Intermediate-layer DNN Adaptation for Offline and Session-based Iterative Speaker Adaptation

Kshitiz Kumar, Chaojun Liu, Kaisheng Yao, Yifan Gong

Microsoft Corporation, Redmond, WA

{kshitiz.kumar, chaojunl, kaisheny, yifan.gong}@microsoft.com

10.21437/Interspeech.2015-288

Abstract

In this work we present intermediate-layer deep neural network adaptation (DNN) techniques upon which we build offline as well as iterative speaker adaptation for online applications. We motivate our online work for task completion in Microsoft personal voice assistant, where we present different adaptation styles in a speech session *e.g.*, (a) adapt the speaker-independent (SI) model on the current utterance, (b) recursively adapt an incremental speaker-dependent (SD) model in the session for just the previous utterance, (c) adapt the SI model for all past utterances in the session. We considered a number of adaptation techniques and demonstrated that the intermediate-layer approach with inserting-and-adapting a linear layer on top of an intermediate singular-value-decomposition layer provides the best results for offline adaptation, where we obtained respectively 22.6% and 12% relative reduction in word-error-rate (WER) for supervised and unsupervised adaptation on 100-utterances. An alternative intermediate-layer recursive adaptation in a 5-utterances session provided 6% relative-reduction in WER for online applications.

Index Terms: Second-pass decoding, DNN, Speaker Adaptation, Confidences

1. Introduction

Recent progress in deep learning for automatic speech recognition (ASR) has produced record-setting performances. The strong modeling capability of deep neural networks (DNNs) [1, 2] has posed challenges to speaker adaptation, which aims at improving ASR performance on targeted users. Several methods have recently been proposed for DNN adaptation. In general, there are three broad approaches. The first approach is based on affine transformation of DNN parameters. For example, methods in [3] apply affine transformation on the top hidden layer activities. The second approach uses conservative training methods. For example, Kullback-Leibler divergence to constrain parameter updates was proposed in [4]. A recent improvement in [5] uses singular value decomposition to further reduce the number of adaptation parameters to achieve low-footprint adaptation. The last approach uses subspace methods. This approach constructs an adapted DNN as a point in a space that is formed from analysis on many speaker-dependent models. Popular methods in this approach uses principle component analysis [6], i-vectors [7, 8], and GMM fMLLR based approach in [9]. A relevant recent work applies speaker adaptive training in [10]. Most of the above described methods [3, 4] require sufficient number of speaker adaptation utterances to be effective. However, in practice, it is hard to have a large number of adaptation utterances. The assumption in [6, 7] may not hold when the user space is not inclusive.

Of our particular interests is a mobile personal assistant. This application consists of a short 4-6 utterances sequence of user-computer interactions in a session, where a user asks for information from a server. The server responds to the user's questions with answers after ASR and spoken language understanding (SLU). For example, a user asks for "bakeries in Redmond" and the system displays corresponding best results. Subsequently user can ask for reviews, driving directions etc. Though specific, the above example is a typical scenario that pose key challenges to ASR. A method or framework for DNN adaptation needs to be effective with small amount of adaptation utterances, and no assumption of known speakers.

This paper presents methods for intermediate-layer DNN adaptation where we additionally insert a linear layer on top of our singular-value-decomposition (SVD) [5] layers. We apply our work to both offline and online applications where for online adaptation we use a framework that effectively uses only available adaptation data in a session. The online framework consists of iterative adaptation styles that improve ASR performance yet they are implemented in real time.

2. Speaker adaptation techniques

This sections presents the adaptation methods we implemented with respect to a representative DNN model in Fig. 1(a). In practice our DNNs consist of 5 hidden layers but for figurative description in Fig. 1 we consider a DNN with 2 hidden layers in (D0, D1) and an output layer. Our baseline DNN architecture includes singular value decomposition (SVD) layers (S0, S1) in Fig. 1 [5]. We use notation "L-D0" to indicate DNN layer weights that input to D0, using which we compute output of layer D0, similarly L-S0 and L-O respectively indicates layers that input to S0 and *Output*.

2.1. Intermediate layer DNN adaptation

Recent work in [3] proposed adaptation on the top hidden layer. The motivation is treating the outputs from the top hidden layer as features to the top log-linear model. It uses an affine transformation $\mathbf{F}$ that has size of $O(NM)$ where N and M respectively denote the number of nodes for the top layer (S1) and the output layer (O). For our DNN architecture in Fig. 1(a), this method is equivalent to adapting layer L-O but adapting it has two key limitations:

1. it can adapt only seen senones - since $\mathbf{F}$ changes with error signals from observed senones only, this method requires sufficient observations on all senones to be effective.
2. large parameter size - for a typical N as 256 and M as 6000 for our DNN in Fig. 1(a), this method adapts approximately 1536k parameters, here "k" indicates order

of 1000, thus adapting layer L-O can be very prohibitive for limited data scenarios.

