technique constructs virtual training examples by combining linearly both input features and output labels. Mixup demonstrated impressive effectiveness in various machine learning tasks, such as image data classification (ImageNet-2012 [8], CIFAR-10/100 [9]), tabular data classification (UCI [10]), and speech recognition (Google commands [11]). However, the speech recognition task considered in [4] was very small and simple: it consisted of 65,000 1-second long utterances representing one of 30 speech commands. On the other hand, actual ASR tasks containing hundreds of hours of continuous speech are much more challenging.

This paper is focused on exploring mixup technique for acoustic model training on large-scale ASR tasks. Three popular ASR benchmarks representing various aspects of speech recognition are considered: Switchboard-1 [12] (conversational telephone speech), TED-LIUM [13] (lectures), and AMI [14] (meetings, distant conversational speech). We report consistent WER reduction on all these tasks.

Today most widely used acoustic models are based on recurrent neural networks (e.g. Long Short-Term Memory networks, LSTM [15–17]), which are trained to classify Hidden Markov Model (HMM) states on long sequences instead of independent frames. Therefore, a sequence of features and corresponding supervision is considered as a training example. For CE training, supervision is just a sequence of class labels. However, speech recognition is inherently a sequence classification problem, so acoustic models trained using CE criterion are suboptimal in terms of WER. Sequence-discriminative training designed to handle this problem obtains significant WER improvement in many tasks [18–20]. Traditionally this is performed by retraining CE-trained model using one of sequence-discriminative criteria, for example state-level Minimum Bayes Risk (sMBR) [18]. On the other hand, the recently proposed Lattice-Free Maximum Mutual Information (LF-MMI) [21] approach allows to carry out sequence-discriminative training from scratch, without CE pre-training stage. LF-MMI has a number of advantages over traditional sequence-discriminative training, and it provides better recognition accuracy. Various schemes of employing mixup training in both CE and LF-MMI frameworks are explored in this paper.

Considering mixup as both data augmentation and regularization technique, we compared it with widely used speed perturbation [5] data augmentation technique and dropout [7] regularization technique. Other important aspects of applying mixup in training of ASR acoustic model are studied as well.

The rest of the paper is organized as follows. In Section 2, we discuss mixup and similar approaches. Strategies for applying mixup to acoustic model training are described in Section 3. Section 4 describes experimental setup. Results of experiments are presented in Section 5. Finally, Section 6 concludes the paper and discusses future work.

2. Related Work

Lack of generalization is one of the most important problems of neural networks. Due to this, performance may degrade drastically on test data unseen in the training process. There are numerous approaches aimed at increasing the generalization ability of neural networks. These approaches can be divided into two major groups.

The first group consists of data augmentation techniques which try to increase the amount of training data through various modifications of original training dataset. This usually leads to more precise representation of the input data distribu-

10.21437/Interspeech.2018-2191

tion, thereby improving the generalization ability of the trained models.

The second group of approaches is regularization. Unlike the data augmentation, these approaches operate on the training process level. As an example, the dropout technique randomly removes part of connections in a neural network during the training and thereby helps to avoid overfitting (memorization). There are different schemes of applying the dropout technique to various machine learning tasks [7, 22]. Another popular regularization technique is batch normalization [23] which normalizes the outputs of hidden layers.

Mixup is a new technique aimed at solving the overfitting problem [4]. It combines arbitrarily chosen training samples and their labels to generate new training data:

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j$$
$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j$$

where x_i, x_j are raw input vectors, and y_i, y_j are one-hot label encodings. Due to its nature, mixup encourages linear behavior of neural networks on between-class data. Mixup can be considered as both data augmentation and regularization technique. Moreover, it is additive to all aforementioned approaches and can benefit from the combination with them.

3. Acoustic model training with mixup

This section describes various schemes for employing mixup in acoustic model training. As already mentioned, we consider a sequence of features and corresponding supervision as a training example, instead of a features-label pair for individual frame.

3.1. Cross-Entropy Training

Supervision for cross-entropy acoustic model training is a sequence of 1-hot labels representing tied HMM states. Simplest way to apply mixup for CE training is frame-by-frame interpolation of features and labels for two sequences, sharing the same mixture weight across all frames (global scheme). The second scheme is mixing of each frame in a sequence with a randomly chosen frame near to the current, e.g. in 3 frames (local scheme). We expect performance improvement using the local scheme, because it allows smoother transitions between the states. We also tried mixing features in a sequence with random features having the same label (class scheme), and mixing a sequence with itself shifted by several frames (shift scheme).

