End-to-end Training of a Large Vocabulary End-to-end Speech Recognition System

Chanwoo Kim, Sungsoo Kim, Kwangyoun Kim, Mehul Kumar, Jiyeon Kim, Kyungmin Lee, Changwoo Han, Abhinav Garg, Eunhyang Kim, Minkyoo Shin, Shatrughan Singh, Larry Heck, Dhananjaya Gowda

Samsung Research

{chanw.com, ss216.kim, ky85.kim, mehul3.kumar, jstacey7.kim, k.m.lee, cw1105.han, abhinav.garg, sc.ehkim.jin, mk0211.shin, shatrughan.s, larry.h, d.gowda}@samsung.com

Abstract

In this paper, we present an end-to-end training framework and strategies for building state-of-the-art end-to-end (E2E) speech recognition systems. Our training system utilizes a cluster of Central Processing Units (CPUs) and Graphics Processing Units (GPUs). The entire data reading, large scale data augmentation, neural network parameter updates are all performed on-the-fly. We use vocal tract length perturbation and an acoustic simulator for data augmentation. The processed features and labels are sent to the GPU cluster. Horovod `allreduce` approach is employed to train neural network parameters. We evaluated the effectiveness of our system on the standard Librispeech corpus [1] and 10,000-hr anonymized Bixby English dataset. Our end-to-end speech recognition system built using this training infrastructure showed 2.85 % WER on `test-clean` for the LibriSpeech corpus after shallow fusion with a recurrent neural network (RNN) language model (LM). For English Bixby open domain test set, we obtained a WER of 7.92 % using a bidirectional full attention (BFA) E2E model after shallow fusion with an RNN-LM. When the monotonic chunckwise attention (MoCha) based approach is employed for streaming speech recognition, we obtained a WER of 9.95 % on the same Bixby open domain test set.

Index Terms: end-to-end speech recognition, distributed training, example server, data augmentation, acoustic simulation

1. Introduction

In recent years, deep learning techniques have significantly improved speech recognition accuracy [2, 3, 4, 5, 6, 7]. This improvement has come about from the shift from Gaussian Mixture Model (GMM) to the Feed-Forward Deep Neural Networks (FF-DNNs), FF-DNNs to Recurrent Neural Network (RNN) and in particular the Long Short-Term Memory (LSTM) networks. Thanks to these advances, voice assistant devices such as Google Home [8, 9] or Amazon Alexa or Samsung Bixby are being used at many homes and on personal devices.

Recently there has been increasing interest in switching from the conventional Weighted Finite State Transducer (WFST) based decoder using an Acoustic Model (AM) and a Language Model (LM) to a complete end-to-end all-neural speech recognition systems [10, 11, 12]. These complete end-to-end systems have started surpassing the performance of the conventional WFST-based decoders with a very large training database, a better choice of target unit such as Byte Pair Encoded (BPE) subword units, and an improved training methodology such as Minimum Word Error Rate (MWER) training.

Another important aspect of these recent improvements in speech recognition performance is data. To build high performance speech recognition systems for conversational speech, we need to use a large amount of speech data covering various domains. In [13], it has been shown that we need a very large training set ($\sim$125,000 hours of semi-supervised speech data) to achieve high speech recognition accuracy for difficult tasks like video captioning. To train neural networks using such large amounts of speech data, we usually need multiple Central Processing Units (CPUs) or Graphics Processing Units (GPUs) or GPU clusters [14, 15].

With widespread adoption of voice assistant speakers, far-field speech recognition has become very important. In far-field speech recognition, the impacts of reverberation and noise are much larger than those in near-field cases. Traditional approaches to far-field speech recognition include noise robust feature extraction algorithms [16, 17], or multi-microphone approaches [18, 19, 20, 21, 22]. More recently, approaches using data augmentation has been gaining popularity for far-field speech recognition [23, 24, 25]. An "acoustic simulator" [8, 24] is used to generate simulated speech utterances for millions of different room dimensions, a wide distribution of reverberation time and signal-to-noise ratio. In a similar spirit, Vocal Tract Length Perturbation (VTLP) has been proposed [26] to tackle the speaker variability issue. As shown in our recent paper [27], VTLP is especially useful when the speaker variability in the training database is not sufficient. For these kinds of data augmentation, processing on CPUs is more desirable than processing on GPUs. Due to this, we have proposed an end-to-end training approach using Example Servers (ES) and workers. Example servers are typically run on the CPU cluster performing data reading, data augmentation, and feature extraction [24].

