GEO-LOCATION DEPENDENT DEEP NEURAL NETWORK ACOUSTIC MODEL FOR SPEECH RECOGNITION

Guoli Ye[1], *Chaojun Liu*[2], *Yifan Gong*[2]

[1]Microsoft Search Technology Center Asia, Beijing, China
[2]Microsoft Corporation, One Microsoft Way, Redmond, WA 98052

{guoye; chaojunl; ygong}@microsoft.com

ABSTRACT

Users from the same geo-location region exhibit similar acoustic characteristics, e.g., they have similar accent; even more, they may have similar preference to device. In this paper, we propose to build geo-location dependent deep neural network for speech recognition, where the geo-location signal is inferred from users' GPS. During runtime, the server will base on a user's geo-location to select the right model to recognize his voice. We tackle three major issues associated with this model: high train/deployment cost, large model size, and train data sparsity. Our solution is featured by its low cost, thus practical for production modeling. We also discuss the reliability of GPS signal in practical use. The proposed model is evaluated on Microsoft Chinese voice search and Cortana live test set. Among 12 provinces, it shows an overall 4.8% relative character error rate reduction, over a strong baseline production-level model, with only 50% model size increase. The gain is larger for the low-resource provinces, with relative error rate reduction up to 9%.

Index Terms— geo-location, acoustic modeling, speech recognition

1. INTRODUCTION

Deep neural network hidden Markov model (DNN) [1] is more robust than Gaussian mixture hidden Markov model (GMM), for different accent, speakers, and noise. However, it is still beneficial to consider these variations during model building: by adapting the model to different accent [2, 3], and different speakers [4, 5]; or by explicitly augmenting the input with noise signal [6].

We observe that users from the same geo-location region have similar acoustic characteristics, e.g., they have similar accent; even more, they may have similar preference to device. Thus, instead of using one DNN to handle users' voice from different geo-locations, we propose to build geo-location dependent DNN. The geo-location signal of a user has different levels of granularity: GPS, and its derived city, province (state), and country. In this paper, we will use province (state) level signal. In runtime, there will be a set of DNNs in speech server, each for one province. When a user calls the service, his current province inferred from GPS will be used as the signal to choose the right model to recognize his voice.

Geo-location information has been used to build language model, and shown good gain [7]. In acoustic model, the most related research is accent model [2, 3, 8], i.e., to build one model for each accent region. One way to get accent signal is to ask the user to specify his accent when using the application. However, users are not always cooperative in practice. A more feasible way is to automatically identify the users' accent [9]. The accent identification module introduces additional runtime cost, and is not always correct. Also, to train a robust identification module requires a lot of accent labeled data, which is costly. Unlike accent model, our method directly derives a user' province from his GPS, which is zero-cost in both runtime and data labeling.

The number of provinces in a country is usually large. Take China for example, there are 34 provinces. That means we need to build 34 geo-location dependent DNNs (GLD-DNN). Compared with geo-location independent DNN (GLI-DNN), it poses several challenges. Firstly, it will increase significantly the model size, training and deployment cost. Secondly, the train data of each GLD-DNN is only a small subset of that of GLI-DNN, which causes data sparsity issue.

To reduce the training cost, we propose a simple training recipe to update the baseline multi-style sequential trained GLI-DNN by a sequential adaptation with specific geo-location data. The adaptation will re-use most of the files already generated by GLD-DNN, e.g., feature files and sequential training lattices. Thus, the additional training cost on top of GLI-DNN is very small. Furthermore, starting from a robustly trained GLI-DNN, the adapted GLD-DNN is less likely to deviate from a good model, relieving the data sparsity issue. Section 3 describes the training recipe in detail.

To reduce model size and tackle data sparsity, we apply singular value decomposition (SVD) bottleneck adaptation. SVD bottleneck adaptation was originally proposed in [4] for speaker adaptation, which updates only a small part of DNN parameters and requires less data. As a result, each GLD-DNN only needs to store the small amount of adapted parameters. Section 4 introduces SVD bottleneck adaptation, and compares it with other adaptation methods.

In practice, we found that the data from some provinces are similar to each other. In Section 5, we propose a simple way to cluster the training data based on cross-test results. This technique further relieves the data sparsity problem, and helps a lot in limited-resource provinces. The geo-location and accent are considered related. In Section 6, the performance of GLD-DNN on accent speech is evaluated. In Section 7, we discuss the reliability of the geo-location signal. Specifically, what happens when people travel from one province to another.

