MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks

Chen, Z.; Watanabe, S.; Erdogan, H.; Hershey, J.R.

TR2015-100 September 2015

Abstract

Long Short-Term Memory (LSTM) recurrent neural network has proven effective in modeling speech and has achieved outstanding performance in both speech enhancement (SE) and automatic speech recognition (ASR). To further improve the performance of noise-robust speech recognition, a combination of speech enhancement and recognition was shown to be promising in earlier work. This paper aims to explore options for consistent integration of SE and ASR using LSTM networks. Since SE and ASR have different objective criteria, it is not clear what kind of integration would finally lead to the best word error rate for noise-robust ASR tasks. In this work, several integration architectures are proposed and tested, including: (1) a pipeline architecture of LSTM-based SE and ASR with sequence training, (2) an alternating estimation architecture, and (3) a multi-task hybrid LSTM network architecture. The proposed models were evaluated on the 2nd CHiME speech separation and recognition challenge task, and show significant improvements relative to prior results.

Interspeech 2015

Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks

Zhuo Chen[1,2]*, Shinji Watanabe*[1]*, Hakan Erdogan*[1,3]*, John R. Hershey*[1]

[1]Mitsubishi Electric Research Laboratories (MERL), 201 Broadway, Cambridge MA, 02139, USA
[2]Columbia University, New York, NY, USA
[3]Sabanci University, Orhanli Tuzla, 34956, Istanbul, Turkey

zchen@merl.com, watanabe@merl.com, haerdogan@sabanciuniv.edu, hershey@merl.com

Abstract

Long Short-Term Memory (LSTM) recurrent neural network has proven effective in modeling speech and has achieved outstanding performance in both speech enhancement (SE) and automatic speech recognition (ASR). To further improve the performance of noise-robust speech recognition, a combination of speech enhancement and recognition was shown to be promising in earlier work. This paper aims to explore options for consistent integration of SE and ASR using LSTM networks. Since SE and ASR have different objective criteria, it is not clear what kind of integration would finally lead to the best word error rate for noise-robust ASR tasks. In this work, several integration architectures are proposed and tested, including: (1) a pipeline architecture of LSTM-based SE and ASR with sequence training, (2) an alternating estimation architecture, and (3) a multi-task hybrid LSTM network architecture. The proposed models were evaluated on the 2nd CHiME speech separation and recognition challenge task, and show significant improvements relative to prior results.

Index Terms: noisy speech recognition, speech enhancement, LSTM, integration, sequence training

1. Introduction

Unlike a feed forward deep neural network(DNN), a recurrent neural network(RNN) has a feedback structure that enables the RNN to consider previous activations when estimating the current ones. Such property makes it especially appropriate in modeling audio signals, which usually have strong dependencies across time. Long short-term memory(LSTM) [1] is one specially designed architecture for RNN. In an LSTM neural network, the neuron in each hidden layer is replaced with a special unit called memory block. The memory blocks usually contain a self-connected memory cell to remember the temporal state and several gates to control the information and gradient flow. By opening and closing the gates, an LSTM network can control the entrance of previous information into its memory cells as well as the emittance of information out of the cells, and thus enables itself to learn dependencies across long time contexts, alleviating the well-known "gradient vanishing" problem of RNN.

LSTM has been successfully applied in the problem of speech enhancement(SE). In [2], the LSTM was trained to remove additive noise, using noisy mel-filterbank features as input and targeting a soft-mask that is used to multiply noisy spectra, and reported results were much better than non-negative matrix factorization based models. In [3], the authors further improved the model by incorporating speech recognition information and using a phase-sensitive objective, and achieved state of the art performance in 2nd Chime Challenge separation task. In [4], the LSTM was applied to enhance the reverberated speech.

The LSTM architecture has also been explored in the problem of automatic speech recognition (ASR). The LSTM is usually used as the acoustic model, where the training target is prediction of the HMM state [5]. In [6], the authors applied an LSTM acoustic model for a large-vocabulary continuous speech recognition (LVCSR) task and showed that the LSTM can achieve better performance than a DNN, using fewer parameters. In [7], a sequence-discriminative training objective was used in addition to the cross-entropy one, achieving state-of-art performance for LVCSR problems. In [8], the LSTM was applied in a medium-vocabulary scenario, where the authors directly used the phoneme label as the prediction target and form the HMM posterior with the recognized phoneme confusion matrix, and the performance reported was also better than the DNN baseline.

The successes of the application in SE and ASR naturally makes the LSTM a good candidate for the problem of noisy speech recognition, where the noisy speech is usually first enhanced with the speech enhancement system, then the speech recognizer is trained with the enhanced speech. However, one well-known issue for noisy speech recognition is that the criteria of speech enhancement and speech recognition are not the same. Therefore, it is usually not clear what kind of combination would help the overall performance the most. For example, in [9], the authors reported, NMF-based speech enhancement would not help a DNN-based speech recognizer.

