Noise and Speech Estimation as Auxiliary Tasks for Robust Speech Recognition

Gueorgui Pironkov[1](✉), Stéphane Dupont[1], Sean U.N. Wood[2], and Thierry Dutoit[1]

[1] Circuit Theory and Signal Processing Lab, University of Mons,
Boulevard Dolez 31, 7000 Mons, Belgium
{gueorgui.pironkov,stephane.dupont,thierry.dutoit}@umons.ac.be
[2] NECOTIS, Department of Electrical and Computer Engineering,
University of Sherbrooke, 2500 Boulevard de l'Université,
QC, Sherbrooke J1K 2R1, Canada
sean.wood@usherbrooke.ca

Abstract. Dealing with noise deteriorating the speech is still a major problem for automatic speech recognition. An interesting approach to tackle this problem consists of using multi-task learning. In this case, an efficient auxiliary task is clean-speech generation. This auxiliary task is trained in addition to the main speech recognition task and its goal is to help improve the results of the main task. In this paper, we investigate this idea further by generating features extracted directly from the audio file containing only the noise, instead of the clean-speech. After demonstrating that an improvement can be obtained through this multi-task learning auxiliary task, we also show that using both noise and clean-speech estimation auxiliary tasks leads to a 4% relative word error rate improvement in comparison to the classic single-task learning on the CHiME4 dataset.

Keywords: Speech recognition · Multi-task learning · Robust ASR · Noise estimation · CHiME4

1 Introduction

In recent years, Deep Neural Networks (DNN) have proven their efficiency in solving a wide variety of classification and regression tasks [14]. In particular, DNNs have been used as acoustic models for Automatic Speech Recognition (ASR), significantly outperforming the previous state-of-the-art methods based on Gaussian Mixture Models (GMM) [9]. Improvements brought by neural networks have progressively reduced the Word Error Rate (WER) to a level where some studies argue that ASR can now achieve near human-level performance [31]. Despite these recent improvements, dealing with noisy and reverberant conditions is still a major challenge for ASR [29]. Several techniques have been developed to address this problem, including feature enhancement for example, where features are *cleaned* at the front-end of the ASR system.

N. Camelin et al. (Eds.): SLSP 2017, LNAI 10583, pp. 181–192, 2017.
DOI: 10.1007/978-3-319-68456-7_15

In this work, we use Multi-Task Learning (MTL) to improve ASR performance in the noisy and reverberant acoustic context. MTL consists of training a single system, specifically a DNN, to solve multiple tasks that are different but related, as opposed to the traditional Single-Task Learning (STL) architecture where the system is trained on only one task [2]. MTL has previously been applied in a variety of situations where ASR is the main task and different auxiliary tasks are added. In most cases, however, few MTL auxiliary tasks have been found to be helpful for the main ASR task when speech is corrupted by noise and reverberation. Generating the clean-speech feature as an auxiliary task is one of the most efficient such approaches [5,15,17,23]. We explore this idea further here by generating the noise features alone as an auxiliary task, as well as generating the noise and clean-speech features separately as two additional auxiliary tasks. The core idea is to increase the acoustic model's awareness of the noisy environment, and how it corrupts speech. To evaluate these auxiliary tasks, we use the simulated part of the CHiME4 dataset [29]. While the CHiME4 dataset contains both real and simulated data, only the simulated part may be used here since we need to extract clean-speech and noise features to train the MTL system.

This paper is organized as follows. First, we present the state-of-the-art in MTL for ASR in Sect. 2. We then describe the MTL mechanism in depth in Sect. 3. Details of the experimental setup used to evaluate the noise estimation auxiliary task are presented in Sect. 4, with the results and analysis presented in Sect. 5. Finally, the conclusion and ideas for future work are discussed in Sect. 6.

