Multi-task learning and Weighted Cross-entropy for DNN-based Keyword Spotting

Sankaran Panchapagesan[†]*, Ming Sun, Aparna Khare, Spyros Matsoukas*
Arindam Mandal, Björn Hoffmeister, Shiv Vitaladevuni

Amazon.com
[†]panchi@amazon.com

Abstract

We propose improved Deep Neural Network (DNN) training loss functions for more accurate single keyword spotting on resource-constrained embedded devices. The loss function modifications consist of a combination of multi-task training and weighted cross entropy. In the multi-task architecture, the keyword DNN acoustic model is trained with two tasks in parallel - the main task of predicting the keyword-specific phone states, and an auxiliary task of predicting LVCSR senones. We show that multi-task learning leads to comparable accuracy over a previously proposed transfer learning approach where the keyword DNN training is initialized by an LVCSR DNN of the same input and hidden layer sizes. The combination of LVCSR-initialization and Multi-task training gives improved keyword detection accuracy compared to either technique alone. We also propose modifying the loss function to give a higher weight on input frames corresponding to keyword phone targets, with a motivation to balance the keyword and background training data. We show that weighted cross-entropy results in additional accuracy improvements. Finally, we show that the combination of 3 techniques - LVCSR-initialization, multi-task training and weighted cross-entropy gives the best results, with significantly lower False Alarm Rate than the LVCSR-initialization technique alone, across a wide range of Miss Rates.

Index Terms: keyword spotting, DNN, Deep Neural Network, multi-task learning, weighted cross entropy

1. Introduction

Deep Neural Network (DNN) acoustic models have quickly become the state of the art in speech recognition in recent years [1]. They have generally been found to be more robust to speaker and environmental variations than the earlier widely used Gaussian Mixture Model (GMM) based acoustic models [2]. DNNs are able to make effective use of parameters to learn powerful hidden layer representations, so that with increasing amounts of training data and larger networks, the test accuracy on the given task can generally be continuously improved. In a practical speech recognition system there are limits on the size of the network that can be deployed due to computational cost and desired latency considerations. It is possible to deploy larger networks using faster implementations that exploit quantization and SIMD instructions on CPUs [3]. However, when dealing with practical constraints on size of the network or the amount of data available for a task, it is necessary to make use of improved training techniques to obtain robust models that have better generalization performance. Some improved training techniques include transfer learning, multi-task learning, and knowledge distillation [4, 5, 7].

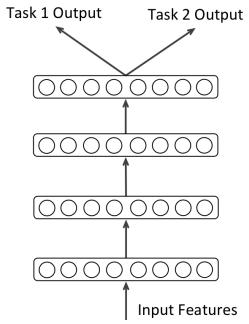

Figure 1: *Multi-task architecture*

Transfer learning is a general approach in machine learning for improving the performance on a main task by knowledge transfer from a related task that was previously learned. In the context of DNNs, the input and hidden layers of the network for the main task are initialized from a network of the same input and hidden layer sizes that has been trained for the related task [5, 6]. A better initialization of the training of the main task results in a better optimum and improves generalization performance.

In multi-task learning (MTL), two or more related tasks are jointly learned with the goal of improving generalization and performance on one or more of the tasks [4]. For DNNs, MTL is achieved by sharing hidden layers between the tasks, and having a separate output layer branch for each task, as shown in figure 1 for two tasks. MTL prevents overfitting by learning representations in the shared hidden layers of the network that generalize better as they are useful in performing more than one task. At evaluation time, the additional layers for the auxiliary tasks that are not shared with the main task can be stripped away as they are no longer needed. Therefore, while MTL leads to increased training time due to the additional parameters needed for the auxiliary tasks, there is no increase in computation at test time. MTL and transfer learning are finding numerous applications within speech research, including general ASR [10, 9], multilingual ASR [5, 8], keyword spotting [6], robust ASR [11, 13] and speech separation [12].

