Building Real-time Speech Recognition without CMVN

Thai Son Nguyen, Matthias Sperber, Sebastian Stüker, Alex Waibel

Institute for Anthropomatics and Robotics Karlsruhe Institute of Technology
thai.nguyen@kit.edu

Abstract. Estimating cepstral mean and variance normalization (CMVN) in run-on and real-time settings poses several challenges. Using a moving average for variance and mean estimation requires a comparatively long history of data from a speaker which is not appropriate for short utterances or conversations. Using a pre-estimated global CMVN for speakers instead reduces the recognition performance due to potential mismatch between training and testing data. This paper investigates how to build a real-time run-on speech recognition system using acoustic features without applying CMVN. We propose a feature extraction architecture which can transform unnormalized log mel features to normalized bottleneck features without using historical data. We empirically show that mean and variance normalization is not critical for training neural networks on speech data. Using the proposed feature extraction, we achieved 4.1% word error rate reduction compared to global CMVN on the Skype conversations test set. We also reveal many cases when features without zero-mean can be learnt well by neural networks which stands in contrast to prior work.

Keywords: real-time speech recognition · feature normalization · neural network.

1 Introduction

Ceptral mean and variance normalization (CMVN) [22] and other normalization techniques (e.g., Cepstral mean normalization (CMN) [7]) are widely adopted in many neural network speech recognition systems due to several advantages. First, these techniques as shown in [22] make the recognizer more robust by canceling out environmental changes. Second, they help reducing the environment mismatch (e.g. background noises or microphones) between training and testing conditions. Last, the acoustic features after normalization have zero mean which is found critical for neural network training [13].

In offline situations, CMVN is usually applied at the utterance level or more ideally at the speaker level when many utterances of the same speaker are available. However, these approaches are not appropriate for real-time situations, because they require a certain amount of history to be available for the current speaker, and cannot handle unexpected speaker changes. Instead, mean and

variance can be continuously computed over a moving window of some hundred frames (e.g., 3 seconds [17, 1]). However, moving windows require the availability of historical data of at least a window-size, so that a delay must be introduced to handle the beginning of a new utterance. A third approach, computing mean and variance globally (e.g., [26, 19]) for all training and test data, avoids the delay but reduces the recognition performance due to potential data mismatch. CMVN can also be recursively updated in real-time as in [17], but this approach does not handle multiple speakers. Peddinti et al. [14] proposed to use mel-frequency cepstral coefficients (MFCC) without normalization for real-time speech recognition, as currently implemented in the Kaldi toolkit [15]. In their approach, i-vectors [3] which supply the information about the mean offset of the speaker's data are provided to every input so that the network itself can do feature normalization. However, i-vectors still require a certain amount of data of about 6 seconds per speaker.

In this paper, we investigate and employ feature extraction approaches which exhibit comparable performance to CMVN but do not require speaker historical data and are therefore better suited for real-time situations. Our contributions are summarized as follows:

- We contrast different CMVN methods and point out their respective advantages and limitations in a real-time feature extraction setting. We conclude that global CMVN is most desirable regarding real-time properties, although utterance- or speaker-based CMVN yield best recognition accuracy.
- We propose to use a two-step transformation method that is empirically shown to transform unnormalized log mel-filterbank (FBANK) features into suitable acoustic model inputs, without requiring historical data of the current speaker. Using this transformation, we show that the acoustic models trained on the new feature domain significantly outperform global CMVN.
- We identify a potential mismatch between training and testing data when acoustic models are trained on unnormalized data and propose to use data augmentation as a solution. We empirically show that retraining the feature extraction and the systems on a volume perturbation dataset can avoid the mismatch of audio volume and increase the recognition performance by up to 3.1%.
- We also observe and discuss cases when features without zero-mean can be learnt well by neural networks, which stands in contrast to prior work.

Without waiting for acoustic features being normalized correctly, a run-on speech recognition using our proposed feature extraction can process utterances of arbitrary length (or shortness). It can also handle situations where multiple speakers are sharing a single microphone.

