ADAPTATION OF MULTILINGUAL STACKED BOTTLE-NECK NEURAL NETWORK STRUCTURE FOR NEW LANGUAGE

Frantisek Grézl, Martin Karafiát and Karel Veselý *

Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Brno, Czech Republic

ABSTRACT

The neural network based features became an inseparable part of state-of-the-art LVCSR systems. In order to perform well, the network has to be trained on a large amount of in-domain data. With the increasing emphasis on fast development of ASR system on limited resources, there is an effort to alleviate the need of in-domain data. To evaluate the effectiveness of other resources, we have trained the Stacked Bottle-Neck neural networks structure on multilingual data investigating several training strategies while treating the target language as the unseen one. Further, the systems were adapted to the target language by re-training. Finally, we evaluated the effect of adaptation of individual NNs in the Stacked Bottle-Neck structure to find out the optimal adaptation strategy. We have shown that the adaptation can significantly improve system performance over both, the multilingual network and network trained only on target data. The experiments were performed on Babel Year 1 data.

Index Terms— feature extraction, Bottle-neck features, neural network adaptation, multilingual neural networks, Stacked Bottle-Neck structure

1. INTRODUCTION

Quick delivery of ASR system for a new language is one of the challenges in the community. Hand in hand with the quick delivery comes limitation of available resources. Such scenario calls not only for automated construction of systems, that have been carefully designed and crafted "by hand" so far, but also for effective use of available resources. This is particularly important for features obtained from neural networks (NNs). In order to perform well, neural networks need to be trained on large amount of data. Moreover, this data needs to be transcribed to provide training targets. Unfortunately, the data collection and annotation is the most time- and money-consuming procedure.

This naturally raises the question whether features of sufficient quality can be obtained from different sources. The first study of portability of NN-based features was done in [1] where NNs trained on English data were applied on Mandarin and Levantine Arabic to produce probabilistic features. Consistent word error rate (WER) reduction was observed for both languages. In both cases however, the amount of training data would be itself sufficient for training good neural networks (100 and 70 hours respectively).

Our work [2] studied the possibility to train a multilingual neural network to be used to derive features for a new language. Several approaches to create the target set for the multilingual training were explored. We have shown that concatenation of phoneme sets is a safe and simple approach (further denoted as *one softmax*). However, performing merging on phoneme sets of individual languages can be beneficial depending on the language set and desired features.

The integral way of obtaining multilingual (or language independent) NN based (bottle-neck) features is presented in [3, 4]. Here, the NN last – output – softmax layer is divided into language-specific parts which makes the main body of the NN language-independent. Only one part of the output (N^{th}) layer corresponding to the language of a particular input-output training pair is active. Thus the outputs of the $(N-1)^{th}$ layer provide information which should be equally useful for classification of any of the language-specific targets used in the training. This leads to truly multilingually trained weights in NN except for the language-specific parts of the output layer. This approach will be further denoted as *block softmax*. This technique was modified by Heigold et al. in [5] and tested in multilingual DNN hybrid system.

When comparing the two approaches, we should note that one output layer (*one softmax*) for all language-specific targets performs, together with classification of the input vectors, indirectly also language identification as it has to distinguish between similar (or the same) targets from different languages.

All the above techniques assume no data for the target language, which is somewhat unrealistic scenario as there has to be some transcribed data to train the acoustic model. And since there is the data, forced alignment can be done on them and the input-output pairs can be used to adapt the neural network for feature extraction.

It should be also noted, that none of the techniques above led to significant improvement over the monolingual NN trained on the target language data only. On the other hand, adding target language data to multilingual training brought consistent improvement. This shows how important it is to present the target acoustic space during the NN training.

The adaptation on target language brings issues with language specific phonemes. Vu et al. [6, 7] suggest to solve this problem by approximation of such phonemes by several phonemes from the source languages that, in combination, have the characteristic of the target phoneme. Then, NN is retrained on target language using only the outputs (phonemes) belonging to it.

