SVD-based Universal DNN Modeling for Multiple Scenarios

Changliang Liu [1], *Jinyu Li*[2], *Yifan Gong*[2]

[1]Microsoft Search Technology Center Asia, Beijing, China
[2]Microsoft Corporation, One Microsoft Way, Redmond, WA 98052
{chanliu; jinyli; ygong}@microsoft.com

Abstract

Speech recognition scenarios (aka tasks) differ from each other in acoustic transducers, acoustic environments, and speaking style etc. Building one acoustic model per task is one common practice in industry. However, this limits training data sharing across scenarios thus may not give highest possible accuracy. Based on the deep neural network (DNN) technique, we propose to build a universal acoustic model for all scenarios by utilizing all the data together. Two advantages are obtained: 1) leveraging more data sources to improve the recognition accuracy, 2) reducing substantially service deployment and maintenance costs. We achieve this by extending the singular value decomposition (SVD) structure of DNNs. The data from all scenarios are used to first train a single SVD-DNN model. Then a series of scenario-dependent linear square matrices are added on top of each SVD layer and updated with only scenario-related data. At the recognition time, a flag indicates different scenarios and guides the recognizer to use the scenario-dependent matrices together with the scenario-independent matrices in the universal DNN for acoustic score evaluation. In our experiments on Microsoft Winphone/Skype/Xbox data sets, the universal DNN model is better than traditional trained isolated models, with up to 15.5% relative word error rate reduction.

Index Terms: speech recognition, singular value decomposition (SVD), universal DNN model

1. Introduction

Commercial speech recognition services typically support a number of products for different application scenarios. For example, Microsoft provides voice search and short message dictation on Windows phone, speech to speech translation on Skype, and marketplace voice search on Xbox. Google has voice search on Andriod and speech transcription on Youtube. Speech signals from different scenarios may be subject to different channels, such as far-field or near field recording; different speaking styles, such as reading or spontaneous style; different devices, such as mobile phone, desktop or Xbox Kinect, and so on. Traditional wisdom to deal with different scenarios is to build different acoustic models using scenario-dependent data [1][2].

The choice of building scenario-dependent acoustic model with only scenario-specific data comes from the GMM era, in which multi-style training [3] may not be a good choice for pooling large amount of training data from lots of aforementioned different scenarios together. One reason is that the GMM model obtained from multi-style training exhibits very broad distribution because it needs to cover all the acoustic environments, speaking styles, and recording devices, etc. However, this situation changes with the recent success of deep neural network (DNN) [4][5][6][7][8]. As shown in [9][10], the DNN training provides a layer-by-layer feature extraction strategy that automatically derives powerful features from heterogeneous data for senone classification. Therefore, it is time now to examine whether a universal acoustic model can be built by pooling the training data from all scenarios.

In this paper, we propose an approach to build a single acoustic model, universal across all scenarios to fully utilize the training data from all scenarios. Most of the parameters in this universal acoustic model are shared across all scenarios. Meanwhile, there are some small scenario-dependent parameter sets. The proposed method is based on the DNN structure derived from singular value decomposition (SVD) [11][12]. First, a DNN model is trained with the training data from all scenarios. Then, we do SVD reconstruction on this model. After that, scenario-dependent linear square matrices are inserted on top of each of the SVD layers. Finally, we fine tune the scenario dependent square matrices using the scenario-related data while fixing the rest parameters of the model. At the recognition time, the decoder will decide which square matrices to use given the scenario ID along with the input audio data.

In the remaining of this paper, we will first introduce the SVD-based DNN technique briefly in Section 2. The details of the universal DNN modeling are described in Section 3. In Section 4, several experiments and results are presented. Finally, the conclusion and some discussion can be found in the Section 5.

2. SVD-based DNN

A DNN is a feed-forward artificial neural network with multiple hidden layers. Usually the network is fully connected between adjacent layers. DNNs provide significant accuracy improvements over Gaussian mixture models (GMMs) as acoustic models. However, they require much more parameters than traditional GMM systems, incurring very large computational cost during online evaluation. Utilizing the low-rank property of DNN matrices [13], SVD-based DNN is proposed to reduce the DNN model size as largely as 80% while maintaining the accuracy improvements in [11]. In this method, an SVD on the weight matrices is applied to the DNN, and then the model is restructured based on the inherent low-rank property of the original matrices. After restructuring, the DNN model size is significantly reduced with negligible accuracy loss.