In this work we build on the work in [3] and motivate techniques that address above issues as well as improve accuracy We propose to individually adapt other intermediate DNN layers in L-S1, L-D1 etc. Adapting these internal layers under a regularization constraint has broader impact than simply adapting L-O and affects all senones due to subsequent feature transformation through nonlinear or SVD layers. Furthermore for ASR, we rationalize first few DNN layers as feature normalization steps, where device and speaker-dependent features get normalized; we think of middle-layers as higher-order feature synthesis, where we encapsulate normalized features into abstract speech bases; and finally top-layer is a classification layer that classifies speech into physical triphones or senone states. Thus individually adapting different DNN layers provides unique adaptation techniques.

2.2. Insert-and-adapt a linear layer on top of SVD layer

The number of adaptation parameters for above intermediate-layer adaptation techniques are typically 2048 x 256 = 524k, approximately $1/3^{rd}$ of that for adapting L-O but still substantially large, and doesn't completely address the parameter-size issue. A smaller parameter-size significantly helps with data storage requirements and fetching speaker-dependent parameters for decoding in our speech deployment in service. In order to address parameter-size, we propose to leverage SVD layers and insert-and-adapt a linear layer on top of the SVD layer. This was motivated by our prior work in [5], where we demonstrated a low-footprint DNN adaptation. We refer to the work in [5] as "SVDAll". A key difference between "SVDAll" and our current work is that "SVDAll" inserts-and-adapts linear layers on top of all SVD layers in the DNN architecture, whereas, we insert a linear layer to just one of the SVD layers. The location of insertion is chosen experimentally. Thus our work requires $1/5^{th}$ of the parameters required in [5], and provides significantly better results as noted in Fig. 2.

We illustrate our approach of inserting and adapting in Fig. 1(b); we insert a linear network I0 on top of the SVD layer S0, and adapt corresponding layer L-I0. The number of associated parameters is 256 x 256 = 65k, *i.e.*, only 4.2% of that required for L-O. A smaller parameter-size can provide powerful adaptation algorithms in limited data scenarios. Similarly we may choose to insert a layer I1 and adapt corresponding L-I1. Our work finds some similarities with a very recent work in [11]. However our approach of SVD-decomposition-based DNN along with inserting-and-adapting layers clearly distinguishes our work.

3. Experiments for offline intermediate-layer adaptation

In this section we present results for offline adaptation using techniques described in Sec. 2. We conducted our experiments on a US English short-message-dictation (SMD) dataset. Results on this task have also been noted in [3, 5]. This task consists of 6 speakers, each speaker has 100 adaptation utterances (11 minutes). Test results were averaged across 1200 utterances (2.2 hours). The total number of test set words is 20k. There is no overlap between the adaptation and test data. Our baseline deep neural network (DNN) acoustic model (AM) was trained from over 300 hours of voice-search and SMD data with 66-dim dynamic log-MelFilterbank features and a context window of

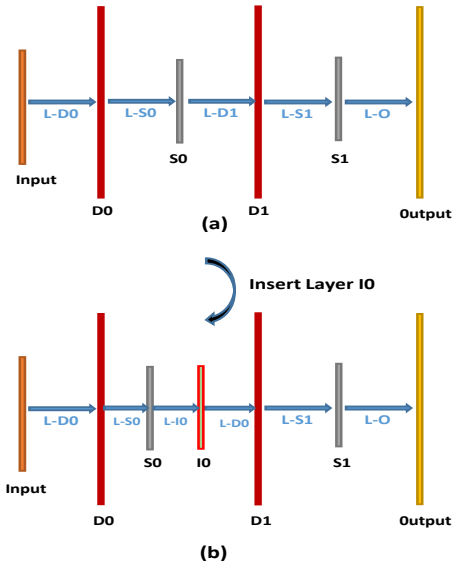

Figure 1: *(a) A representative SVD-based DNN architecture; layers in (D0, D1) indicate usual deep non-linear layers; layers in (S0, S1) indicate linear SVD layers, (b) Inserting a compact linear layer I0 on top of SVD layer S0.*

11 frames, forming an input vector of 726-dim. DNN had 5 hidden layers with 2048 nodes each, 5 SVD layers with about 256 nodes each, along with 6000 output units. The hidden layers apply sigmoid nonlinearity; output layer applies softmax. We convert the full-rank DNN model to low-rank model by doing SVD on all the matrices except the one between the input and the first hidden layer [5]. We then retrained the low-rank model and obtained accuracy comparable to the full-rank model.