2. EXPERIMENTAL SETTINGS

2.1. Data

We are working on Chinese geo-location models, though this technique could be applied to other languages as well. The data are hand-transcribed anonymous utterances from Microsoft voice search and Cortana traffic in China market. Each utterance is annotated with its user's province information, obtained from the user's query log. The query log is strictly anonymous.

All the training utterances are used to train GLI-DNN. To build GLD-DNN, the data is partitioned into groups, each representing a province. We choose the top twelve largest groups (provinces), and build a GLD-DNN for each of them. The twelve provinces represent the top markets in China, which contribute half of the whole data traffic. Users from the rest of the provinces will still use the GLI-DNN. The training and test data statistics are listed in Table 1. On average, each utterance has a duration of 3.1 seconds.

Table 1. *Data statistics*

Province	#Train Hours	#Train Utterances	#Test Utterances
Guangdong	409	473,616	8,680
Beijing	355	408,487	7,772
Shandong	182	210,120	2,718
Jiangsu	130	152,868	2,580
Zhejiang	123	142,413	2,420
Hebei	105	122,115	2,476
Sichuan	84	96,824	1,984
Shanghai	85	98,292	1,686
Hubei	73	85,585	1,446
Hunan	53	66,223	966
Tianjin	53	62,459	1,016
Liaoning	44	57,779	1,024
All	1696	1,976,781	34,768

2.2. Language Model

A 4-gram language model is used. The vocabulary size is around 200K. The number of *n*-grams is about 40 million.

2.3. Acoustic Model

The DNN model has 6715 nodes in the output layer. The input feature contains 74 dimensions: 22-dimension log-filter-bank with up to the 2nd order derivative, plus 8-dimension pitch related feature. The feature is computed every 10ms over a 25ms window. We also augment the feature vectors with previous and next 5 frames (5-1-5). The DNN is SVD based, the detailed configuration is given in the next section.

3. GLD-DNN TRAINING RECIPE

The GLD-DNN training recipe is shown in Figure 1. Data from all provinces is used to train GLI-DNN. The model is then adapted by each province's data to get the corresponding GLD-DNN.

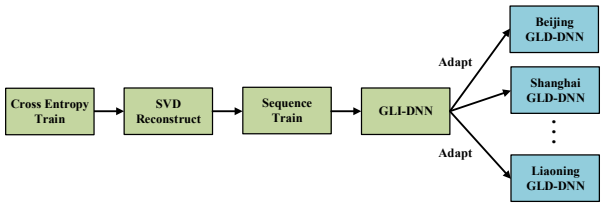

Figure 1: GLD-DNN training recipe

3.1. GLI-DNN Training

The model is first trained with cross entropy (CE) criterion. The resulting DNN has 5 hidden layers, each with 2048 units. SVD reconstruction is then applied, which reduces the model size by 80% and keeps the same accuracy. This resulting model is SVD structured. Finally, the sequential training with maximum mutual information (MMI) criterion [10, 11] is applied to the SVD DNN, with a learning rate of 5E-4. F-smoothing is used [10] with weight 0.05 assigned to CE in the objective function.

3.2. SVD Reconstruction for GLI-DNN

SVD reconstruction was first proposed in [12]. It utilizes the low-rank property of DNN matrices to reduce the DNN model size while maintaining the accuracy. This method applies SVD [13] to each weight matrix $\boldsymbol{A}$ in DNN to get:

$$\boldsymbol{A_{m\times m} = U_{m\times m}\Sigma_{m\times m}V^T_{m\times m}}, \quad (1)$$

where $\boldsymbol{\Sigma}$ is a diagonal matrix with $\boldsymbol{A}$'s singular values on the diagonal in decreasing order. By keeping $\boldsymbol{k}$ biggest singular values of $\boldsymbol{A}$, Equation (1) becomes

$$\boldsymbol{A_{m\times m} \approx U_{m\times k}\Sigma_{k\times k}V^T_{k\times m} = U_{m\times k}N_{k\times m}}, \quad (2)$$

where $\boldsymbol{N_{k\times m} = \Sigma_{k\times k}V^T_{k\times m}}$. In this way, the weight matrix $\boldsymbol{A}$ is decomposed into two smaller matrices $\boldsymbol{U}$ and $\boldsymbol{N}$. As shown in Figure 2, the SVD reconstructed DNN introduces a small SVD bottleneck layer with $\boldsymbol{k}$ neurons between two large hidden layers with size $\boldsymbol{m}$ in the original model. And the number of parameters in weight matrices is changed from the original $\boldsymbol{m \times m}$ to $\boldsymbol{2 \times m \times k}$. Usually, $\boldsymbol{k}$ is much smaller than $\boldsymbol{m}$. In our case, $\boldsymbol{m}$ is 2048, and $\boldsymbol{k}$ is around 300. Therefore, the number of parameters is significantly reduced.