In this paper, we describe the structure of our end-to-end training system to train an end-to-end speech recognition system. This training system has several advantages over previous systems described in [24]. First, instead of using the `QueueRunner`, we use a more efficient data queue using `tf.data` in Tensorflow [28]. Second, instead of precalculating information about room configurations and room impulse responses in the acoustic simulator, these are calculated on-the-fly. Thus, the entire training system runs on -the-fly. Additionally, instead of using the parameter server-worker structure, we use an `allreduce` approach implemented in the Horovod [29] distributed training framework, which has been shown to be more efficient. The system described in [14], is

Thanks to Samsung Electronics for funding this research. The authors are thankful to Executive Vice President Seunghwan Cho and members of Speech Processing Lab at Samsung Research.

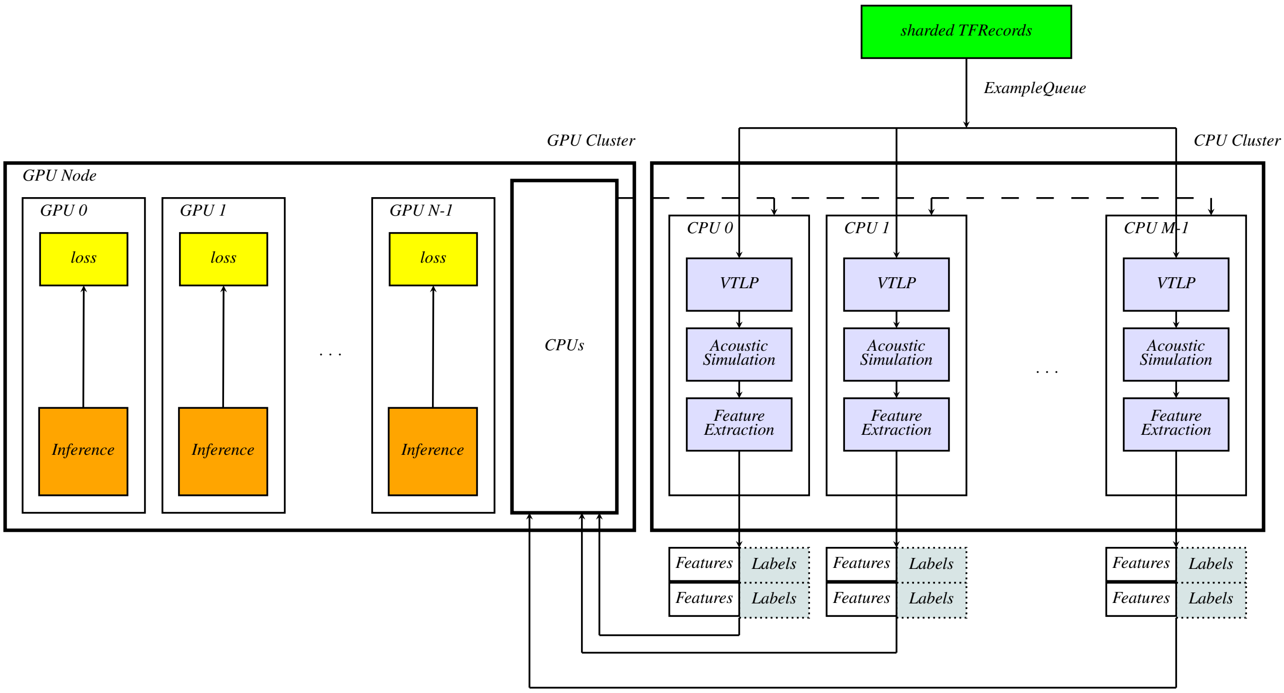

Figure 1: *The Samsung Research end-to-end training framework for building an end-to-end speech recognition system with multi CPU-GPU clusters and on-the-fly data processing and augmentation pipeline.*

designed to train the acoustic model part of the speech recognition system where as our training system trains the complete end-to-end speech recognition system.