In this paper, we propose a novel application of MTL to improve the accuracy of keyword spotting from speech using DNNs. We compare it against an earlier proposed transfer learning method [6] and show that it gives comparable performance. Both the transfer learning approach and the multi-task approach give significant accuracy improvements over a standard training recipe that includes only initialization by layer-wise pre-training. We also identify the need to balance the key-

word and background speech data in the training set, and propose to weight the cross-entropy loss higher for feature frames from keywords. We show that the combination of transfer learning, MTL and weighted cross entropy gives the best results.

The paper is organized as follows. In Section 2 we give a brief review of the literature on the keyword spotting problem. In Section 3 we describe the proposed modifications to the loss function, including the multi-task, weighted cross-entropy and their combination. In Section 4 detailed experimental results with the proposed loss functions are presented. We conclude with a brief summary in Section 5.

2. Keyword Spotting

Keyword (KW) spotting in continuous speech has been an area of research for more than two decades [14]-[23]. In much of recent work, latency and computation are not concerns, and offline large vocabulary speech recognition systems can be used to decode the audio utterances and create transcripts or lattices of words and/or phones which can then be searched for the presence of the keyword(s) of interest [14]-[16]. An earlier approach, which still finds application in online low-latency and computation-constrained systems, uses Hidden Markov Models (HMMs) for both the keyword and the background non-keyword speech or noise audio [17, 18, 19]. The background model is also sometimes called the filler or garbage model, and may be a simple speech/non-speech loop HMM [18], or may involve a loop over phones or words [21]. With the growing success of deep learning in recent years, novel techniques using Deep Neural Networks (DNNs) and Recurrent Neural Networks (RNNs) that do not involve HMMs have also been proposed [22, 6, 23].

In this paper, we are concerned with single KW spotting systems on resource-constrained embedded devices.

In this context, we study KW spotting using the well known HMM based approach with KW and filler/background HMMs [17, 18, 19]. An example KW spotting decoding FST for the KW "*Alexa*" is shown in Figure 2, with six phones in its pronunciation. Note that single state HMMs for the phones are shown for simplicity.

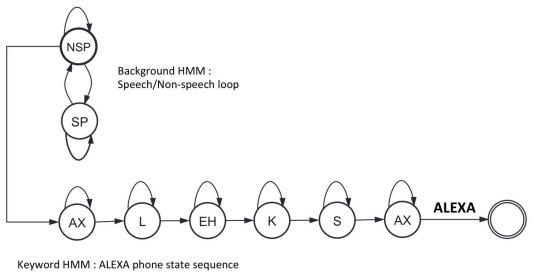

Figure 2: *HMM-based Keyword Spotting*

The HMM decoder uses a Deep Neural Network (DNN) acoustic model and a KW is hypothesized if the final state of the KW FST is reached. The output layer of the KW DNN models the HMM states of the keyword(s) of interest (i.e., KW-specific phone state distributions), and the two 1-state background phones - Speech and Non-speech. Various transition parameters and exit penalties in the KW and Background HMMs can be tuned for better accuracy, and a Detection Error Tradeoff (DET) curve can be obtained by plotting the lowest achievable False Alarm Rate (FAR) at a given Miss Rate (MR)/False Reject Rate (FRR). The DET curves in Figures 3-7 were obtained in this manner.

In our system, audio is divided into overlapping frames of 25 ms with a frame shift of 10 ms. The basic acoustic features per frame are the well known Log mel-Filter-Bank Energies (LFBEs) which are obtained by passing the magnitude squared of the FFT of the windowed signal through the Mel filterbank and taking the logarithm [24]. The input to the DNN acoustic model typically consists of several stacked frames of LFBEs.

