A STUDY OF RANK-CONSTRAINED MULTILINGUAL DNNS FOR LOW-RESOURCE ASR

Reza Sahraeian, Dirk Van Compernolle[*].

KU Leuven - ESAT
Kasteelpark Arenberg 10, 3001 Heverlee, Belgium

ABSTRACT

Multilingual Deep Neural Networks (DNNs) have been successfully used to exploit out-of-language data to improve under-resourced ASR. In this paper, we improve on a multilingual DNN by utilizing low-rank factorization (LRF) of weight matrices via Singular Value Decomposition (SVD) to sparsify a multilingual DNN. LRF was previously used for monolingual DNNs, yielding large computational savings without a significant loss in recognition accuracy. In this work, we show that properly applying LRF on a multilingual DNN can improve recognition accuracy for multiple low-resource ASR configurations. First, only the final weight layer is factorized. Since the output weight layer needs to be trained with language specific data, reducing the number of parameters is beneficial for under-resourced languages. It is common in multilingual DNN speech recognition, to further adapt the full neural network through retraining of the multilingual DNN on target language data. Again we observe that in low-resource situations, this adaptation can bring significant improvement if LRF is applied to all hidden layers. We demonstrate the positive effect of LRF in two very different scenarios: one is a phone recognition task for two related languages and the other is a word recognition task using five different languages from the GlobalPhone dataset.

Index Terms— Multilingual deep neural network, low-rank factorization, low-resource ASR

1. INTRODUCTION

Recently, there has been significant interest in the area of multilingual acoustic modeling in the context of deep neural networks (DNNs) in particular for under-resourced languages [1, 2, 3, 4]. In the DNN, the hidden layers can be considered as a universal complex feature transformation which can be successfully used for different languages [5]. Hence, the hidden layers can be trained simultaneously for different languages to benefit from each other. More specifically, to bootstrap acoustic modeling for a low-resource language, the auxiliary data from other high resource language(s) can be used to train the multilingual DNN and then only the softmax layer is trained with the low-resource target language [6, 2]. Additional improvement can be obtained by further adjusting the whole DNN which is often termed as DNN *adaptation* [7, 8]. Universal phoneme sets [9, 10] and phone targets mapped across a small set of languages [11] have both been used as a multilingual phoneme set during training. There are generally two popular configurations of multilingual DNN systems: the first one is the conventional multilingual DNN based on a *hybrid* system [12] ; the second model configuration is based on a *tandem* which exploits the DNNs to perform nonlinear discriminative feature transformation [13]. The transformed features are then used as inputs for another GMM or DNN based model. The latter allows multiple DNNs to be stacked and is referred to as a *stacked hybrid* system [4].

DNN training involves a large number of parameters which makes it slow and can lead to overfitting. Therefore, it is of interest to develop techniques that reduce the number of parameters without hurting the performance; two popular approaches to achieve this are dropout [14] and maxout [15]. In [16], it is shown that a large portion of the weight parameters in a DNN are very small and have a negligible effect on the output values of each layer which exploits the sparseness in DNN. Other techniques have been proposed which change the DNN architecture; for example, [17] proposed to shrink the hidden layers gradually from bottom to upper layers. Moreover, low-rank matrix factorization for DNN weight matrices using linear bottleneck [18] and SVD [19] was proposed to reduce the overall training time while recognition accuracy was not significantly affected. The aforementioned studies deal with model size reduction and accelerating the DNN training and test time for monolingual systems; while no significant improvement is achieved. However, [20] showed that LRF of the final weight layer improves the performance of a monolingual stacked hybrid system and the low-rank linear layer outperforms sigmoid layer to extract bottleneck features. Very recently, LRF of the last weight layer has been utilized in the framework of multilingual DNNs and improved performance was reported for a very resource-constrained setting [21].

Our work extends the use of LRF for multilingual DNN by exploring several scenarios in which not only the final weight layer, but also other weight layers are factorized, and we show that DNN adaptation benefits from this factorization. The rest of the paper is organized as follows. In section 2, we describe the multilingual DNN training with shared hidden layers. Then, the low-rank factorization technique is explained in section 3. The experimental setup and results are presented in sections 4 and 5. Finally, we have concluding remarks.

