SINGULAR VALUE DECOMPOSITION BASED LOW-FOOTPRINT SPEAKER ADAPTATION AND PERSONALIZATION FOR DEEP NEURAL NETWORK

Jian Xue, Jinyu Li, Dong Yu, Mike Seltzer, and Yifan Gong

Microsoft Corporation, One Microsoft Way, Redmond, WA 98052, USA

{jianxue; jinyli; dongyu; mseltzer; ygong}@microsoft.com

ABSTRACT

The large number of parameters in deep neural networks (DNN) for automatic speech recognition (ASR) makes speaker adaptation very challenging. It also limits the use of speaker personalization due to the huge storage cost in large-scale deployments. In this paper we address DNN adaptation and personalization issues by presenting two methods based on the singular value decomposition (SVD). The first method uses an SVD to replace the weight matrix of a speaker independent DNN by the product of two low rank matrices. Adaptation is then performed by updating a square matrix inserted between the two low-rank matrices. In the second method, we adapt the full weight matrix but only store the delta matrix – the difference between the original and adapted weight matrices. We decrease the footprint of the adapted model by storing a reduced rank version of the delta matrix via an SVD. The proposed methods were evaluated on short message dictation task. Experimental results show that we can obtain similar accuracy improvements as the previously proposed Kullback-Leibler divergence (KLD) regularized method with far fewer parameters, which only requires 0.89% of the original model storage.

Index Terms— deep neural network, speaker adaptation, speaker personalization, singular value decomposition

1. INTRODUCTION

Recent progress in deep learning has attracted a lot of interest in automatic speech recognition (ASR) [1][2][3][4][5][6]. The discovery of the strong modeling capabilities of deep neural networks (DNN) and the availability of high-speed hardware has made it feasible to train huge networks with tens of millions of parameters. In the framework of context-dependent DNN hidden-Markov-models (CD-DNN-HMM) [1], the conventional Gaussian Mixture Model (GMM) is replaced by a DNN to evaluate the senone log-likelihood. Besides CD-DNN-HMMs, a DNN can also be used to provide the bottle-neck features for a GMM-HMM system [7][8]. In both applications of a DNN in ASR, significant accuracy improvement was achieved.

However, the outstanding performance of CD-DNN-HMM requires huge number of parameters, which makes adaptation very challenging, especially with limited adaptation data. Several methods for DNN adaptation have previously been proposed [9]. For example, affine transformations are applied to the inputs and outputs of a neural network, where a linear layer is added before the original input layer and after the output layer [10]. Another approach applies a linear transformation to the activations of the internal hidden layers [11]. The shape of the activation function was changed to better fit the speaker-specific features in [12]. Recently, a regularized adaptation technique was proposed, which adapts the model conservatively by forcing the senone distributions estimated from the adapted model to be close to that estimated from the speaker independent (SI) model [13]. In [14], a separate small size of speaker code is learned for each individual speaker while the large adaptation network is learned from the whole training set. Factorized adaptation is studied in [15] by using limited number of parameters to take into account of the underlying factors that contribute to the distorted speech signal.

Besides DNN adaptation, speaker personalization with a DNN model creates a storage issue. It is not practical to store an entire DNN model for each individual speaker during deployment due to the high storage cost. None of the previous methods addressed this issue. In this paper we present two low-footprint DNN adaptation and personalization methods based on a singular value decomposition (SVD). The first method, called SVD bottleneck adaptation, uses a recently proposed format for the SI model where the weight matrix in each layer is approximated as the product of two low-rank matrices [16] [17]. Between the two low-rank matrices in each layer, we insert an additional square matrix, initialized to identity. We only update and store these matrices during adaptation. These small matrices have much fewer parameters compared to the original DNN model, which significantly reduces the footprint of personalized DNN model. In the second method, called SVD delta compression, we adapt the weight matrix in each layer but only store the difference between the adapted and SI weights, which we refer to as the delta matrix. We assume these delta matrices have low rank because the adapted model doesn't deviate significantly from the SI model. Thus, we can reduce the footprint of the delta matrices by using an SVD and only

storing the decomposed matrices, which have a much smaller number of parameters. In both methods we use the Kullback-Leibler divergence (KLD) regularized optimization criterion [13] during adaptation, which prevents overfitting to the adaptation data by preventing the adapted model from staying too far from the SI model.

The rest of the paper is organized as follows. Section 2 briefly reviews the KLD regularized algorithm [13]. Section 3 describes our new SVD-based DNN adaptation and personalization methods, including SVD bottleneck adaptation and delta compression. Experimental results are presented in Section 4 to show the effectiveness of the methods. We conclude our work in Section 5.