2 Related Work

Many speech and language processing problems including speech synthesis [10,30], speaker verification [4], and spoken language understanding [16] have benefited form MTL training. In the case of ASR, whether applying an STL or MTL architecture, the main task consists of training the acoustic model to estimate the phone-state posterior probabilities. These probabilities are then fed as input to a Hidden-Markov Model (HMM) that deals with the temporality of speech. The use of MTL for ASR has already been tested with a variety of auxiliary tasks. Early studies used MTL with gender classification as an auxiliary task [17,26], the goal being to increase the acoustic model's awareness of the impact of the speaker gender on the speech. As explained previously, the goal of the main task is to predict phone-state probabilities; some studies investigate a broader level of classes as the auxiliary task, as they try to directly predict the phone probability instead of the probability of the HMM state [1,25]. A related auxiliary task consists of classifying even broader phonetic classes (e.g. fricative, plosive, nasal,. . .) but has shown poor performance [26]. Another approach consists of classifying graphemes as auxiliary task, where graphemes are the symbolic representation of speech (e.g. any alphabet), as opposed to the phonemes that directly describe the sound [3,26]. In order to increase the generalization ability of the network, recent studies have also focused on increasing its speaker-awareness. This is done by recognizing the speaker or by estimating the associated i-vector [6] of each speaker as auxiliary task [19,20,27,28], instead of concatenating the i-vector to the input features. Adapting the acoustic model to a particular speaker can also benefit from MTL [11]. Additional information about these methods can be found in [18].

Most of the previously cited methods do not particularly focus on ASR in noisy and reverberant conditions, nonetheless robust ASR is a field of interest as well. Some studies have focused solely on improving ASR in reverberant acoustic environment by generating de-reverberated speech as auxiliary task, using reverberated speech as input during training [8,22]. Another approach that tackles the noise problem in ASR with MTL consists of recognizing the type of noise corrupting the speech, where a single noise type among several possible types is added for each sentence of the clean speech [12,24]. This approach does not seem to have a real positive impact on the main ASR task, however. The MTL task that shows the highest improvement consists of generating the clean-speech features as auxiliary task [15,17,23]. Of course, in order to generate the targets needed to train this auxiliary task, access to the clean speech is required to extract the features, and this can only be done with simulated noisy and reverberant data. It is also possible to use an MTL system as a feature extractor for robust ASR, where a bottle-neck layer is used, the goal being to use the activations of the bottle-neck layer as input of a traditional STL/ASR system [13].

Though previous studies have proposed recognizing the type of noise, or generating the clean-speech features, to the best of our knowledge, there have been no attempts to estimate the noise features alone as an auxiliary task, or to estimate both the noise and speech features separately in an MTL setup.

3 Multi-Task Learning

Initially introduced in 1997, the core idea of multi-task learning consists of training a single system (a neural network here) to solve multiple tasks that are different but still related [2]. In the MTL nomenclature, the *main task* is the principal task, i.e. the task that would be initially used for a STL architecture, whereas at least one *auxiliary task* is added to help improve the network's convergence to the benefit of the main task. An MTL architecture with one main task and N auxiliary tasks is shown in Fig. 1 as an example.

All MTL systems share two essential characteristics: (a) The same input features are used for training both the main and the auxiliary tasks. (b) The parameters (weights and biases) of all neurons, and more generally the internal structure of the network, are shared among the main and auxiliary tasks, with the exception of the output layer. Furthermore, these parameters are updated by backpropagating a mixture of the error associated with each task, with a term:

$$\epsilon_{MTL} = \epsilon_{Main} + \sum_{n=1}^{N} \lambda_n * \epsilon_{Auxiliary_n}, \tag{1}$$

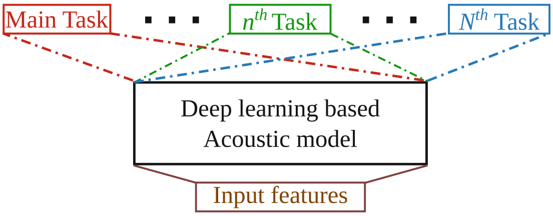

Fig. 1. A Multi-Task Learning system with one main task and N auxiliary tasks.

where ϵ_{MTL} is the sum of all the task errors to be minimized, ϵ_{Main} and $\epsilon_{Auxiliary_n}$ are the errors obtained from the *main* and *auxiliary* tasks respectively, λ_n is a nonnegative weight associated with each of the auxiliary tasks, and N is the total number of auxiliary tasks added to the main task. The value λ_n controls the influence of the auxiliary task with respect to the main task. If the n^{th} auxiliary task has a λ_n close to 1, the main task and the auxiliary task will contribute equally to the error estimation. On the other hand, if λ_n is close to 0, a single-task learning system could be obtained due to the very small (or nonexistent) influence of the auxiliary task. The auxiliary task is frequently removed during testing, keeping only the main task. Selecting a relevant auxiliary task with respect to the main task is the crucial point leading to convergence of the main task. Instead of computing and training each task independently, sharing the parameters of the system among multiple tasks may lead to better results than an independent processing of each task [2].