2. MULTILINGUAL DNN

Deep Neural Networks can be considered as a cascaded sequence of nonlinear feature extractors followed by a classifier at the output layer. The neural net is trained to predict the posterior probability of each context-dependent state determined by standard clustering algorithms from previously trained HMMs [22]. Neural networks usually employ a sigmoid or tanh nonlinearity function; however, it has been shown that rectifier linear unit (ReLU) can improve the performance of DNN [23]. In this work, we use the ReLU nonlinearity: $u_j^{(l)} = max(h_j^{(l)}, 0)$; where $\mathbf{u}^{(l)} = [u_1^{(l)}, u_2^{(l)}, ..., u_{n_H}^{(l)}]$ is a set of activations in layer l, and $h_j^{(l)}$ is the jth element of $\mathbf{h}^{(l)} = \mathbf{W}^{(l)}\mathbf{u}^{(l-1)} + \mathbf{b}^{(l)}$. $\mathbf{W}^{(l)}$ is the matrix of connection weights be-

[*]This research was supported by the fund for scientific research of Flanders (FWO) under project AMODA GA122.10N.

tween the $(l-1)$th and lth layers, $\mathbf{b}^{(l)}$ is the bias vector at the lth layer.

Multilingual DNNs rely on the assumption that the hidden layers are universal feature extractors that are transferable between languages and domains [6, 1]. The whole procedure may be summarized as follows [7]: First, a DNN is trained using multilingual training data. Then, for a novel target language the hidden layers are reused and only the softmax layer is trained with target language data. In a last phase, further adaptation of all parameters in the DNN may be performed. The number of epochs for retraining the whole network depends on the amount of available data from the target language. In the multilingual target layer, each language can have its own output layer or a common output layer may be used; in the latter, we need to provide a universal phoneme set.

3. LOW-RANK FACTORIZATION

The use of low-rank matrix factorization for DNN training is proposed in [18] and [19] to reduce computational and space complexity for monolingual DNNs. To this end, each connection weight matrix can be factorized into smaller matrices and thereby the number of parameters in the network is significantly reduced. Especially when DNNs are trained with a large number of output targets, [18] shows that LRF of the last weight layer reduces the number of parameters of the DNN significantly. In the case of low-resource ASR using multilingual DNN, LRF is particularly attractive as it reduces the number of independent parameters that should be estimated or adapted with low-resource data.

In a first configuration, we only factorize the weight matrix of the final layer. Let us denote the final weight matrix for language L by A^L with dimensions $n_H \times n_T^L$ where n_H is the number of units in the last shared hidden layer and n_T^L is the number of output targets for language L. Note that when a common output target using a universal phone set is used, there is only one output weight layer. In both scenarios, if there is a rank n_r for the final weight matrix, then there exists a factorization $A^L = B^L \times C^L$ where B^L and C^L are full rank matrices of size $n_H \times n_r$ and $n_r \times n_T^L$ respectively. Now, in a multilingual low-resource scenario we may want to further reduce the number of language dependent parameters by incorporating the matrix B^L in the layers that are shared across languages and thus $B^L = B$ for all languages as shown in Fig. 1 [21]. Then, for a language L', we only need to train an output weight matrix of dimensions $n_r \times n_T^{L'}$, which is much smaller than $n_H \times n_T^{L'}$. It is worth noting that in this approach there exists one extra weight layer in the shared components compared to the typical multilingual DNN; however, we show in the experiments that this is not very relevant.

Secondly, we propose to extend LRF to other weight layers which leads to a huge reduction of the number of parameters in the multilingual DNN system. The LRF is applied after initial training of the multilingual DNN and before adaptation with resource-constrained target language data. It is true that after low-rank factorization of all layers the multilingual DNN may have moved away from its optimal trained state and that convergence may not be achieved during retraining, given limited data and number of training passes. Thus, it is of great interest to investigate if the possible gain by adaptation can overcome the convergence issue.

