LESSONS FROM BUILDING ACOUSTIC MODELS WITH A MILLION HOURS OF SPEECH

Sree Hari Krishnan Parthasarathi, Nikko Strom

Amazon.com, USA.

{sparta,nikko}@amazon.com

ABSTRACT

This is a report of our lessons learned building acoustic models from 1 Million hours of unlabeled speech, while labeled speech is restricted to 7,000 hours. We employ student/teacher training on unlabeled data, helping scale out target generation in comparison to confidence model based methods, which require a decoder and a confidence model. To optimize storage and to parallelize target generation, we store high valued logits from the teacher model. Introducing the notion of scheduled learning, we interleave learning on unlabeled and labeled data. To scale distributed training across a large number of GPUs, we use BMUF with 64 GPUs, while performing sequence training only on labeled data with gradient threshold compression SGD using 16 GPUs. Our experiments show that extremely large amounts of data are indeed useful; with little hyper-parameter tuning, we obtain relative WER improvements in the 10 to 20% range, with higher gains in noisier conditions.

Index Terms: Speech recognition, acoustic models, large scale semi-supervised learning.

1. INTRODUCTION

A well-known maxim in the speech community is *there is no data like more data* [1]. Increasing the size of the training data by an order of magnitude have consistently led to substantial improvements in accuracy ([2], [3]). In this paper we push the envelope in building acoustic models (AM) on extremely large amounts of data. Specifically, we report our lessons in building acoustic models on 1 Million hours of unlabeled speech, while using only 7,000 hours of labeled speech data.

Taking a historic perspective; in the 1920's Radio Rex[1] used thresholds on formant energies to recognize the word "Rex". Since then, automatic speech recognition (ASR) systems have become ever more complex and used increasing amounts of speech data. Over the decades, corpora have grown from a few tens of hours of speech (TIDIGITS [5], TIMIT [6], WSJ [7]), to a few hundred hours (Switchboard [8]), to a few thousand hours of speech (Fisher corpus [9]). In symbiosis with this growth of data and more powerful computing hardware, a similar evolution in model complexity and algorithms can be traced, from the hard-wired analog signal processing of Radio Rex, via template based pattern matching and dynamic time-warping [10], to hidden Markov models [11], and the current prevalent deep neural networks [12].

Recently, with deep learning models, training data sizes on the order of ten thousand hours of speech are not unusual ([13], [14]). Building an AM from a hundred thousand hours is still rare, but [15] showed that increasing from several thousand hours of training data to a hundred thousand hours of lightly supervised data can yield substantial accuracy improvements. As a note, the large dataset used in our work is fully unlabeled.

Semi-supervised learning (SSL) has a long history in ASR ([16], [17], [18]). Self-training is the most commonly used approach where typically there is a smaller labeled dataset, and a much larger unlabeled dataset. The labeled data is used to train a seed model from a powerful model family, which is used to decode the unlabeled data at the second stage (often large beam sizes are used). The most reliable hypotheses are selected based on confidence measures [19] and the speech data with the selected hypotheses are used for re-training the AM.

Self-training requires good confidence measures, which has been a challenge for SSL ([19], [20]). Several methods to estimate word and frame level token confidence from speech lattices or hypotheses have been developed ([21], [22]). With models that have high memorization capability such as LSTM AMs, label quality becomes even more important [23]. Another challenge for the scale of data we consider in this paper, is an efficient inference mechanism to not only generate lattices/hypotheses, but also to estimate token confidence and use it for hypothesis selection. A further challenge is applying sequence discriminative training, where label errors have a larger detrimental effect ([24], [25]).