2 Improving real-time feature extraction

We propose performing two steps to learn robust feature extraction for real-time speech recognition systems. First, the traditional mel-filter features are

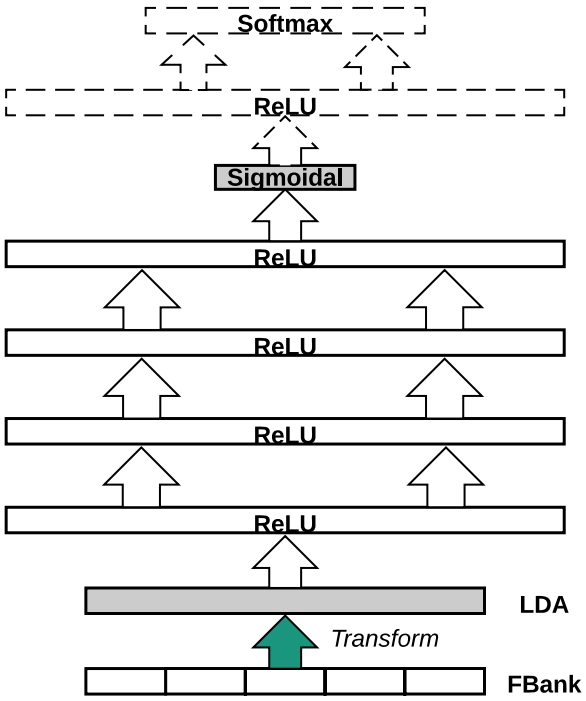

Fig. 1. Real-time feature extraction.

transformed into LDA domain and then fed into a bottleneck network to have final features which are value-normalized and easier to exploit by neural network models. Second, in order to increase the system's robustness, data augmentation is used for retraining both the feature extraction and the network model.

2.1 Using LDA transformed features

The most popular acoustic features such as MFCC or FBANK without normalization are problematic input for neural networks to learn. MFCC features usually span a wide range in every dimnension, e.g., [-93, 363] on typical data, while FBANK features only have positive values, e.g. in the range [0, 11.66]. We attempt to find a transformed domain such that the transformation can be performed in real-time. Linear Discriminant Analysis (LDA) [4] is usually used for dimensionality reduction, but here we propose to use it only for feature transformation. Using LDA, we compute a $d \times d$ linear transformation matrix which projects d-dimensional *FBANK* into a new domain with the same dimensionality. In this LDA domain, the features maintain the class-discriminatory information and can be mapped with their class-separability magnitudes according to the associated eigenvectors and eigenvalues. When used for dimensionality reduction, LDA is applied by keeping only k (much smaller than d) features

with largest magnitudes. We, however, use all d-dimensional features in our in models because we observed better system performance.

2.2 Using normalized bottleneck features

As will be experimentally shown, optimizing single network models on unnormalized data can be hard. Dealing with this situation, our idea is to train a first network model for extracting *length-normalized features*. Later we can use a second network to perform the real classification task. Figure 1 illustrates our proposed feature extraction architecture. The input of the network can be unnormalized FBANK or LDA-transformed features. We employ some rectifier [25] layers on top of the input layer, followed by a narrow (bottleneck) layer of 42 sigmoidal units. Two last layers which will be discarded after the training include one rectifier and the final softmax. Since the training of this feature extraction optimizes phonemes classification, the extracted features at the bottleneck layer are supposed to be significant for class-discrimination. When using a sigmoidal activiation function, we can obtain bottleneck features that are normalized to be in a small range which can be easier handled by the second network. We experimented with sigmoidal functions, the logistic function which has range if [0,1] and the hyperbolic tangent which produces features in range [-1,1].

Different from [8, 24], the proposed feature extraction is able to handle both normalized and unnormalized inputs. It does not suffer from vanishing gradients and does not need pre-training which significantly reduces the training time. Applying this feature extraction in real-time can be considered as adding more hidden units to the classification network, which linearly increases the computation time (i.e. 25% in our experiments).

2.3 Increasing robustness by data augmentation

As will be explored in this paper, the neural network systems trained on unnormalized features potentially need to deal with environment mismatch between training and testing. In speech recognition, mismatches such as different speech variations, background noises or microphones, can lead to a significant drop of recognition performance. In this paper, we analyze the robustness of our proposed feature extraction against the mismatch of audio volume conditions and improve it with the help of data augmentation.