In this paper, low-rank factorization of the weight layers is done by using SVD based model restructuring method in which a $n_H \times n_T^L$ weight matrix layer A^L is decomposed as:

$$A^L_{n_H \times n_T^L} \approx U_{n_H \times n_r} \Sigma_{n_r \times n_r} V^T_{n_T^L \times n_r} \quad (1)$$

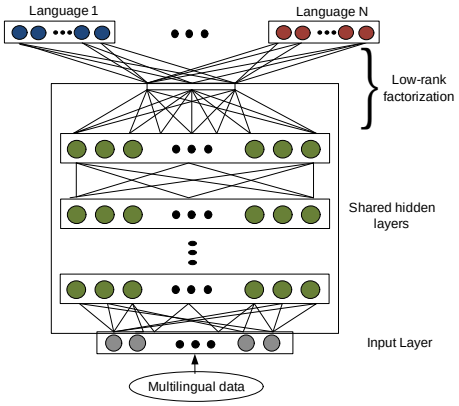

Fig. 1. Multilingual DNN training with LRF in the final weight layer.

Then, we consider $B = U_{n_H \times n_r}$ and $C^L = \Sigma_{n_r \times n_r} V^T_{n_T^L \times n_r}$ and replace A^L with these two smaller matrices as described in [19].

LRF can also be accomplished by configuring the DNN with a linear bottleneck and let the factorization being learned during DNN training, and since parameters of DNN is reduced before training, the overall training time can be reduced as well [18]. The downside of this method, however, is that the bottleneck dimension has to be defined beforehand and for a new dimensionality we need to train a new DNN. However, the main goal of this paper is to improve the accuracy rather than decreasing training time; thus, SVD is applied to factorize the weight layers so that n_r can be tuned with less computational complexity.

4. EXPERIMENTAL SETUP

4.1. ASR systems

Monolingual reference systems were built using target language data only. First, Gaussian mixture model systems were built using a 39-dimensional MFCC feature vector with 13 cepstral coefficients, and their first and second derivatives. Speaker based cepstral mean and variance normalization (CMVN) was applied and features were spliced in time taking a context size of 7 frames (i.e.,$\pm$3), followed by decorrelation and dimensionality reduction to 40 using LDA and further decorrelation using MLLT [24]. The number of gaussians and tied states for GMM based modeling was tuned over the development set. The derived states were used as targets in the DNN systems.

Then, monolingual DNNs were trained on mean and variance normalized 24-dimensional FBANK features being concatenated with 7 left and 7 right neighbor frames to yield an input feature vector size of 360; we observed that FBANK features outperform MFCCs as input features for DNN. The multilingual systems were based on multi-task learning of DNNs. The neural network's input features and the learning rates were the same as those used in the monolingual DNNs except that normalization was not applied. More details about the implementations are provided in the experiment section.

All the DNNs used in this study were trained using a ReLU nonlinearity based on greedy layerwise supervised training [25]. The initial and final learning rates were specified by hand and equal to 0.01 and 0.001 respectively.

The Kaldi ASR toolkit [26] is used for both GMM and DNN based acoustic modeling.

4.2. Flemish-Afrikaans

First, we perform experiments with two closely related languages: Flemish and Afrikaans; in this setting, Afrikaans plays the role of under-resourced target language, and Flemish takes on the role as well resourced donor language. We used component-o from the spoken Dutch corpus (Corpus Gesproken Nederlands, CGN) [27]. This dataset contains 38 hours of speech sampled at 16KHz and we have taken 36hr for the training and 2hr for the evaluation. In this work, we used only the training part including 36 hours (produced by 150 speakers) as donor data. The CGN pronunciation dictionary uses an alphabet of 47 phonemes.

The Afrikaans data is taken from the NCHLT corpus consisting of 210 speakers, including broadband speech sampled at 16 kHz [28]. The phoneme set contains 38 phonemes, including silence. All repeated utterances were removed from the original dataset. In our setting, to simulate various low resource conditions, we consider one hour of data, five hours of data and the full training set including about 10.7 hours [29]. We used the default evaluation and development sets including 2.2hr and 1hr data respectively[1]. We used the standard HLT test scenario which is a phone recognition task and the results are presented as phone error rate (PER). A bi-gram phoneme language model is trained on the training set.