2. KLD REGULARIZED ADAPTATION

The parameters of DNNs are typically trained to maximize the negative cross entropy

$$\bar{D} = \frac{1}{N}\sum_{t=1}^{N} D(x_t) = \frac{1}{N}\sum_{t=1}^{N}\sum_{y_t=1}^{S} \tilde{p}(y_t|x_t)\log p(y_t|x_t) \quad (1)$$

where x_t is the input feature vector, y_t is the output at time t, S is the total number of senones, N is the number of samples in the training set, and $\tilde{p}(y_t|x_t)$ is the target probability.

A straightforward approach to adapt a DNN is to adjust the DNN parameters with the adaptation data, starting from the SI model. However, doing so may destroy previously learned information and over-fit the model to the adaptation data, especially when the adaptation set is small. To prevent over-training, model updating needs to be done conservatively. A regularized adaptation method was proposed to address this issue [13]. The idea is that the posterior senone distribution estimated from the adapted model should not deviate too far from the one estimated with the SI model. The deviation is measured by KLD. By adding this divergence as a regularization term to the objective function used in regular DNN training and adaption, like cross entropy, and removing the terms unrelated to the model parameters, we get a new regularized optimization criterion

$$\widehat{D} = \frac{1}{N}\sum_{t=1}^{N}\sum_{y_t=1}^{S} \hat{p}(y_t|x_t) log p(y_t|x_t) \quad (2)$$

where $\hat{p}(y_t|x_t)$ equals $(1-\rho)\tilde{p}(y_t|x_t) + \rho p^{SI}(y_t|x_t)$, ρ is the regularization weight, and $p^{SI}(y_t|x_t)$ is the posterior probability estimated from the SI model. By comparing formulae (1) and (2) we can see that applying KLD regularization is equivalent to changing the target probability distribution to $\hat{p}(y_t|x_t)$, which is a linear interpolation of the distribution estimated from the SI model and the ground truth alignment of the adaptation data.

The normal back-propagation algorithm can be directly applied to adapt the model. The only difference is the error signal at the output layer, which is now defined based on $\hat{p}(y_t|x_t)$. More details can be found in [13].

3. METHODS DESCRIPTION

A DNN used for ASR system typically has 5-8 hidden layers, and each layer consists of a few thousand units. With the same amount of training data, the DNN model usually has 2 to 10 times more parameters than the traditional GMM model. The huge number of parameters in DNN presents big challenges for speaker adaptation and personalization. We recently presented a SVD based DNN model restructuring method in [17]. With this method we can convert the original large full-rank DNN model to a much smaller low-rank model without loss of accuracy. Most of the previously proposed DNN adaptation methods can also apply this low-rank DNN model. But all of these methods still need to update and store a large amount of parameters. Although in some methods only some of the DNN layers need to be adjusted, the updated portion is still large due to the high dimensionality of each layer.

Here we presented two SVD based DNN adaptation and personalization methods. The goal of both of these methods is to be able to adapt and store a small-footprint model representation that still obtains large accuracy improvements, which will reduce the cost in deployment.

3.1. SVD Bottleneck adaptation

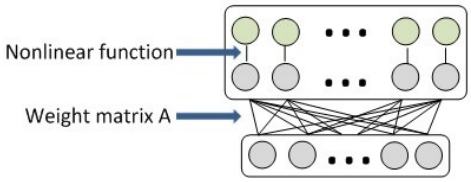
As described in [17], for a $m \times n$ (m ≥ n) weight matrix A in DNN, apply SVD on it, we get

$$A_{m\times n} = U_{m\times n}\sum_{n\times n} V^T_{n\times n} \quad (3)$$

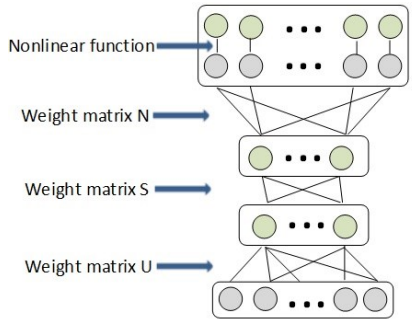
where $\sum$ is a diagonal matrix with A's singular values on the diagonal. If A is a sparse matrix, the number of A's non-zero singular values will be k, where $k << n$. Then we can rewrite (3) as

$$A_{m\times n} \approx U_{m\times k}\sum_{k\times k} V^T_{n\times k} = U_{m\times k} N_{k\times n} \quad (4)$$

Applying the decomposition to the DNN model, it acts as if a linear bottleneck layer with much fewer units has been added between the original layers.



a) One layer in original DNN model



b) Three corresponding layers in new DNN model

Figure 1 Model conversion in low-rank DNN

To do SVD bottleneck adaptation, we add an additional linear layer to the bottleneck linear layer with k units. Applying this change to (4), we get

$$A_{m\times n} \approx U_{m\times k}N_{k\times n} = U_{m\times k}S_{k\times k}N_{k\times n} \quad (5)$$

where $S_{k\times k}$ is initialized to be identity matrix $I_{k\times k}$. Figure 1 describes how we modify the model structure.