In this work, to achieve better performance and answer the questions above, we integrate LSTM speech enhancement and LSTM speech recognition systems. We design and evaluate several network architectures for the problem of noisy speech recognition. First, we use a pipe-line structure, where an LSTM SE is followed by an LSTM ASR. To further improve the performance, we also explore discriminative sequence training [10, 11] for the LSTM on the medium-vocabulary scenario. Second, we propose an alternative optimization scheme between SE and ASR. The ASR uses the SE's output to get the enhanced speech. And the SE also uses the ASR information for further enhancement. And finally, we propose a multitask LSTM network architecture, in which a unified objective that considers both the SE quality and ASR accuracy is used. The proposed models were evaluated on the medium-vocabulary track in 2nd CHiME speech separation and recognition challenge[12]. And we show that the proposed models outperform the best reported result and a DNN baseline.

As discussed above, several related models were proposed in previous works. In [7], LSTM was applied for cross-entropy and sequence training, but their target is clean speech and large vocabulary scenario and no speech enhancement was applied. In [8], the authors were dealing with a similar problem. However, their recognition model was phoneme-based without sequence training, which is different from the proposed model. To the best of our knowledge, there is no similar work to the one introduced in this paper.

2. Bi-directional LSTM

In this section, we briefly introduce bi-directional LSTM networks by explaining the forward computations in a layer of an LSTM network followed by some discussion about the final layer processing. We consider an LSTM memory block at nth layer with an input vector $\boldsymbol{h}_t^{n-1}$ and output activation $\boldsymbol{h}_t^n$ at frame t (here, we omit the utterance index). Note that the input vector at the first layer corresponds to the observation vector, i.e., $\boldsymbol{h}_t^0 = \boldsymbol{y}_t$. We first define the concatenated vector of output activation $\boldsymbol{h}_{t-1}^n$ at previous time frame $t-1$ and the $n-1$th layer output activation $\boldsymbol{h}_t^{n-1}$ at current time frame t as $\boldsymbol{m}_t^n \triangleq [(\boldsymbol{h}_t^{n-1})^\top, (\boldsymbol{h}_{t-1}^n)^\top]^\top$. Then, the LSTM memory block has a memory cell (return: $\boldsymbol{c}_t$), which are obtained from the input gate (return: $\boldsymbol{i}_t$) and forget gate (return: $\boldsymbol{f}_t$):

$$\begin{aligned}
\boldsymbol{i}_t^n &= \sigma(\boldsymbol{W}_{im}^n \boldsymbol{m}_t^n + \boldsymbol{W}_{ic}^n \boldsymbol{c}_{t-1}^n + \boldsymbol{b}_i^n), \\
\boldsymbol{f}_t^n &= \sigma(\boldsymbol{W}_{fm}^n \boldsymbol{m}_t^n + \boldsymbol{W}_{fc}^n \boldsymbol{c}_{t-1}^n + \boldsymbol{b}_f^n), \\
\boldsymbol{c}_t^n &= \boldsymbol{f}_t^n \odot \boldsymbol{c}_{t-1}^n + \boldsymbol{i}_t^n \odot \tanh(\boldsymbol{W}_{cm}^n \boldsymbol{m}_t^n + \boldsymbol{b}_c^n),
\end{aligned} \tag{1}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are affine transformation parameters to be estimated at the training step. $\odot$, $\sigma(\cdot)$, and $\tanh(\cdot)$ denote the element-wise product operation, sigmoid function, and hyperbolic tangent function, respectively. The memory cell and input and forget gates are calculated from the concatenated activation vector $\boldsymbol{m}_t^n$ and the cell vector $\boldsymbol{c}_{t-1}^n$ at the previous frame. The relationship between $\boldsymbol{c}_t^n$ and $\boldsymbol{c}_{t-1}^n$ is controlled by the forget gate $\boldsymbol{f}_t^n$ dynamically, which enables to retain the long-range dependency of the cell, unlike the hidden state in standard RNNs.

Once we obtain cell vector $\boldsymbol{c}_t^n$, we can calculate output gate vector $\boldsymbol{o}_t^n$, and finally calculate output activation $\boldsymbol{h}_t^n$ as follows:

$$\begin{aligned}
\boldsymbol{o}_t^n &= \sigma(\boldsymbol{W}_{om}^n \boldsymbol{m}_t^n + \boldsymbol{W}_{oc}^n \boldsymbol{c}_t^n + \boldsymbol{b}_o^n), \\
\boldsymbol{h}_t^n &= \boldsymbol{o}_t^n \odot \tanh(\boldsymbol{c}_t^n).
\end{aligned} \tag{2}$$

A set of these equations is a basic feed-forward operation of the LSTM memory block at nth layer. At the top layer (N), output activation $\boldsymbol{h}_t^N$ is further calculated by the following affine transformation:

$$\hat{\boldsymbol{h}}_t = \boldsymbol{W}^N \boldsymbol{h}_t^N + \boldsymbol{b}^N. \tag{3}$$

This final activation $\hat{\boldsymbol{h}}_t$ would be used for the regression, classification through the softmax operation, or masking function through the sigmoid operation.