4.3. GlobalPhone

Next, we extend our experiments to a multilingual case where five languages are used from the GlobalPhone dataset [30] with German as the target language, and the other four as donor languages. The GlobalPhone corpus is a multilingual text and speech corpus that covers speech data from 20 languages [30]. In our experiments, German (GE) was used as the target language, and Spanish (SP), Portuguese (PO), Russian (RU) and French (FR) as the auxiliary languages. The detailed statistics for these languages from the Globalphone corpus are presented in [30]. The recognition task is a standard word recognition task using a trigram language model obtained from Karlsruhe University[2].

The full German database consists of 14.85 hours by 65 speakers. To simulate low-resource conditions, we constructed two subsets containing 1 hour (8 speakers) and 5 hours (40 speakers) of data, both using randomly selected 7-8 minutes of speech for each of the selected speakers. The development and evaluation set include 1.95 and 1.45 hour data and each of them consists of 6 speakers. For the multilingual experiments, we used respectively following amounts of data of the donor languages: 22.74hr for FR, 22.71hr for PO, 21.10hr for RU and 17.55hr for SP [6].

5. EXPERIMENTS

5.1. Flemish-Afrikaans

The monolingual reference experiments yielded 505, 1380 and 2281 context-dependent states for 1hr, 5hr and 10.7hr training data respectively. The number of hidden layers and neurons per layer were tuned; the optimal number of hidden layers were 2, 3, 4, and the number of hidden units in each layer were 200, 400 and 500 for the respective settings. The first two rows in Table 1 show the PERs using GMM and DNN based acoustic modeling.

Table 1. Comparing PERs(%) for Afrikaans using monolingual and multilingual systems with and without LRF for the final weight layer.

Systems		Afrikaans data		
		1hr	5hr	10.7hr
Monolingual	HMM/GMM	23.09	16.87	14.81
	HMM/DNN	23.56	15.20	12.06
Multilingual DNN	Not adapted	18.66	12.83	10.89
	Adapted	18.52	12.64	10.79
Adapted multilingual DNN with LRF for the final layer	$n_r = 100$	18.69	12.68	10.36
	$n_r = 200$	17.76	12.30	10.29
	$n_r = 500$	17.47	12.29	10.44

For the multilingual DNNs we examined two possible types of multilingual phoneme targets. In the first one, the phoneme targets for Flemish and Afrikaans are kept separate. In the second scenario, a universal phoneme set was created by applying a knowledge-based phoneme mapping [31]. We observed in our experiments that the latter outperforms and thus we only report the results of the second scenario in this paper. In this case, a common multilingual target was used and therefore the whole multilingual DNN, including the softmax layer, was trained using both Flemish and Afrikaans data and the output target included 4131, 4778 and 5422 tied-states for multilingual HMM/GMM systems with 1hr, 5hr and 10hr Afrikaans included respectively. Then, these hidden layers were reused to train a softmax layer using only the Afrikaans data. Furthermore, adaptation of the full DNN to Afrikaans data only was performed. Table 1 also shows the PERs obtained using the multilingual DNN system; the optimal number of hidden layers in each setting were 7, 8, and 8 for 1hr, 5hr, and 10.7hr Afrkaans respectively with the number of hidden units per layer equal to 1000. Table 1 also compares the PERs obtained by only training the softmax layer with those achieved after updating all layers (i.e. adaptation is applied). The following observation can be made from Table 1: first, the performance for target language (Afrikaans) is improved when Flemish is included for multilingual DNN training in all settings. Moreover, it is also always beneficial to apply adaptation by further updating the whole DNN.

To investigate the effectiveness of LRF in the multilingual DNN, SVD is applied to factorize the last weight layer after which the whole DNN is fine tuned. In this experiment, we retrained the multilingual DNN with multilingual data for 5 epochs. Then, all the hidden layers together with the first weight matrix of size $1000 \times n_r$ are transferred with further adaptation to bootstrap the acoustic modeling for the Afrikaans language. The PERs for different choices of n_r are shown in Table 1.