We built an SSL infrastructure that can train models on 1 Million hours of audio with a quick turnaround time. This paper reports our lessons in terms of the design choices made while building models at this scale. We based our training on the student/teacher paradigm. Recently, student/teacher training has become popular in the speech community for model compression ([26], [27], [28]). Here, instead student/teacher training is applied to produce soft targets for unlabeled data, which leads to efficient target generation ([29], [30]). Further, we introduce a particular learning schedule – interleaving training on labeled data with training on unlabeled data. Sequence training is also used, but only using labeled data. Finally, in our large-scale experiments we contrast two types of distributed training.

The remainder of the paper is organized as follows: starting with a description of the baseline fully supervised AM system in Section 2, we discuss the semi-supervision design choices in Section 3. Next, we cover the experimental setup in Section 4, and validation results exploring the design choices in Section 5. The final 1 Million hour results with analyses are described in Section 6, followed by our conclusions in Section 7.

2. BASELINE SUPERVISED ACOUSTIC MODEL

We use an HMM-LSTM hybrid. The HMM models low-frame rate single state triphone units [31]. States are clustered down to 3,183 senones using phonetic decision trees. The acoustic features consist of 64-dimensional log mel-warped energies computed on audio signals every 10 ms with a 25 ms analysis window ([32], [33]). These are stacked three at a time and sub-sampled to a 30 ms advance. A

[1] Arguably the first speech recognition system [4].

causal mean estimate is computed and subtracted, and finally global mean and variance normalization is applied. To compensate for subsampling, features are created at three different offsets for each utterance.

The LSTM model is a stack of five layers, each consisting of 768 units resulting in about 24 M parameters. The model has a three-frame look-ahead. The training data is 7,000 hours of labeled US English data drawn from the Echo family of devices. The models are trained first with the cross-entropy criterion (CE), using alignments computed on the labeled data. First, we follow an exponential learning rate decay for ten epochs, with chunked BPTT for greater parallelization efficiency [34]. In this technique, utterances are split into smaller sub-sequence chunks (here, 32 frames) and the sub-sequences are randomized. For each epoch we cycle through a different feature offset. Then the models are fine-tuned using full sequence CE BPTT for two more epochs. Finally, three epochs of the sequence discriminative criterion state-level minimum Bayes risk (sMBR) is applied.

We employ distributed training using synchronous SGD on two p3.16xlarge instances (16 Tesla V100 GPU cards). Gradient Threshold Compression [35] is used for efficient peer-to-peer weight updates after every minibatch.

3. LARGE-SCALE SEMI-SUPERVISED LEARNING

At the scale of 1 Million hours, certain design choices were crucial for experiment turnaround time, while also obtaining significant accuracy improvements. This section presents various design choices and their considerations.

3.1. Data Selection and Feature Extraction

We drew data according to a device distribution roughly similar to that of the labeled data. Within each device, we drew samples randomly from the production data firehose. We did not filter data with confidence models nor for background speech/noise. Our hypothesis was that well-calibrated posteriors from the teacher model would mitigate poorly selected data.

To speed up parallel feature generation we did not require a pre-roll of utterances for initialization as described in [36]. We developed a feature pipeline that uses an efficient hashing mechanism to cluster speakers and sort utterances belonging to a speaker for performing running cepstral mean normalization. This could then be parallelized over several thousand CPU cores.

3.2. Student-Teacher Learning

A key design choice was to employ the student/teacher learning paradigm, thus taking the ASR decoder out of the SSL recipe. In essence, for each feature vector, the teacher network outputs a probability distribution over senones. The student network also estimates the probabilities over the senones given the same feature vector, and the learning objective optimizes the CE loss between these two distributions. The student models are identical to the LSTMs described in the previous section, but the teacher models have five bi-directional LSTM layers, each of size 768 units (amounting to a total of 78 M model parameters). The training of the teacher on the labeled data follows the same recipe as the regular LSTMs, discussed in Section 2.