The rest of the paper is organized as follows: We describe the entire training system structure in detail in Sec. 2. The structure of the end-to-end speech recognition system is described in Sec. 3. Experimental results that demonstrates the effectiveness of our speech recognition system is presented in Sec. 4. We conclude in Sec. 5.

2. Overall structure of the end-to-end speech recognition

In this section, we describe the overall structure of our end-to-end training system. Fig. 1 shows how the entire system is structured. Our system consists of a cluster of CPUs and a cluster of GPUs. Each GPU node of the GPU cluster has eight nVidia™P-40 GPUs and two Intel E5-2690 v4 CPUs. Each of these CPUs has 14 cores. The large box on the left hand side of Fig. 1 denoted "GPU cluster" shows a typical GPU node with N GPUs. The large box on the right shows a "CPU cluster" of M CPUs, each running an independent data pipeline.

2.1. Training job launch

The main process of the training system runs on one of CPU cores of the GPU cluster. This CPU portion of the GPU node is represented as a box in the right hand side of the GPU node box. When the training job starts, this main training process launches multiple example server jobs on the CPU cluster using the `IBM Platform LSF` [30]. In Fig. 1, this launching process is represented by a dashed arrow from the CPU portion of the GPU node to the CPU cluster.

2.2. Data reading using an example queue

In the CPU cluster, each CPU runs one example server which reads speech utterance and transcript data from sharded `TFRecords` defined in Tensorflow [28]. The TFRecord format is a simple format in Tensorflow for storing a sequence of binary records. To support efficient reading using multiple CPUs, we use sharded `TFRecords`.

To read large-scale data efficiently in parallel, we use an example queue shown in the left side of Fig. 1. The original speech waveform data, transcripts, and meta data are stored in sharded `TFRecords`. The data pipeline is implemented using `tf.data` in Tensorflow [28], and contains the data augmentation and feature extraction blocks. These `tf.data` APIs are efficient in building complex pipelines by applying a series of elementary operations. We perform data interleaving and parallel reading using `tf.contrib.data.parallel_interleave`, shuffling using `tf.data.Datatset.shuffle`, and padding using `tf.data.Dataset.padded_batch`.

2.3. Data augmentation and feature extraction

To improve robustness against speaker variability, we apply an on-the-fly VTLP algorithm on the input waveform. The warping factor is generated randomly for each input utterance. Unlike conventional VTLP approaches in [31, 26], we resynthesize the processed speech. The purpose of doing this is to apply VTLP before applying the acoustic simulator. One more advantage is that this resynthesis approach enables us to use a window length optimal for VTLP different from that used in feature processing. In the bilinear transformation, the relation between the input and output discrete-time frequencies is given by:

$$\omega_k' = \omega_k + 2\tan^{-1}\left(\frac{(1-\alpha)\sin(\omega_k)}{1-(1-\alpha)\cos(\omega_k)}\right). \quad (1)$$

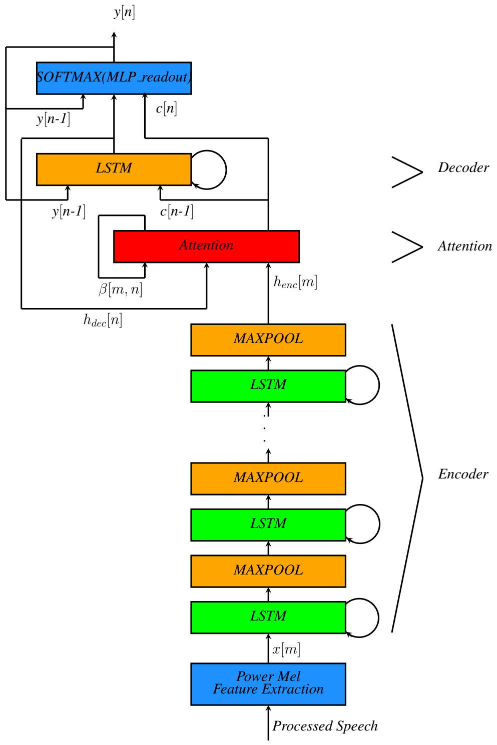

Figure 2: *The structure of the entire end-to-end speech recognition system.*

where $w_k = 2\pi k/K$ is the discrete-time frequency and K is the DFT size. More details about our VTLP algorithm can be found in our another paper [27]. The acoustic simulator in Fig. 1 is similar to what we described in [8, 24]. One difference compared to our previous one in [8] is that we do not pre-calculate room impulse responses, but instead they are calculated on-the-fly. For feature processing we use `tf.data.Dataset.map` API. Instead of using the more conventional log-mel or MFCC features, we use the power mel filterbank energies, since it shows slightly better performance [32].