3.2. LF-MMI training

Supervision in LF-MMI is a numerator Finite State Acceptor (FSA) representing alternative pronunciation of the training utterances and allowing a little "wiggle room" to vary from reference phone positions. Details on LF-MMI numerator graphs can be found in paper [21] and in Kaldi "Chain" models documentation [24].

We suppose that, unlike CE case, combining of training examples is not sufficient to carry out mixup training for LF-MMI models completely correctly. We propose the training procedure called gradient mixup consisting of the following steps:

1. Doing forward propagation and computing numerator occupancies separately for both combined sequences.
2. Mixing input sequences with weight λ and doing forward propagation.
3. Computing denominator occupancies using posterior probabilities corresponding to the mixed sequence.
4. Calculating error signals for both sequences and mixing these errors with weight λ.
5. Back-propagating the resulting error signal.

The main disadvantage of this procedure is a very high computational cost. It can be simplified by omitting separate forward propagation in the first step. In this case, numerator occupancies for both sequences are calculated using mixed sequence posterior probabilities (simplified gradient mixup).

We also considered applying mixup on the level of LF-MMI training examples. In this approach features for 2 sequences are combined identically to the global scheme described in Subsection 3.1. Numerator graphs are merged into one graph in a parallel way if both mixture weights are higher than the threshold. Otherwise, the FSA corresponding to the sequence with higher weight is taken. This approach provides less control on supervision mixing, but we assume that weighting is applied implicitly during the forward-backward step. The following schemes for weights scaling in the combined numerator graph are considered:

- No scaling: does not change weights of hypotheses in the resulting FSA.
- Default scaling: modifies all hypotheses corresponding to both combined sequences by adding penalties equal to $-\ln\lambda$ and $-\ln(1-\lambda)$ respectively, where λ and $1-\lambda$ are mixture weights for these sequences.
- Balanced scaling: penalizes hypotheses corresponding to a lower mixture weight, while rewards the ones corresponding to higher mixture weight. Denoting the lower weight as λ, the penalty (positive value) is equal to $0.5\,(\ln(1-\lambda) - \ln\lambda)$ and the reward (negative value) is equal to $0.5\,(\ln\lambda - \ln(1-\lambda))$.

This approach is easy to implement, as it does not require any modifications in the training procedure. Furthermore, it is almost computationally free.

4. Experimental Setup

4.1. Datasets

Main experiments are conducted on the 300 hour Switchboard English conversational telephone speech task [12] being the most studied ASR benchmark today [2, 3, 19, 21, 25–27]. We used Switchboard-1 Release 2 (LDC97S62) as the training set. Results are reported on the Hub5 2000 (LDC2002S09) evaluation set containing 20 ten-minute conversations from Switchboard (SW) and 20 ten-minute conversations from CallHome English (CH). It should be noted that the SW part is quite similar to the training data, while the CH part differs significantly and is much harder to recognize due to this.

The second corpus used for the experiments in this paper is TED-LIUM [13]. We used the last (second) release of this corpus [28]. This publicly available data set contains 1495 TED talks that amount to 207 hours of speech data from 1242 speakers recorded in 16kHz. The training, development and two test sets were chosen in the same way as described in [29]

We also present results on the AMI meeting corpus [14] for individual headset microphone (IHM) and single distant microphone (SDM) tasks.

Table 1: *Comparison of mixup schemes for CE BLSTMP model on the Switchboard task*

Scheme	Details	WER		
		SW	CH	ALL
baseline	—	10.2	20.1	15.2
local	range of 3 frames	10.1	20.0	15.1
class	max weight 0.1	10.4	20.8	15.7
shift	1-3 frames	10.0	19.9	15.0
global	—	**9.8**	**18.8**	**14.3**

Table 2: *Comparison of mixup schemes for LF-MMI TDNN-LSTMP model trained without dropout on the Switchboard task*

Scheme	Threshold	Epochs	WER		
			SW	CH	ALL
baseline	—	2	9.6	20.1	14.9
no scaling	0.001	2	**8.9**	17.7	**13.4**
default	0.001	2	9.3	18.0	13.7
balanced	0.001	2	9.4	**17.5**	13.5
no scaling	0.1	2	9.0	17.6	13.4
no scaling	0.2	2	9.0	17.6	13.4

