A Perceptually Inspired Data Augmentation Method for Noise Robust CNN Acoustic Models

László Tóth[1(✉)], György Kovács[1,2,3], and Dirk Van Compernolle[4]

[1] Department of Informatics, University of Szeged, Szeged, Hungary
{tothl,gykovacs}@inf.u-szeged.hu
[2] MTA-SZTE Research Group on Artificial Intelligence, Szeged, Hungary
[3] MTA Research Institute for Linguistics, Budapest, Hungary
[4] KU Leuven Department of Electrical Engineering (ESAT), Leuven, Belgium
compi@esat.kuleuven.be

Abstract. Here, we present a data augmentation method that improves the robustness of convolutional neural network-based speech recognizers to additive noise. The proposed technique has its roots in the input dropout method because it discards a subset of the input features. However, instead of doing this in a completely random fashion, we introduce two simple heuristics that select the less reliable components of the spectrum of the speech signal as candidates for dropout. The first heuristic retains spectro-temporal maxima, while the second is based on a crude estimation of spectral dominance. The selected components are always retained, while the dropout step discards or retains the unselected ones in a probabilistic manner. Due to the randomness involved in dropout, the whole process may be interpreted as a data augmentation method that perturbs the data by creating new data instances from the existing ones on the fly. We evaluated the method on the Aurora-4 corpus just using the clean training data set, and we got relative word error rate reductions between 22% and 25%.

Keywords: Robust speech recognition
Convolutional neural networks · Data augmentation · Input dropout

1 Introduction

In the context of machine learning, data augmentation refers to methods that seek to increase the quantity of training data. In the field of speech recognition, some authors use a broad definition that also interprets the involvement of unsupervised and other-language data as data augmentation [29]. Here, however, we focus on the stricter, and more usual definition where the augmented training data is obtained by artificial transformations and perturbations of the original data set. The general purpose of data augmentation is to increase the amount of training data, and thus alleviate the problems related to overfitting in a

A. Karpov et al. (Eds.): SPECOM 2018, LNAI 11096, pp. 697–706, 2018.
https://doi.org/10.1007/978-3-319-99579-3_71

low-resource scenario [21,29]. However, if we know that our training data underrepresents the variance of real-life data from a certain aspect, then we can apply a dedicated transformation that perturbs the data with respect to that given parameter. Jaitly and Hinton applied vocal tract length perturbation (VTLP) to decrease the sensitivity of the recognizer to speaker characteristics [20]. A more general form of elastic spectral distortion was proposed by Kanda et al. with a similar goal [21]. VTLP was applied by several other authors for large vocabulary recognition tasks as well [8,29]. Ko et al. manipulated the speed of speech signals, and in spite of the simplicity of this technique, they obtained significant gains on large vocabulary tasks [22]. In another paper, the same team simulated reverberant conditions in a far-field task using data augmentation [23]. Hartmann et al. applied noise addition in combination with speed and speaker-based perturbations [13]. Some recent papers also used a combination of the data augmentation methods discussed above [17,28].

In this study, we apply data augmentation in the framework of convolutional neural networks (CNNs), and our goal is to handle additive noise. In such cases, a reasonable strategy would be to augment the training data by adding noise to it. Optimally, that would require some sort of prior knowledge about the type of noise. Otherwise, the best one could do is to add various types of artificial or real-life noises with various signal-to-noise ratios [17]. However, our modeling assumption here is that we do not have samples of the actual noise. In accord with this, we will train our models using only clean training data, and we consider the training scenario that makes use of noisy samples as well to be a different task.

Our data augmentation approach is motivated be the observation that data augmentation and dropout are closely related techniques [5]. The dropout method increases the robustness of a deep neural network by randomly discarding neurons during training, thus forcing them to rely on each other to a lesser extent. Dropout can be applied to the input of the network as well, which in the general case corresponds to discarding input components in a total random fashion [15]. While it is hard to recommend any better dropout strategy for a general machine learning task, but in the case of speech recognition we do have some prior knowledge about the role of the various spectro-temporal input features. Hence, instead of randomly choosing the dropped features as usual, here we propose to discard those input values that we assume to be more vulnerable to noise. We introduce two perceptually motivated strategies to select the noise-robust components of the mel-spectral input representation of our CNN. One of these simply retains the spectro-temporal maxima, as spectral peaks are known to be essential for speech intelligibility [14]. The other heuristic looks for spectrally dominant components, and we expect it to retain formant-like information [11]. The proposed dropout scheme retains the input components suggested by the heuristic, while it discards or keeps the remaining ones with a predefined probability. Hence, instead of adding noise, we perturb the data by discarding those spectral pixels that we assume to be noisy. The stochasticity of dropout guarantees the variability of the artificially generated data. The augmentation

process is performed online for the actual batch of training data, and requires only negligible additional computation.