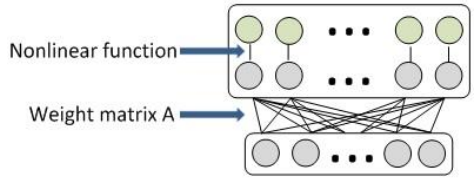
Figure 1 shows how SVD is applied to a standard DNN. For a $m \times m$ weight matrix A, if we apply SVD on it, we get

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

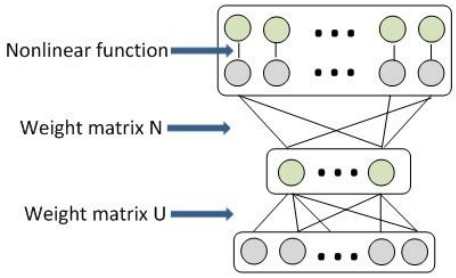
where Σ is a diagonal matrix with A's singular values on the diagonal in the decreasing order. Since A is a low-rank matrix, a large part of A's singular values should be very small. Assume we only keep k biggest singular values of A, we can rewrite formula (1) as

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

where $N_{k\times m} = \Sigma_{k\times k}V^T_{k\times m}$. In this way the matrix A is decomposed into two smaller matrices U and N. Figure 1(a) shows a layer in original DNN with weight matrix $A_{m\times m}$. After SVD reconstruction, a bottleneck *SVD layer* is inserted between two large hidden layers, shown in Figure 1(b). The weight matrices becomes $U_{m\times k}$ and $N_{k\times m}$. Usually, k is much smaller than m. Therefore the number of parameters is significantly reduced.



(a) One layer in an original DNN model



b) Two corresponding layers in a new DNN model

Figure 1: Model conversion in a restructured DNN by SVD

3. SVD-based Universal DNN Modeling

To build a universal DNN model for several scenarios, the most straightforward way is to do multi-style training with all kinds of data. However, given enough training data, the accuracy of such a model is usually worse than the isolated-trained models because the model discriminating ability is hurt by the confusion of different channels, different speaking styles, etc. A better design to model multiple scenarios is to factorize the DNN parameters, such that most parameters in this universal model are used to characterize the whole training data while some scenario-dependent parameters carry the scenario-related information. In this way, both the benefits of data sharing and modeling sharpness can be achieved.

To build such a universal DNN model, we leverage the SVD bottleneck layer as Figure 2. An additional linear layer is added on top of the SVD layer as

$$U_{m\times k}N_{k\times m} = U_{m\times k}S_{k\times k}N_{k\times m}$$

where $S_{k\times k}$ is a square matrix which is initialized to be identity matrix $I_{k\times k}$.

To model the scenario-dependent information, there will be a set of linear square matrices, for example, S_{winp}, S_{skype}, S_{xbox}. They are put together in parallel on top of the SVD layer. During scenario-dependent modeling, S_{winp}, S_{skype}, S_{xbox} are updated separately by its related data. The number of parameters for matrix S is k^2. k is usually very small and these scenario-dependent matrices are much smaller than the scenario-independent matrices (N and U) in the DNN model.

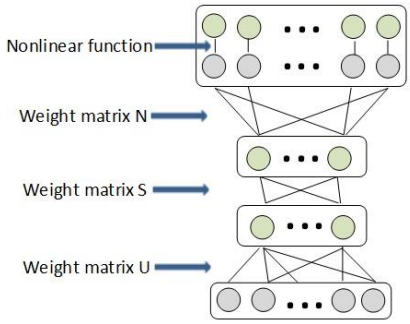


Figure 2: One layer with a square matrix on top of SVD

The full network is shown in Figure 3, where scenario-dependent linear layers are added on top of each SVD layer, although we only explicitly plotted one scenario-dependent linear layer. These scenario-dependent layers are expected to model the scenario-specific information, like acoustic channel, speaking style, etc. Besides the scenario-dependent layers, other parameters are shared across all scenarios.