4.2. Training details

We performed all experiments using the Kaldi ASR Toolkit[1] [30]. Baseline recipes are *swbd/s5c* for the Switchboard task and *ami/s5b* for the AMI IHM/SDM tasks. Acoustic models considered are Bidirectional LSTM with projections (BLSTMP) [16] for cross-entropy training (only on the Switchboard task) and a mixture of Time Delay Neural Network and unidirectional LSTM with projections (TDNN-LSTMP) [31] for LF-MMI training. All these models are trained in the *nnet3* Kaldi setup using 40-dimensional Mel-frequency cepstral coefficients (MFCC) without cepstral truncation. Configurations of neural networks are exactly the same as described in [22] (see also local/chain/tuning/run_tdnn_lstm_11.sh in the Kaldi recipes).

The following techniques are also applied: speaker adaptation using i-vectors [32], speed perturbation for 3-fold data augmentation [5], dropout regularization [7] (for some TDNN-LSTMP models). For the latter, the schedule '0,0@0.2,p@0.5,0' described in [22] was used. We varied only the peak dropout probability p in the experiments.

In contrast to the Kaldi recipes, for most of experiments we reduced maximum number of GPUs from default values of 12–16 to 4 due to hardware limitations. This led to minor performance degradation for LF-MMI models, which can be partly compensated by reducing number of epochs. Nevertheless, we also trained final LF-MMI models for the Switchboard task in 16-GPU setup in order to compare with the results of the baseline Kaldi recipe (see also [22]).

For TED-LIUM experiments, there are several differences from the Kaldi recipe *tedlium/s5_r2*. First, datasets are not the same as in the recipe (see Subsection 4.1 for details). Second, we did not use neither i-vectors, nor speed perturbation in these experiments.

[1] Version 5.3.78∼1-d883e

Table 3: *Comparison of mixup schemes for LF-MMI TDNN-LSTMP model trained without dropout on the TED-LIUM task*

Scheme	Threshold	Epochs	WER		
			dev	test1	test2
baseline	—	4	11.5	8.6	11.1
no scaling	0.001	6	10.7	7.8	**10.1**
default	0.001	6	10.7	7.9	**10.1**
balanced	0.001	6	**10.4**	**7.6**	10.2

4.3. Mixup details

In this paper we experimented with mixup on the level of training examples only. We are going to investigate the gradient mixup approach for LF-MMI in the future work.

Mixup examples construction schemes described in Section 3 are implemented in our Kaldi-compatible tools[2] which are used instead of *nnet3-copy-egs* in CE training and *nnet3-chain-copy-egs* in LF-MMI training. Training examples stored in the archive are processed sequentially with a randomly sampled mixture weight and chosen mixup scheme. If the mixup scheme requires the second sequence, it is chosen randomly from the same archive.

The original work [4] uses symmetric Beta distribution for sampling of mixture weights. However, in this case some sequences in the training data will be dominated by the other sequences. In order to prevent this situation, the weight for a sequentially taken example is forced to be in $[0.5, 1.0]$ range. Our preliminary experiments shown that this restriction leads to better performance. We also found that the Beta distribution parameter equal to 1 (which means uniform distribution) is close to optimal value. So, uniform distribution with the aforementioned restriction was used. In all experiments, mixup was omitted for 10% of training examples chosen randomly.

5. Experiments

This section presents our experiments on applying mixup for acoustic model training. Results on AMI are obtained with 3-gram language models. Results on Switchboard are reported after rescoring of word lattices with a 4-gram language model. For TED-LIUM, a 4-gram language model is used.

5.1. Experiments with mixup schemes

The first set of experiments was aimed at empirical evaluation of mixup training schemes described in Section 3.

Table 1 presents results for CE training on the Switchboard task (all models have been trained with 8 epochs). It can be seen that only the global scheme improves WER significantly, while local and shift schemes almost do not affect model performance. The class scheme harms significantly, probably due to corruption of the original sequence with independent frames belonging to different speakers. A way to improve this scheme is using equally labeled chunks instead of individual frames.

