Distinct Triphone Acoustic Modeling Using Deep Neural Networks

Dongpeng Chen, Brian Mak

Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
{dpchen,mak}@cse.ust.hk

Abstract

To strike a balance between robust parameter estimation and detailed modeling, most automatic speech recognition systems are built using tied-state continuous density hidden Markov models (CDHMM). Consequently, states that are tied together in a tied-state are not distinguishable, introducing quantization errors inevitably. It has been shown that it is possible to model (almost) all distinct triphones effectively by using a basis approach; previously two methods were proposed: eigentriphone modeling and reference model weighting (RMW) in CDHMM using Gaussian-mixture states. In this paper, we investigate distinct triphone modeling under the state-of-the-art deep neural network (DNN) framework. Due to the large number of DNN model parameters, regularization is necessary. Multi-task learning (MTL) is first used to train distinct triphone states together with carefully chosen related tasks which serve as a regularizer. The RMW approach is then applied to linearly combine the neural network weight vectors of member triphones of each tied-state before the output softmax activation for each distinct triphone state. The method successfully improves phoneme recognition in TIMIT and word recognition in the Wall Street Journal task.

Index Terms: distinct triphone acoustic modeling, multi-task learning, deep neural networks

1. Introduction

Context-dependent (CD) acoustic units are used in almost all the current automatic speech recognition (ASR) systems. Their distribution, however, is very uneven. It was shown in [1] that the 80–20 Pareto rule seems also true for the distribution of triphones in the 84-hour Wall Street Journal training corpora: about 80% of all triphone occurrences in the corpus come from only 20% of the most common distinct triphones. Direct estimation of the acoustic model parameters for the infrequent triphones will yield poor models and hurt the overall recognition performance. The problem is more severe for low-resource languages with small amounts of training data [1].

Different methods have been proposed for the robust parameter estimation of context-dependent acoustic models. For example, triphone-by-decomposition [2, 3], parameter tying [4] and the basis approach [5, 6] were explored and proved to be effective for acoustic models using hidden Markov models (HMM) with Gaussian-mixture (GMM) states. Among these methods, state tying [7] is most popular in modern ASR systems due to its simplicity and effectiveness in solving the unseen triphones problem. Nevertheless, when states are tied, the member states in a tied-state (or senone) are indistinguishable, and quantization errors are inevitably introduced.

Lately we investigate the following problem: Is it possible to model each distinct triphone (that has ever appeared) robustly? For GMM-HMM acoustic models, we had proposed two methods: eigentriphone modeling (ETM) [8] and reference model weighting (RMW) [9] for distinct triphone modeling. The idea is to treat the distinct triphone modeling problem as an adaptation problem, and both methods derive a set of basis vectors so that the (distinct or untied) member states of a senone can be projected onto the low-dimensional space spanned by them. Eigentriphone modeling is analogous to eigenvoice and reference modeling weighting is analogous to reference speaker weighting in speaker adaptation. Moreover, eigentriphone modeling derives an orthogonal basis using weighted principal component analysis (PCA) whereas the basis derived by reference model weighting is generally non-orthogonal as it is composed simply from the supervectors of the member states. Both methods were found effective and improved recognition performance for both low-resource speech recognition and large-vocabulary speech recognition tasks.

With the emergence of the deep neural network (DNN) for acoustic modeling, we would like to investigate if distinct triphone modeling is also feasible in DNN-HMM. The major challenge is the large increase of DNN parameters required to train all the distinct triphones. For example, for the TIMIT task, there are about 600 output units (or senones) in a typical triphone DNN-HMM, but the number increases to almost 10,000 if all the distinct triphone states (DTS) of TIMIT are to be modeled. The weight vector for each DNN output commonly consists of 1024–2048 weights, whereas ETM or RMW only employs only a basis of several tens or a hundred of basis vectors to model the Gaussian means of the DTS's. In this paper, we propose to train DTS DNN-HMM together with the training of senone DNN-HMM in the multi-task learning (MTL) framework, so that the task of senone training will act as a regularizer for the task of DTS training. Then the RMW method originally designed for GMM-HMM is modified to transform the activation of output units of each DTS to further improve their performance. Experimental evaluations on TIMIT phoneme recognition and WSJ0 word recognition show that the proposed method is effective.