Of course, lot of papers apply some sort of spectral subtraction or spectral peak selection to increase noise robustness (e.g. [10,19,25]). The key difference is that these solutions typically apply the same (usually signal processing-based) method during both training and testing, while in our case the spectrum is manipulated only in the training phase. That is, any performance gain on the test data will not be a direct result of the input processing method, but will rather be due to the network that distilled some extra knowledge from the additional samples created during data augmentation.

2 Perceptually Inspired Data Augmentation

The input of our CNN is a standard mel-spectral time-frequency representation [1,30], which we call the FBANK features. We used 42 filterbanks spanning the full frequency range, a 25 msec frame length, and a 10 msec frame shift. The adjacent bands were grouped to form wider channels, which are processed by separate sets of convolutional neurons in the CNN. We formed 9 such channels each covering 9 mel-bands, with an overlap of 5 bands. As the network processes 9 neighboring frames as one block of input data, this means that the convolutional filters operate on 9×9 spectro-temporal patches. More implementation details about our CNN can be found in our earlier articles [24,32].

To increase the robustness of the network to additive noise, one standard augmentation approach is to perturb the training set by adding noise to it. Here, without prior knowledge about the noise, the best we could do would be to perform the noise injection using random noise. Instead of this, our proposed augmentation strategy relies on the very simple assumption that spectral peaks are less vulnerable to additive noise. Furthermore, we exploit the finding that data augmentation and dropout are two closely related techniques [5]. Combining these two observations, we propose an augmentation strategy that resembles input dropout in the sense that it randomly deletes a subset of the input features. However, the selection of the dropped pixels is not completely random, but is governed by a perceptually motivated heuristic. This heuristic labels each component of the actual block of input (i.e. 9×9 spectro-temporal features) as 'vulnerable' or 'not vulnerable'. While this decision is not probabilistic, preserving the randomness of dropout is crucial for the variability of the generated data. Here, we introduce randomness by either deleting or retaining *all vulnerable pixels* with probability p.

2.1 Strategy A

We shall assume that our training data consists of clean recordings, and we seek strategies to select those spectral components that are presumably the least affected by noise. By retaining these parts and randomly dropping the rest during training, we force the network to focus on the more reliable components.

Our first strategy exploits the simple fact that spectral peaks are less sensitive to additive noise than spectral valleys. It is also well known that the spectral peaks carry the bulk of information required for speech intelligibility [14]. Hence, our first strategy simply retains those input values that have the highest amplitude. We experimented with preserving just 10% or 20% of the components from each 9×9 spectro-temporal block. In our detailed evaluation, we decided to work with 10% for two reasons. First, it gave us slightly better results; and second, this way the number of features retained was about the same as that for our second strategy (see below).

2.2 Strategy B

Spectral masking plays a crucial role in human speech perception [27]. Masking tells us that some spectral components may be discarded, and this is heavily exploited by low-bitrate speech coding algorithms (e.g. [31,33]). Moreover, there is psychoacoustic evidence that speech features are distributed over spectral bands as wide as one octave [3]. In vowel perception, formant integration over 3.5-4 Bark wide bands has been observed [7]. Although the size of our spectro-temporal windows was tuned experimentally, their height of 9 mel-filters roughly coincides with the values mentioned above (the distance between the centers of our mel-filters is about 0.5 Bark). Based on this, our second strategy imitated this spectral dominance effect by just keeping the highest-amplitude component in each column of the 9×9 spectro-temporal windows. With this approach, the 9 windows that were used to cover the whole frequency range retained exactly 9 spectral components at each time instance. As a comparison, the low-bitrate speech coder of Wan et al. retains 8 spectral lines per frame [33]. This coder applies the In-Synchrony-Bands-Spectrum (SBS) auditory model of Ghitza to select the spectral components to be preserved [11]. Although we could have