3.2.1. Confidence Modeling

There is evidence that even unfiltered data can lead to significant SSL improvements ([17], [37]). Further, as neural networks have improved, the estimated probabilities become better calibrated [38]. Our hypothesis was that the teacher's posteriors are calibrated well enough to act as the confidence measure for the student training. However, in a traditional self-learning system, the language model is also providing additional information during the decoding, which is not present in our system. We hypothesize that this is partially mitigated by the bi-directional LSTM model, which has more context than the student.

3.2.2. Target Generation

The senone output distribution is large, and generating targets from the teacher model on-the-fly can slow down training. To reduce bandwidth and storage requirements as we parallelize across multiple GPUs, we store only the k highest valued logits. During the student model training, full posteriors are reconstructed by filling the missing logits with large negative values. While this reconstruction is lossy, we found empirically that the probability mass is dominated by the top few posteriors. We found storing the top-20 values for k to be sufficient from the standpoint of not having a WER degradation, while yielding a huge gain in storage.

3.3. Scheduled Learning

While we primarily train on unlabeled data, the limited labeled data is also used. Learning on unlabeled and labeled data is interleaved, with slightly higher learning rates on the labeled data.

We used two unlabeled training datasets (100khrs and 1Mhrs), as will be discussed in Section 4. Given the large amounts of data, our design was to perform just one learning pass through the data. We divided the data into a number of *sub-epochs*, with a sub-epoch defined as 25,000 and 55,000 hours for the 100khr and 1Mhr datasets respectively. We decayed the learning rate as we passed through the sub-epochs, following an exponential learning rate decay.

For the 100khrs, after each sub-epoch through the unlabeled data, we perform CE training on the labeled data, with a rotation through the feature offsets (refer to Section 2). For the 1Mhr data, after every five sub-epochs through the unlabeled data, we perform CE training on the labeled data, rotating through the feature offsets.

As discussed in Section 2 we employ sequence chunked BPTT for training speed. On the 100khrs set, chunked training is used for the first three sub-epochs (including the corresponding passes through the labeled data), followed by a full sequence BPTT on the last sub-epoch on the unlabeled data. On the 1Mhrs data, we apply chunked training for the first 15 sub-epochs, and then do fine-tuning during the last three sub-epochs.

3.4. Sequence Training for SSL

Sequence discriminative training often yields large accuracy gains (commonly, around 10% relative). However, it is also a difficult problem for SSL ([39], [24]), since it is particularly sensitive to noisy references during training. We chose to perform sequence training only on labeled data. There was evidence [40] that the accuracy gains may be relatively small in such a setup. However, our hypothesis was that this result was due to a smaller labeled dataset, and using our full 7,000 hour labeled data would still recover large gains from sequence training.

3.5. Distributed Training

For the scale of data we want to learn from, our design goal was to parallelize beyond a few tens of GPUs. We explored the Gradient Threshold Compression method (GTC) [35] and Blockwise Model-Update Filtering (BMUF) [41].

With high-end GPUs like Tesla V100s, gradient compression based training scales well up to 16 GPU cards, but efficiency tapers off at higher scale. In this work, we used two p3.16xlarge instances (16 Tesla V100 GPU cards spread over two hosts).

The BMUF training scales nearly linearly with GPUs, at least in terms of throughput, because the per-worker model updates happen much more infrequently. However, it can come at a cost in accuracy. The Nesterov-like momentum updates at block level recover some of these losses [42], but we still see some degradation (Table 2). For our experiments with BMUF we used eight p3.16xlarge instances (64 Tesla V100 GPUs).

4. EXPERIMENTAL SETUP

We discussed a number of system level details in Section 2. In this section we give the details with regard to our experimental setup.

4.1. Training Datasets

For our experiments we used three far-field training datasets drawn from production data of the Alexa family of devices from the US English locale: (a) a 7,000 hour fully labeled dataset (b) 100,000 hours of unlabeled data for prototyping and validating design choices, and (c) a 1 Million hour unlabeled dataset for the final model build.