We provide adaptation results for both unsupervised and supervised techniques. For unsupervised we use the SI model to decode and then align data against decoded hypotheses. We present results from unsupervised adaptation in Fig. 2(a). Unsupervised technique does not require human transcriptions, hence cost-effective and readily applicable than supervised adaptation. All of our adaptation methods use Kullback-Leibler-divergence (KLD) with regularization coefficient of 0.1. KLD regularization [4] belongs to the conservative training category for DNN adaptations, where we adapt models conservatively by forcing the senone distribution estimated from the adapted model to be close to that from the unadapted model. Our baseline SI model word error rate (WER) is 19.9%. From Fig. 2(a), we note that adapting either of L-I1 or L-I2 provides the best WER of 17.5%, this is statistically significant over 17.75% for "SVDAll", while requiring $1/5^{th}$ of the parameters in "SVDAll". We noted parameter-size across adaptation techniques in Table 1. Furthermore, we note that the WER for adapting one of intermediate-layers broadly obeys a "U-shaped" curve, where layers in between D1 and D4 provide the best adaptation performance. We specifically note that adapting intermediate layer provide huge benefits over adapting either the top-layer L-O or input layer L-D0 with respect to both accuracy and parameter-size requirements. Thus adapting intermediate layers addresses the two key issues we noted in Sec. 2.1.

Fig. 2(b) presents results for supervised adaptation. It has the same setup as that for unsupervised adaptation in Fig. 2(a)

Table 1: *Number of required parameters across adaptation techniques.*

Adaptation technique	No. of Parameters (in 1000's)
Top layer L-O	1536
Input layer L-D0	1486
Individual layers in (L-S0, L-D0 etc.)	524
SVDAll [5]	327
Individual inserted layers in (L-I0 etc.)	65

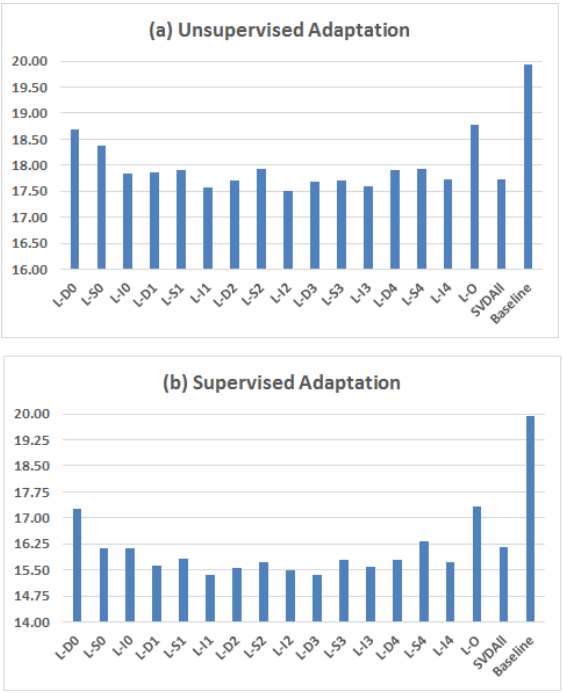

Figure 2: *(a) WER for adapting one of the intermediate layers with unsupervised transcription, (b) WER for adapting intermediate layers with supervised transcription.*

except that we used ground-truth transcriptions for alignment. Results show that the presented intermediate-layer adaptation methods are also effective for supervised adaptation. For example, adapting layer L-D3 or L-I1 improved baseline WER from 19.9% to 15.4%, this corresponds to 22.6% word-error relative reduction (WERR). This also provided 5% WERR on top of "SVDAll" while requiring $1/5th$ of adaptation parameters in "SVDAll". The shape of the WER curve is again broadly "U-shaped" across adapting one of intermediate layers in between D1 and D4.

Overall the approach of inserting-and-adapting a linear layer on top of an intermediate SVD layer provides accuracy comparable or better than directly adapting an existing intermediate layers. Furthermore, it requires the fewest number of parameters. We have recently obtained additional savings in the number of speaker-specific parameters for layer L-I1 by only adapting weights corresponding to top-N output units of L-S1; this works due to an inherent ranking in the SVD layer L-S1 output units. We also experimented with feature-space discriminative linear regression (fDLR) [12]; it applies a linear transformation to the input features. In our experiments, we found the WER for fDLR to be similar to that for adapting layer L-D0. We also experimented with adapting combinations of intermediate layers but so far we have not seen the gain to be significant in our task. However adapting multiple layers may help for substantially larger dataset.