3. Loss functions for DNN Training

The basic loss function that is optimized during the training of DNN acoustic models is the cross-entropy loss [25]. We consider loss functions for a single frame n for simplicity of notation. The cross-entropy loss for feature vector $\mathbf{x}_n$ is given by:

$$\mathcal{L}_n(\mathbf{W}) = -\log \mathbf{y}_{c_n}(\mathbf{x}_n, \mathbf{W}) \tag{1}$$

where $\mathbf{W}$ are the parameters of the DNN, $\mathbf{y}_k(\mathbf{x}_n, \mathbf{W})$ is the kth output of the final softmax layer of the DNN, and c_n is the class label assumed to be in the range $\{1, 2, \ldots, K\}$.

3.1. Multi-task Training

In multi-task training with two tasks, let $\mathcal{L}_n^{(1)}(\mathbf{W})$ and $\mathcal{L}_n^{(2)}(\mathbf{W})$ be loss functions for the two tasks, defined similar to Eq. 1. We use a multi-task loss function of the form:

$$\mathcal{L}_n(\mathbf{W}) = \gamma\, \mathcal{L}_n^{(1)}(\mathbf{W}) + (1-\gamma)\, \mathcal{L}_n^{(2)}(\mathbf{W}) \tag{2}$$

where $0 \leq \gamma \leq 1$.

3.2. Class-weighted cross-entropy

In the case of KW spotting, the amount of data available for the KW is usually much less than the amount of data available for the background speech and non-speech. It is also not clear a-priori which of the background data would be useful, and therefore it is preferable to use all the available data for training and not filter out background data. However, since the background data dominates the training set, it may be desirable to weight the KW data higher in the training. One simple way of achieving this is by weighting the loss function for a frame higher if the label for the frame belongs to one of the KW phone states. More generally, we define a weight vector $\mathbf{w} \in \mathbb{R}^k$ with elements $w_k > 0$ defined over the range of class labels $k \in \{1, 2, \ldots, K\}$. We then define the *class-weighted cross-entropy* (CW-XENT) as follows:

$$\mathcal{L}_n(\mathbf{W}) = -w_{c_n} \log \mathbf{y}_{c_n}(\mathbf{x}_n, \mathbf{W}) \tag{3}$$

3.3. Combined loss function

We can also consider loss function that is a combination of the multi-task and class-weighted cross-entropy loss functions as follows:

$$\mathcal{L}_n(\mathbf{W}) = -\gamma w_{c_n} \log \mathbf{y}_{c_n}^{(1)}(\mathbf{x}_n, \mathbf{W}) - (1-\gamma) \log \mathbf{y}_{l_n}^{(2)}(\mathbf{x}_n, \mathbf{W}) \tag{4}$$

Here, we have included class-specific weights only for the main task as that is what we have studied in this paper.

4. Experimental Results

Our KW spotting system is tuned and evaluated on Dev and Test sets that contain data collected under different conditions.

Some audio streams in the Dev and Test sets contain the KW and others contain speech utterances without the KW, or just background speech and noise. The evaluation sets contain several thousand occurrences of the keyword(s) of interest, so that the results are statistically significant.

The GPU-based distributed DNN trainer described in [26] was utilized for many of the experiments reported here. In the results, "Random-initialization" of DNNs refers to light supervised pre-training in a layer-wise manner on a small subset of training data.

We present results in the form of DET curves along with Area Under the Curve (AUC) numbers, which will allow comparison of various systems in terms of performance impact. Note that since we present AUC numbers for DET curves instead of ROC curves, lower AUC numbers correspond to better performance.

All DET curves in this paper only show false alarm rates up to a multiplicative constant, due to the sensitive nature of this information. The plots and the AUC values still accurately preserve the relative performance improvements between different systems across a range of reasonable operating points.