Table 1 reveals further trends: first, using low-rank decomposition of the multilingual DNN improves the performance compared to the conventional multilingual DNNs. Moreover, the PER reduction is more pronounced when the target language is more under-resourced. However, the reasonable questions which might arise are that how the nonlinear bottleneck would perform? and as mentioned in section 3, the low-rank network has an extra weight layer compared to the multilingual baseline system so is the obtained improvement because of this extra layer? To answer these questions, we consider a scenario with 1hr Afrikaans training data; the simplest approach is to train a conventional multilingual DNN with 8 hidden layers where a nonlinear bottleneck constraint is applied on the last layer to have the width of n_r. Thus, the total number of weights in this multilingual system is the same as the 7-layer multilingual DNN with LRF of the last weight layer. The PER obtained for this system when $n_r = 500$ is 18.13% which is higher than the corresponding

[1]The authors are thankful to the HLT group at Meraka for providing us with the training, test and validation sets and Afrikaans dictionary.

[2]http://csl.ira.uka.de/GlobalPhone/

Table 2. WER(%) for German using monolingual and multilingual DNN systems.

Settings		Monolingual		Multilingual DNN	
		GMM	DNN	Not adapted	Adapted
1hr	Dev.	22.84	21.41	18.74	18.78
	Eval.	35.38	34.90	32.54	32.57
5hr	Dev.	15.70	13.40	12.74	12.76
	Eval.	24.41	22.93	22.13	22.04
14.85hr	Dev.	13.95	11.56	11.15	11.02
	Eval.	21.36	19.49	18.78	18.36

PER presented in Table 1 which is 17.47%.

5.2. GlobalPhone

In this set of experiments, we used LRF in a more multilingual environment where German plays the role of low-resource target language as described in section 4.3. First, we constructed baseline systems for the three training sets in a monolingual fashion using HMM/GMM and HMM/DNN acoustic modeling. The number of context-dependent triphone states were 700, 1200 and 3100 with an average of 4, 9 and 13 Gaussian components per state for 1hr, 5hr, and 14.85hr German training data respectively. These parameters were tuned on the development set. Word error rates (WER) for both development (Dev.) and evaluation (Eval.) sets are presented in Table 2 for HMM/GMM systems as well as HMM/DNN ones. The optimal number of hidden layers were 4, 4, 5 and the number of hidden units in each layer were 50, 200, and 300 for 1hr, 5hr and 14.85hr of training data respectively.

Then, a multilingual DNN was trained with a dedicated softmax layer for each language while the hidden and input layers were shared. Following the setup of the authors in [6], the number of target context-dependent states were set to 3100 for each auxiliary language. The number of hidden layers and units per layer were tuned. We used a DNN with 7 layers for the setting including 1hr of German data and 8 layers for the two other settings; the number of nodes was 1500 per layer in all DNNs. The performance of the multilingual systems with and without adaptation is presented in Table 2. It is observed that no improvement was obtained by adaptation when only 1hr or 5hr of German data was available. With more available German data, we can see that adaptation yields a small improvement. This is typical behavior for a multilingual DNN with a large number of parameters.

Next, the final weight layer of the best multilingual DNN for each scenario was factorized using SVD and afterwards the whole network was fine tuned with multilingual data. The WERs for $n_r = 500$ are presented in Table 3 for both development and evaluation sets. We also tried other bottleneck dimensions like 700 and 200 and we observed that $n_r = 500$ is a reasonable choice. The number of epochs needed for convergence depends on the settings and n_r. For the German 1hr data, the system improves for development set by just doing LRF which is the same behavior we observed in Afrikaans-Flemish experiments (Table 1); we attribute this improvement to the fact that the number of parameters that should be estimated with under-resourced language specific data has decreased. Moreover, when adaptation was applied, we observed improvements in all settings. This is most likely due to the fact that DNN model size is reduced. For example, for the setting with 14.85hr German data the number of the parameters in the multilingual DNN is reduced by a factor of 0.88. Finally, we experimented with factorization of ALL hidden weight layers. To this end, we took the best model obtained from the previous experiment; since the last weight layer of this model was already factorized, SVD was applied only on the hidden weight layers and the input weight layer was kept intact. In our experiment, $n_H = 1500$ and thus n_r needs to be chosen such that $(1500 \times n_r + n_r \times 1500) < 1500 \times 1500$. We set $n_r = 500$; so the number of parameters in each hidden weight layer is reduced by a factor of 0.66; afterwards, the whole network is retrained with multilingual data for 5 epochs. Table 4 compares the WERs for different factorized models before and after adaptation. From Table 4 we observe that the factorization of all hidden weight layers initially degrades the performance when 1hr and 5hr of data is available for German. This is not surprising as by applying LRF, a small amount of noise is added to all weight matrices and hence the network has moved away from the local optimum that was reached during training. However, when adapting the network from this starting point we ultimately reach a significantly better performance. This can be understood by the reasoning that LRF has created a network with fewer, but more relevant parameters. In 14.85hr scenario, we observe that the lost information after LRF of all layers can be well retrieved by multilingual retraining due to the availability of enough target language training data. Moreover, further improvement is achieved by adaptation like the other two scenarios. It is also important to note that the choice of learning rate in the adaptation phase is crucial; in our work, it was set to 0.0001.