Data augmentation applied to speech recognition has been explored in many studies. In [10, 16], corrupting clean training speech with noise improved the speech recognizer against noisy speech. Using vocal tract length perturbation [11] has shown gains on TIMIT. In [12, 14], training with speed and volume perturbations datasets increased the system performance on several LVCSR tasks. In this paper, we only consider data augmentation by performing volume perturbation.

3 Experimental setup

3.1 Training and test data

In our experiments we used a large training set of 460 hours. This dataset is the result of combining TED-LIUM [18], Quaero [21] and Broadcast News [9] corpora. Our three evaluation sets include TED-LIUM test, tst2013 from the IWSLT evaluation campaign [2] and the English set from the MSLT corpus [5] which contains conversations over Skype.

The volume perturbations were done as suggested by [14] where each recording was scaled with a random variable using *sox*. We set the random variable within the range [0.2,2] for all recordings in the training data set. Then they were added to the original training set to form the augmented dataset. To investigate the robustness against volume mismatch, we used the ranges [0.2, 0.6] and [1.6, 2.0] for the all recordings of the tst2013 set to create a perturbed test set.

3.2 System description

All the network models used roughly same number of input features (i.e, 440 FBANK and 462 LDA or bottleneck features) and were trained using the cross-entropy loss function to predict 8,000 context-dependent phonemes. Rectifier networks were constructed of 6 hidden layers with 1,600 units per layer. For sigmoidal networks, we used 5 hidden layers of 2,000 units and performed pre-training with denoising auto-encoders [23]. For our convolution neural network (CNN), we used the best architecture from [20] which includes two convolutional layers of 256 hidden units with filter size 9 and a max pool size of 3, followed by 4 fully connected layers with 1,024 units. However, we did not use delta and delta-delta features for consistent comparisons between models.

The tests were performed with the Janus Recognition Toolkit (JRTK) [6] with a 4-gram language model and a vocabulary of more than 150,000 words.

4 Results

4.1 Using normalized and unnormalized features

In Table 1, we compare the systems using different CMVN methods against various systems trained on unnormalized FBANK features. Using our training data, CMVN systems performance depends on the amount of available speaker historical data. Normalization at speaker level yielded the best performance, followed by utterance level normalization and normalizations with windows 300 frames in length. The results on the perturbed test set show an interesting fact that these normalizations produce robust features to the changes of audio volume. Global CMVN is less optimal than other normalizations (7.1% rel. increase in WER compared to speaker level). However, real-time system may have to adopt this method, in order to achieve acceptable latency.

For the normalized features, the gap between sigmoidal and rectifier [25] networks appears small. However, when using the features without normalization which have only positive values in a large range [0, 11.66], optimizing sigmoidal networks for good convergence becomes difficult. We had to reduce the initial learning rate by a factor of ten compared to normalized features. The training then converged at a poor local minimum and caused worse classification performance. The situation changed with the rectifier network. We were able to keep the same learning rate and the training converged with the same pattern. However, it suffers from a 7.3% rel. increase in WER compared to global CMVN. Switching to a CNN network gave a further improvements, however its result is still not good as that of the CMVN systems. These results demonstrate the difficulties when training single network models on unnormalized FBANK features.

The increase in WER of the systems using unnormalized features and global normalized features on the perturbed test set indicates that they may be sensitive to volume mismatch between training and test data.

Table 1. Word error rates of various systems using 40 log mel-filter bank features with and without CMVN

CMVN	**Network Type**	**tst2013**	**tst2013-vp**
Speaker	sigmoid	15.5	15.5
Utterance	sigmoid	15.8	15.8
Window	sigmoid	16.2	16.4
Global	sigmoid	16.6	17.3
Global	rectifier	16.5	17.1
none	sigmoid	22.3	23.2
none	rectifier	17.7	18.0
none	rectifier (CNN)	17.1	17.6

4.2 Using LDA transformed features

Table 2 compares the efficiency of different LDA transformations applied to unnormalized features. Such a conventional approach which reduces dimensionality of 440 features of 11 consecutive frames down to 42 and then stacks again for 11 frames, does not show clear improvements. When transforming 40 FBANK features without reduction and stacking 11 adjacent frames of LDA features as the network input, the systems improved. Further improvement was achieved when transforming 440 features of 11 consecutive frames via LDA and using them as network input. Interestingly, the transformed features which are in the range [-14.95, 14.50] without zero-mean are better than FBANK with global CMVN. When applying global mean and variance normalization again on these LDA features, the performance even got worse showing that the normalization is unnecessary for this training data.