4.1. Baseline results with LVCSR-initialization

The baseline we use is the transfer learning approach proposed in [6], where the KW DNN is initialized by an LVCSR DNN of the same input and hidden layer sizes. An output layer of the appropriate size is lightly pre-trained in a supervised manner, before fine-tuning of the full network with back-propagation. While the DNN architecture and KW detection algorithm in [6] are different, we found the transfer learning approach to be effective, as shown in Figure 3. Over a range of FA Rates of interest, we find that LVCSR-initialization gives around 15-20%

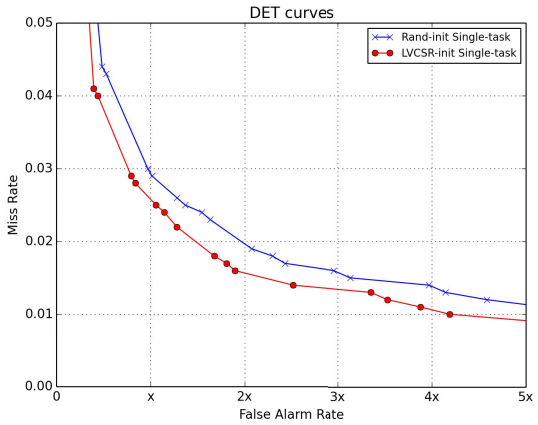

Figure 3: *Effect of transfer learning by LVCSR-initialization of KW DNN: DET curves on the held-out Test Set.*

4.2. Results with multi-task architecture

We studied multi-task learning (MTL) with KW and LVCSR targets, with two different topologies for the networks. The first multi-task topology had 4 shared hidden layers between the two tasks and only separate output layers as was shown in Figure 1. The second multi-task topology had 3 shared hidden layers between the two task and one separate hidden layer for each of the tasks before their respective output layers. The output layers are affine transforms followed by a softmax as is typical.

For each architecture, the weight of the KW task (γ in Equation 2) was varied and the DET curve results were plotted on the Dev set. Recall from Equation 2 that the weight on

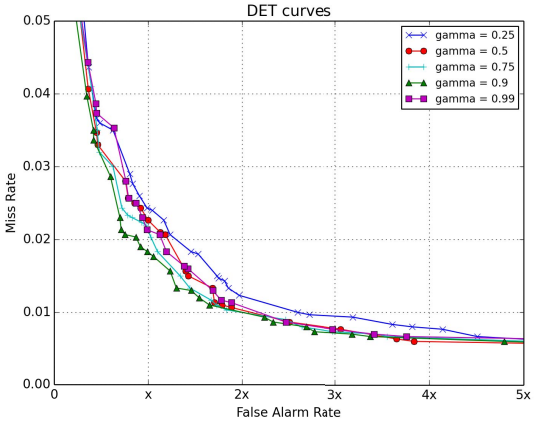

Figure 4: *Results of varying weight γ on the KW task in multi-task training; 4 shared hidden layers; DET curves on Dev Set.*

the auxiliary LVCSR task is $(1-\gamma)$. Figure 4 shows the results for the multi-task topology with 4 shared layers, where $\gamma = 0.9$ was found to be optimal. For the multi-task topology with 3 shared layers, $\gamma = 0.75$ found to be optimal for smaller training data sets, while $\gamma = 0.9$ was found to be optimal on larger training data sets. In general, we also observed that 3 shared hidden layers gives slightly better results than 4 shared hidden layers on different training sets, although the optimal relative weights of the two tasks vary slightly.

4.3. Comparison of LVCSR-initialization and MTL

Figure 5 shows DET curves comparing the transfer learning approach proposed in [6], i.e. LVCSR-initialization of the KW DNN vs. the (KW, LVCSR) multi-task DNN proposed in this paper. This investigation was on a larger training data set, where the multi-task configuration was fixed at 3 shared and 1 separate hidden layers, and $\gamma = 0.9$ in the multi-task loss function, based on tuning experiments on the Dev set. It is seen that the multi-task DNN is comparable to or performs slightly worse than the LVCSR-initialized DNN.