4 Experimental Setup

In this section, we will present the tools and methods used to evaluate the new auxiliary task that we propose for robust ASR.

4.1 Database

In order to evaluate noise estimation as an auxiliary task for robust ASR, we use the CHiME4 database [29]. This database was released in 2016 for a speech recognition and separation challenge in reverberant and noisy environments. This database is composed of 1-channel, 2-channel, and 6-channel microphone array recordings. Four different noisy environments (café, street junction, public transport, and pedestrian area) were used to record real acoustic mixtures through a tablet device with 6-channel microphones. The WSJ0 database [7] is used to create simulated data. WSJ0 contains clean-speech recordings to which noise is added. The noise is recorded from the four noisy environments described above. For the noise estimation auxiliary task, we use features extracted from these recordings containing only noise as targets for training. As we cannot obtain these targets for real data, we only use the simulated data in this study.

All datasets (training, development, and test sets) consist of 16 bit wav files sampled at 16 kHz. The training set consists of 83 speakers uttering 7138 simulated sentences, which is the equivalent of ∼15 h of training data. The development set consists of 1640 utterances (∼2.8 h) uttered by 4 speakers. Finally, 4 additional speakers compose the test set with 1320 utterances corresponding to approximately 4.5 h of recordings.

In this work, we investigate noise and clean-speech estimation as auxiliary tasks, therefore we use only the noise recorded from a single channel during training (channel no 5). The test and development set noises are randomly selected from all channels, making the task harder but also challenging the generalization ability of the setup.

4.2 Features

The features used as input for training the MTL system as well as targets for the noise and/or clean-speech estimation tasks are obtained through the following traditional ASR pipeline:

1. Using the raw audio wav files, 13-dimensional Mel-Frequency Cepstral Coefficients (MFCC) features are extracted and normalized through Cepstral Mean-Variance Normalization (CMVN).
2. For each frame, the adjacent ±3 frames are spliced.
3. These 91-dimensional feature vectors are reduced through a Linear Discriminative Analysis (LDA) transformation to a 40-dimensional feature space.
4. The final step consists of projecting the features through a feature-space speaker adaptation transformation known as feature-space Maximum Likelihood Linear Regression (fMLLR).

Finally, the 40-dimensional features that are computed through this pipeline are spliced one more time with the surrounding ±5 frames for the input features fed to the acoustic model, thus giving additional temporal context to the network during training. For the auxiliary tasks' targets, the same pipeline is followed to generate the clean-speech and noise features but there is no ±5 splicing at the final stage. Alignments from the clean-speech are reused for the transformations applied on noisy features.

4.3 Training the Acoustic Model

Training and testing this MTL auxiliary tasks was done using the *nnet3* version of the Kaldi toolbox [21].

We use a classic feed-forward deep neural network acoustic model to evaluate the performance of this new auxiliary task. The DNN is composed of 4 hidden layers, each of them consisting of 1024 neurons activated through Rectified Linear Units (ReLU). The main task used for STL and MTL computes 1972 phone-state posterior probabilities after a softmax output layer. The training of the DNN is done through 14 epochs using the cross-entropy loss function for the main task, and quadratic loss function for the auxiliary tasks (as they are regression issues), with an initial learning rate starting at 0.0015 that is progressively reduced to 0.00015. Stochastic gradient descent (SDG) is used to update the parameters of the network through the backpropagation of the error derivatives. The size of the mini-batch used to process the input features is set 512. These parameters were selected through empirical observations.

The same experiments were also conducted using other deep learning algorithms including Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) cells and Time-Delay Neural Networks (TDNN). However, the feed-forward DNN showed similar or better results than these more complex architectures on the simulated data of CHiME4. Also, the computational time for the RNN-LSTM network was much higher than for the feed-forward DNN. While the complexity and temporarily of the main and auxiliary tasks did not require a more complex acoustic model here, we note that for some auxiliary tasks, having a more complex network can be crucial for the convergence of the auxiliary task, as is the case for speaker classification for instance [19].