Similar to the LSTM, Bi-directional LSTM has the same memory block as the basic unit. Instead of propagating the information in one time direction, in BLSTM layer, there are two separated propagation sequences from the both time directions. Therefore, unlike equation (3), the BLSTM neural network obtains the final activation $\hat{\boldsymbol{h}}_t$ by using both the final activations from the past $\boldsymbol{h}_t^{N\rightarrow}$ and future $\boldsymbol{h}_t^{N\leftarrow}$, as follows:

$$\hat{\boldsymbol{h}}_t = \boldsymbol{W}^{N\rightarrow} \boldsymbol{h}_t^{\rightarrow} + \boldsymbol{W}^{N\leftarrow} \boldsymbol{h}_t^{\leftarrow} + \boldsymbol{b}^N. \tag{4}$$

This property enables the BLSTM network further explore the connection within contexts, and often lead to better performance than LSTM.

3. Pipeline Architecture

3.1. Speech enhancement

Speech enhancement LSTM and BLSTM networks are trained to predict a real valued mask or filter function that multiplies the noisy signal's STFT for each time-frequency bin. This mask function is constrained to be in the range $[0, 1]$ similar to a Wiener filter or ideal ratio filter by using a sigmoid activation at the output of the network. We believe instead of direct prediction of the clean spectrum, mask prediction approach is better since it is easier to predict the mask and it fits with a sigmoid output layer.

We use magnitude spectrum approximation (MSA) or phase-sensitive spectrum approximation (PSA) loss functions which were shown to be superior to the mask approximation (MA) loss function in earlier work [2, 3]. The input to the network is log-Mel-filterbank energies of the noisy signal with 100 bins which gave the best result among alternatives in [2]. The PSA loss function for training the network is given as follows:

$$\mathcal{L}_{\text{se}} = \frac{1}{N} \sum_{u,t,f} |\hat{a}_{u,t,f} y_{u,t,f} - s_{u,t,f}|^2, \tag{5}$$

where the network predicts the real masking function $\hat{a}$ from the final activation $\hat{h}$ in eqn (4), and the sum is over utterances (u) and time-frequency bins (t, f). Here y and s represent complex domain noisy signal and speech signal STFTs respectively. N is the number of all frames in the training data.

Whenever used, ASR information is added to the input by using the one-best state alignment information received from the recognizer. The input feature used is typically an average of the log-Mel-filterbank feature vectors aligning to each state in the training data [3].

3.2. Cross Entropy training

After the speech enhancement step, a BLSTM-based acoustic model is trained with enhanced feature $\hat{\boldsymbol{s}}_{u,t}$. Similar to a DNN-HMM hybrid system[13, 14], we use HMM state $r_{u,t}$ at frame t in utterance u as a training target, which is obtained from the reference with the Viterbi alignment. The cross entropy based cost function is represented as follows:

$$\mathcal{L}_{\text{ce}} = -\frac{1}{N} \sum_{u,t} \log p(r_{u,t} | \hat{\boldsymbol{s}}_{u,t}), \tag{6}$$

The HMM state posterior $p(r_{u,t}|\hat{\boldsymbol{s}}_{u,t})$ is obtained from the final activation $\hat{h}$ in eqn (4) of the BLSTM with the softmax operation, i.e., $p(r_{u,t} = k|\hat{\boldsymbol{s}}_{u,t}) = \exp(\hat{h}_{u,t,k}) / \sum_{k'} \exp(\hat{h}_{u,t,k'})$ for state index k.

3.3. Discriminative sequence training

Different from the cross entropy objective function that measures the error at frame level, the discriminative sequence training treats the whole sequence as target in objective function. With uth enhanced feature sequence $\hat{\boldsymbol{S}}_u = \{\hat{\boldsymbol{s}}_{u,t} | t = 1, \dots\}$ and uth reference word sequence $\mathcal{W}_u$, the objective function based on the state-level minimum Bayes risk crition[15] is rep-