The rest part of the paper is organized as follows. We first introduce our MTL-DNN training method of distinct triphone states in Section 2. Then in Section 3, after a quick review of RMW, the RMW transformation of the output activations of DTS in DNN is described. This is followed by experimental evaluations in Section 4. Finally, we give the concluding remarks in Section 5.

[1] Collecting more data does not completely avoid the problem because there will be more seen triphones in the larger data set and the Pareto rule still applies. However, it will partly solve the problem because a larger data set will probably give a better coverage of the more commonly used triphones in the test set.

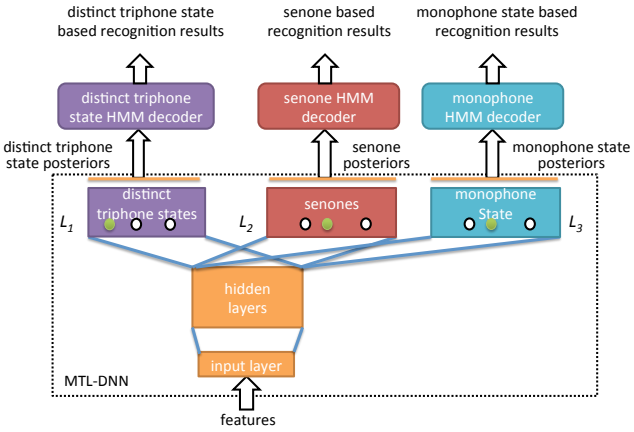

Figure 1: The MTL-DNN used for the joint training of monophone states, senones (or tied states) and distinct triphone states (DTS). The horizon bars represent the softmax activation.

2. Multi-task Learning of Distinct Triphones Using DNNs

Multitask learning (MTL) [10] is a machine learning approach that is used to jointly learn multiple tasks together. If the multiple tasks are *related* and if they can share some internal representation, then through learning them together, they are able to transfer knowledge to one another to improve their generalization performance. From the regularization perspective, related tasks serve as the regularizers for each other during MTL training. In [10], Caruana postulates some requirements for related tasks if their joint learning in the MTL approach is to work well:

(a) related tasks must share input features, and

(b) related tasks must share hidden units to benefit each other when trained with MTL-backprop.

The hidden layers of a neural network naturally capture learned knowledge that can be readily transferred or shared across multiple tasks. In fact, the hidden layers of DNN act as a common feature extractor for all learning tasks. We will call a DNN trained by MTL algorithm as an MTL-DNN. MTL-DNN has been applied successfully in fields of speech, language [11], and image/vision [12]. In ASR, MTL-DNN has been used to improve TIMIT phoneme recognition [13, 14], cross- and multilingual speech recognition [15, 16], and low-resource speech recognition [17, 18].

2.1. MTL-DNN for Training Distinct Triphone States

Due to the large number of distinct triphone states (DTS) in common tasks (when compared with the number of tied states or senones), standard single-task learning (STL) of DNNs does not work well, resulting in overfit models with poor recognition performance on unseen test data. Regularization is required, and in this paper, we investigate the use of highly related task(s) as the regularizer(s) under the framework of multi-task learning of DNN. The resulting networks will be called MTL-DNNs. Our approach strictly follows Caruana's two MTL requirements. That is, for each training input, multiple output units are activated and multiple tasks are learned in parallel. Furthermore, the tasks share the same acoustic observations and all hidden layers in a DNN.

[18] suggests a heuristic guideline to select the extra task(s) for a primary task in MTL: select task(s) that may exploit extra information from the training data of the primary task. Here, DTS modeling is taken as the primary task $\mathcal{T}_1$), and the following two extra tasks are selected for its MTL training: senones modeling task $\mathcal{T}_2$, and monophone states modeling task $\mathcal{T}_3$. The three tasks are obviously related as they all represent acoustic modeling at different phonetic resolutions. The extra information is the implicit membership of the DTS's in the senones and monophones.