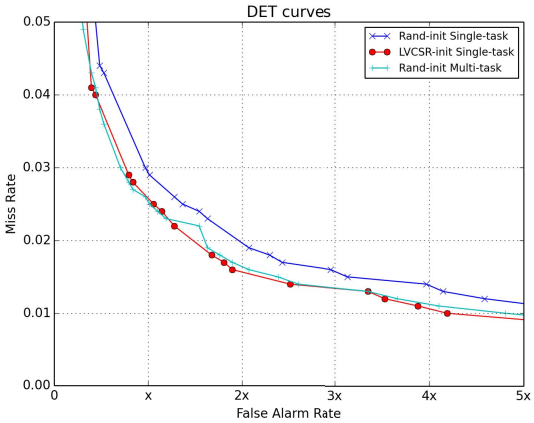

Figure 5: *Comparison of LVCSR-initialization vs. Multi-task training; DET curves on the held-out Test Set.*

4.4. Combination of LVCSR-initialization and MTL

We also investigated the combination of LVCSR-initialization followed by multi-task training.The DET curve results are shown in Figure 6, over a range of miss rates of interest. Corre-

sponding Area Under the Curve (AUC) measures were obtained for the plots in Figure 6 over a range of FA rates, and the numbers given in Table 1.

First, we see that the proposed Multi-task approach is comparable to or slightly worse (1.5% relative in AUC) than LVCSR-initialization, on the held-out test set. We also find that the combination of LVCSR-initialization and Multi-task training can give an accuracy improvement over either technique alone, with relative reductions in AUC of 7.5% and 8.9% over LVCSR-initialization only and Multi-task training only, respectively.

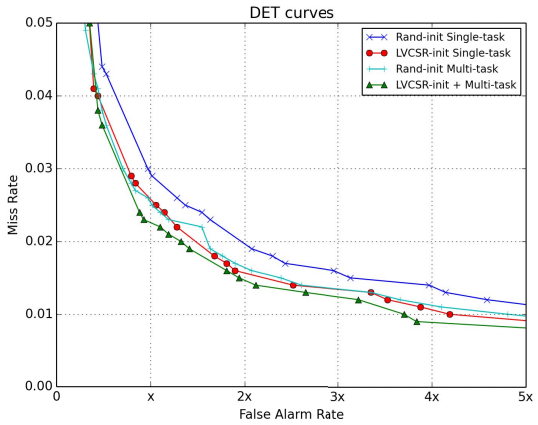

Figure 6: *Results on held-out Test set, with combination of LVCSR-initialization and Multi-task training.*

4.5. Results of combination with class-weighted cross-entropy training

We also investigated the class-weighted cross-entropy loss function that was described in Section 3.2, in the context of LVCSR-initialization and MTL with (KW, LVCSR) targets. In general, we find that for training data sets that are unbalanced with respect to the numbers of KW and non-KW utterances, i.e., where the number of KW utterances is relatively small, CW-XENT with a higher weight on the KW frames gives large improvements. On the data set that is considered here, the imbalance between KW and non-KW data is not as large, and the improvements are smaller. We first tuned the weight on the KW classes in CW-XENT on the dev set, and then evaluated the optimal KW weight of 1.5 on the held out Test set. The resulting DET curve and corresponding AUC number on the Test set are shown in Figure 7 and Table 1 respectively.

We see that CW-XENT can give an additional small improvement in accuracy on top of LVCSR-init + Multi-task. Overall, the combination of the three techniques: LVCSR-init + Multi-task + CW-XENT, gives 11.6% relative reduction in AUC over LVCSR-init alone, and on the held out Test set. The total relative reduction with respect to a randomly-initialized single task DNN was 26%. Thus, the multi-task training and CW-XENT loss functions proposed in this paper are seen to be effective in improving accuracy for KW detection.