Tables 2 and 3 show results for LF-MMI training on the Switchboard and TED-LIUM tasks respectively. We notice that the results are close for all considered mixup schemes: no scaling one is slightly better on Switchboard, whereas balanced one performs better on TED-LIUM. Numerator mixing threshold also does not consistently affect the performance. Thus, in the

[2] The source code is available at https://github.com/speechpro/mixup

Table 4: *Comparison of mixup and speed perturbation (SP) for CE BLSTMP model on the Switchboard task*

Mixup	SP	Epochs	WER		
			SW	CH	ALL
−	−	5	11.2	21.4	16.4
−	+	8	10.2	20.1	15.2
+	−	16	10.4	20.0	15.2
+	+	8	9.8	18.8	14.3
+	+	12	9.7	18.4	14.1
+	+	16	**9.4**	**18.1**	**13.8**

Table 5: *Comparison of mixup and speed perturbation (SP) for LF-MMI TDNN-LSTMP model on the Switchboard task*

Mixup	SP	Epochs	WER		
			SW	CH	ALL
−	−	2	10.0	20.9	15.5
−	+	2	9.6	20.1	14.9
+	−	5	9.6	18.6	14.0
+	+	2	9.0	17.6	13.4
+	+	3	**9.0**	**17.4**	**13.3**

further experiments we used the simplest mixup scheme with no scaling and default threshold value of 0.001.

5.2. Comparing mixup and speed perturbation

The second set of experiments compares mixup with popular speed perturbation data augmentation technique [5] on the Switchboard task. Optimal number of epochs was tuned for each model. The results are given in Table 4 and Table 5 for CE BLSTMP and LF-MMI TDNN-LSTMP models respectively. It can be seen that mixup performs as well as speed perturbation for the CE model, while outperforming it significantly for the LF-MMI model. Moreover, these techniques are highly complementary.

5.3. Comparing mixup and dropout

The next experiment compares mixup with dropout regularization technique [7] for TDNN-LSTMP LF-MMI model on the Switchboard task. As already mentioned in Subsection 4.2, we used dropout schedule described in [22] and varied only the peak dropout probability. As shown in Table 6, mixup outperforms dropout. Combining mixup and dropout with default peak probability leads to performance degradation. However, dropout with small probability provides some improvement in combination with mixup.

5.4. Other experiments

Table 7 shows the results of mixup training for LF-MMI TDNN-LSTMP model on the AMI IHM/SDM tasks. Significant WER reduction is observed in both of these tasks as well.

Finally, the last experiment was conducted in order to make an exact comparison with actual Kaldi LF-MMI baseline results [22] obtained using 16 GPUs. Table 8 shows the results of the comparison. As can be seen, mixup does not help on the Switchboard subset. However, it reduces WER on the Call-Home data significantly, which means improved robustness to a mismatch between training and test conditions.

Table 6: *Comparison of mixup and dropout for LF-MMI TDNN-LSTMP model on the Switchboard task*

Mixup	Dropout prob.	Epochs	WER		
			SW	CH	ALL
−	0.0	2	9.6	20.1	14.9
−	0.3	2	9.0	18.6	13.9
+	0.0	2	8.9	17.7	13.4
+	0.3	2	9.1	18.0	13.7
+	0.1	2	9.0	17.6	13.4
+	0.05	2	9.0	**17.2**	**13.2**
+	0.05	3	**8.8**	17.4	**13.2**

Table 7: *Results of mixup-trained LF-MMI TDNN-LSTMP model on the AMI IHM/SDM tasks*

Model	Dropout prob.	Epochs	WER	
			dev	eval
IHM baseline	0.3	3	20.3	20.1
mixup IHM	0.05	3	**19.4**	**18.8**
SDM baseline	0.3	3	36.1	40.2
mixup SDM	0.05	3	**34.8**	**38.4**

6. Conclusions and Future Work

In this paper we applied mixup technique to ASR acoustic model training and found it to be highly effective for cross-entropy as well as LF-MMI scenarios. Relative WER reduction up to 13% was obtained on various ASR tasks. The main advantages of mixup are:

- Significant performance improvement in mismatched test conditions.
- Low implementation cost.
- Minimal impact on training time.

Mixup performs as well or better than speed perturbation data augmentation technique, and outperforms dropout regularization technique. Furthermore, these techniques are found to be complementary.

Our future work will focus on further studying of simple mixup schemes as well as implementing and exploring the proposed gradient mixup training scheme for LF-MMI. It is also interesting to investigate various mixture weights distributions and scheduling variants. Finally, we are going to employ mixup training for neural network based language models for ASR.