Fig. 1 shows an MTL-DNN used for training the three tasks together. Basically, three single-task learning DNNs (STL-DNNs) which estimate the posterior probabilities of monophone states, senones, and distinct triphone states (DTS) are merged together so that they their inputs and all hidden layers, while each of them keeps its own output layer (L_1, L_2 and L_3). The MTL objective function is to minimize the sum of cross entropies of the three tasks over the training set $\mathcal{D}$:

$$\mathcal{E}(\mathcal{D}, \Lambda) = -\sum_{\mathbf{x}\in\mathcal{D}} \left(\sum_{j=1}^{3} \sum_{i=1}^{N_j} d_i^{(j)} \log P(s_i^{(j)}|\mathbf{x}; \lambda_0, \lambda_j) \right) \tag{1}$$

where $d_i^{(j)}$ is the target value of the ith state in $\mathcal{T}_j$, λ_0 consists of model parameters that are shared by *all* tasks (i.e., the hidden layers) and λ_j consists of model parameters specific to only task $\mathcal{T}_j$ (i.e., the output layers). During training, for each input acoustic vector, 3 output units, one for each task, will be activated. Thus, the shared hidden layers help the tasks regularize each other to achieve better generalization.

In practice, starting from a conventional GMM-HMM, we go through the following steps to construct and train the proposed MTL-DNN.

STEP 1 : A conventional senones (tied-states) GMM-HMM system is first built.

STEP 2 : The senone GMM-HMM system is used to forced-align the training and development data to get the frame labels for DNN training. Each frame is assigned 3 labels: `<DTS, DTS_index>`, `<senone, senone_index>`, and `<monophone, monophone_state_index>`.

STEP 3 : Pre-train a DBN by contrast divergence [19].

STEP 4 : Add an output softmax layer of senones, and train the senones posteriors by standard DNN back-propagation.

STEP 5 : Add an output softmax layer of monophone states and initialize it with random weights.

STEP 6 : Treat each senone as a state cluster and untie it to get all its member DTS's. Add an output softmax layer consisting of all those DTS's that have at least 10 training samples[2]. The output weight vector of a DTS is initialized by the well-trained weight vector of its corresponding senone.

STEP 7 : Train the resulting MTL-DNN by standard back-propagation like an STL-DNN except that the learning rate of the hidden layers is set to $1/3$ of that for STL-DNN, since 3 error signals are now propagated back to the hidden layers. The learning rate of the output layers remains the same.

STEP 8 : During decoding, posteriors for the three kinds of output units are generated separately and fed into their corresponding HMM decoders.

[2]For unseen triphone states and those DTS's with fewer than 10 samples, they will be still represented by the appropriate senones.

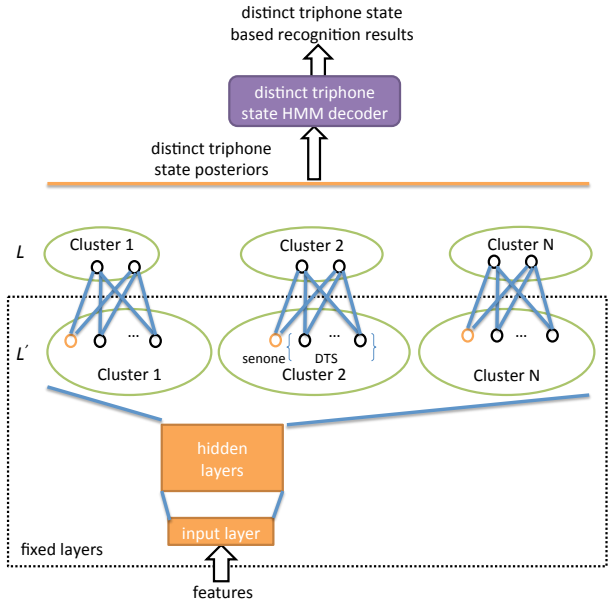

Figure 2: Transforming the activation of each DTS using the RMW technique.

3. Transformation of DTS Activations by RMW

When reference model weighting (RMW) is applied to distinct triphone modeling using GMM-HMMs, each triphone state is represented as a supervector created by stacking up the mean vectors of its Gaussian mixtures. Then for each state cluster (which is equivalent to a tied state in practice), all or a subset of its member DTS supervectors are collected to form a basis and all DTS's of the cluster can then be expressed as a linear combination of the basis vectors. Since the number of such basis vectors is much smaller than the dimension of the state supervectors, the combination weights may be estimated robustly even with very small amount of training samples.