The large degradation (5.4% rel. in WER) of the performance on the perturbed test set presents the need of a method for improving LDA features against possible environment mismatch.

Table 2. The systems with LDA features.

LDA Feature	CMVN	DNN	tst2013	tst2013-vp
Reduction	none	rectifier	17.5	17.8
Full-40	none	rectifier	16.8	17.4
Full-440	none	rectifier	**16.2**	**17.0**
Full-440	Global	rectifier	16.5	17.2
Full-440	none	sigmoid	16.8	17.7

4.3 Using normalized bottleneck features

The proposed bottleneck feature extraction shows its advantages when applied to both unnormalized FBANK and LDA features and produces improved features. The same networks trained on the bottleneck features showed relative reduction of 7.4% and 4.9% as shown in Table 3. The extracted bottleneck features are in a normalized range [0, 1] or [-1, 1], so a sigmoid network can be trained well showing again that we do not need to apply mean normalization.

When evaluating against the mismatch test set, we found that the extracted features are more stable to speech variations indicating the normalized bottleneck network may be automatically forced to learn robust features.

Table 3. The system with normalized bottleneck (BN).

Feature	BN Type	DNN	tst2013	tst2013-vp
FBANK	*sigmoid*	rectifier	16.4	16.6
FBANK	*sigmoid*	sigmoid	16.5	16.8
LDA	*sigmoid*	rectifier	**15.5**	**15.8**
LDA	*tanh*	rectifier	15.5	15.8

4.4 Using data augmentation

When retraining the feature extraction and the systems on the augmented dataset, we obtained improvements on both test sets as presented in Table 4. Now, there are only small gaps between the two test sets indicating robustness of the models and the effectiveness of the proposed data augmentation. Retraining improves the recognition performance in general (i.e. 3.1% rel. for the bottleneck system using FBANK). We could only achieve small gains for the systems using LDA features. This can be the case when we only retrained the feature extractions and the systems without re-estimating the LDA transformation.

Table 4. The systems trained with data augmentation

Feature	tst2013	tst2013-vp
FBANK	17.3 *(2.3%)*	17.4 *(3.3%)*
LDA	16.1 *(0.6%)*	16.3 *(4.7%)*
BN-FBANK	15.9 *(3.1%)*	15.9 *(4.2%)*
BN-LDA	15.4 *(0.7%)*	15.6 *(1.3%)*

4.5 Comparison on different test sets

Table 5 compares the results of our systems on two different test sets. TED-LIUM set contains 11 TED talks while the MSLT set is a collection of 3,000 utterances of recorded Skype conversations. There is no speaker information for the MSLT set and more than a half of the utterances are less than 3 seconds.

In different online domains, our proposed feature extraction can reduce the WER by 12.1% relative compared to the global CMVN. Comparing to another system of the same complexity which uses the bottleneck architecture from [8], we also achieve a significant improvement.

Table 5. Results on the TED-LIUM and MSLT test sets

Feature	CMVN	TED-LIUM	MSLT2016
FBANK	Global	9.8	33.9
BN [8]	Global	9.2	30.9
FBANK	*none*	10.3	35.0
BN-LDA	*none*	**8.7**	**29.8**

5 Conclusions

We have presented a novel and effective feature extraction for real-time and run-on speech recognition. Our proposed two-step transformation is able to transform unnormalized log mel-filterbank features into useful value-normalized features. These features can be used directly for neural networks or Gaussian mixture models without further normalization. Applying this feature extraction approach hides the involvement of explicit normalization such as CMVN. Other real-time speech applications (such as speaker recognition) can also benefit from our method.