The adaptation of NN trained on large amount of data from one language to target domain with little data by final fine-tuning was proposed in [8] and extended to multilingual NN in [9]. This approach eliminates the necessity of identification and approximation of new phonemes.

Our work aims on adaptation of Stacked Bottle-Neck (SBN)

*This work supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0013. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government. The work was also supported by the IT4Innovations Centre of Excellence CZ.1.05/1.1.00/02.0070. M. Karafiat was supported by Grant Agency of the Czech Republic post-doctoral project No. P202/12/P604.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

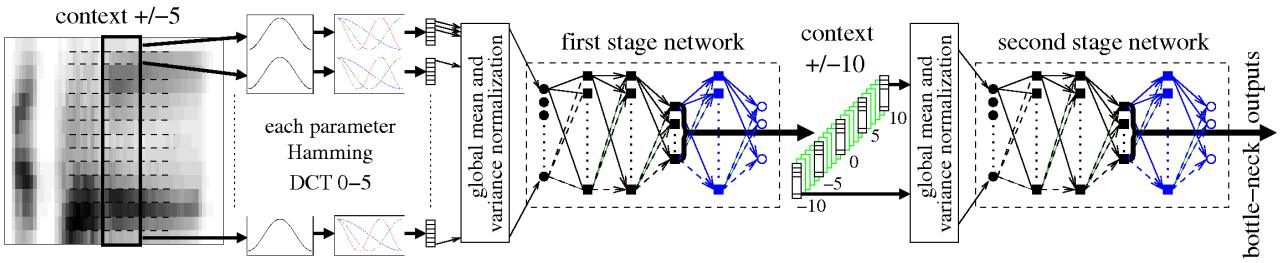

Fig. 1. Block diagram of Bottle-Neck feature extraction. The blue parts of NNs are used only during the training. The green frames in context gathering between the NNs are skipped. Only frames with shift -10, -5, 0, 5, 10 form the input to the second stage NN.

(originally called Universal Context) NN structure [10] with performance superior to just one NN. The fine-tuning approach [8] was applied. We investigated different methods of multilingual training as well as the adaptation (fine-tuning) of the individual parts of the SBN structure. Since the SBN is a structure of two NNs, the question which naturally arises is which NN in the structure should be adapted or if one of them can be trained on target data only.

The adaptation approach should benefit from large amount of data available for other languages that allows for proper training of sufficiently big NN. This network will serve as starting point for training a NN for the target language. As the amount of target data will be small, the retraining will be fast even for large NN which is the second benefit of the adaptation approach. At the end, a NN trained on large amount of diverse acoustic data but focused on target scenario will be available.

2. EXPERIMENTAL SETUP

2.1. Data

The IARPA Babel Program data[1] simulate a case of what one could collect in limited time from a completely new language: it consists of two parts: scripted (speakers read text through telephone channel) and conversational (spontaneous telephone conversations). The *dev* data contains conversational speech only. In this work we have used 4 language collections released during the first year of the project: Cantonese IARPA-babel101-v0.4c (CA), Pashto IARPA-babel104b-v0.4aY (PA), Turkish IARPA-babel105-v0.6 (TU) and Tagalog IARPA-babel106-v0.2g (TA). Two training scenarios are defined for each language – Full Language Pack (FLP), where all collected data are available for training; and Limited Language Pack (LLP) which consist only of one tenth of FLP. For multilingual training, the data from three FLP sets are used. The role of the target language is played by the LLP of the forth language. We have selected Pashto and Turkish as target languages for our experiments.

Vocabulary and language model (LM) training data are also defined with respect to the scenario. They basically consists of transcript of the given data pack. The overview of numbers of speakers and amounts of data is given in Tab. 1. The Turkish language is extensively agglutinative which increases number of words and in turn leads to high OOV rate.

Note, the amounts of raw audio are given, which in case of conversational speech, includes one recording for each side of the conversation. Thus the data contains huge portion of silence useless for training. The amounts of data used for training are given in Tab. 2.