7. Acknowledgements

This work was financially supported by the Ministry of Education and Science of the Russian Federation, Contract 14.575.21.0132 (IDRFMEFI57517X0132).

Table 8: *Results of final mixup-trained LF-MMI model (16-GPU setup) on the Switchboard task*

Model	Dropout prob.	Epochs	WER		
			SW	CH	ALL
Kaldi baseline	0.3	4	**8.8**	18.1	13.5
mixup	0.05	4	**8.8**	**16.7**	**12.8**

8. References

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, 2012.

[2] G. Saon and M. Picheny, "Recent advances in conversational speech recognition using convolutional and recurrent neural networks," *IBM Journal of Research and Development*, vol. 61, no. 4, pp. 1:1–1:10, 2017.

[3] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, "The Microsoft 2017 Conversational Speech Recognition System," *arXiv preprint arXiv:1708.06073*, 2017.

[4] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[5] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio Augmentation for Speech Recognition," *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2015.

[6] N. Jaitly and G. Hinton, "Vocal tract length perturbation (VTLP) improves speech recognition," *International Conference on Machine Learning (ICML) Workshop on Deep Learning for Audio, Speech, and Language Processing*, 2013.

[7] N. Srivastava, H. G., A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[8] B. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[9] A. Krizhevsky, "Learning multiple layers of features from tiny images," Master's thesis, Department of Computer Science, University of Toronto, 2009.

[10] M. Lichman, "UCI machine learning repository," https://archive.ics.uci.edu/ml/index.php, 2013.

[11] P. Warden, "Launching the Speech Commands Dataset," https://research.googleblog.com/2017/08/launching-speech-commands-dataset.html, 2017.

[12] J. Godfrey, E. Holliman, and J. McDaniel, "Switchboard: telephone speech corpus for research and development," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, pp. 517–520, 1992.

[13] A. Rousseau, P. Deléglise, and Y. Estève, "TED-LIUM: an automatic speech recognition dedicated corpus," *LREC*, 2012.

[14] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, and V. e. a. Karaiskos, "The AMI Meeting Corpus," *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, vol. 88, 2005.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] H. Sak, A. Senior, and F. Beaufays, "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling," *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2014.

[17] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition," *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 1468–1472, 2015.

[18] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3761–3764, 2016.

[19] K. Vesely, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2013.

[20] H. Sak, O. Vinyals, G. Heigold, A. Senior, E. McDermott, R. Monga, and M. Mao, "Sequence Discriminative Distributed Training of Long Short-Term Memory Recurrent Neural Networks," *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2014.

[21] D. Povey, V. Peddinti, D. Galvez, P. Ghahrmani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 2751–2755, 2016.

[22] G. Cheng, V. Peddinti, D. Povey, V. Manohar, S. Khudanpur, and Y. Yan, "An Exploration of Dropout with LSTMs," *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 1586–1590, 2017.

[23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *in Proc. of ICML*, p. 448456, 2015.

[24] http://kaldi-asr.org/doc/chain.html.

[25] I. Medennikov, A. Prudnikov, and A. Zatvornitskiy, "Improving English Conversational Telephone Speech Recognition," *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 2–6, 2016.

[26] W. Hartmann, H. R., T. Ng, J. Ma, F. Keith, and M.-H. Siu, "Improved Single System Conversational Telephone Speech Recognition with VGG Bottleneck Features," *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 112–116, 2017.

[27] Z. Tüske, W. Michel, R. Schlüter, and H. Ney, "Parallel Neural Network Features for Improved Tandem Acoustic Modeling," *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 1651–1655, 2017.

[28] "TED-LIUM corpus release 2," http://www.openslr.org/resources/19/TEDLIUM_release2.tar.gz.

[29] N. Tomashenko, Y. Khokhlov, and Y. Esteve, "On the Use of Gaussian Mixture Model Framework to Improve Speaker Adaptation of Deep Neural Network Acoustic Models," in *INTERSPEECH*, 2016, pp. 3788–3792.

[30] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 1–4, 2011.

[31] V. Peddinti, Y. Wang, D. Povey, and S. Khudanpur, "Low latency acoustic modeling using temporal convolution and lstms," *IEEE Signal Processing Letters*, vol. 25, no. 3, pp. 373–377, March 2018.

[32] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 55–59, 2013.