4.2. Test Datasets

We used several test sets in this work: (a) a validation test set (referred to as VAL), which consisted of about 30 hours of data, (b) acoustically difficult audio data collected in a real room with about 5,000 utterances roughly equally spread among five device placements. The first device placement (DP1) in the center of the room was the easiest, while other conditions (DP2 to DP5) were more challenging, and (c) a 30 hours independent test set (referred to as TEST). The TEST set was also divided into native (TST-NATIVE) and non-native (TST-NON-NAT) speakers as judged by the annotators.

4.3. Decoding Setup and Scoring

All decoding on the VAL test set use a 4-gram statistical language model (LM). The acoustic model scale factor was tuned on this test set. For the decoding runs on all other test sets, the statistical LM was combined with a set of domain-specific grammars. We report results as relative Word Error Rate Reduction (WERR) compared the strong baseline supervised learning system.

5. EXPERIMENTS ON 100,000 HOURS

In this section, we validate the key design choices by training models on the 100,000 hour unlabeled data and decoding on the VAL test set. The key elements are: (a) scheduled learning and its interaction with sMBR trained teacher, (b) sequence training of the student model, and (c) choice of distributed training method.

5.1. Scheduled Learning

We perform our analysis with and without scheduled learning; we also consider its interaction with and without sMBR trained teacher. Table 1 presents accuracy for the four different options relative to a baseline LSTM AM that is trained with the CE criterion on the fully labeled 7,000 training data.

It can be seen from the table that scheduled learning, i.e., interleaving labeled data in the learning, helps the student models both in the case of CE trained as well as sequence trained teachers. However, the gain with scheduled learning is more with students trained with the CE-teacher.

Table 1: *On VAL test set, relative WER (%) reduction for SSL student models trained on 100,000 hour dataset: with and without scheduled learning (SL); with and without sMBR trained teachers. The WER reduction is computed against a baseline LSTM AM that is trained with CE criterion on the fully labeled 7,000 hour training data.*

	without sMBR teacher	with sMBR teacher
without SL	1.0	8.8
with SL	6.8	**10.8**

5.2. Scaling Number of GPUs to 64

Student models used in Table 1 were trained using the GTC trainer with 16 GPUs. Using the best model configuration from Table 1, i.e., with scheduled learning and with sMBR trained teachers, we now investigate the effect of BMUF trainer on student models. For student training with BMUF trainer, we use 64 GPUs. Note that the objective here is not to compare the BMUF and the GTC trainers (which would involve an extensive search over hyper-parameters of both trainers), but to obtain an estimate of the WER gain or loss in scaling up the number of GPUs in training (for which we use BMUF). With 64 GPUs, we obtain a relative WER reduction of 7.8% over the baseline LSTM AM that is trained with the CE criterion on the fully labeled 7,000 training data (compared to 10.8% in Table 1). Thus, in attempting to scale to 64 GPUs, we lose some of the gains due to SSL.

Table 2: *On VAL test set, relative WER (%) reduction for sequence training of SSL students. The WER reduction is computed against a baseline LSTM AM that is trained with CE criterion on the fully labeled 7,000 hour training data. sMBR is performed with GTC trainer.*

System	CE Trainer	WERR (%)
Baseline labeled CE	GTC	0
Baseline labeled CE + sMBR	GTC	10.7
SSL + SL + sMBR	GTC	18.6
SSL + SL + sMBR	BMUF	15.6

5.3. Sequence Training

Recall that our strategy is to perform sequence training of student models only on the 7,000 hour labeled dataset. We compare if the gains we obtained at the CE stage also carry over to the sMBR stage. Since the labeled dataset is much smaller, we use the GTC trainer with 16 GPUs for all models. From Table 2, compared to a fully supervised CE trained AM, sMBR training yields a 10.7% relative improvement in WER. In comparison to this sMBR model, we now compare two SSL student models on which sMBR training is performed, i.e., GTC and BMUF trained SSL student models. Sequence training only on labeled data still gives a good gain for SSL students (WER reductions of 18.6% and 15.6%, respectively over baseline labeled CE model), translating still into relative WER reductions of 8.2% and 5.4%, respectively over the fully supervised sMBR model. It is interesting to note that the effect of training with BMUF using 64 GPUs still does not fully recover after sMBR training with the GTC trainer using 16 GPUs, but we select this option for speed.