[1]Collected by Appen `http://www.appenbutlerhill.com`

Table 1. *Data analysis*

Language	CA	PA	TU	TA
FLP speakers	952	1189	980	1096
FLP hours	194.9	194.3	192.7	193.9
LLP speakers	120	126	121	123
LLP hours	22.3	21.0	22.1	21.7
LM sentences	12043	9536	12025	12503
LM words	98569	108025	67706	60001
dictionary	7305	7025	12124	6295
dev speakers	20	121	18	120
dev hours	3.1	20.0	2.8	19.7
num. of words	13512	101803	11366	64489
OOV rate [%]	5.2	4.2	12.2	8.0

2.2. NNs for feature extraction

The features obtained using Neural Networks are the Bottle-Neck (BN) features. A structure of two 6-layer NNs is employed according to [10]. It is depicted in Fig. 1.

The NN input features are based on critical band energies (squared FFT magnitudes binned by Mel-scaled filter-bank and logarithmized) concatenated with estimates of F0 and probability of voicing. The estimation of F0 (implemented according to [11]) is based on normalized cross-correlation function. Dynamic programming is used for smoothing the estimates. Although it might seem not necessary to use the F0 and probability of voicing parameters for non-tonal languages, it turns out that these features are useful and their incorporation brings nice improvement of the final systems [12].

The conversation-side based mean subtraction is applied on the whole feature vector. 11 frames are stacked together. Hamming window followed by DCT consisting of 0th to 5th base are applied on the time trajectory of each parameter resulting in 102 coefficients on the first stage NN input.

The first stage NN has four hidden layers with 1500 units each except the BN layer. The BN layer is the third hidden layer and its size is 80 neurons. Its outputs are stacked over 21 frames and downsampled before entering the second stage NN. This NN has the same structure and sizes of hidden layers as the first one. The size of BN layer is 30 neurons and its outputs are the final outputs forming the BN features for GMM-HMM recognition system.

Neurons in both BN layers have linear activation functions as they were reported to provide better performance [13]. Before the features enter the NNs' input layer, global mean and variance normalization is performed.

The NN targets are phoneme states obtained by forced alignment of training data. The numbers of targets for individual lan-

Table 2. Baseline results for target languages (LLP). Only the SBN system is trained on FLP in one case and on LLP in the other.

Lang. Pack		# targets (phn. states)	hours	WER [%] HLDA-PLP	WER [%] BN
PA	FLP	216	64.7	76.9	62.0
	LLP	126	7.1		71.4
TU	FLP	126	56.6	75.5	61.9
	LLP	126	7.3		69.5

guage packs are given in Tab. 2. The forced alignments were generated with provided segmentations, however it was found that they still contain large portion of silence (50%–60%). Therefore, the silence on the ends of segments was stripped. Also segment was splitted when there was part of silence longer then 300 ms. This new segmentation reduced the amount of silence in the training data to 15%-20%. The final amounts of data used for NN training are also given in Tab. 2.

2.3. Recognition system

The training data for GMM-HMM system consists of LLP data of given language only.

First, a system based on standard Mel-PLP features is created. 13 PLP coefficients are generated together with first, second and third order derivatives. HLDA is estimated with Gaussian components as classes to reduce the dimensionality to 39. Then the conversation-side based mean and variance normalization is applied. Based on these features, baseline HLDA-PLP speech recognition system is trained using LLP data only. It is HMM-based cross-word tied-states triphone system, with approximately 4500 tied states and 18 Gaussian mixture components per state for all languages. It is trained from scratch using mix-up maximum likelihood training. The HLDA-PLP system is used for alignment of training data for NN training.

To train the system on Bottle-Neck features, the BN outputs are transformed by Maximum Likelihood Linear Transform (MLLT), which considers HMM states as classes. Then, new models are trained by single-pass retraining from HLDA-PLP baseline system. 12 Gaussian components per state were found to be sufficient for BN features trained from single-pass retraining. Next, 12 maximum likelihood iterations follow to better settle new HMMs in the new feature space.