As can be seen in Equation (2), the SVD reconstruction gives only an approximation of the original weight matrix, so the resulting model still has some accuracy degradation. In practice, we retrain the reconstructed SVD DNN to update weights, which usually will get back the accuracy loss.

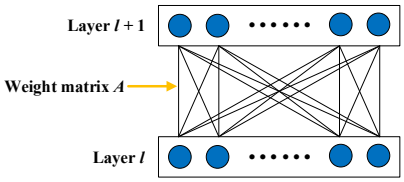

Figure 2(a): Original DNN model

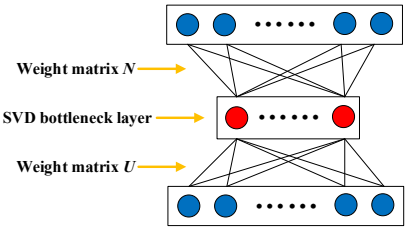

Figure 2(b): SVD reconstructed DNN model

3.3. GLD-DNN Training

Each GLD-DNN is adapted from GLI-DNN with its own province's data. The adaptation criterion is also MMI. Compared with GLI-DNN, due to the limited amount of data, a smaller learning rate of 1E-4 is used. The F-smoothing weight is the same as GLI-DNN, with 0.05 weight assigned to CE in the objective function. We also tried KL divergence regularization [14] and different F-smoothing weights, but did not find better results. It is likely due to our learning rate is very small, which already acts like regularization.

The feature files and lattices used by GLD-DNN adaptation are already generated during GLI-DNN training. So, we could reuse them. As a result, the adaptation is very fast, and the training cost on top of GLI-DNN is small.

4. GLD-DNN ADAPTATION

Despite GLI-DNN is SVD based and already smaller compared with conventional DNN, we still can't afford to update all the parameters during GLD-DNN adaptation. In this section, we propose to adapt only a small part of the parameters in the network, while keeping other parameters unchanged. The adapted parameters are considered to model the province dependent information, while the unchanged ones capture the province independent information. Doing in this way is also good for deployment. The speech server only needs to store one set of province independent parameters, and 12 sets of province dependent parameters. In runtime, the user's province signal will be used to select the province dependent parameters, which will be assembled with province independent parameters to form the final DNN for recognition. This section compares 3 different ways to do GLD-DNN adaptation: (1) top layer adaptation (2) SVD bottleneck adaptation (3) hybrid adaptation.

4.1. Top Layer Adaptation

It was found in [2, 3] that the DNN top layer has well captured the accent information. Since geo-location is closely related to accent, it is reasonable to try only adapting the top layer. Specifically, for the SVD DNN in our system, only the two matrices $\boldsymbol{U_{m\times k}}$ and $\boldsymbol{N_{k\times m}}$ in top layer will get adapted.

4.2. SVD Bottleneck (BN) Adaptation

SVD bottleneck (BN) adaptation was first proposed in [4] for speaker adaptation. It adds an additional linear layer on top of the original SVD bottleneck layer as shown in Figure 3. This introduces an additional square matrix $S_{k\times k}$. We initialize the matrix $S_{k\times k}$ to be an identity matrix, such that the resulting model is equivalent to the original model as shown in Equation (3).

$$\boldsymbol{U_{m\times k}N_{k\times m} = U_{m\times k}S_{k\times k}N_{k\times m}}, \quad (3)$$