During decoding, the most likely transcriptions are obtained through the phone-state probabilities estimated by the feed-forward network, and used by the HMM system and associated with a language model. The language model is the 3-gram KN language model trained on the WSJ 5K standard corpus.

4.4 Baseline

The baseline of our system is obtained by training the setup presented in the previous section in single-task learning manner. We compute the word error rate for both the development and test sets over all four noisy environments for the simulated data of CHiME4. The results are shown in Table 1. A very significant mismatch coming from the recording environments between the development and test set can be noticed, explaining the higher WER for the test set. For the rest of this paper we display only the *Average* results as the trends and evolutions of the WER are similar over all four noisy environments.

Table 1. Word error rate in % on the development and test sets of CHiME4 dataset used as baseline. *Average* is the mean WER of all 4 environmental noises and *Overall* is the mean WER over the development and test sets.

	Average	Bus	Café	Pedestrian	Street
Dev set	18.54	16.55	22.05	15.03	20.52
Test set	26.82	21.44	30.99	26.90	27.96
Overall	22.68	19.00	26.52	20.97	24.24

5 Results

In this section, we investigate the improvement brought by the new MTL auxiliary task, namely regenerating the noise contained in the corrupted sentence, in comparison to STL. We also combine this auxiliary task with the more traditional clean-speech generation auxiliary task.

5.1 Noise Features Estimation

In order to evaluate the impact of estimating the noise features as an auxiliary task in our MTL setup, we vary the value of λ_{noise}, thus varying the influence of this auxiliary task with respect to main ASR task. The obtained results for values of λ_{noise} varying between 0 (STL) and 0.5 are presented in Table 2. There is a small but persistent improvement of the WER for $\lambda_{noise} = 0.05$, over both the development and test sets. For smaller values ($\lambda_{noise} = 0.01$), the improvement is nearly insignificant as the value of λ_{noise} brings the training too close to STL ($\lambda_{noise} = 0$), while for values of λ_{noise} too high ($\lambda_{noise} \geq 0.15$), the WER is worse than for STL as the influence of the auxiliary task overshadows the main ASR task.

Table 2. Average word error rate (in %) of the Multi-Task Learning architecture when the auxiliary task is noise feature estimation, where λ_{noise} is the weight attributed to the noise estimation auxiliary task during training. The baseline, which is the Single-Task Learning architecture, is obtained for $\lambda_{noise} = 0$. The *Overall* values are computed over both datasets.

λ_{noise}	*0 (STL)*	0.01	0.05	0.1	0.15	0.2	0.3	0.5
Dev set	18.54	18.43	**18.19**	18.31	18.65	18.82	19.59	20.83
Test set	26.82	26.63	**26.50**	26.55	26.85	27.08	28.01	29.89
Overall	22.68	22.53	**22.35**	22.43	22.75	22.95	23.80	25.36

In order to further highlight these observations, we present the relative WER improvement brought by MTL in comparison to STL in Fig. 2. An improvement is obtained for values of λ_{noise} between 0.01 and 0.1. The highest improvement is obtained for $\lambda_{noise} = 0.05$, with a relative improvement in comparison to STL going up to 1.9% on the development set for instance. Larger values of λ_{noise} degrade performance on the main speech recognition task.

As discussed in Sect. 4.3, training is done over 14 epochs. In order to prove the ASR improvement is not only the result of the introduction of a small noise into the system, but rather that both tasks are converging, we present the error over these 14 epochs in Fig. 3, highlighting in this way the error reduction obtained on both tasks loss functions over time.

Despite the persistence of the relative improvement for small values of λ_{noise}, it can be noted that this improvement is quite small. This can be explained by