References

1. Alam, M.J., Ouellet, P., Kenny, P., OShaughnessy, D.: Comparative evaluation of feature normalization techniques for speaker verification. In: International Conference on Nonlinear Speech Processing. pp. 246–253. Springer (2011)

2. Cettolo, M., Niehues, J., Stüker, S., Bentivogli, L., Frederico, M.: Report on the 10th IWSLT evaluation campaign. In: The International Workshop on Spoken Language Translation (IWSLT) 2013 (2013)
3. Dehak, N., Kenny, P.J., Dehak, R., Dumouchel, P., Ouellet, P.: Front-end factor analysis for speaker verification. IEEE Transactions on Audio, Speech, and Language Processing **19**(4), 788–798 (2011)
4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification. Wiley-Interscience (2000)
5. Federmann, C., Lewis, W.D.: Microsoft speech language translation (MSLT) corpus: The IWSLT 2016 release for english, french and german. In: The International Workshop on Spoken Language Translation (IWSLT) 2016 (2016)
6. Finke, M., Geutner, P., Hild, H., Kemp, T., es, K.R., Westphal, M.: The karlsruhe VERBMOBIL speech recognition engine. In: Proc. of ICASSP (1997)
7. Furui, S.: Cepstral analysis technique for automatic speaker verification. IEEE Transactions on Acoustics, Speech, and Signal Processing **29**(2), 254–272 (1981)
8. Gehring, J., Miao, Y., Metze, F., Waibel, A.: Extracting deep bottleneck features using stacked auto-encoders. In: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. pp. 3377–3381. IEEE (2013)
9. Graff, D.: The 1996 broadcast news speech and language-model corpus. In: Proceedings of the DARPA Workshop on Spoken Language technology. (1997)
10. Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., et al.: Deep speech: Scaling up end-to-end speech recognition. arXiv preprint arXiv:1412.5567 (2014)
11. Jaitly, N., Hinton, G.E.: Vocal tract length perturbation (VTLP) improves speech recognition. In: Proc. ICML Workshop on Deep Learning for Audio, Speech and Language (2013)
12. Ko, T., Peddinti, V., Povey, D., Khudanpur, S.: Audio augmentation for speech recognition. In: INTERSPEECH. pp. 3586–3589 (2015)
13. LeCun, Y.A., Bottou, L., Orr, G.B., Müller, K.R.: Efficient backprop. In: Neural networks: Tricks of the trade, pp. 9–48. Springer (2012)
14. Peddinti, V., Povey, D., Khudanpur, S.: A time delay neural network architecture for efficient modeling of long temporal contexts. In: INTERSPEECH. pp. 3214–3218 (2015)
15. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K.: The Kaldi speech recognition toolkit. In: IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society (Dec 2011)
16. Prisyach, T., Mendelev, V., Ubskiy, D.: Data augmentation for training of noise robust acoustic models. In: International Conference on Analysis of Images, Social Networks and Texts. pp. 17–25. Springer (2016)
17. Pujol, P., Macho, D., Nadeu, C.: On real-time mean-and-variance normalization of speech recognition features. In: Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on. vol. 1, pp. I–I. IEEE (2006)
18. Rousseau, A., Deléglise, P., Estève, Y.: Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks. In: Proc. of LREC (2014)
19. Sainath, T.N., Kingsbury, B., Mohamed, A.r., Ramabhadran, B.: Learning filter banks within a deep neural network framework. In: Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on. pp. 297–302. IEEE (2013)

20. Sainath, T.N., Mohamed, A.r., Kingsbury, B., Ramabhadran, B.: Deep convolutional neural networks for LVCSR. In: Acoustics, speech and signal processing (ICASSP), 2013 IEEE international conference on. pp. 8614–8618. IEEE (2013)
21. Stüker, S., Kilgour, K., Kraft, F.: Quaero 2010 speech-to-text evaluation systems. In: High Performance Computing in Science and Engineering'11, pp. 607–618. Springer (2012)
22. Viikki, O., Laurila, K.: Cepstral domain segmental feature vector normalization for noise robust speech recognition. Speech Communication **25**(1), 133–147 (1998)
23. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: The 25th international conference on Machine learning. pp. 1096–1103. ACM (2008)
24. Yu, D., Seltzer, M.L.: Improved bottleneck features using pretrained deep neural networks. In: Interspeech. vol. 237, p. 240 (2011)
25. Zeiler, M.D., Ranzato, M., Monga, R., Mao, M., Yang, K., Le, Q.V., Nguyen, P., Senior, A., Vanhoucke, V., Dean, J., et al.: On rectified linear units for speech processing. In: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. pp. 3517–3521. IEEE (2013)
26. Zeyer, A., Schlüter, R., Ney, H.: Towards online-recognition with deep bidirectional LSTM acoustic models. Interspeech 2016 pp. 3424–3428 (2016)