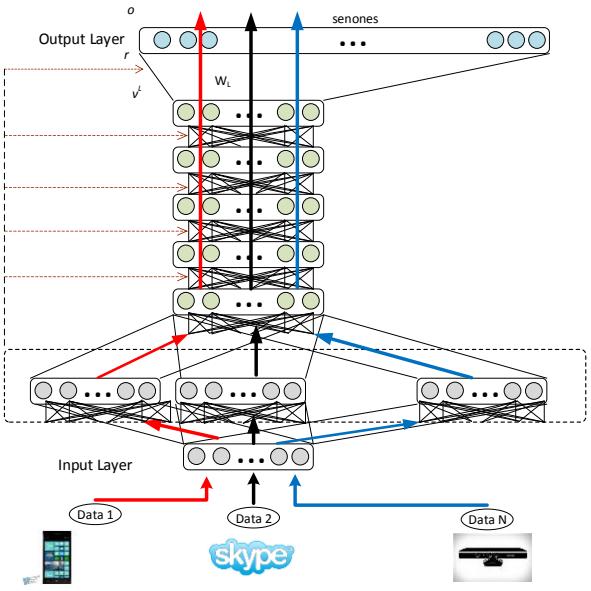


Figure 3: The full network of a universal DNN model

The training procedure for the universal DNN is as follows.

- Train a full size DNN with the data from all scenarios;
- Apply SVD model reconstruction and then fine tune the SVD-based DNN;
- Add scenario-dependent layers into the SVD-based DNN. All the scenario-dependent matrices are initialized as $I_{k\times k}$.

- Update the scenario-dependent matrices using the scenario-related data while fixing the shared layers.

At the recognition time, according to its scenario ID, the input audio will be passed into different scenario-dependent matrices as well as into the shared scenario-independent matrices.

There are three advantages of this method compared with the traditional one. Firstly, with data and parameter sharing, the model benefits from all kinds of data with better triphone coverage, resulting in a better accuracy than separately-trained models. Secondly, the small scenario-dependent parameter carries the information of that scenario and makes the whole model actually scenario-dependent while only increasing the model size by a small percentage. Thirdly, a single universal model saves deployment and maintenance cost because we would have to maintain different models for different scenarios otherwise.

4. Experimental Evaluation

The proposed method is evaluated using the data from three Microsoft speech recognition services: voice search and short message dictation on Windows phone (Winp), speech to speech translation on Skype, and market place voice search on Xbox. The language is English in United States. Speech Signals from the above services are different due to acoustic channels, speaking styles, and recording environments. Table 1 shows a brief description of their major difference.

Table 1. *Major difference of Winp, Skype and Xbox audio*

	Channel	Far field talking	Speaking style	Recording environment
Winp	Mobile Phone	No	Prepared	Various
Skype	Desktop, Laptop	No	Spontaneous & conversational	Home/Office
Xbox	Kinect sensor	Yes	Prepared	Home

As training data, we use around 400 hours for each scenario. Therefore, totally we have 1200 hours training audio. In GMM model training, the input feature is MFCC with up to third-order derivatives. The final feature dimension is reduced by heteroscedastic linear discriminant analysis (HLDA) [14]. Each GMM consists of 32 Gaussians and the senone number is 4500. In DNN model training, the input feature is 22-dimension log-filter-bank feature with up to the second-order derivatives. We augment the feature vectors with previous and next 5 frames (5-1-5). The full size DNN has 5 hidden layers with 2048 units for each. The output layer size is 4500.

4.1. Baseline GMM and DNN modeling

In this section, we will talk about the baseline models of this study and discuss why it is possible to build a universal model by DNN. Three types of models were trained in our experiments:

1. Isolated training. Three models are trained for Winp, Skype and Xbox respectively with only the corresponding scenario-specific data.
2. Multi-style training. A single model is trained with the data from all scenarios and is used to recognize utterances from these scenarios.
3. Model update with scenario-related data. In this training, we also get three models which are updated from Multi-style trained model with data from Winp, Skype, and Xbox, respectively. They have exactly the same model structure as the Multi-style trained model.