4. Online Adaptation

In this section we present the key motivations, applications and broad framework for online adaptation. We consider application scenarios where (a) we may not have sufficient data to train speaker or environment models, (b) we may have data but it may not generalize across devices or acoustic environments at runtime, (c) we may have unseen speakers, (d) we may operate in a session with single or multiple utterances that we may incrementally leverage for adaptation.

We further motivate with application to Cortana, a Microsoft personal voice assistant. There we introduced a key notion of a session where a user can accomplish certain tasks *e.g.*, driving directions, setting up meetings, calling or texting etc. A session typically has multiple utterances, we refer to 1^{st} utterance as turn-1 ($T1$) utterance and similarly turn-n utterance as Tn. Sessions are typically short with 4-6 utterances. We apply our adaptation work to this key scenario to improve user experience with incremental adaptation. Our work does not require any prior information about speaker or acoustic environment. Furthermore, training speaker or environment dependent models for millions of Cortana users can be prohibitive, requiring dedicated infrastructure, engineering and testing resources, whereas, online adaptation requires minimal support. We next present different adaptation styles.

4.1. Incremental adaptation on an intermediate SD model for just the previous utterance

We use SI model to obtain ASR results for turn-1 ($T1$) utterance. Simultaneously we use $T1$ utterance to adapt SI model and produce an incremental speaker-dependent (SD) SD-$T1$ model, this replaces the SI model for decoding $T2$ utterance. Similarly the approach can be applied to Tn utterance, as also noted in Fig. 3(a). This works in real-time as the model adaptation for SD-$T1$ can be completed while we communicate and display ASR results for $T1$ to user, and hear back from user with $T2$.

4.2. Iterative adaptation from SI model using all past utterances in a session

A session in Cortana consists of a few utterances, thus an opportunity for incremental adaptation within the session. We present this adaptation in Fig. 3(b). Similar to Sec. 4.1, we use SI model to produce ASR results for $T1$ utterance and use $T1$ to obtain an adapted model in SD-$T1$. Similarly we obtain SD-Tn by adapting SI model using all utterances in turns $[1 \ldots n]$, and use SD-Tn to decode $(n+1)^{th}$ utterance. This approach works in real-time, the rational is identical to that described in Sec. 3(a).

4.3. Adaptation Framework for Second-pass decoding

We also considered second-pass decoding based adaptation framework as described in Fig. 3(c). The system produces results from 1^{st}-pass decoding in the usual way from a given set of acoustic features and *Baseline* acoustic model (AM). There,

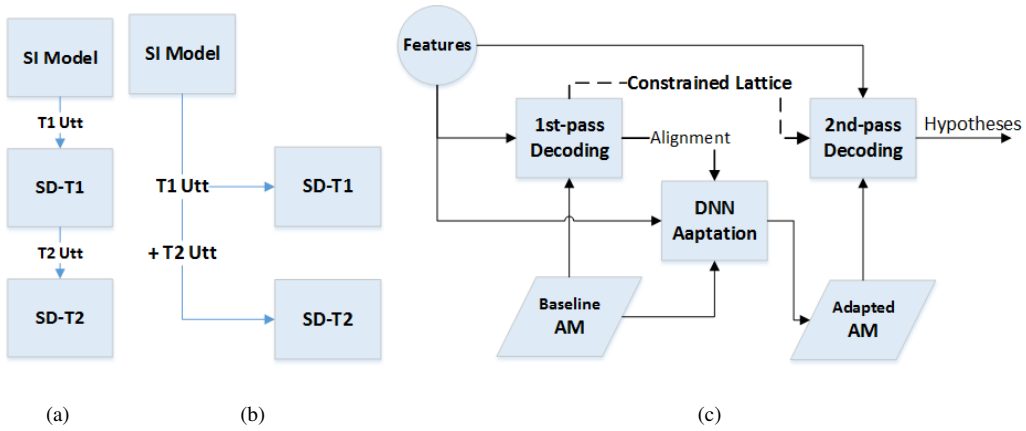

Figure 3: Adaptation styles (a) Incrementally adapt intermediate SD model for just the previous turn utterance, (b) Iteratively SI Model using all past data utterances, (e) Adaptation framework for second-pass decoding.

we additionally output alignments and optionally a constrained lattice, that we use for adaptation, and obtain *Adapted* AM. Finally we obtain hypothesis by decoding against the adapted AM, where we can use constrained lattice for decoding. We have engineered this method to be $1.1x$ real time. A key difference between this approach and those in Secs. 4.1 and 4.1 is that here for $T1$ utterance we will redecode from adapated model $SD\text{-}T1$, and communicate these results to user, and similarly we for other Tn utterances.