Final word transcriptions are decoded using 3gram LM trained only on the transcriptions of LLP training data[2].

The results obtained with baseline HLDA-PLP systems are given in Tab. 2. The rather poor performance is given by the limited amount of data for acoustic as well as language model. Also note, that Turkish has quite high OOV rate.

3. EXPERIMENTS

3.1. Full and Limited language packs

To obtain the baselines, the NNs were first trained in monolingual manner on both, LLP and FLP. Training on LLP will provide us with the lower bound - if the evaluated technique does not exceed this threshold, it is not worth the effort. The FLP result will serve as upper bound. The closer the results will be to this value, the more the technique benefits from other resources. The results obtained with monolingual NNs are shown in Tab. 2.

[2]This is coherent to BABEL rules, where *the provided data only* can be used for system training.

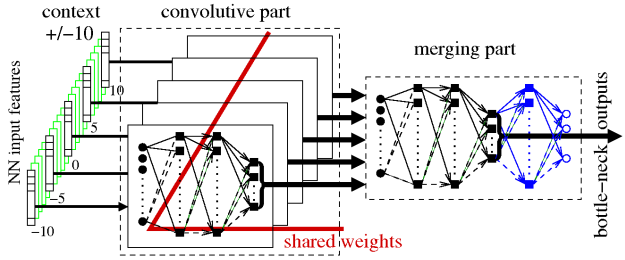

Fig. 2. Block diagram of Convolutive Bottle-Neck neural network.

We can see a dramatic drop in system performance when the NNs are trained just on LLP data. Compared with HLDA-PLP, the BN system achieves better performance even when NNs are trained on small amount of data. Note, that HMM systems are trained on the LLP data. The FLP data are used for training the SBN NNs only.

3.2. Multilingual NN

The next set of experiments is focused on the performance of BN features obtained from multilingual NNs. This case provides the starting point for the adaptation. The NNs were trained on FLPs of three languages leaving the target one out. Two approaches to train multilingual NNs are evaluated:

The first one – *one softmax* – discriminates between all targets of all languages. No mapping or clustering of phonemes was done. This simple approach turned out to perform the best in [2]. Thus the resulting NN has quite a large output layer containing all phonemes from all languages with one softmax activation function.

The second approach – *block softmax* – divides the output layer into parts according to individual languages. During the training, only the part of the output layer corresponding to the language the given target belongs to, is activated. This approach was successfully used in [4].

During our various experiments, we have observed that the *block softmax* approach has sometimes problem with training convergence and requires a learning rate reduction in order to get trained properly. To prevent this behavior, the trained *one softmax* NN was used as initialization for *block softmax* NN training. This initialization was done only for the first stage NN.

As the goal was to adapt the feature extraction NNs, the *Convolutive Bottleneck Network* (CBN) [13] technique which would allow to retrain the whole structure in one step was considered. This approach moves the context from the position between networks in front of the first one and the global normalization before the second stage NN is omitted. The first stage NN is present in the convolutive structure five times, but all its instances share the same weights. The block diagram of the CBN is shown in Fig. 2.

The *Convolutive Bottleneck Network* and *block softmax* approaches were used together to obtain multilingual NN. Since the initialization done in *block softmax* NN training proved to be useful, the training of CBN feature extractor is done in three phases: first, the first stage NN is trained in standard way with *one softmax*. Second, the NN is retrained with *block softmax*. Third, the whole structure is created, the trained first stage NN is appended with a random initialization of the second stage NN and trained. This approach is denoted as *Convolutive block softmax* further.