The trainings were conducted both on GMM and DNN models. The results in terms of word error rates (WERs) are given in Table 2 and Table 3 respectively. From Table 2, we can see that in GMM modeling, the multi-style trained model GB performs much worse on most scenarios than isolated trained model GA. The WER degradation is 18.5% relative on Winp and 16.1% relative on Skype, respectively. Even after model update with scenario-related data, the WRE loss cannot be fully recovered. Model GC is still much worse than model GA on Winp and Skype data. Model GC is nearly the upper bound we can get with the same model structure. This indicates that the universal modeling by GMM is in-effective. However, the situation is very different on DNN modeling, as shown in Table 3.

Table 2. *WERs of baseline GMM models for Winp, Skype and Xbox*

Model	Training	Winp	Skype	Xbox
GA	Isolated training	30.36	35.31	26.69
GB	Multi-style training	35.98	40.20	26.75
GC	+ update with scenario-dependent data	31.93	37.33	25.34

Table 3. *WERs of baseline DNN models for Winp, Skype and Xbox*

Model	Training	Winp	Skype	Xbox
A	Isolated training	22.44	27.12	19.84
B	Multi-style training	22.90	29.30	18.48
C	+ update with scenario-dependent data	20.72	27.04	16.88

From Table 3, we can first see that the DNN model is much better than GMM model. Comparing model A with model GA, the relative WER reduction ranges from 22% to 27% on these three scenarios. In DNN modeling, the gap between isolated trained model A and multi-style trained model B is not as large as in GMM modeling. The significant degradation is only observed on Skype data. But it is still only 8%, much smaller than in GMM. Hence, a DNN really does a better job on normalizing different channel, speaking styles, etc. than a GMM. Comparing model C and A, we can see that after updating with scenario-related data, the gap is totally removed and even the accuracy is significantly better than the separately trained model on Winp and Xbox data. This indicates that universal modeling is effective with DNN modeling. In our study, model A is treated as the baseline while model C is the upper bound of the proposed universal DNN model. Note that although model C gets superior accuracy, we are more interested in a model that has same senone set and most shared parameters across all scenarios.

Such a model can significantly reduce the product deployment cost and is the initiative of this study.

By comparing model B and A, we can also see that the DNN works better on dealing with channel mismatch than speaking style mismatch. Winp and Xbox are from different channels but with similar speaking style: prepared speech. In contrast, Skype is with spontaneous style. In DNNs, the input features from different channels are normalized very well after a series of nonlinear transforms from multiple hidden layers. In this way, both Winp and Xbox can benefit from each other by data sharing. That's why model B performs much better than model A on Xbox data. However, the major difference of prepared speech and spontaneous speech are on pronunciation variations. So, Skype cannot benefit from Winp and Xbox data. This is why model B performs worse on the Skype test set than model A. We expect larger senone set size can partially resolve this issue and the experimental results will be shown in Section 4.3.

4.2. SVD-based universal DNN modeling

In Table 4, model D is generated by applying SVD restructuring to model B with the dimension of SVD layer around 300. In general, it has very similar accuracy as model B. Model E is the proposed universal DNN model. In its training, we just fine tune the scenario-dependent linear square matrices on top of each SVD layer with individual scenario-related data. It outperforms A on both Winp and Xbox test sets significantly, and has similar accuracy on Skype scenario. Compared with model C, the upper bound, the gap is very small. This means that just updating the small linear matrices on top of each SVD layer achieves similar accuracy improvement as updating the full model. There are around 0.5M parameters in scenario-dependent matrices, compared to 8M parameters in scenario-independent matrices. This is the most critical advantage of the proposed universal modeling.

Table 4. *WERs of the universal DNN model with 4500 senones*

Model	Training	Winp	Skype	Xbox
D	SVD reconstruction from model B	22.50	28.47	18.50
E	update model D with scenario-dependent data	21.13	27.23	16.77

These results are consistent with what we expect from the universal DNN modeling in two aspects:

- The universal DNN model benefits from data sharing across different scenarios.
- It's feasible to just use a set of small scenario-dependent linear matrices to carry the scenario-dependent information.

In the universal DNN model, a large number of parameters are shared across scenarios. This is consistent with our intuition. The variability for phonemes in different scenarios is actually small and should also be shared across scenarios in the DNN modeling. On the other hand, there is some scenario-specific phoneme variation in different scenarios. The scenario-dependent linear matrices are just designed to model this kind of difference in a much lower-dimension space. Hence a small number of parameters are good for these scenario-dependent matrices**.**

4.3. Enlarge the senone set

When comparing model E and model A, we can see that both Winp and Xbox models benefit a lot from data and parameter sharing. The only exception is Skype model. As we analyzed in Section 4.1, Skype is mostly different from other two scenarios on speaking style. It is spontaneous speech which has more pronunciation variations. It is difficult to benefit from Winp and Xbox data which are prepared speech. As we expect, the pronunciation variations can be better modeled if we enlarge the senone number.