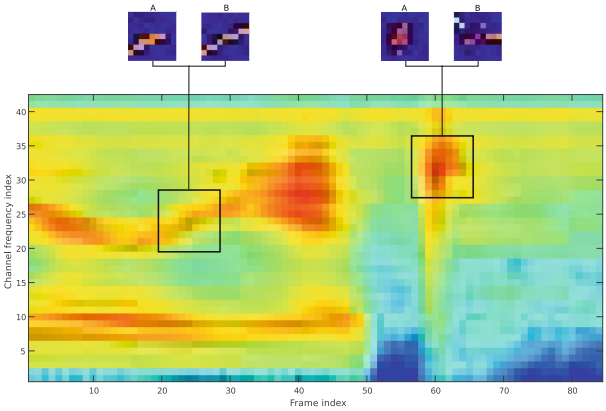

Fig. 1. An illustration of the result of data augmentation strategies A and B for a formant and for a burst.

applied a more sophisticated auditory model here, we first wanted to prove that the concept is viable, and leave the use of auditory models for future refinement.

Figure 1 shows two examples of what is retained from a local spectro-temporal block after being processed by the two different strategies. As the first example, we chose a position where the mel-spectrogram shows a clearly observable formant movement. In this case, the spectral dominance-based method (Strategy B) nicely follows the track of the formant, while Strategy A results in a thicker, but not continuous line. In the second example the window is fitted on a burst. In this case the output of strategy A is much closer to what our intuition suggests as optimal. Later, in Sect. 4, we will see experimentally which strategy proves to be better in practice.

2.3 CNN Training with Online Data Augmentation

The generation of the augmented samples is integrated into the training process in an online manner. After reading in the next batch of training data, we first decide whether to perform data augmentation on *all* the data vectors within the given batch or leave it unaltered. This is similar to a modified version of dropout which applies the same dropout mask within a given mini-batch [12]. The decision is made in a random manner, and the probability that the actual batch should be transformed will be denoted by p. Next, we decide on the number of the convolutional channels to be modified. For this, we generate another random number in the range of 1 to N (where $N = 9$ equals the total number of channels). Lastly, if p selected the actual batch for modification, then we perform the spectral manipulation (Strategy A or Strategy B) on N randomly chosen spectro-temporal input windows. The optimal values for the parameters p and N will be found experimentally later on.

Data augmentation by definition means that the augmented training set contains more data instances than the original one. In our implementation, generating more instances corresponds simply to let the algorithm perform more iterations through the data. Thanks to the stochastic nature of the augmentation procedure, the risk of overfitting the training set is much smaller than for the original data set. In spite of the assumed advantage of allowing more iterations, here in the experiments we did not increase the number of training steps, which can be interpreted as augmenting the data set, and then downsampling it to the original size. While this resulted in a slight performance loss, this way the time cost of the training remained the same as before, so we can claim that our proposed technique has no or only negligible processing time overhead.

3 Experimental Set-Up

We evaluated the proposed method on the Aurora-4 database [16]. The test set of Aurora-4 consists of 4620 utterances, with a subset recorded using a Sennheiser close-talking microphone, and a subset recorded using a set of secondary microphones. Both of these subsets contain a clean subset and one consisting of the

noise-corrupted versions of the same utterances. The final subsets are called test set A (clean recordings with the Sennheiser microphone), set B (noise-corrupted version of set A), set C (clean recordings with secondary microphones), and set D (noise-corrupted version of set C). The database contains two training sets, namely the clean set and the multi-condition set, both consisting of 7138 utterances. The multi-condition set contains samples from the secondary microphones and the various types of noisy conditions, while the clean training set consists of only the clean training data from the Sennheiser microphone. Here we will train our CNN using the clean training set, as our whole concept is based on the assumption that we have neither noisy training data, nor samples from the noise, and the goal of our data augmentation methods is to train the network which spectral components of the clean data are reliable.

We used the Kaldi toolkit and its Aurora-4 recipe to train a HMM/GMM model. We performed forced alignment with this model, and utilized the acquired frame-level state labels as training targets to replace Kaldi's DNN with our in-house CNN implementation. The decoding step was again performed with Kaldi, using the standard tri-gram language model and the 5k word vocabulary. Our CNN was trained with backpropagation using the frame-level cross-entropy error function. A random 10% of the training set was held out as the development set used for the early stopping of training, and for tuning the meta-parameters.