The results of various multilingual approaches are given in Tab. 3. It can be seen that all multilingual BN features slightly outperformed the LLP monolingual ones. This observation suggests that when the target domain becomes under-resourced, the use of

Table 3. WER [%] of target languages LLP systems with BN features obtained from multilingual SBN systems

target language	BN WER PA	TU
monolingual SBN	71.4	69.5
one softmax	69.9	68.6
block softmax	69.1	69.0
init block softmax	69.2	68.8
Convolutive block softmax	69.2	68.1

Table 4. WER [%] of target languages LLP systems with BN features obtained from adapted multilingual SBN system

target language	BN WER [%] PA	TU
monolingual SBN	71.4	69.5
one softmax	67.9	67.2
block softmax	67.1	66.9
init block softmax	67.1	66.3
Convolutive block softmax	66.7	65.6

other data is more efficient and multilingual system can perform well even on "unseen language" without the adaptation.

3.3. Adaptation of Multilingual NN

The adaptation of NN is done through retraining of already trained NN on target language data. The trained multilingual NN guarantees a good starting point which already produces good features as can be seen from Tab. 3. Retraining allows to shift the weights towards the acoustic space of target data.

Our approach to NN adaptation has two phases: First, only the last layer is trained. Since our initial NN is multilingual, the output layer has large number of units. We need to initialize the output layer randomly with the proper number of outputs matching the target language phoneme set. If the whole NN was retrained now, the error caused by the random weights in the last layer could be propagated deeper in the NN and the training could drift apart from the optimum. This is why the rest of NN is fixed and only the last layer is trained. In the second phase, the other layers are released and the whole NN is retrained once more. Since this retraining starts from an already trained network, the learning rate for this phase is set to one tenth of its original value[3].

Having several methods to obtain multilingual NN and several ways to apply the adaptation leads to two questions:

What to adapt? This set of experiments should tell us which system is the best for the subsequent adaptation. Here, all multilingual SBN feature extractors were adapted to target domain by fine-tuning of the second stage NN.

The results obtained are shown in Tab. 4. It can be seen that the WER decreases consistently with the increasing complexity of multilingual training approach. Adapted *Convolutive block softmax* NN reaches half of the interval given by the FLP and LLP monolingual systems. This shows that the technique is very efficient and can boost the system performance for domains with little data.

Where to adapt? This set of experiments should tell us which NN in the SBN system should be kept multilingual or be adapted or be trained only on target domain data. The most simple multilingual system was selected for these experiments.

[3]Learning rate of 0.004 is used for training from random weights, and 0.0004 for the retraining.

Table 5. WER [%] of target languages LLP systems with BN features obtained from various adaptation schemes of *one softmax* multilingual SBN system

SBN NN 1^{st} stage	2^{nd} stage	BN WER [%] PA	TU
LLP only	LLP only	71.4	69.5
multilingual	multilingual	69.9	68.6
multilingual	adapted	67.9	67.2
multilingual	LLP only	68.3	66.5
adapted	adapted	67.7	66.9
adapted	LLP only	68.0	65.0

Tab. 5 summarizes the possible combination of differently trained NNs in SBN system. The conclusion here is not so straight forward as in previous case. The different strategies to adapt a SBN system for Pashto give more or less the same results but we obtain considerable differences for Turkish. Moreover, the tendencies go in opposite directions – if the first stage NN is kept multilingual (or adapted) and the second stage NN is changed from adapted to purely target domain (LLP only) the WER increases for Pashto and decreases for Turkish. The only conclusion seems to be that it is beneficial to adapt the first stage NN. The second stage NN should be either adapted or trained on target data only. Note that for Turkish, considerable improvement over the best multilingual system – *convolutive block softmax* – was achieved.

Unfortunately, there are too many differences in the two systems to analyze this behavior in detail with the given setup – it is not possible to tell whether the behavior origins from the multilingual systems or is inherent in the target data set. A way to study this phenomenon at least to some extent would be to select another language on which both multilingual systems could be used and evaluated. If the trends continue to be opposite, the behavior is caused by the multilingual systems, if similar, then it is caused by the target data characteristics.