Table 3. WER(%) for different multilingual settings where the last weight layer is factorized with $n_r = 500$.

Adaptation	German data					
	1hr		5hr		14.85hr	
	Dev.	Eval.	Dev.	Eval.	Dev.	Eval.
No	18.49	32.67	12.87	22.28	11.10	18.53
Yes	18.33	32.19	12.69	22.19	10.82	17.99

Table 4. Comparing WER(%) for German data using multilingual DNN where SVD is applied on ALL hidden layers ($n_r = 500$).

Set:	Dev.		Eval	
Adaptation	Yes	No	Yes	No
1hr	**16.86**	20.14	**29.72**	35.17
5hr	**11.82**	14.93	**19.81**	23.25
14.85hr	**10.16**	11.04	**16.79**	17.74

6. CONCLUSIONS

In this paper, LRF of multilingual DNN was studied for improving low-resource ASR. We examined different settings with different amount of data from target under-resourced language. First, we evaluate the impact of LRF of a multilingual DNN for two related languages, Flemish and Afrikaans, in a phone recognition task. Moreover, we conducted a word recognition task where German was the target language and four donor languages were taken from GlobalPhone dataset. From the combined set of experiments we may draw following conclusions: (i) In all scenarios, using extra data from donor languages improved the recognition results with 10% and more relatively, whereas the proximity of the donor language to the target language did not seem to be important. (ii) Low-Rank Factorization of the final weight layer gives a further improvement of 3-6% relative if followed by adaptation; (iii) Low-rank factorization of ALL hidden layers in combination with adaptation can boost the results with 7-10% relative in comparison with the normal multilingual DNN.

7. REFERENCES

[1] Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *ICASSP*. IEEE, 2013, pp. 7304–7308.

[2] Frantisek Grezl, Martin Karafiát, and Milos Janda, "Study of probabilistic and bottle-neck features in multilingual environment," in *ASRU*. IEEE, 2011, pp. 359–364.

[3] Mark JF Gales, Kate M Knill, Anton Ragni, and Shakti P Rath, "Speech recognition and keyword spotting for low resource languages: babel project research at CUED," in *Spoken Language Technologies for Under-Resourced Languages*, 2014, pp. 16–23.

[4] KM Knill, Mark JF Gales, Satish Prasad Rath, Philip C Woodland, Chenghui Zhang, and S-X Zhang, "Investigation of multilingual deep neural networks for spoken term detection," in *ASRU*. IEEE, 2013, pp. 138–143.

[5] Karel Veselỳ, Martin Karafiát, Frantisek Grézl, Milos Janda, and Ekaterina Egorova, "The language-independent bottleneck features," in *Workshop on Spoken Language Technology (SLT)*, 2012, pp. 336–341.

[6] Arnab Ghoshal, Pawel Swietojanski, and Steve Renals, "Multilingual training of deep neural networks," in *ICASSP*. IEEE, 2013, pp. 7319–7323.

[7] Frantisek Grézl, Martin Karafiát, and Karel Vesely, "Adaptation of multilingual stacked bottle-neck neural network structure for new language," in *ICASSP*. IEEE, 2014, pp. 7654–7658.

[8] Stephan Thomas, Michael L Seltzer, Kenneth Church, and Hynek Hermansky, "Deep neural network features and semi-supervised training for low resource speech recognition," in *ICASSP*. IEEE, 2013, pp. 6704–6708.