6. RESULTS ON 1 MILLION HOURS

In this section we present our final 1 Million hour model. We compare this model against the fully supervised sMBR model on (a) the acoustically difficult test sets with five device positions DP1 to DP5, and (b) TEST test set, along with sub-dividing it in two dimensions: nativity (TST-NATIVE, TST-NON-NAT) and SNR levels. We present these results in Tables 3, 4.

6.1. Training Convergence

For the final 1 Million hour semi-supervised training we are using BMUF with 64 GPUs, using sMBR trained teacher, and employing scheduled learning. Figure 1 plots convergence as WER reduction on the VAL set, relative to an LSTM AM that is trained with CE criterion on the fully labeled 7,000 hour training data with GTC trainer. The x-axis represents sub-epochs (each sub-epoch is about 55,000 hours of data), adding up all the way up to 1 Million hours. It can be seen that the student model keeps improving up to the 14^{th} sub-epoch (i.e. up to 770,000 hours of data). Sub-epochs 16 to 18 are fine-tuning epochs, and the gains are larger. Note that the decrease in WER is not monotonic (sometimes there is even a slight increase), and we have not extensively tuned the learning hyper-parameters. From Figure 1, the WER reduction after training on the full 1 Million hours, at the CE stage, is 13.7% – significantly better than the corresponding 100,000 hour result (10.8%) in Table 1.

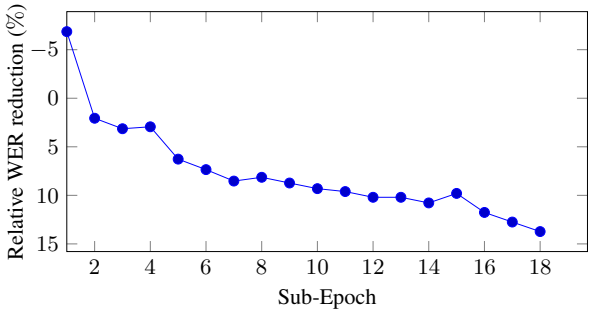

Fig. 1: *On VAL test set, relative WER reduction per sub-epoch of the 1 Million hour SSL model against a baseline LSTM AM that is trained with CE criterion on the fully labeled 7,000 hour training data. Each sub-epoch corresponds to about 55,000 hours of data.*

6.2. Final Results

The final results including the final sMBR training can be seen in Table 3 and Table 4. Except for the easiest device position (DP1), and the easiest noise condition (SNR>25dB), relative WER reductions are all greater than 10%, and consistently the improvement is greater for harder conditions. Note also that the improvement is greater for non-native speakers. We take this as validation that large scale SSL can not only significantly improve accuracy overall (11.6% error reduction), but also yield an out-sized improvement for the most challenging conditions.

Table 3: *On the acoustically difficult test set (in DP1 to DP5), relative WER reduction (%) of the final 1 Million hour model against a baseline LSTM AM that is sMBR trained on the fully labeled 7,000 hour training data.*

Test Conditions	WERR (%)
DP1	9.8
DP2	22.2
DP3	21.8
DP4	16.5
DP5	18.9

Table 4: *On TEST test set, relative WER reduction (%) of the final 1 Million hour model against a baseline LSTM AM that is sMBR trained on the fully labeled 7,000 hour training data.*

Test Conditions	WERR (%)
TEST	11.6
TST-NATIVE	11.6
TST-NON-NAT	13.0
TEST, SNR: <5 dB	13.3
TEST, SNR: 5-10 dB	14.5
TEST, SNR: 10-15 dB	10.7
TEST, SNR: 15-20 dB	11.2
TEST, SNR: 20-25 dB	12.9
TEST, SNR: >25 dB	6.7