3.1. RMW of the DTS Weight Vectors

In the DTS DNN, each output node represents a DTS. Let's assume that there are N_k DTS's $\{s_i^{(k)} : i = 1, 2, \ldots, N_k\}$ in the state cluster represented by the kth senone; there are H hidden units in the last hidden layer, and their outputs (including the bias) are represented by $\mathbf{y} = [1, y_1, y_2, ..., y_H]'$. Let's also denote the DNN weight vector connecting the last hidden layer to $s_i^{(k)}$ as $\mathbf{w}_i^{(k)} \in \mathbb{R}^{H+1}$. Thus, its activation is given by

$$a_i^{(k)} = \mathbf{w}_i^{(k)\prime} \mathbf{y} \,. \tag{2}$$

When applying RMW for robust training of DTS DNNs, the set of weight vectors of the kth state cluster, $\{\mathbf{w}_0^{(k)}\} \cup \{\mathbf{w}_i^{(k)} : i = 1, 2, \ldots, N_k\}$, where $\mathbf{w}_0^{(k)}$ is the DNN weight vector of the kth senone, is treated as a basis. Then the DNN weight vector of each member DTS is re-modeled as a linear combination of the basis vectors as follows:

$$\hat{\mathbf{w}}_i^{(k)} = \mathbf{W}^{(k)} \mathbf{u}_i^{(k)} \,, \tag{3}$$

with the new activation $\hat{a}_i^{(k)} = \left(\mathbf{W}^{(k)} \mathbf{u}_i^{(k)}\right)' \mathbf{y}$, where $\mathbf{W}^{(k)} = [\mathbf{w}_0^{(k)} \mathbf{w}_1^{(k)} \ldots] \in \mathbb{R}^{(H+1)\times(N_k+1)}$ and $\mathbf{u}_i^{(k)} \in \mathbb{R}^{(N_k+1)}$ is the combination weights for the DTS unit $s_i^{(k)}$. Hence, if we represent the activations to all $\{s_i^{(k)}\}$ by $\mathbf{a}^{(k)} \in \mathbb{R}^{N_k}$, and all their combination weight vectors by $\mathbf{U}^{(k)} \in \mathbb{R}^{(N_k+1)\times N_k}$, then the new activation vector after RMW is given by

$$\hat{\mathbf{a}}^{(k)} = \left(\mathbf{W}^{(k)} \mathbf{U}^{(k)}\right)' \mathbf{y} = \mathbf{U}^{(k)\prime} \left(\mathbf{W}^{(k)\prime} \mathbf{y}\right) = \mathbf{U}^{(k)\prime} \begin{bmatrix} a_0^{(k)} \\ \mathbf{a}^{(k)} \end{bmatrix} \tag{4}$$

where $a_0^{(k)}$ is the activation due to the kth senone.

Eq. (4) demonstrates that the application of RMW on the connection weight vectors between the last hidden layer and the DTS output layer is equivalent to applying a linear transform $\mathbf{U}^{(k)}$ on the activations to the DTS output units. The proposed RMW method can be implemented by adding an additional RMW layer between the last hidden layer and the DTS output layer of the DNN as shown in Fig. 2. Our method differs from the ensemble learning method in [20] where model integration is performed *after* the softmax function.

3.2. Estimation of the Combination Weights

To estimate the additional RMW combination weights for each state cluster, all the network weights of a well-trained DNN from the input layer to the layer L' of Fig. 2 are fixed. Each RMW combination weight vector is initialized as $\mathbf{u}_i^{(k)} = [1, 0, 0, \ldots]$ where the unity value corresponds to the combination weight to the senone vector $\mathbf{w}_0^{(k)}$ which will also be fixed during the estimation. Since there are many parameters in the additional layer, L2 regularization was tried. In some preliminary TIMIT experiments, we empirically found that similar results could be achieved by simply re-estimating each DTS weight vector as a linear combination of its original vector and its senone weight vector as below:

$$\hat{\mathbf{w}}_i^{(k)} = \mathbf{w}_0^{(k)} + \alpha \mathbf{w}_i^{(k)} \tag{5}$$

where α is a global combination weight for all DTS's which is tuned using a development set. Therefore we employed this simple method in the following experiments.

4. Experiments

The proposed robust distinct triphone modeling method based on DNNs as described in Section 2 and 3 was tested on the TIMIT phone recognition task and the WSJ0 word recognition task. Their performance is compared with their respective GMM-HMM and STL-DNN baselines which were prepared as described in [9].