4. CONCLUSIONS

In our paper, we addressed multilingual training of Stacked Bottle-Neck neural network structure for feature extraction. While for languages with plentiful resources, the optimal approach is to train the BN-NN on the target data, limited resources call for re-using data from other languages. We have evaluated several techniques for multilingual training of neural networks. Two of them seem to be important: the first, *block softmax* separates the phone sets of individual languages in output layer and adds the phonemes of the target language to the final layer only, the second, *convolutive block softmax*, uses the same trick on a convolutive bottle-neck network defined in [13]. Both multilingual systems perform comparably well. The *block softmax* is easier to implement while the *convolutive* version provides possibility of adapting the whole structure at once.

The experiments where different parts of SBN system were trained/adapted differently revealed importance of proper adaptation scheme. Adapting the simplest *one softmax* SBN system in different ways results in better performance than adaptation of the most complex *convolutive block softmax* one for Turkish data.

Our future experiments will focus on the "Where to adapt" question with the goal of explanation of the opposite tendencies for different languages. We would also like to see if similar situation will occur in case of adapting differently trained multilingual systems. Our further interest lies in combination of several NNs approaches into one more complex structure of NNs.

5. REFERENCES

[1] A. Stolcke, F. Grézl, M.Y. Hwang, X. Lei, N. Morgan, and D. Vergyri, "Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons," in *Proceedings of ICASSP 2006*, Toulouse, FR, 2006, pp. 321–324.

[2] F. Grézl, M. Karafiát, and M Janda, "Study of probabilistic and bottle-neck features in multilingual environment," in *Proceedings of ASRU 2011*, 2011, pp. 359–364.

[3] Stefano Scanzio, Pietro Laface, Luciano Fissore, Roberto Gemello, and Franco Mana, "On the use of a multilingual neural network front-end," in *Proceedings of INTERSPEECH-2008*, 2008, pp. 2711–2714.

[4] Karel Veselý, Martin Karafiát, František Grézl, Miloš Janda, and Ekaterina Egorova, "The language-independent bottleneck features," in *Proceedings of IEEE 2012 Workshop on Spoken Language Technology*. 2012, pp. 336–341, IEEE Signal Processing Society.

[5] Georg Heigold, Vincent Vanhoucke, Andrew W. Senior, Patrick Nguyen, Marc'Aurelio Ranzato, Matthieu Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in *ICASSP*, 2013, pp. 8619–8623.

[6] Ngoc Thang Vu, Florian Metze, and Tanja Schultz, "Multilingual bottleneck features and its application for under-resourced languages," in *Proc. of The third International Workshop on Spoken Languages Technologies for Under-resourced Languages (SLTU'12)*, Cape Town, South Africa, 2012.

[7] Ngoc Thang Vu, Wojtek Breiter, Florian Metze, and Tanja Schultz, "An investigation on initialization schemes for multilayer perceptron training using multilingual data and their effect on ASR performance," in *Proc. Interspeech*, 2012.

[8] Samuel Thomas, Sriram Ganapathy, and Hynek Hermansky, "Multilingual MLP features for low-resource LVCSR systems," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4269–4272.

[9] Samuel Thomas, Michael Seltzer, Kenneth Church, and Hynek Hermansky, "Deep neural network features and semi-supervised training for low resource speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, Vancouver, Canada, may 2013, IEEE.

[10] F. Grézl, M. Karafiát, and L. Burget, "Investigation into bottleneck features for meeting speech recognition," in *Proc. Interspeech 2009*, Sep 2009, pp. 294–2950.

[11] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," in *Speech Coding and Synthesis*, W. B. Kleijn and K. Paliwal, Eds., New York, 1995, Elseviever.

[12] Martin Karafiát, František Grézl, Mirko Hannemann, Karel Veselý, and Jan "Honza" Černocký, "BUT BABEL System for Spontaneous Cantonese," in *Proceedings of Interspeech 2013*. 2013, number 8, pp. 2589–2593, International Speech Communication Association.

[13] Karel Veselý, Martin Karafiát, and František Grézl, "Convolutive bottleneck network features for LVCSR," in *Proceedings of ASRU 2011*. 2011, pp. 42–47, IEEE Signal Processing Society.