4 Results and Discussion

First, we evaluated the two data augmentation strategies by varying the meta-parameter values p and N. Table 1 shows the frame-level error rates obtained on the train and development sets for various values of p and N using strategy A (the baseline score is given in the $p = 0$ column). We see that the error rate on the train set grows steadily when we increase p and N. However, on the development set the increase of the error rate is much smaller, and its value is quite stable in the $p = 0.7 - 0.8, N = 7 - 9$ range. The same analysis for augmentation strategy B (see Table 2) shows similar trends, although in this case the error rate increases with a slower rate, showing a wider plateau.

Normally, we select those meta-parameter values for testing which yield the best performance on the development set. However, in this case we know that the development set does not represent the testing conditions faithfully, as it contains

Table 1. The frame error rates (%) on the train, development and multi-conditional development sets using augmentation strategy A.

	Train					Clean Dev					Noisy Dev				
N \ P	*0*	*0.6*	*0.7*	*0.8*	*0.9*	*0*	*0.6*	*0.7*	*0.8*	*0.9*	*0*	*0.6*	*0.7*	*0.8*	*0.9*
7		28.5	29.7	31.2	31.1		29.2	29.7	**30.4**	**30.3**		53.9	54.0	54.6	54.2
8	22.9	29.8	31.1	31.5	32.6	28.9	29.4	**30.0**	**30.1**	30.6	56.7	53.7	53.7	53.6	54.1
9		31.5	33.2	34.1	35.2		29.5	**30.2**	30.5	30.9		53.5	**53.5**	53.7	53.9

Table 2. The frame error rates (%) on the train, development and multi-conditional development sets using augmentation strategy B.

	Train					Clean Dev					Noisy Dev				
N \ P	*0*	*0.7*	*0.8*	*0.9*	*1.0*	*0*	*0.7*	*0.8*	*0.9*	*1.0*	*0*	*0.7*	*0.8*	*0.9*	*1.0*
7		29.0	29.4	30.0	29.1		29.7	29.8	**30.2**	**29.8**		54.4	54.6	54.8	54.1
8	22.9	29.7	30.1	30.7	30.1	28.9	29.8	**30.0**	**30.3**	**30.0**	56.7	54.6	54.3	55.0	54.4
9		30.4	30.9	30.5	32.0		**30.0**	**30.2**	**30.0**	30.9		54.3	54.4	**54.0**	55.0

only clean recordings, while test set will be noisy. Moreover, we are willing to sacrifice some accuracy under clean conditions, if it improves the results under noisy conditions.

As candidates for p and N, in the tables we marked those scores in bold where the error surface on the development set has a plateau, and the error rate increase over the baseline is below 0.5%. However, we found no convincing strategy to select just one p and N value from among these values. Hence, we evaluated our models on the development set of the multi-conditional training scenario, which contains noisy samples as well. The resulting frame error rates are shown on the right hand side of Tables 1 and 2. Similar to the development set, the error rates obtained are convincingly stable with respect to p and N, and in this case they also beat the baseline. Favoring larger p and N values, for the final tests we chose $p = 0.7, N = 9$ for Strategy A, and $p = 0.9, N = 9$ for Strategy B. With these parameter values the word error rate on the test set for Strategy A was 25.6%, which corresponds to a relative error rate reduction of 24% over the baseline of 33.7%. Strategy B performed slightly worse, attaining a word error rate of 26.0%. Having chosen the meta-parameter values, we peeked into the test data (for Strategy A) and, similar to the case of the multi-conditional development set, we found that the results are quite stable with respect to the actual choice of p and N. For the $p = 0.7 - 0.9, N = 7 - 9$ range, all scores fell between 25.3% and 26.2%, which correspond to relative error rate reductions of 22–25%.

Next, we performed a more detailed analysis to see whether the improvement is different for the four subsets of the test data. Table 3 shows the relative improvement of the error rate for the subsets A, B, C and D. According to the

Table 3. The word error rate and its relative improvement for the four test subsets, for strategies A and B.

Data set	Baseline	Data augmentation			
		Strategy A	*Improvement*	Strategy B	*improvement*
Set A	3.6%	3.5%	*2.6%*	3.9%	*−6.2%*
Set B	23.5%	16.1%	*31.2%*	16.5%	*29.5%*
Set C	35.7%	30.9%	*13.5%*	30.1%	*15.7%*
Set D	48.6%	37.9%	*22.1%*	38.5%	*20.9%*
Overall	33.7%	25.6%	*24.0%*	26.0%	*22.9%*

Table 4. Word error rates got using the baseline system, with data augmentation and with input dropout.