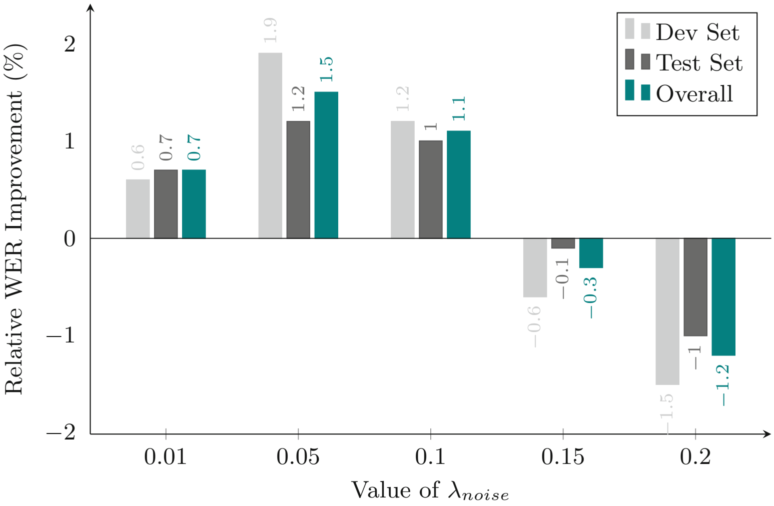

Fig. 2. Evaluation of the relative improvement of the word error rate brought by multi-task learning in comparison to single-task learning, with λ_{noise} the weight attributed to the noise estimation auxiliary task. The *Overall* values are computed over both the development and test datasets.

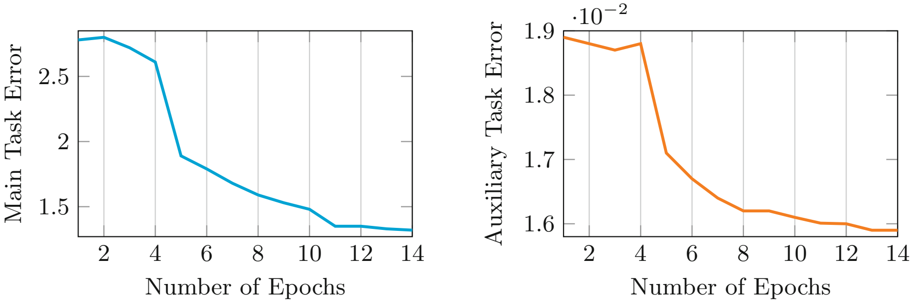

Fig. 3. Evolution of the tasks errors over training epochs. The *Main Task* is the speech recognition error computed through the cross-entropy loss function, whereas the *Auxiliary Task* corresponds to the noise estimation error obtained through the quadratic loss function.

several considerations. First, this auxiliary task is less directly related to the main task than for instance clean speech generation, meaning that the convergence of the auxiliary task may not significantly help the main task. Another consideration is that the auxiliary task is in fact quite a hard task here as the Signal-to-Noise Ratio (SNR) is always in favor of the clean-speech and not the noise, making it hard to estimate the noise alone. Finally, the suitability of the features extracted following the pipeline presented in Sect. 4.2, as well as using fMLLR transformation in this context, is most likely not optimal for noise.

Despite these considerations, using noise estimation as auxiliary task seems to be helpful for the main ASR task when λ_{noise} is properly selected. Additionally, using a MTL setup is easy to implement and does not require extensive computational time in comparison to STL (as the same network is trained for both tasks). Finally, the targets for this particular auxiliary task, noise estimation, are easy to get as we have access to the noise when generating the simulated data.

5.2 Combining Noise and Clean-Speech Features Estimation

Instead of separately generating clean-speech or noise as auxiliary tasks, we investigate here the combination of both tasks in the MTL framework. In order to do that, we first repeat the same experiment as in Sect. 5.1 but where we generate only the clean-speech features as the auxiliary task. After varying the

Table 3. Average word error rate in % on the development and test sets of CHiME4 dataset, when different auxiliary tasks are applied. *Overall* is the mean WER over the development and test sets data.

Auxiliary task(s)	Dev set	Test set	Overall
None (STL)	18.54	26.82	22.68
Noise estimation ($\lambda_{noise} = 0.05$)	18.19	26.50	22.35
Clean-speech estimation ($\lambda_{speech} = 0.15$)	17.99	26.06	22.03
Noise + clean-speech estimation	**17.79**	**25.78**	**21.79**