2.4. Parameter calculation and update

The features and the target label are sent to the GPU cluster using the ZeroMQ [33] asynchronous messaging queue. Each example server sends these data asynchronously to the CPU portion of the GPU node as shown in Fig 1. Using these data, neural network parameters are calculated and updated using an Adam optimizer and the Horovod [29] `allreduce` approach.

3. Structure of the end-to-end speech recognition system

We have adopted the RETURNN speech recognition system [34, 35] for training our end-to-end system with various modifications. Some of the important modifications are: replacing the input data pipeline with our proposed on-the-fly example server based pipeline with support for VTLP and acoustic simulation, implementing the Monotonic Chunkwise Attention (MoChA) [36] for online streaming E2E speech recognition, minimum Word Error Rate (mWER) training, support for handling Korean language or script, our own scoring and Inverse Text Normalization modules, support for power mel filterbank features, Maximally Uniform Distribution (MUD) preprocessing of data, etc.

The structure of our entire end-to-end speech recognition system with MUD processing [32] is shown in Fig. 2. $\boldsymbol{x}[m]$ and $\boldsymbol{y}[n]$ are the input power mel filterbank energy vector and the output label, respectively. m is the input frame index and n is the decoder output step index. $\boldsymbol{c}[n]$ is the attention context vector calculated as a weighted sum of the encoder hidden state vectors denoted as $\boldsymbol{h}_{enc}[m]$. The attention weights are computed as a softmax of energies computed as a function of the encoder hidden state $\boldsymbol{h}_{enc}[m]$, the decoder hidden state $\boldsymbol{h}_{dec}[n]$, and the attention weight feedback $\boldsymbol{\beta}[m,n]$ [35].

In [35], the peak value of the speech waveform is normalized to be one. However, since finding the peak sample value is not possible for online feature extraction, we do not perform this normalization. We modified the input pipeline so that the online feature generation can be performed. We disabled the clipping of feature range between -3 and 3, which is the default setting for the Librispeech experiment using MFCC features in [35]. We conducted experiments using both the uni-directional and bi-directional Long Short-Term Memories (LSTMs) [37] in the encoder. However, uni-directional LSTMs are used in the decoder. For online speech recognition experiments, we used the MoChA models [36] with a chunk size of 2. For better stability in LSTM training, we use the gradient clipping by global norm [38], which is implemented as `tf.clip_by_global_norm` API in Tensorflow [28]. We use six layers of encoders and one layer of decoder followed by a softmax layer.

4. Experimental Results

In this section, we present a summary of experimental results obtained for our E2E speech recognition systems built using the proposed Samsung Research end-to-end training framework.

Table 1: *Word Error Rates (WERs) obtained using MFCC implemented in [39] and power mel filterbank coefficients on the Librispeech corpus [1]. For each WER number, the same experiment was conducted twice and averaged.*

Cell Size		MFCC	Power Mel Filterbank Coefficients
1536 cell	test-clean	4.06 %	3.94 %
	test-other	13.97 %	13.56 %
	average	9.02 %	8.75 %

Table 2: *Word Error Rates (WERs) obtained with VTLP processing with different warping factor α distribution, and with and without an RNN LM. The warping factor α is the constant controlling warping in* (1).

Warping Factor		0.7 ~ 1.3	0.8 ~ 1.2	0.9 ~ 1.1
Without RNN-LM	test-clean	3.82 %	3.66 %	3.86 %
	test-other	12.50 %	12.39 %	12.35 %
	average	8.16 %	8.03 %	8.11 %
With RNN-LM	test-clean	2.93 %	2.85 %	2.96 %
	test-other	10.40 %	10.25 %	10.13 %
	average	6.67 %	6.55 %	6.55 %

For near-field speech recognition experiments, we use the open source Librispeech database [1], as well as our in house Bixby Usage training and test corpuses for English. The LibriSpeech dataset consists of around 960 hours of training data consisting of 281,241 utterances. The evaluation set consists of the official 5.4 hours `test-clean` and 5.1 hours `test-other` data. The Bixby Usage train corpus consists of approximately 10,000 hours of anonymized Bixby usage data. The evaluation set consists of around 1000 open domain utterances.