7. CONCLUSIONS

This paper reported on our lessons learned in building acoustic models on 1 Million hours of unlabeled speech data, in conjunction with 7,000 hours of labeled data. Using student-teacher learning, we simplified target generation without the need for decoding and confidence modeling. To optimize storage and to parallelize the target generation, we stored high valued logits from the teacher model. We introduced the notion of scheduled learning, interleaving learning on unlabeled and labeled data. This approach gave gains with CE and sMBR trained teacher models, but yielded bigger WER gains for CE trained teacher models. To scale distributed training to 64 GPUs we used BMUF, while performing sequence training only on the labeled data using GTC training with 16 GPUs. Our experiments showed that extremely large amounts of data are indeed useful; with little hyper-parameter tuning, we obtained relative WER improvements in the 10 to 20% range, with much higher gains in more difficult conditions, acoustically or in terms of speakers.

Acknowledgements

We would like to thank Nitin Sivakrishnan for help with speeding up the data pipeline, Pranav Ladkat for implementing BMUF, Gautham Kollu for implementing an optimized forward propagation, Xing Fan for providing the setup for baseline models, and Harish Mallidi for help with the decoding infrastructure.

8. REFERENCES

[1] F. Jelinek, "Some of my best friends are linguists," in *LREC*, 2004.

[2] R. K. Moore, "A comparison of the data requirements of automatic speech recognition systems and human listeners," in *Proc. of Eighth European Conference on Speech Communication and Technology*, 2003.

[3] C. Chelba, P. Xu, F. Pereira, and T. Richardson, "Large scale distributed acoustic modeling with back-off n-grams," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 6, pp. 1158–1169, 2013.

[4] J. H. Martin and D. Jurafsky, *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson/Prentice Hall, 2009.

[5] R. G. Leonard and G. Doddington, "TIDIGITS speech corpus," *Texas Instruments, Inc*, 1993.

[6] J. S. Garofolo, L. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continuous speech corpus," *NASA STI/Recon technical report*, vol. 93, 1993.

[7] D. B. Paul and J. M. Baker, "The design for the Wall Street Journal-based CSR corpus," in *Proc. of Workshop on Speech and Natural Language*, 1992.

[8] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proc. of ICASSP*.

[9] C. Cieri, D. Miller, and K. Walker, "The Fisher Corpus: a resource for the next generations of speech-to-text." in *LREC*, vol. 4, 2004, pp. 69–71.

[10] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE transactions on acoustics, speech, and signal processing*, vol. 26, no. 1, pp. 43–49, 1978.

[11] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. of the IEEE*, 1989.

[12] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[13] D. Amodei, S. Ananthanarayanan *et al.*, "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *Proc. of ICML*, 2016.

[14] Y. Huang, Y. Wang, and Y. Gong, "Semi-supervised training in deep learning acoustic model." in *Proc. of Interspeech*, 2016.

[15] H. Soltau, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition," *arXiv preprint arXiv:1610.09975*, 2016.

[16] T. Kemp and A. Waibel, "Unsupervised training of a speech recognizer: recent experiments." in *Proc. of Eurospeech*, 1999.

[17] L. Lamel, J.-L. Gauvain, and G. Adda, "Lightly supervised and unsupervised acoustic model training," *Computer Speech & Language*, vol. 16, no. 1, pp. 115–129, 2002.

[18] J. Ma, S. Matsoukas, O. Kimball, and R. Schwartz, "Unsupervised training on large amounts of broadcast news data," in *Proc. of ICASSP*, 2006.

[19] M.-h. Siu, H. Gish, and F. Richardson, "Improved estimation, evaluation and applications of confidence measures for speech recognition," in *Proc. of Fifth European Conference on Speech Communication and Technology*, 1997.