5. Summary

In this paper we have proposed and studied a combination of multi-task training and weighted cross entropy DNN training loss functions for more accurate keyword (KW) spotting. In the multi-task architecture, the KW DNN acoustic model is trained with two tasks in parallel - the main task of predicting the KW-specific phone states, and an auxiliary task of predicting LVCSR senones. We first showed that this multi-task approach leads to comparable accuracy with respect to a previously proposed transfer learning approach where the KW DNN is initialized by an LVCSR DNN of the same input and hidden layer sizes. We also showed that combination of LVCSR-initialization and (KW, LVCSR) Multi-task training gives accuracy improvements over LVCSR-initialization alone, with the relative reduction in AUC being around 7.5% on the training data sets studied.

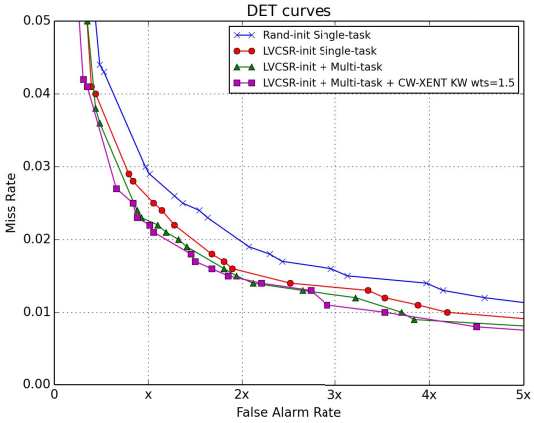

Figure 7: *Results on held-out Test set with combination of 3 techniques: LVCSR-initialization, Multi-task training and Class-weighted cross-entropy.*

Table 1: *Area Under the Curve (AUC) comparison of DET plots from Figure 7, on held out Test set. (Range of FARs used to compute AUC was 0.5x to 5x)*

Model	**AUC**	**Relative Reduction**
Random-init Single-task	0.239x	0%
LVCSR-init Single-task	0.199x	17%
Random-init Multi-task,	0.202x	16%
LVCSR-init + Multi-task	0.184x	23%
LVCSR-init + Multi-task + CW-XENT, KW wts=1.5	0.176x	26%

We also proposed modifying the DNN training loss function to give a higher weight on input frames corresponding to KW phone targets, with a motivation to balance the KW and background training data. We showed that weighted cross-entropy results in additional accuracy improvements. We showed that the combination of 3 techniques: LVCSR-initialization, Multi-task training and Class-weighted cross-entropy training gives the best results. Compared to the LVCSR-initialization technique alone, the combination of 3 techniques results in significantly lower Miss Rate over a range of False Alarm Rates, with a relative reduction in AUC of 11.6% on a held out Test set. Thus, the multi-task training and weighted cross-entropy loss functions proposed in this paper are seen to be very effective in improving accuracy for KW detection.

6. Acknowledgements

We wish to thank Ryan Thomas, George Tucker, Gengshen Fu, Nikko Strom and Rohit Prasad for discussions and suggestions.

7. References

[1] Geoffrey Hinton, Li Deng, Dong Yu, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath George Dahl, and Brian 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, pp. 82-97, November 2012

[2] Dong Yu, Michael L. Seltzer, Jinyu Li, Jui-Ting Huang, Frank Seide, "Feature Learning in Deep Neural Networks - Studies on Speech Recognition Tasks", ICLR 2013.

[3] Vincent Vanhoucke, Andrew Senior, Mark Z. Mao "Improving the speed of neural networks on CPUs," Deep Learning and Unsupervised Feature Learning Workshop, NIPS 2011

[4] Rich Caruana, "Multitask learning," *Machine Learning*, vol. 28, pp. 4175, 1997

[5] Georg Heigold, Vincent Vanhoucke, Andrew Senior, Patrick Nguyen, MarcAurelio Ranazaot, Matthieu Devin, and Jeff Dean, Multilingual acoustic models using distributed deep neural networks, in *ICASSP 2013*.

[6] Guoguo Chen, Carolina Parada, Georg Heigold. "Small-footprint keyword spotting using deep neural networks," ICASSP 2014.