In Table 1, we compare the performance between the baseline MFCC and the power-law of $(\cdot)^{\frac{1}{15}}$ features for a bidirectional full attention (BFA) E2E model with an encoder LSTM cell size of 1536. Especially for `test-other`, which is a more difficult task, the power mel filterbank coefficients shows better performance than the baseline MFCC.

In Table 2, we show Word Error Rates (WERs) for a BFA model using different window sizes and warping coefficient distributions, with and without using an external Recurrent Neural Network (RNN) Language Model (LM) [35] built using the standard LibriSpeech LM corpus. The best performance was achieved when the window length is 50 *ms* and the warping coefficients are uniformly distributed between 0.8 and 1.2. We obtained 3.66 % WER on the *test-clean* database and 12.39 % WER on the *test-other* database without using an LM. To best of our knowledge, this is the best result on this database without using any language models. Using this shallow-fusion technique with an RNN-LM, we achieved WERs of 2.85 % and 10.25 % on the Librispeech `test-clean` and `test-other` databases, respectively. To the best of our knowledge, the 2.85 % WER on the `test-clean` is the best result ever reported on this database using any approach, and 10.25 % WER on the `test-other` is the best result on this database using an end-to-end recognition system.

In Table 3, we summarize our WER results for both the LibriSpeech and Bixby near-field E2E ASR models with and without using an external RNN-LM trained using around 65GB of Bixby LM corpus with an architecture exactly similar to the LibriSpeech LM model used in [35]. All the models mentioned in the table have 1024 LSTM cells in each encoder layer. For comparison, the best WFST based conventional LSTM-HMM based ASR system gives a WER of 8.85% on the Bixby same open domain test set. We can see that our current Bixby E2E BFA model is ∼10% better, while our MoChA streaming model is ∼10% poorer compared to the conventional WFST based DNN-HMM system.

Table 3: *Summary of Word Error Rates (WERs) obtained for different LibriSpeech and Bixby near-field E2E ASR models with and without an RNN LM.*

Models		BFA	MOCHA
LibriSpeech	w/o LM	3.66 %	6.78 %
	RNN-LM	2.85 %	5.54 %
Bixby	w/o LM	8.25 %	10.77 %
	RNN-LM	7.92 %	9.95 %

The performance our far-field E2E ASR model trained using the proposed data pipeline with example servers and acoustic simulator is shown in Fig. 3. The performance of the far-field models are evaluated on a English Commands test set with 900 utterances and recorded at 4 different distances: 0.5m, 1m, 3m and 5m.

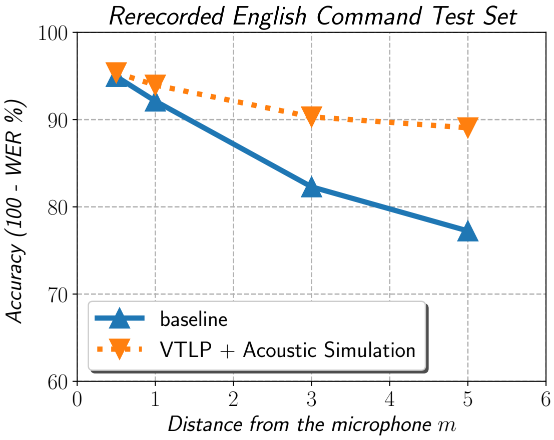

Figure 3: *Speech recognition accuracy with respect to the distance between the microphone and the speaker.*

5. Conclusions

We presented a new end-to-end training framework and strategies for training state-of-the-art end-to-end speech recognition systems. Our training system utilizes a cluster of Central Processing Units (CPUs) and Graphics Processing Units (GPUs). The entire data reading, large scale data augmentation, neural network parameter updates are performed on-the-fly using example servers and sharded `TFRecords` and `tf.data`. We use vocal tract length perturbation and an acoustic simulator for data augmentation. Horovod `allreduce` approach is employed to train the neural network parameters using Adam optimizer. We evaluated the effectiveness of our system on the standard Librispeech corpus [1] and 10,000-hr anonymized Bixby English training and test sets both in near-field as well as far-field scenarios.