Table 5 describes the results of the models with 9000 senones. The difference between model H and model E is only the number of senones. Model H gets relative 3.7% WER reduction on the Skype task from model E. There is no benefit on the Winp and Xbox tasks. As expected, spontaneous speech can benefit more from the increase of senone size than prepared speech.

Table 5. *WERs of the universal DNN model with 9000 senones*

Model	Training	Winp	Skype	Xbox
F	Multi-condition training	22.38	27.89	18.33
G	SVD reconstruction from model H	22.08	27.22	18.32
H	Update model G with scenario-dependent data	20.77	26.23	16.90

5. Conclusions

In this paper, we present a method to build a universal model to host all kinds of scenarios in speech recognition services. This is nearly an impossible task in GMM model era. From our observation, the GMM model performs very badly on dealing with multiple channels or speaking styles. Due to the linear-by-linear normalization power in DNNs, it becomes possible to build such a universal model with DNN modeling. To achieve this, a series of scenario-dependent linear square matrices are inserted on top of SVD layers. Other layers are all shared across scenarios. In training, only the scenario-dependent matrices are trained with the scenario-related data separately, while other layers are trained with the combined data in a multi-style training way. The number of parameters of the scenario-dependent matrices is much smaller than that of the scenario-independent matrices. The universal model performs much better than traditional isolated-trained models. The relative WER reduction is 5.8% and 15.5% on Microsoft Windows phone and Xbox data sets, respectively. We also tried to enlarge the output layer size in the universal modeling. After doubling the output layer of the universal DNN, significant WER reduction is got on spontaneous style speech (Skype speech to speech translation).

Currently, the scenario-dependent matrices are updated after the scenario-independent matrices have been trained. In the future, we will jointly train both the scenario-dependent matrices and the scenario-dependent matrices together. We are also applying this method to the product-scale training with tens of thousands of audio data.

6. References

[1] N. Jaitly, P. Nguyen, A. W. Senior, V. Vanhoucke, "Application of pretrained deep neural networks to large vocabulary speech recognition," *Proc. Interspeech*, 2012.

[2] J.-T. Huang, J. Li, and Y. Gong, "An analysis of convolutional neural networks for speech recognition," *Proc. ICASSP*, 2015.

[3] R. Lippmann, E. Martin, and D. Paul. Multi-style training for robust isolated-word speech recognition. *Proc. ICASSP*, pages 705–708, 1987.

[4] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," *Proc. Interspeech*, pp. 437-440, 2011.

[5] T. N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, A.-r. Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," *Proc. ASRU*, pp. 30-35, 2011.

[6] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Large vocabulary continuous speech recognition with context-dependent DBN-HMMs," *Proc. ICASSP*, pp. 4688-4691, 2011.

[7] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio Speech and Language Process.*, vol. 20, no. 1, pp. 14-22, Jan. 2012.

[8] L. Deng, J. Li, J.-T. Huang et. al. "Recent advances in deep learning for speech research at Microsoft," *Proc. ICASSP*, 2013.

[9] J. Li, D. Yu, J. T. Huang, and Y. Gong. "Improving wideband speech recognition using mixed-bandwidth training data in CD-DNN-HMM," *Proc. IEEE Spoken Language Technology Workshop*, pages 131–136, 2012.

[10] D. Yu, M. Seltzer, J. Li, J.-T. Huang, and F. Seide, "Feature learning in deep neural networks-studies on speech recognition tasks," *Proc. International Conference on Learning Representations*, 2013.

[11] J. Xue, J. Li, and Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition," *Proc. Interspeech*, 2013.

[12] J. Xue, J. Li, and Y. Gong, "Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network," *Proc. ICASSP*, 2014.

[13] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," *Proc. ICASSP*, pp. 6655-6659, 2013.

[14] N. Kumar and A. G. Andreou. Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition. Speech Communication, 26(4):283–297, 1998.