In all GMM-HMM acoustic models, cross-word triphones were employed and modeled as 3-state CDHMMs. Feature vectors were standard 39-dimensional MFCC acoustic vectors, and they were extracted from the training speech data every 10ms over a window of 25ms. On the other hand, all DNNs in our experiments were pre-trained with a deep belief network consisting of 4 hidden layers with 2048 nodes per layer, and the mini-batch size was set to 128. Afterward, an output softmax layer of the required modeling states was added, which correspond to the states of their respective GMM-HMM baselines. The resulting DNNs were fine-tuned using standard back-propagation.

4.1. TIMIT Phone Recognition

4.1.1. Acoustic Modeling

A conventional state-tied GMM-HMM baseline system was first trained. It has 587 senones and 16 mixtures per state, and

achieves a phone recognition accuracy of 72.04% on the standard test set.

Table 1: Phone recognition accuracies (%) of different DNN systems. M, S and D represent systems using monophone states, senones and distinct triphone states as output units respectively.

Models	M output	S output	D output
STL-DNN (M)	78.12	—	—
STL-DNN (S)	—	78.89	—
STL-DNN (D)	—	—	75.30
MTL-DNN (MS)	78.36	79.18	—
MTL-DNN (SD)	—	79.60	76.72
MTL-DNN (MSD)	78.42	80.01	77.74
+ RMW activations	—	—	80.30

Then, single-task learning DNN (STL-DNN) baseline systems were built using 40-dimensional log filter-bank features and the energy coefficient as well as their first- and second-order derivatives. STL-DNNs were trained to classify the central frame of each 15-frame acoustic context window. STL-DNNs for monophone states and senones were first trained. Then triphone states with at least 10 training samples were untied from their senones. At the end, a set of 9823 distinct triphone states (DTS) were obtained which is $\sim$17 times of the optimal number of senones. The network weights of the DTS STL-DNN were initialized from the senone STL-DNN and trained as describe in Section 2. MTL-DNNs were then built by jointly training at least two of the three kinds of acoustic units (monophone states, senones, and DTS's) together. During testing, outputs of different acoustic units were computed independently and fed into corresponding decoders.

Finally, the DTS activations were further transformed using the RMW technique as described by Eq. (5). The optimal value of the global parameter α was determined using the development data, and 0.1 was found to give the best results.

4.1.2. Experiment Results and Discussion

Table 1 summarizes the recognition performance of various DNN systems. We have the following observations:

- As expected, the STL-DNN baselines cuts the error rates of the senone GMM-HMM baseline by nearly 22%!
- Among three STL-DNNs, senone STL-DNN has the best performance, as it successfully trades off detailed modeling between monophone states and DTS's.
- Joint training of different kinds of acoustic units using MTL-DNNs between both the primary task as well as the extra task(s). DTS's were more robustly trained due to the regularization effect from the senones and monophone states which were well trained and thus might provide a representation bias towards a better local optimum for DTS modeling. On the other hand, senones were also better trained due to the extra contextual information provided by the DTS's.
- Although after MTL training, DTS's still perform the worst, the proposed RMW-based transformation of the DTS activations could successfully re-estimate their parameter robustly. Compared with the senone DNN baseline, the final DTS DNN system reduces the phone recognition errors by 6.7%.

Table 2: WSJ0: Word recognition accuracies (%) of various DNN-HMM systems using bigram/trigram language models.

Models	M output	S output	D output
STL-DNN (M)	91.93/—	—	—
STL-DNN (S)	—	94.90/96.79	—
STL-DNN (D)	—	—	94.83/96.54
MTL-DNN (MSD)	92.78/—	95.25/97.20	95.52/97.24
+ RMW activations	—	—	95.67/97.36

4.2. WSJ0 Reading Speech Recognition

4.2.1. Acoustic Modeling

The training procedure of the GMM-HMM baseline followed that of TIMIT. The optimal system has 1254 senones and 32 Gaussian mixtures per state. It has a word recognition accuracy of 93.29% on the Nov92 test set using a bigram language model.

The training procedure for the DNN systems was also the same as that of the TIMIT task, except that input features now are the standard 39-dimensional MFCC vectors. There are 22407 DTS's that have at least 10 training samples. This number is about 18 times the number of senones. Moreover, during recognition, the posterior probabilities of the output units were scaled by the their statistical priors counted from the training data before they were fed into their corresponding decoders.