[9] David Imseng, Hervé Bourlard, et al., "Towards mixed language speech recognition systems," in *INTERSPEECH*, 2010, pp. 278–281.

[10] Tanja Schultz and Alex Waibel, "Fast bootstrapping of LVCSR systems with multilingual phoneme sets.," in *Eurospeech*, 1997.

[11] Ekaterina Egorova, K Vesely, Martin Karafiát, Milos Janda, and J Cernocky, "Manual and semi-automatic approaches to building a multilingual phoneme set," in *ICASSP*. IEEE, 2013, pp. 7324–7328.

[12] George E Dahl, Dong Yu, Li Deng, and Alex 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, 2012.

[13] Frantisek Grézl, Martin Karafiát, Stanislav Kontár, and Jan Cernocky, "Probabilistic and bottle-neck features for LVCSR of meetings," in *ICASSP*. IEEE, 2007, vol. 4, pp. IV–757.

[14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[15] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio, "Maxout networks," in *ICML*, 2013, pp. 1319–1327.

[16] Dong Yu, Frank Seide, Gang Li, and Li Deng, "Exploiting sparseness in deep neural networks for large vocabulary speech recognition," in *ICASSP*. IEEE, 2012, pp. 4409–4412.

[17] Shiliang Zhang, Yebo Bao, Pan Zhou, Hui Jiang, and Lirong Dai, "Improving deep neural networks for LVCSR using dropout and shrinking structure," in *ICASSP*. IEEE, 2014, pp. 6849–6853.

[18] Tara N Sainath, Brian Kingsbury, Vikas Sindhwani, Ebru Arisoy, and Bhuvana Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *ICASSP*. IEEE, 2013, pp. 6655–6659.

[19] Jian Xue, Jinyu Li, and Yifan Gong, "Restructuring of deep neural network acoustic models with singular value decomposition.," in *INTERSPEECH*, 2013, pp. 2365–2369.

[20] Yu Zhang, Ekapol Chuangsuwanich, and James Glass, "Extracting deep neural network bottleneck features using low-rank matrix factorization," in *ICASSP*, 2014, pp. 185–189.

[21] Aanchan Mohan and Richard Rose, "Multi-lingual speech recognition with low-rank multi-task deep neural networks," in *ICASSP*. IEEE, 2015, pp. 4994–4998.

[22] Hervé Bourlard, Nelson Morgan, Chuck Wooters, and Steve Renals, "CDNN: A context dependent neural network for continuous speech recognition," in *ICASSP*. IEEE, 1992, vol. 2, pp. 349–352.

[23] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, 2013, vol. 30.

[24] Mark JF Gales, "Semi-tied covariance matrices for hidden markov models," *Speech and Audio Processing, IEEE Transactions on*, vol. 7, no. 3, pp. 272–281, 1999.

[25] Daniel Povey, Xiaohui Zhang, and Sanjeev Khudanpur, "Parallel training of deep neural networks with natural gradient and parameter averaging," *arXiv preprint arXiv:1410.7455*, 2014.

[26] Daniel Povey et al., "The KALDI speech recognition toolkit," in *ASRU*, 2011, pp. 1–4.

[27] Nelleke Oostdijk, "The spoken Dutch corpus. overview and first evaluation," in *International Conference on Language Resources and Evaluation*, 2000, pp. 887–894.

[28] Etienne Barnard, Marelie H Davel, Charl van Heerden, Febe de Wet, and Jaco Badenhorst, "The NCHLT speech corpus of the South African languages," in *SLTU*, St Peterburg, Russia, May 2014, pp. 194–200.

[29] Reza Sahraeian, Dirk Van Compernolle, and Febe de Wet, "Under-resourced speech recognition based on the speech manifold," in *INTERSPEECH*, 2015, pp. 1255–1259.

[30] Tanja Schultz, Ngoc Thang Vu, and Tim Schlippe, "Globalphone: A multilingual text & speech database in 20 languages," in *ICASSP*. IEEE, 2013, pp. 8126–8130.

[31] Reza Sahraeian, Dirk Van Compernolle, and Febe de Wet, "Using generalized maxout networks and phoneme mapping for low resource ASR- a case study on Flemish-Afrikaans," in *Pattern Recognition Association of South Africa*, 2015, pp. 112–117.