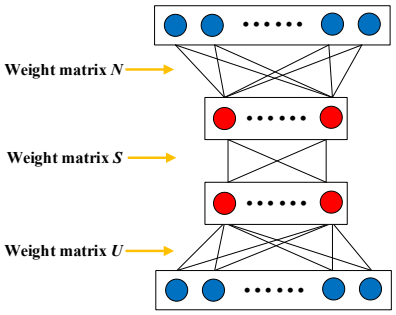

Figure 3: SVD bottleneck adaptation

During GLD-DNN adaptation, we only update the parameters in $\boldsymbol{S_{k\times k}}$, while keeping the other parameters in $\boldsymbol{U_{m\times k}}$ and $\boldsymbol{N_{k\times m}}$ unchanged. In our case with $\boldsymbol{m}$ to be 2048, and $\boldsymbol{k}$ to be 300, this reduces the number of adapted parameters: from $\boldsymbol{2\times 2048\times 300}$ to $\boldsymbol{300\times 300}$. This parameter reduction enables us to update all five layers' $\boldsymbol{S_{k\times k}}$, and still has much smaller number of parameters compared with top layer adaptation in Section 4.1: $\boldsymbol{300\times 300\times 5}$ for SVD BN adaptation, and $\boldsymbol{300\times 2048+300\times 6715}$ for top layer adaption, with 6715 to be the output layer size.

4.3. Hybrid Adaptation

This method basically combines the top layer adaptation and SVD BN adaptation. The only difference compared with SVD BN adaptation is that more parameter budget is given to the top layer, to emphasize its importance. Specifically, for the top layer, we update all 3 matrices $\boldsymbol{S_{k\times k}}$, $\boldsymbol{U_{m\times k}}$ and $\boldsymbol{N_{k\times m}}$. For the rest 4 layers, same as SVD BN adaptation, we only update matrix $\boldsymbol{S_{k\times k}}$. The adapted number of parameters for this method is the sum of the above 2 methods.

4.4. Comparison of Adaptation Methods

The character error rate (CER) of GLI-DNN and GLD-DNN by different adaptation methods is shown in Table 2. The CER reduction (CERR) is over the CER of GLI-DNN.

Table 2. *Evaluation of different adaptation methods*

Province	GLI-DNN	Top Layer		SVD BN		Hybrid	
	CER	CER	CERR	CER	CERR	CER	CERR
Guangdong	14.99	14.58	2.7%	14.44	3.7%	14.37	4.1%
Beijing	14.6	14.42	1.3%	14.17	3.0%	14.15	3.1%
Shandong	14.21	13.51	4.9%	13.33	6.2%	13.38	5.8%
Jiangsu	12.98	12.63	2.7%	12.38	4.6%	12.43	4.2%
Zhejiang	14.75	14.47	1.9%	14.07	4.6%	14.11	4.5%
Hebei	13.44	13.03	3.1%	12.88	4.2%	12.81	4.8%
Sichuan	13.81	13.12	5.0%	13.07	5.4%	13.13	4.9%
Shanghai	13.37	12.79	4.3%	12.74	4.7%	12.69	5.1%
Hubei	13.71	13.52	1.4%	13.57	1.0%	13.53	1.3%
Hunan	14.43	13.98	3.1%	13.57	6.0%	13.67	5.3%
Tianjin	14.39	13.74	4.5%	13.99	2.8%	13.87	3.6%
Liaoning	12.08	12.04	0.3%	11.21	7.2%	11.15	7.7%
All	14.25	13.88	2.6%	13.68	4.0%	13.66	4.2%

SVD BN adaptation consistently outperforms top layer adaptation, which indicates that top layer alone is not sufficient to capture all the information in geo-location. Indeed, geo-location contains richer information than accent. For example, people from the same province tend to buy similar devices. This low-level device/channel information is known to be better captured by layers near input. Hybrid adaptation is slightly better than SVD BN, but with much more adapted parameters. As a tradeoff between accuracy and model size, we choose SVD BN adaptation. By this method, deploying 12 provinces' GLD-DNNs only requires 50% model size increase over baseline GLI-DNN.

5. DATA CLUSTERING

In practice, we observe that the users from some provinces may have similar acoustic characteristics, esp. for the provinces that are close to each other in geo-location. This section studies the data clustering to further solve the data sparsity issue for GLD-DNN. Both knowledge and data driven methods are tried.

5.1. Clustering by Accent Region

Linguistics divide China into several accent regions. Since geo-location and accent are well correlated, we borrow the accent region definition to divide our 12 provinces into 4 disjoint accent regions in Table 3. As a result, the number of GLD-DNNs is reduced from 12 to 4.