[20] Y. Huang, D. Yu, Y. Gong, and C. Liu, "Semi-supervised GMM and DNN acoustic model training with multi-system combination and confidence re-calibration." in *Proc. of Interspeech*, 2013.

[21] S. Thomas, M. L. Seltzer, K. Church, and H. Hermansky, "Deep neural network features and semi-supervised training for low resource speech recognition," in *Proc. of ICASSP*, 2013.

[22] H. Liao, E. McDermott, and A. Senior, "Large scale deep neural network acoustic modeling with semi-supervised training data for YouTube video transcription," in *Proc. of ASRU*, 2013.

[23] Y. Huang, Y. Wang, and Y. Gong, "Semi-supervised training in deep learning acoustic model." in *Proc. of Interspeech*, 2016.

[24] V. Manohar, D. Povey, and S. Khudanpur, "Semi-supervised maximum mutual information training of deep neural network acoustic models." in *Proc. of Interspeech*, 2015.

[25] J.-T. Huang and M. Hasegawa-Johnson, "Maximum mutual information estimation with unlabeled data for phonetic classification." in *Proc. of Interspeech*, 2008.

[26] L. Lu, M. Guo, and S. Renals, "Knowledge distillation for small-footprint highway networks," in *Proc. of ICASSP*, 2017.

[27] G. Tucker, M. Wu, M. Sun, S. Panchapagesan, G. Fu, and S. Vitaladevuni, "Model compression applied to small-footprint keyword spotting." in *Proc. of Interspeech*, 2016.

[28] J. Li, M. L. Seltzer, X. Wang, R. Zhao, and Y. Gong, "Large-scale domain adaptation via teacher-student learning," in *Proc. of Interspeech*, 2017.

[29] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Proc. of Advances in NIPS*, 2014.

[30] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[31] G. Pundak and T. N. Sainath, "Lower frame rate neural network acoustic models." in *Proc. of Interspeech*, 2016.

[32] S. H. K. 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.

[33] S. Garimella, A. Mandal, N. Strom, B. Hoffmeister, S. Matsoukas, and S. H. K. Parthasarathi, "Robust i-vector based adaptation of DNN acoustic model for speech recognition," in *Proc. of Interspeech*, 2015.

[34] P. Doetsch, M. Kozielski, and H. Ney, "Fast and robust training of recurrent neural networks for offline handwriting recognition," in *Proc. of ICFHR*, 2014.

[35] N. Strom, "Scalable distributed DNN training using commodity GPU cloud computing," in *Proc. of Interspeech*, 2015.

[36] B. King, I.-F. Chen, Y. Vaizman, Y. Liu, R. Maas, S. H. K. Parthasarathi, and B. Hoffmeister, "Robust speech recognition via anchor word representations," in *Proc. of Interspeech*, 2017.

[37] L. Lamel, J.-L. Gauvain, and G. Adda, "Unsupervised acoustic model training," in *Proc. of ICASSP*, 2002.

[38] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," *arXiv preprint arXiv:1706.04599*, 2017.

[39] J.-T. Huang and M. Hasegawa-Johnson, "Semi-supervised training of gaussian mixture models by conditional entropy minimization," *Optimization*, vol. 4, p. 5, 2010.

[40] N. Kanda, S. Harada, X. Lu, and H. Kawai, "Investigation of semi-supervised acoustic model training based on the committee of heterogeneous neural networks." in *Proc. of Interspeech*, 2016.

[41] K. Chen and Q. Huo, "Scalable training of deep learning machines by incremental block training with intra-block parallel optimization and blockwise model-update filtering," in *Proc. of ICASSP*, 2016.

[42] W. Li, B. Zhang, L. Xie, and D. Yu, "Empirical evaluation of parallel training algorithms on acoustic modeling," *arXiv preprint arXiv:1703.05880*, 2017.