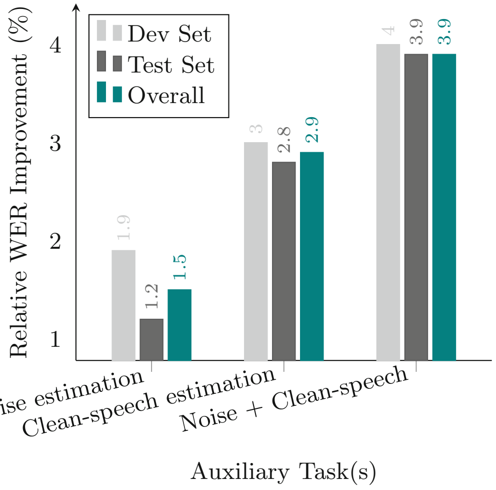

Fig. 4. Evaluation of the relative improvement of the word error rate brought by multi-task learning in comparison to single-task learning, with different auxiliary tasks. The *Overall* values are computed over both the development and test datasets.

value of λ_{speech} we found the best WER is obtained for $\lambda_{speech} = 0.15$. The obtained results are depicted in Table 3 and, as in the previous section, we compute the relative improvement brought by the different auxiliary tasks (plus their combination) in comparison to STL in Fig. 4.

The results show that, as expected, a better WER is obtained when using clean-speech estimation as auxiliary task in comparison to noise estimation, with an overall relative improvement of 2.9% (while it was 1.5% in the previous experiment). Interestingly however, using both the clean-speech and noise estimation auxiliary tasks lead to even better performance, with 3.9% overall relative improvement and more than 1% absolute improvement on the test set. This result highlights the fact that the network is learning different and valuable information from both auxiliary tasks in order to improve the main task. Once again, implementing these auxiliary tasks is simple and does not require significant additional computational time in comparison to classic single-task learning architectures.

6 Conclusion

In this paper, we have studied multi-task learning acoustic modeling for robust speech recognition. While most previous studies focus on clean-speech generation as auxiliary task, we propose and investigate here another different but related auxiliary task: noise estimation. This auxiliary task consists of generating the features extracted from the audio file containing only the noise that is later added to the clean-speech to create the simulated noisy data. After showing that an improvement can be obtained with this auxiliary task, we combined it with the clean-speech estimation auxiliary task, resulting in one main task and two auxiliary tasks. A relative WER improvement of 4% can be obtained thanks to the association of these two auxiliary tasks in comparison to the classic single-task learning architecture. Training and testing here was done only on the simulated data taken from the CHiME4 dataset, as the clean-speech and noise audio are required separately for the auxiliary tasks training, thus making it impossible to train with real data. In future work, we would like to find a way to integrate real data to the training, and re-evaluate the impact of these two auxiliary tasks. We would also like to use other types of features which may be more suitable to capture the noise variations, as the features we are currently using are designed to best capture the diversity of speech.

Acknowledgments. This work has been partly funded by the Walloon Region of Belgium through the SPW-DGO6 Wallinov Program no 1610152.

References

1. Bell, P., Renals, S.: Regularization of context-dependent deep neural networks with context-independent multi-task training. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4290–4294. IEEE (2015)