Table 3. *The division by accent regions*

Accent Region	Provinces
Xiang	Hunan
Cantonese	Guangdong
Wu	Jiangsu, Zhejiang, Shanghai
Northern	Beijing, Shandong, Hebei, Sichuan, Hubei, Tianjin, Liaoning

5.2. Clustering by Cross Test Result

We propose a very simple cross-test result driven method to cluster the training data. This method assumes we have already trained the baseline GLD-DNNs, one per province.

To find which other provinces' data is helpful to train GLD-DNN of province *A*, we test all other 11 provinces' GLD-DNNs on the test data *A* of province *A*. The 11 provinces are sorted based on its test accuracy on set *A* in decreasing order. The top *n* (usually one or two) provinces' data is considered to be helpful for building province *A*'s model, and will be combined with *A*'s own data to update the GLD-DNN for province *A*. The choice of the number *n* depends on how many data the province *A* already has, and how good is the cross test accuracy.

This method does not reduce the number of GLD-DNNs. Also, it is not a strict hard-clustering, as the province *C*'s data may be used to train both province *A* and *B*'s models by this method. It is worth mentioning that we also tried hard data clustering with similar data driven technique, but did not get better results than this method.

5.3. Comparison of Clustering Methods

The CERR in Table 4 is over the baseline GLD-DNN (one per province, no clustering). Clustering by accent region turns out to degrade the performance, while clustering by cross test results gives consistent CERR among different provinces. The provinces in the table are sorted in decreasing order of its training data size. It is clear to see that the cross-test clustering method helps more on low resource provinces.

The last column of the table shows the clustered GLD-DNN error reduction over the baseline GLI-DNN, with an overall error reduction of 4.8%. Better error reduction is found in low resource provinces (e.g., 8% for Sichuan, 9% for Hunan, and 9.1% for Liaoning). Since the baseline GLI-DNN is a strong production model, and the GLD-DNN does not require more train data and is also cheap to train and deploy, we consider this as a nice gain.

Table 4. *Evaluation of different clustering methods*

Province	GLD-DNN	Clustering by Accent Region		Clustering by Cross Test Result		
	CER	CER	CERR	CER	CERR	CERR over GLI-DNN
Guangdong	14.44	14.44	0.0%	14.44	0.0%	3.7%
Beijing	14.17	14.43	-1.8%	14.17	0.0%	3.0%
Shandong	13.33	13.66	-2.5%	13.12	1.6%	7.7%
Jiangsu	12.38	12.58	-1.6%	12.38	0.0%	4.6%
Zhejiang	14.07	14.28	-1.5%	14.07	0.0%	4.6%
Hebei	12.88	12.95	-0.5%	12.64	1.9%	6.0%
Sichuan	13.07	13.31	-1.8%	12.7	2.8%	8.0%
Shanghai	12.74	12.7	0.3%	12.7	0.3%	5.0%
Hubei	13.57	13.94	-2.7%	13.02	4.1%	5.0%
Hunan	13.57	13.57	0.0%	13.13	3.2%	9.0%
Tianjin	13.99	13.78	1.5%	13.63	2.6%	5.3%
Liaoning	11.21	11.86	-5.8%	10.98	2.1%	9.1%
All	13.68	13.84	-1.2%	13.57	0.8%	4.8%

6. IMPACT ON ACCENT RECOGNITION

To further verify the relationship between GLD-DNN and accent, we collected some heavy accent Guangdong test data, and evaluated the models. This is a small test set with 465 utterances (the number of characters is 3066).

Table 5 shows that Guangdong GLD-DNN could get 8% CERR on heavy accent data. The gain on this set is even larger than that in Guangdong province data (3.7% CERR in Table 4). One difference between the two sets is that this data is heavy accent data, and the previous Guangdong province data is randomly sampled live data, with various level of accent. It seems to suggest that the GLD-DNN is more pronounced for heavy accent users with bad WER. However, since the test set is small, we are caution to make the conclusion. Collecting more and larger accent data sets on different provinces is needed to further confirm the findings.

Table 5. *Evaluation on Guangdong heavy accent data*

Test Set	GLI-DNN	GLD-DNN	
	CER	CER	CERR
Guangdong Heavy Accent	28.08	25.86	8%

7. RELIABILITY OF GEO-LOCATION SIGNAL

One common worry is the reliability of GPS inferred geo-location signal. Since a user does not always stay in the same province, this signal will change and could be noisy.