We use the square matrix S to store the speaker information. In the SI model, S is set to the identity matrix so that (4) and (5) are identical. During adaptation we only update the square matrix S. The number of parameters for matrix S is k^2, which is much smaller than the original number $m \times n$.

The advantage of this approach is that only a couple of small matrices need to be updated for each speaker. This dramatically reduces the deployment cost for speaker personalization. Also, it potentially reduces the required amount of adaptation data for each new speaker.

3.2. SVD delta compression

Usually the adaptation set is small for each individual speaker, so the adapted model should not deviate far away from the SI model. This is the foundation of regularized adaptation. From this point of view, we believe that the delta matrices between the adapted matrices and SI matrices should have low ranks. If we apply SVD decomposition on the delta matrices and only keep a small number of non-zero singular values for each one, the delta matrices should be largely unchanged. So we can convert each delta matrix into the product of two low-rank matrices in the same way as before

$$D_{m\times n} = A_{m\times n}^{adapted} - A_{m\times n}^{SI} = U_{m\times n}\sum_{n\times n}V_{n\times n}^{T} \approx U_{m\times k}\sum_{k\times k}V_{k\times n}^{T} = U_{m\times k}N_{k\times n} \quad (6)$$

We only need to store matrices U and N for each individual speaker. The total number of parameters changes from mn to $(m+n)k$. During runtime we multiply U and N, add the result back to $A_{m\times n}^{SI}$ to resynthesize the adapted model.

3.3 Combination of SVD bottleneck adaptation and delta compression

The two methods introduced above address the DNN adaptation and personalization issues from different perspectives. We can combine them together if the delta matrices for the square matrices in low-rank DNN model still have the low-rank property. We investigate this in section 4.

4. EXPERIMENTS

The proposed methods were evaluated on a short message dictation (SMD) task. The baseline SI models were trained with 300hours voice search and SMD data. The input feature to CD-DNN-HMM system is a 24-dimension mean-normalized log-filter bank feature with up to second-order derivatives and a context window of 11 frames, forming a vector of 792-dimension (72×11) input. On top of the input layer there are 5 hidden layers with 2048 units for each. The output layer has a dimension of 5976. To apply SVD bottleneck adaptation method, we first convert the full-rank DNN model to low-rank model by doing SVD on the matrices above all the hidden layers and keeping 40% of total singular values. Then the numbers of units for the linear layers after SVD are 208, 184, 176, 200, and 344 accordingly, from bottom to top. We then retrained the low-rank model and obtained comparable accuracy to the full-rank model. More details of SVD based low-rank DNN model training can be found in [17].

The evaluation was conducted on data from 9 speakers. The total number of test set words is 26433. There is no overlap among training and testing data. The first several rows of Table 1 summarize the results obtained with SI models and with KLD regularized adaptation on the low-rank SI model. The regularization weight ρ is set to 0.5, and the WER shown is averaged on 9 speakers.

Table 1. Results of SI modes, KLD adaptation method, and bottleneck KLD adaptation method

Acoustic model	WER	Number of parameters
Full-rank SI model	25.21%	30M
Low-rank SI model	25.12%	7.4M
Supervised KLD adaptation with 5 utterances	24.30%	7.4M
Supervised KLD adaptation with 100 utterances	20.51%	7.4M
Supervised SVD bottleneck adaptation with 5 utterances	24.23%	266K
Supervised SVD bottleneck adaptation with 100 utterances	19.95%	266K

From Table 1 we observe that supervised KLD regularized adaptation obtains 18.4% and 3.3% relative WER reduction with 100 and 5 utterances of adaptation data.

4.1. Results of SVD bottleneck adaptation

The last two rows of Table 1 show us the results obtained with bottleneck KLD adaptation method, where number of parameters only includes the adapted portion of the models. The supervised SVD bottleneck adaptation obtains 20.6% and 3.5% relative WER reduction with 100 and 5 utterances of adaptation data, respectively, which is a little better than the results obtained with standard KLD regularized adaptation method. However, the number of parameters in the adapted portion reduces to 266K, only 0.89% of the parameters updated in the standard KLD regularized method.

4.2. Results of SVD delta compression

The experiments of SVD delta compression were first conducted on the full-rank model. We do supervised KLD regularized adaptation on full-rank SI model, and then apply SVD delta compression to reduce the storage footprint of the adapted model for each speaker. Table 2 shows the results using SVD delta compression. The values in the first column of last four rows indicate the number of singular values that were kept for each delta matrix, from bottom to top. Number of parameters in the last column shows the stored portion of the model.