Method	WER
Baseline (no augmentation)	33.7%
Data augmentation (Strategy A, $p = 0.7, N = 9$)	25.6%
Input dropout ($p = 0.1$)	31.4%
Input dropout ($p = 0.2$)	31.4%

results, strategy B outperformed strategy A for set C, where the recordings differ only in the channel characteristics, but no additive noise is involved. Strategy A proved better in all other cases, including the clean test set A. While the reason for the different behavior would require a deeper analysis, these results suggest that the proper combination of the two methods may result in a further gain.

As the basic idea of our augmentation technique came from the input dropout method, we performed an experiment to compare the two approaches. In the original dropout paper Hinton et al. applied input dropout with a dropout probability of $p = 0.2$, and reported a moderate gain in the phone error rate on TIMIT [15]. However, later authors found it to be ineffective [9,26]. Here, we evaluated it with $p = 0.1$ and 0.2, and the results are shown in Table 4. Clearly, while there is a significant reduction of 6.8% in the error rate, it is much smaller than the 22–25% got with our method. These results highlight the advantage of choosing the discarded pixels via a heuristic, rather than in a totally random fashion.

Lastly, in Table 5 we compare our scores with some recent results given in the literature. As can be seen, our method is competitive with most of the recently published solutions. Although there are better results now (e.g. [6]), these studies applied a more refined input representation, and so a totally fair comparison cannot be made.

Table 5. Comparison of our best result with some recent results given in the literature for Aurora-4, using the clean training set.

Method	WER
CNN with FBANK features [18]	28.9%
DNN with exemplar-based enhancement [4]	26.8%
GMM with auditory spectral enhancement [2]	25.5%
CNN with PNS features and Gabor filter kernels [6]	22.9%
CNN with data augmentation (this paper)	25.6%

5 Conclusions

We presented a data augmentation algorithm that is based on the concept of input dropout. However, instead of dropping spectral components randomly, we

proposed two perceptually inspired strategies to select the least noise-robust parts of the spectrogram, and perturb the data by randomly dropping these components. Incorporating this strategy into the training process of our CNN, we got relative WER reductions of 22–25% using the clean training set of Aurora-4. In the future, we plan to refine our method by applying a more sophisticated strategy and auditory model in the selection of the reliable spectral parts.

Acknowledgments. This research was partially supported by the EU-funded Hungarian grant EFOP-3.6.1-16-2016-00008, and by the National Research, Development and Innovation Office of Hungary (FK 124584). László Tóth was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

References

1. Abdel-Hamid, O., Mohamed, A., Jiang, H., Penn, G.: Applying convolutional neural network concepts to hybrid NN-HMM model for speech recognition. In: Proceedings of ICASSP, pp. 4277–4280 (2012)
2. Alam, M.J., Kenny, P., O'Shaughnessy, D.: Robust feature extraction based on an asymmetric level-dependent auditory filterbank and a subband spectrum enhancement technique. Digital Signal Process. **29**, 147–157 (2014)
3. Allen, J.B.: How do humans process and recognize speech? IEEE Trans. Speech Audio Proc. **2**(4), 567–577 (1994)
4. Baby, D., Gemmeke, J.F., Virtanen, T., Van Hamme, H.: Exemplar-based speech enhancement for deep neural network based automatic speech recognition. In: Proceedings of ICASSP, pp. 4485–4489 (2015)
5. Bouthillier, X., Konda, K., Vincent, P., Memisevic, R.: Dropout as data augmentation. ArXiv e-prints (2015)
6. Chang, S.Y., Morgan, N.: Robust CNN-based speech recognition with Gabor filter kernels. In: Proceedings of Interspeech, pp. 905–909 (2014)
7. Chistovich, L., Lublinskaja, V.: The center of gravity effect in vowel spectra and critical distance between the formants. Hear. Res. **1**, 185–195 (1979)
8. Cui, X., Goel, V., Kingsbury, B.: Data augmentation for deep neural network acoustic modeling. In: Proceedings of ICASSP, pp. 5619–5623 (2014)
9. Deng, L., Abdel-Hamid, O., Yu, D.: A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion. In: Proceedings of ICASSP, pp. 6669–6673 (2013)
10. Flores, J., Young, S.: Continuous speech recognition in noise using spectral subtraction and HMM adaptation. In: Proceedings of ICASSP, pp. 409–412 (1994)
11. Ghitza, O.: Auditory nerve representation criteria for speech analysis/synthesis. IEEE Trans. ASSP **35**(6), 736–740 (1987)
12. Graham, B., Reizenstein, J., Robinson, L.: Efficient batchwise dropout training using submatrices. ArXiv e-prints, February 2015
13. Hartmann, W., Ng, T., Hsiao, R., Tsakalidis, S., Schwartz, R.M.: Two-stage data augmentation for low-resourced speech recognition. In: Proceedings of Interspeech, pp. 2378–2382 (2016)
14. Hillenbrand, J.M., Houde, R.A., Gayvert, R.T.: Speech perception based on spectral peaks versus spectral shape. J. Acoust. Soc. Am. **119**(6), 4041–4054 (2006)