We argue that as long as the region is large (in our case, province), most of the time, GPS location represents the place people live. Our internal data analysis reveals that: among all the queries of a specific user, 90% of them occur in the same province. In other words, at most 10% of the data is noisy. Such a small proportion of outliers could be well handled by DNN, so it won't hurt much for model training.

However, the situation maybe more serious in decoding. For example, when a Beijing user travels to Shanghai, he will end up using the Shanghai model to recognize his voice. To quantify the impact, we conduct a cross test. Specifically, for each province's test data, we test it using all other 11 provinces' GLD-DNNs. The recognition error of test set *A* with GLD-DNN *B* mimics the error a user from province *A* will get, when he travels to province *B*. If this error is 3% relative higher than that tested by GLI-DNN, we consider it to be a serious degradation. Our results show that: among all cross-test $\mathbf{12 \times 11}$ pairs, only 7 pairs get serious degradation, amounting to a ratio of 5%. Consider together with the fact that only 10% of the time, people are in travel, the overall degradation chance is estimated to be only 5/1000. Thus, the impact is small, and the GPS signal is considered to be reliable.

8. CONCLUSIONS & FUTURE WORK

We propose to build geo-location dependent DNN for ASR, where the geo-location signal is inferred from the user's GPS location. The main contributions of this paper are two folds: (1) the novel use of GPS inferred geo-location signal for acoustic modeling, and show the reliability/feasibility of the GPS inferred signal (2) low cost solution to tackle high train/deployment cost, large model size, and data sparsity, thus make it practical for production models.

The idea of GLD-DNN could also be applied to other languages, with a different granularity of geo-location signal. For example, our colleagues have recently used GPS inferred country information to select the Indian users' data from the global English live data traffic. The selected data is used to adapt the native English model to an Indian English model. When evaluating on the Indian users' test data, the Indian English model results in around 30% relative word error rate reduction, compared with the native English model.

Finally, for some applications where user is willing to provide his home information, we could directly use it as the geo-location signal. Since the home signal is provided or confirmed by the users, it is supposed to be more reliable than GPS inferred signal.

9. REFERENCES

[1] Dahl, George E., et al. "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition." Audio, Speech, and Language Processing, IEEE Transactions on 20.1 (2012): 30-42.

[2] Huang, Yan, et al. "Multi-Accent Deep Neural Network Acoustic Model with Accent-Specific Top Layer Using the KLD-Regularized Model Adaptation." Fifteenth Annual Conference of the International Speech Communication Association. 2014.

[3] Chen, Mingming, et al. "Improving Deep Neural Networks Based Multi-Accent Mandarin Speech Recognition Using I-Vectors and Accent-Specific Top Layer." Sixteenth Annual Conference of the International Speech Communication Association. 2015.

[4] Xue, Jian, et al. "Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network." Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE, 2014.

[5] Yu, Dong, et al. "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition." Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013.

[6] Seltzer, Michael L., Dong Yu, and Yongqiang Wang. "An investigation of deep neural networks for noise robust speech recognition." Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013.

[7] Chelba, Ciprian, Xuedong Zhang, and Keith Hall. "Geo-location for Voice Search Language Modeling." Sixteenth Annual Conference of the International Speech Communication Association. 2015.

[8] Huang, Chao, et al. "Accent modeling based on pronunciation dictionary adaptation for large vocabulary Mandarin speech recognition." INTERSPEECH. 2000.

[9] Chen, Tao, et al. "Automatic accent identification using Gaussian mixture models." Automatic Speech Recognition and Understanding, 2001. ASRU'01. IEEE Workshop on. IEEE, 2001.

[10] Su, Hang, et al. "Error back propagation for sequence training of context-dependent deep networks for conversational speech transcription." Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013.

[11] Veselý, Karel, et al. "Sequence-discriminative training of deep neural networks." INTERSPEECH. 2013.

[12] Xue, Jian, Jinyu Li, and Yifan Gong. "Restructuring of deep neural network acoustic models with singular value decomposition." Interspeech. 2013.

[13] Golub, Gene H., and Christian Reinsch. "Singular value decomposition and least squares solutions." Numerische mathematik 14.5 (1970): 403-420.

[14] Huang, Yan, and Yifan Gong. "Regularized Sequence-Level Deep Neural Network Model Adaptation." Sixteenth Annual Conference of the International Speech Communication Association. 2015.