[7] G. Hinton, O. Vinyals, and J. Dean, "Distilling knowledge in a neural network," in *Deep Learning and Representation Learning Workshop*, NIPS, 2014

[8] Peter Bell, Joris Driesen and Steve Renals, "Cross-lingual adaptation with multi-task adaptive networks," *Proc. Interspeech* 2014.

[9] Peter Bell and Steve Renals, "Regularization of context-dependent deep neural networks with context-independent multi-task training," *Proc. ICASSP* 2015.

[10] Michael L. Seltzer and Jasha Droppo, "Multi-task learning in deep neural networks for improved phoneme recognition," *Proc. ICASSP*, 2013.

[11] Yan-Hui Tu, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "Speech separation based on signal-noise-dependent deep neural networks for robust speech recognition," *Proc. ICASSP* 2015.

[12] Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson and Paris Smaragdis, "Deep learning for monaural speech separation," *Proc. ICASSP* 2014.

[13] Ritwik Giri, Michael L. Seltzer, Jasha Droppo, and Dong Yu, "Improving speech recognition in reverberation using a room-aware deep neural network and multi-task learning," *Proc. ICASSP* 2015.

[14] Miller, David RH, Michael Kleber, Chia-Lin Kao, Owen Kimball, Thomas Colthurst, Stephen A. Lowe, Richard M. Schwartz, and Herbert Gish. "Rapid and accurate spoken term detection." In Eighth Annual Conference of the International Speech Communication Association. 2007.

[15] Parlak, Siddika, and Murat Saraclar. "Spoken term detection for Turkish broadcast news." In Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on, pp. 5244-5247. IEEE, 2008.

[16] Tsakalidis, Stavros, Roger Hsiao, Damianos Karakos, Tim Ng, Shivesh Ranjan, Guruprasad Saikumar, Le Zhang, Long Nguyen, Richard Schwartz, and John Makhoul. "The 2013 BBN vietnamese telephone speech keyword spotting system." In Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, pp. 7829-7833. IEEE, 2014.

[17] Rose, Richard C., and Douglas B. Paul. "A hidden Markov model based keyword recognition system." In Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on, pp. 129-132. IEEE, 1990.

[18] Wilpon, Jay G., L. Rabiner, Chin-Hui Lee, and E. R. Goldman. "Automatic recognition of keywords in unconstrained speech using hidden Markov models." Acoustics, Speech and Signal Processing, IEEE Transactions on 38, no. 11 (1990): 1870-1878.

[19] Wilpon, J. G., L. G. Miller, and P. Modi. "Improvements and applications for key word recognition using hidden Markov modeling techniques." In Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on, pp. 309-312. IEEE, 1991.

[20] Ming Sun, Varun Nagaraja, Björn Hoffmeister and Shiv Vitaladevuni, "Model Shrinking for Embedded Keyword Spotting," 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA). IEEE, 2015.

[21] M. Weintraub, "Improved Keyword-Spotting Using SRI'S DECIPHER (TM) Large-Vocabuarly Speech-Recognition System," ACL HLT 1993.

[22] Fernández, Santiago, Alex Graves, and Jürgen Schmidhuber. "An application of recurrent neural networks to discriminative keyword spotting." In Artificial Neural Networks?ICANN 2007, pp. 220-229. Springer Berlin Heidelberg, 2007.

[23] George Tucker, Minhua Wu, Ming Sun, Sankaran Panchapagesan, Gengshen Fu, Shiv Vitaladevuni, "Model compression applied to small-footprint keyword spotting," *Proc. Interspeech* 2016 (to appear).

[24] S.B. Davis, and P. Mermelstein (1980), "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," in IEEE Transactions on Acoustics, Speech, and Signal Processing, 28(4), pp. 357-366.

[25] Christopher Bishop,"Pattern Recognition and Machine Learning," Springer 2006.

[26] Nikko Strom, "Scalable Distributed DNN Training Using Commodity GPU Cloud Computing," *Proc. Interspeech* 2015.