6. References

[1] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An asr corpus based on public domain audio books,” in *IEEE Int. Conf. Acoust., Speech, and Signal Processing*, April 2015, pp. 5206–5210.

[2] M. Seltzer, D. Yu, and Y.-Q. Wang, “An investigation of deep neural networks for noise robust speech recognition,” in *Int. Conf. Acoust. Speech, and Signal Processing*, 2013, pp. 7398–7402.

[3] D. Yu, M. L. Seltzer, J. Li, J.-T. Huang, and F. Seide, “Feature learning in deep neural networks - studies on speech recognition tasks,” in *Proceedings of the International Conference on Learning Representations*, 2013.

[4] V. Vanhoucke, A. Senior, and M. Z. Mao, “Improving the speed of neural networks on CPUs,” in *Deep Learning and Unsupervised Feature Learning NIPS Workshop*, 2011.

[5] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. 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 Processing Magazine*, vol. 29, no. 6, Nov. 2012.

[6] T. Sainath, R. J. Weiss, K. W. Wilson, B. Li, A. Narayanan, E. Variani, M. Bacchiani, I. Shafran, A. Senior, K. Chin, A. Misra, and C. Kim, “Multichannel signal processing with deep neural networks for automatic speech recognition,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, Feb. 2017.

[7] ——, “Raw Multichannel Processing Using Deep Neural Networks,” in *New Era for Robust Speech Recognition: Exploiting Deep Learning*, S. Watanabe, M. Delcroix, F. Metze, and J. R. Hershey, Ed. Springer, Oct. 2017.

[8] C. Kim, A. Misra, K.K. Chin, T. Hughes, A. Narayanan, T. Sainath, and M. Bacchiani, “Generation of simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in Google Home,” in *INTERSPEECH-2017*, Aug. 2017, pp. 379–383.

[9] B. Li, T. Sainath, A. Narayanan, J. Caroselli, M. Bacchiani, A. Misra, I. Shafran, H. Sak, G. Pundak, K. Chin, K-C Sim, R. Weiss, K. Wilson, E. Variani, C. Kim, O. Siohan, M. Weintraub, E. McDermott, R. Rose, and M. Shannon, “Acoustic modeling for Google Home,” in *INTERSPEECH-2017*, Aug. 2017, pp. 399–403.

[10] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 4960–4964.

[11] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, “A comparison of sequence-to-sequence models for speech recognition,” in *Proc. Interspeech 2017*, 2017, pp. 939–943. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-233

[12] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-based models for speech recognition,” in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 577–585. [Online]. Available: http://papers.nips.cc/paper/5847-attention-based-models-for-speech-recognition.pdf

[13] H. Soltau, H. Liao, and H. Sak, “Neural speech recognizer: Acoustic-to-word lstm model for large vocabulary speech recognition,” 2017, pp. 3707–3711. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-1566

[14] E. Variani, T. Bagby, E. McDermott, and M. Bacchiani, “End-to-end training of acoustic models for large vocabulary continuous speech recognition with tensorflow,” in *INTERSPEECH-2017*, 2017, pp. 1641–1645. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-1284

[15] P. Goyal, P. Dollár, R. B. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, “Accurate, large minibatch SGD: training imagenet in 1 hour,” *CoRR*, vol. abs/1706.02677, 2017. [Online]. Available: http://arxiv.org/abs/1706.02677

[16] C. Kim and R. M. Stern, “Power-Normalized Cepstral Coefficients (PNCC) for Robust Speech Recognition,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, pp. 1315–1329, July 2016.

[17] U. H. Yapanel and J. H. L. Hansen, “A new perceptually motivated MVDR-based acoustic front-end (PMVDR) for robust automatic speech recognition,” *Speech Communication*, vol. 50, no. 2, pp. 142–152, Feb. 2008.