Table 2. Results of SVD delta compression with supervised KLD adaptation method

Acoustic model	WER		Number of parameters
	5 utter.	100 utter.	
Supervised KLD on full-rank SI model	24.21%	20.40%	30M
256-512-512-512-512-512	24.21%	20.40%	13.2M
128-256-256-256-256-256	24.20%	20.50%	6.6M
64-128-128-128-128-128	24.22%	20.62%	3.3M
32-64-64-64-64-64	24.22%	20.70%	1.7M

From Table 2 we can see that, in the 5 utterances case we can reduce number of parameters to 1.7M without losing accuracy, which is only 5.6% of the original model size. But if we have 100 utterances of adaptation data, we will start to have more than 1% relative accuracy loss if we reduce the number of parameters to 3.3M. This is because the adapted model deviates further away from SI model with more adaptation data. So the delta matrices will have higher rank and we cannot compress them as aggressively.

4.3. Results of combination

As we indicated in section 3.3, SVD bottleneck adaptation and delta compression methods address the DNN adaptation and personalization issues from different perspectives, so we try to combine them to obtain further improvement. We applied SVD delta compression to the adapted models obtained with SVD bottleneck adaptation and summarized the results in Table 3, where number of parameters only includes the stored portion of the models.

From Table 3 we observe that, combining SVD bottleneck adaptation and delta compression in the 5 utterances case can further reduce number of parameters to 71K without losing accuracy. But for 100 utterances case we cannot combine two methods, since we will at least have more than 2% relative accuracy loss if we do so.

Table 3. Results of combining SVD bottleneck adaptation and delta compression

Acoustic model	WER		Number of parameters
	5 utter.	100 utter.	
Supervised KLD on low-rank SI model	24.23%	19.95%	266K
32-32-32-32-32	24.25%	22.14%	71K
64-64-64-64-64	24.26%	21.42%	142K
96-96-96-96-96	24.21%	20.39%	214K

5. CONCLUSION

In this paper we present two SVD based methods to address the DNN adaptation and personalization issues which requires only small amount of parameters for each speaker. The first method inserts an additional linear layer above each original linear layer in SVD based low-rank DNN model with a small identity matrix, and only updates the inserted small matrices during adaptation. In the second method, we use an SVD to compress and store the delta matrices between the adapted and SI models. We evaluated our methods on a short message dictation task and reduce the speaker-specific parameters to only 0.89% of the original model size without a loss of accuracy. We further investigate the combination of the two proposed methods and we can further reduce the model size if only few adaptation data are available. With more adaptation data performance suffers since the adapted model deviates far away from the SI model on the adapted matrices, violating the low rank assumption of the SVD delta compression method.

6. REFERENCES

[1] G.E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Trans. Audio Speech and Language Process.*, vol. 20, no. 1, pp. 30-42, 2012.

[2] N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke, "An application of pretrained deep neural networks to large vocabulary conversational speech recognition,"*Proc. Interspeech*, 2012.

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

[4] 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.

[5] 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.

[6] 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.

[7] D. Yu, and M. L. Seltzer, "Improved bottleneck features using pretrained deep neural networks," *Proc. Interspeech*, pp. 237-240, 2011.

[8] T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Auto-encoder bottleneck features using deep belief networks," *Proc. ICASSP*, pp. 4153-4156, 2012.

[9] K. Yao, D. Yu, F. Seide, H. Su, L. Deng, and Y. Gong, "Adaptation of context-dependent deep neural networks for

automatic speech recognition," *IEEE workshop on spoken language technology*, pp. 366-369, 2012.

[10] B. Li and K. C. Sim, "Comparison of discriminative input and output transformations for speaker adaptation in the hybrid NN/HMM systems," *Proc. INTERSPEECH*, pp. 526-529, 2010.

[11] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. D. Mori, "Adaptation of hybrid ANN/HMM models using linear hidden transformations and conservative training," *Proc. ICASSP*, pp. 1189-1192, 2006.

[12] S. M. Siniscalchi, J. Li, and C.-H. Lee, "Hermitian polynomial for speaker adaptation of connectionist speech recognition systems," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 21, no. 10, pp. 2152-2161, 2013.

[13] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," *Proc. ICASSP*, pp. 7893-7896, 2013.

[14] O. Abdel-Hamid, and H. Jiang, "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code," *Proc. ICASSP*, pp. 7942-7946, 2013.

[15] J. Li, J.-T. Huang, and Y. Gong, "Factorized adaptation for deep neural network," *Proc. ICASSP*, 2014.

[16] 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.

[17] J. Xue, J. Li, and Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition," *Proc. Interspeech*, pp. 2365-2369, 2013.