5. Experiments for Online Intermediate-layer Adaptation

In this section we present and analyze results for online speaker adaptation discussed in Sec. 4 using intermediate-layer adaptation techniques. Our baseline SI model and DNN architecture is identical to that described in Sec. 3. Our adaptation task consists of SMD data from 100 sessions, where each session consists of 5 utterances (33 secs.).

We apply our framework in Fig. 3(c) to incrementally adapt and decode. Note that we obviously do not have access to true transcriptions during deployment of online adaptation techniques. So we only consider unsupervised adaptation techniques, where we obtain hypotheses from decoding and use that to align data for adaptation. We experimented with a wide range of KL-regularization coefficients and verified 0.1 to be the best. We report results in Table 2. The baseline SI model has a WER of 21.2%. We experimented with adapting different intermediate layers for online adaptation and consistently found adapting L-S2 to be the best. The incremental adaptation work in Sec. 4.1 reduced WER to 20.4%. Next, iterative adaptation in Sec. 4.2 further improved WER to 19.9%, for an overall 6% WERR over the baseline AM. A WERR of 1% is statistically significant in our experiments. These observations are in contrast to offline adaptation experiments in Sec. 3, where we obtained better results with adapting L-I1, while also requiring fewer parameters. We will continue to test these methods across languages and datasets to test the generalization of our current observations.

Note that adaptation with a few utterances is a very challenging task, and to our knowledge our work is among the first to demonstrate strong gains for these scenarios. Furthermore, these techniques work in real time, this is one of the most critical requirements from deployment perspective. We also experimented with fDLR but did not see significant gains. The authors in [9] raised uncertainty on the application of fDLR for adaptation. Our work shows that intermediate-layer techniques can provide huge gains over fDLR. We also experimented with the second-pass framework in Sec. 4.3 but so far we have not see additional gains over that reported in Table 2.

Table 2: *WER for different adaptation styles while adapting layer L-S2.*

Adaptation Styles	WER [%]
SI Baseline Model	21.2
Incrementally adapt SD model on the previous utt. in a session and decode, Sec. 4.1	20.4
Iteratively adapt SI model on all previous utts. and decode, Sec. 4.2	19.9

6. Conclusion

The intermediate-layer approach presented in this work isn't specific to speaker adaptation. The techniques can be applied across other ASR adaptation applications. We are already seeing strong gains while adapting for, (a) accent, (b) environment, (c) device etc. The specific layer to be adapted may in general be application specific. We would like to further understand the role of different intermediate layers in a DNN and develop rational behind adapting one of the layers for best ASR results for particular applications. In this work we also motivated techniques for both offline, and online speaker adaptation for task completion via a few utterances. For online, we propose to leverage utterances in an incremental adaptation framework. We compared different adaptation techniques and established that we can improve accuracy as well as reduce the number of adaptation parameters with intermediate-layer adaptation approaches. We demonstrated a strong 6% relative reduction in WER for a very challenging task of online adaptation with 5 utterances.

7. References

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, 2012.

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

[3] K. Yao, D. Yu, F. Seide, H. Su, L. Deng, and Y. Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition," in *IEEE SLT*, 2012.

[4] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *ICASSP 2013*, 2013.

[5] J. Xue, J. Li, D. Yu, M. Seltzer, and Y. Gong, "Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network," in *ICASSP*, 2014.

[6] S. Dupont and L. Cheboub, "Fast speaker adaptation of artificial neural networks for automatic speech recognition," in *ICASSP*, 2000, pp. 1795–1798.

[7] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *ASRU*, 2003, pp. 55–59.

[8] P. Karanasou, Y. Wang, M. Gales, and P. Woodland, "Adaptation of deep neural network acoustic models using factorised i-vectors," 2014.

[9] X. Lei, H. Lin, and G. Heigold, "deep neural networks with auxiliary gaussian mixture models for real-time speech recognition," in *in Proc. ICASSP*, 2013.

[10] Y. Miao, H. Zhang, and F. Metze, "Towards speaker adaptive training of deep neural network acoustic models," 2014.

[11] T. Ochiai, S. Matsuda, X. Lu, C. Hori, and S. Katagiri, "Speaker adaptive training using deep neural networks," 2014.

[12] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *ASRU*, 2011.