4.2.2. Results and Discussion

Experiment results of the various DNN-HMM systems using bigram and trigram language models are presented in Table 2.

- Different from the experimental results of TIMIT, the performance of DTS DNNs is comparable to senone DNNs and is much better than the monophone state DNN. An analysis of the occurrences of the DTS's over the test sets shows that there are fewer infrequent DTS's in the WSJ0 test set than in the TIMIT test set.
- MTL training consistently reduces the word error rate (WER) of all tasks simultaneously.
- RMW transformation of the DTS activations yields the best performance for DTS DNNs, resulting in a relative WER reduction of 15.1% (17.8%) over the senone STL-DNN baseline using a bigram (trigram) language model.

5. Conclusion

In state-tied ASR systems, quantization error is inevitable because triphone states that are tied together become indistinguishable with each other during decoding. This paper propose a reference model weighting (RMW) technique for robust distinct triphone modeling in a DNN-HMM under the multi-task learning (MTL) framework. Distinct triphone states (DTS's) are jointly trained with senones (and perhaps monophone states) using an MTL-DNN. Their joint training has three benefits. Firstly, the DTS modeling task embeds more contextual information into the hidden layers of the MTL-DNN. Secondly, monophone states modeling and senones modeling tasks provide better inductive bias for the DTS modeling task to reach a better optimum. Lastly, the shared hidden layers make it easy to implement an activation transformation based on the RMW technique by simply adding an additional layer between the last hidden layer and the DTS output layer for each state cluster.

6. References

[1] T. Ko and B. Mak, "Eigentriphones: A basis for context-dependent acoustic modeling," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 4892–4895.

[2] K.-F. Lee, *Automatic Speech Recognition: The Development of the Sphinx Recognition System*. Springer, 1989, vol. 62.

[3] J. Ming, P. O'Boyle, M. Owens, and F. J. Smith, "A Bayesian approach for building triphone models for continuous speech recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 7, no. 6, pp. 678–684, 1999.

[4] S. Takahashi and S. Sagayama, "Four-level tied-structure for efficient representation of acoustic modeling," in *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, vol. 1. IEEE, 1995, pp. 520–523.

[5] X. D. Huang and M. A. Jack, "Semi-continuous hidden Markov models for speech signals," *Computer Speech & Language*, vol. 3, no. 3, pp. 239–251, 1989.

[6] D. Povey, L. Burget, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. K. Goel, M. Karafiát, A. Rastrow *et al.*, "Subspace Gaussian mixture models for speech recognition," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4330–4333.

[7] S. J. Young and P. C. Woodland, "The use of state tying in continuous speech recognition." in *Eurospeech*, 1993.

[8] T. Ko and B. Mak, "Eigentriphones for context-dependent acoustic modeling." *IEEE Transactions on Audio, Speech & Language Processing*, vol. 21, no. 6, pp. 1285–1294, 2013.

[9] D. Chen and B. Mak, "Distinct triphone modeling by reference model weighting," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 7150–7154.

[10] R. Caruana, "Multitask learning," Ph.D. dissertation, Carnegie Mellon University, USA, 1997.

[11] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the International Conference on Machine Learning*. ACM, 2008, pp. 160–167.

[12] Y. Huang, W. Wang, L. Wang, and T. Tan, "Multi-task deep neural network for multi-label learning," in *Proceedings of the IEEE International Conference on Image Processing*, 2013, pp. 2897–2900.

[13] M. Seltzer and J. Droppo, "Multi-task learning in deep neural networks for improved phoneme recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 6965–6968.

[14] D. Chen, B. Mak, and S. Sivadas, "Joint sequence training of phone and grapheme acoustic model based on multi-task learning deep neural networks," in *Proceedings of Interspeech*, 2014, pp. 1083–1087.

[15] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7304–7308.

[16] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7319–7323.

[17] D. Chen, B. Mak, C.-C. Leung, and S. Sivadas, "Joint acoustic modeling of triphones and trigraphemes by multi-task learning deep neural networks for low-resource speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2014.

[18] D. Chen and B. Mak, "Multi-task learning of deep neural networks for low-resource speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, 2015. (To be published)

[19] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[20] L. Deng and J. C. Platt, "Ensemble deep learning for speech recognition," in *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2014.