15. Hinton, G., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Improving neural networks by preventing co-adaptation of feature detectors. CoRR abs/1207.0580 (2012)
16. Hirsch, H.G., Pearce, D.: The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In: ASR2000-Automatic Speech Recognition: Challenges for the New Millenium ISCA Tutorial and Research Workshop (ITRW) (2000)
17. Hsiao, R., Ma, J., Hartmann, W., Karafiat, M., Grezl, F., Burget, L., Szoke, I., Cernocky, J., Watanabe, S., Chen, Z., Mallidi, S., Hermansky, H., Tsakalidis, S., Schwartz, R.: Robust speech recognition in unknown reverberant and noisy conditions. In: Proceedings of ASRU, pp. 533–538. IEEE, December 2015
18. Huang, J.T., Li, J., Gong, Y.: An analysis of convolutional neural networks for speech recognition. In: Proceedings of ICASSP, pp. 4989–4993 (2015)
19. Ikbal, S., Bourlard, H., Magimai-Doss, M.: Peak location estimation for noise robust speech recognition. In: Proceedings of ICASSP, pp. 453–456 (2005)
20. Jaitly, N., Hinton, G.E.: Vocal tract length perturbation (VTLP) improves speech recognition. In: ICML (2013)
21. Kanda, N., Takeda, R., Obuchi, Y.: Elastic spectral distortion for low resource speech recognition with deep neural networks. In: Proceedings of ASRU, pp. 309–314. IEEE (2013)
22. Ko, T., Peddinti, V., Povey, D., Khudanpur, S.: Audio augmentation for speech recognition. In: Proceedings of Interspeech, pp. 3586–3589 (2015)
23. Ko, T., Peddinti, V., Povey, D., Seltzer, M.L., Khudanpur, S.: A study on data augmentation of reverberant speech for robust speech recognition. In: Proceedings of ICASSP (2017)
24. Kovács, G., Tóth, L.: Joint optimization of spectro-temporal features and deep neural nets for robust automatic speech recognition. Acta Cybernetica **22**(1), 117–134 (2015)
25. Lockwood, P., Boudy, J., Blanchet, M.: Non-linear spectral subtraction (NSS) and hidden Markov models for robust speech recognition in car noise environments. In: Proceedings of ICASSP (1992)
26. Miao, Y., Metze, F.: Improving low-resource CD-DNN-HMM using dropout and multilingual DNN training. In: Proceedings of Interspeech, pp. 2237–2241 (2013)
27. Moore, B.C.J.: An Introduction to the Psychology of Hearing. Academic Press, London (1997)
28. Peddinti, V., Chen, G., Manohar, V., Ko, T., Povey, D., Khudanpur, S.: JHU aspire system: robust LVCSR with tdnns, ivector adaptation and RNN-LMS. In: Proceedings of ASRU, pp. 539–546 (2015)
29. Ragni, A., Knill, K.M., Rath, S.P., Gales, M.J.F.: Data augmentation for low resource languages. In: Proceedings of Interspeech, pp. 810–814. ISCA (2014)
30. Sainath, T.N., Mohamed, A., Kingsbury, B., Ramabhadran, B.: Deep convolutional neural networks for LVCSR. In: Proceedings of ICASSP, pp. 8614–8618 (2013)
31. Schroeder, M., Atal, B.S., Hall, J.L.: Optimizing digital speech coders by exploiting masking properties of the human ear. JASA **66**(6), 1647–1652 (1979)
32. Tóth, L.: Phone recognition with hierarchical convolutional deep maxout networks. EURASIP J. Audio Speech Music Process. **25** (2015). https://doi.org/10.1186/s13636-015-0068-3
33. Wan, W., Au, O., Keung, C., Yim, C.: A novel approach of low bit-rate speech coding based on sinusoidal representation and auditory model. In: Proceedings of Eurospeech, pp. 1555–1558 (1999)