2. Caruana, R.: Multitask learning. Mach. learn. **28**(1), 41–75 (1997)
3. Chen, D., Mak, B., Leung, C.C., Sivadas, S.: Joint acoustic modeling of triphones and trigraphemes by multi-task learning deep neural networks for low-resource speech recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5592–5596. IEEE (2014)
4. Chen, N., Qian, Y., Yu, K.: Multi-task learning for text-dependent speaker verification. In: Sixteenth Annual Conference of the International Speech Communication Association (2015)
5. Chen, Z., Watanabe, S., Erdogan, H., Hershey, J.R.: Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks. In: INTERSPEECH, pp. 3274–3278. ISCA (2015)
6. Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., Ouellet, P.: Front-end factor analysis for speaker verification. IEEE Trans. Audio Speech Lang. Process. **19**(4), 788–798 (2011)
7. Garofolo, J., Graff, D., Paul, D., Pallett, D.: CSR-I (WSJ0) Complete LDC93S6A. Web Download. Linguistic Data Consortium, Philadelphia (1993)
8. Giri, R., Seltzer, M.L., Droppo, J., Yu, D.: Improving speech recognition in reverberation using a room-aware deep neural network and multi-task learning. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5014–5018. IEEE (2015)
9. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al.: Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. Sig. Process. Mag. **29**(6), 82–97 (2012)
10. Hu, Q., Wu, Z., Richmond, K., Yamagishi, J., Stylianou, Y., Maia, R.: Fusion of multiple parameterisations for DNN-based sinusoidal speech synthesis with multi-task learning. In: Proceedings of Interspeech (2015)
11. Huang, Z., Li, J., Siniscalchi, S.M., Chen, I.F., Wu, J., Lee, C.H.: Rapid adaptation for deep neural networks through multi-task learning. In: Sixteenth Annual Conference of the International Speech Communication Association (2015)
12. Kim, S., Raj, B., Lane, I.: Environmental noise embeddings for robust speech recognition (2016). arxiv preprint arXiv:1601.02553
13. Kundu, S., Mantena, G., Qian, Y., Tan, T., Delcroix, M., Sim, K.C.: Joint acoustic factor learning for robust deep neural network based automatic speech recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5025–5029. IEEE (2016)
14. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
15. Li, B., Sainath, T.N., Weiss, R.J., Wilson, K.W., Bacchiani, M.: Neural network adaptive beamforming for robust multichannel speech recognition. In: Proceedings of Interspeech (2016)
16. Li, X., Wang, Y.Y., Tur, G.: Multi-task learning for spoken language understanding with shared slots. In: Twelfth Annual Conference of the International Speech Communication Association (2011)
17. Lu, Y., Lu, F., Sehgal, S., Gupta, S., Du, J., Tham, C.H., Green, P., Wan, V.: Multitask learning in connectionist speech recognition. In: Proceedings of the Australian International Conference on Speech Science and Technology (2004)
18. Pironkov, G., Dupont, S., Dutoit, T.: Multi-task learning for speech recognition: an overview. In: Proceedings of the 24th European Symposium on Artificial Neural Networks (ESANN) (2016)

19. Pironkov, G., Dupont, S., Dutoit, T.: Speaker-aware long short-term memory multi-task learning for speech recognition. In: 24th European Signal Processing Conference (EUSIPCO), pp. 1911–1915. IEEE (2016)
20. Pironkov, G., Dupont, S., Dutoit, T.: Speaker-aware multi-task learning for automatic speech recognition. In: 23rd International Conference on Pattern Recognition (ICPR) (2016)
21. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al.: The kaldi speech recognition toolkit. In: IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society (2011)
22. Qian, Y., Tan, T., Yu, D.: An investigation into using parallel data for far-field speech recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5725–5729. IEEE (2016)
23. Qian, Y., Yin, M., You, Y., Yu, K.: Multi-task joint-learning of deep neural networks for robust speech recognition. In: IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), pp. 310–316. IEEE (2015)
24. Sakti, S., Kawanishi, S., Neubig, G., Yoshino, K., Nakamura, S.: Deep bottleneck features and sound-dependent i-vectors for simultaneous recognition of speech and environmental sounds. In: Spoken Language Technology Workshop (SLT), pp. 35–42. IEEE (2016)
25. Seltzer, M.L., Droppo, J.: Multi-task learning in deep neural networks for improved phoneme recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6965–6969. IEEE (2013)
26. Stadermann, J., Koska, W., Rigoll, G.: Multi-task learning strategies for a recurrent neural net in a hybrid tied-posteriors acoustic model. In: INTERSPEECH, pp. 2993–2996 (2005)
27. Tan, T., Qian, Y., Yu, D., Kundu, S., Lu, L., Sim, K.C., Xiao, X., Zhang, Y.: Speaker-aware training of LSTM-RNNS for acoustic modelling. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5280–5284. IEEE (2016)
28. Tang, Z., Li, L., Wang, D.: Multi-task recurrent model for speech and speaker recognition (2016). arxiv preprint arXiv:1603.09643
29. Vincent, E., Watanabe, S., Nugraha, A.A., Barker, J., Marxer, R.: An analysis of environment, microphone and data simulation mismatches in robust speech recognition. Computer Speech & Language (2016)
30. Wu, Z., Valentini-Botinhao, C., Watts, O., King, S.: Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4460–4464. IEEE (2015)
31. Xiong, W., Droppo, J., Huang, X., Seide, F., Seltzer, M., Stolcke, A., Yu, D., Zweig, G.: Achieving human parity in conversational speech recognition (2016). arxiv preprint arXiv:1610.05256