[18] T. Nakatani, N. Ito, T. Higuchi, S. Araki, and K. Kinoshita, “Integrating DNN-based and spatial clustering-based mask estimation for robust MVDR beamforming,” in *IEEE Int. Conf. Acoust., Speech, Signal Processing*, March 2017, pp. 286–290.

[19] T. Higuchi and N. Ito and T. Yoshioka and T. Nakatani, “Robust MVDR beamforming using time-frequency masks for online/offline ASR in noise,” in *IEEE Int. Conf. Acoust., Speech, Signal Processing*, March 2016, pp. 5210–5214.

[20] H. Erdogan, J. R. Hershey, S. Watanabe, M. Mandel, J. Roux, “Improved MVDR Beamforming Using Single-Channel Mask Prediction Networks,” in *INTERSPEECH-2016*, Sept 2016, pp. 1981–1985.

[21] C. Kim, K. Eom, J. Lee, and R. M. Stern, “Automatic selection of thresholds for signal separation algorithms based on interaural delay,” in *INTERSPEECH-2010*, Sept. 2010, pp. 729–732.

[22] C. Kim, C. Khawand, and R. M. Stern, “Two-microphone source separation algorithm based on statistical modeling of angle distributions,” in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, March 2012, pp. 4629–4632.

[23] R. Lippmann, E. Martin, and D. Paul, “Multi-style training for robust isolated-word speech recognition,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 12, Apr 1987, pp. 705–708.

[24] C. Kim, E. Variani, A. Narayanan, and M. Bacchiani, “Efficient implementation of the room simulator for training deep neural network acoustic models,” 2018, pp. 3028–3032. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-2566

[25] W. Hartmann, T. Ng, R. Hsiao, S. Tsakalidis, and R. Schwartz, “Two-stage data augmentation for low-resourced speech recognition,” in *Interspeech 2016*, 2016, pp. 2378–2382. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2016-1386

[26] N. Jaitly and G. E. Hinton, “Vocal tract length perturbation (vtlp) improves speech recognition,” in *Int. Conf. Mach. Learn. (ICML) Workshop Deep Learn. Audio, Speech, Lang. Process.*, 2013.

[27] C. Kim, M. Shin, A. Garg, and D. N. Gowda, “Improved vocal tract length perturbation for a state-of-the-art end-to-end speech recognition system,” Sept. 2019 (submitted).

[28] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, “Tensorflow: A system for large-scale machine learning,” in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. Savannah, GA: USENIX Association, 2016, pp. 265–283. [Online]. Available: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi

[29] A. Sergeev and M. D. Balso, “Horovod: fast and easy distributed deep learning in TensorFlow,” *arXiv preprint arXiv:1802.05799*, 2018.

[30] IBM, *IBM Spectrum LSF, Version 10 Release 1.0, Configuration Reference*.

[31] X. Cui, V. Goel, and B. Kingsbury, “Data augmentation for deep neural network acoustic modeling,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 9, pp. 1469–1477, Sept 2015.

[32] C. Kim, M. Kumar, K. Kim, and D. N. Gowda, “Power-law nonlinearity with maximally uniform distribution criterion for improved neural network training in automatic speech recognition,” Sept. 2019 (submitted).

[33] “Zero mq,” http:zeromq.org.

[34] 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 *IEEE Int. Conf. Acoust., Speech, and Signal Processing*, March 2017, pp. 5345–5349.

[35] A. Zeyer, K. Irie, R. Schlüter, and H. Ney, “Improved training of end-to-end attention models for speech recognition,” in *INTERSPEECH-2018*, 2018, pp. 7–11. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1616

[36] C.-C. Chiu and C. Raffel, “Monotonic chunkwise attention,” in *International Conference on Learning Representations*, Apr. 2018. [Online]. Available: https://openreview.net/forum?id=Hko85plCW

[37] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, no. 9, pp. 1735–1780, Nov. 1997.

[38] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ser. ICML’13. JMLR.org, 2013, pp. III–1310–III–1318. [Online]. Available: http://dl.acm.org/citation.cfm?id=3042817.3043083

[39] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, “librosa: Audio and music signal analysis in python,” in *Proceedings of the 14th Python in Science Conference*, K. Huff and J. Bergstra, Eds., 2015, pp. 18 – 25.