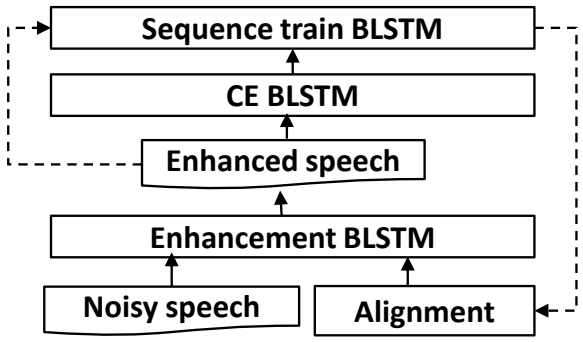

Figure 1: *The Iterative pipeline architecture*

resented as follows:

$$\mathcal{L}_{\text{seq}} = -\frac{1}{N}\sum_{u}\log\frac{\sum_{\mathcal{W}} p_{\text{pseudo}}(\hat{\boldsymbol{S}}_u|\mathcal{R}_{\mathcal{W}_u})p_{\text{lm}}(\mathcal{W})A(\mathcal{W},\mathcal{W}_u)}{\sum_{\mathcal{W}'} p_{\text{pseudo}}(\hat{\boldsymbol{S}}_u|\mathcal{R}_{\mathcal{W}'})p_{\text{lm}}(\mathcal{W}')} \tag{7}$$

where $\mathcal{R}_{\mathcal{W}}$ is a set of HMM state sequence given word sequence $\mathcal{W}$, and $A(\mathcal{W},\mathcal{W}_u)$ proivdes a state-level accuracy. $p_{\text{lm}}(\cdot)$ is a language model, and $p_{\text{pseud}}(\cdot)$ is a pseudo likelihood obtained from the HMM state posterior $p(r_{u,t}|\hat{\boldsymbol{s}}_{u,t})$ used in eqn (6).

3.4. Iterative pipeline architecture

As mentioned above, the speech enhancement in this work requires the alignment information from the data, and more accurate alignment usually leads to a higher quality of speech enhancement. However, the alignment is obtained using an ASR system using HMMs for modeling and the quality of alignments depends on the recognizer. Therefore, SE and ASR in this work depend on each other. To help this "chicken and egg" problem, we design an alternating estimation architecture. In each iteration, we first estimate the speech enhancement given the alignment generated from the previous iteration. Then after the SE step, we train a cross-entropy LSTM using the enhanced speech. Once the CE-LSTM is trained, it would be used as an initialization of the discriminative sequence training LSTM. Finally, the new alignment is generated from the decoder. The full process is shown in Figrue 1.

4. Multi-task architecture

One drawback of the previous architecture is that the SE step and ASR step are independent from each other. As discussed in section 1, the criterion for the speech enhancement and the recognition are different. The optimized the speech enhancement might not lead to the optimum word-error-rate in the recognition step. To further explore this, we designed a multi-task hybrid BLSTM network architecture for the SE step, which consider both the enhancement and the recognition accuracy. The objective function of the proposed network is

$$\mathcal{L}_{multi} = \mathcal{L}_{se} + \alpha\mathcal{L}_{ce}. \tag{8}$$

In (8), α is the weight for recognition. By changing the weight, the objective could focus more on the speech enhancement or recognition. As discussed in previous sections, $\mathcal{L}_{se}$ and $\mathcal{L}_{ce}$ refers to the objective function of the speech enhancement and speech recognition. Different from the traditional neural network, the proposed architecture consists of two objectives, which means the actual network should have two output layers. The network architecture is shown in Figure 2.

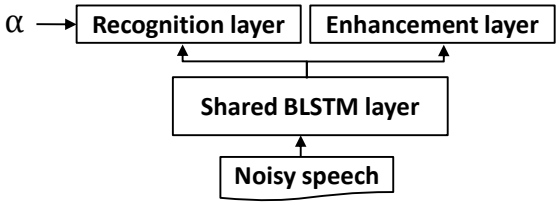

Figure 2: *Multi-task LSTM network*

5. Experiments

5.1. Data

In our experiments, all the systems were evaluated on the medium-vocabulary track in 2nd CHiME speech separation and recognition challenge task. The noisy data in ChiME challenge was constructed from the Wall Street Journal dataset. The clean 16kHz WSJ data was firstly convolved with the room impulse response to model the reverberation. Then the reverberated speech was mixed with the recorded background noise the at 6 different SNRs from $-6dB$ to $9dB$. The training set contains 7138 utterances from 83 speakers, totaling 14.5 hours. The development set contains 4.5 hours data, which consists of 2460 utterances from 10 speakers that are disjoint with the training set. The test set consists of 1980 utterance from 8 speakers, 4 hours in total.

5.2. Pipeline architecture

As discussed in section 3.1, the 100-dimensional noisy logarithmic mel-filter bank was calculated from noisy speech for the speech enhancement. To calculate the spectrogram, the window size of 25 ms and a hop of 10 ms were used. To provide the ASR information to SE, the noisy alignment was generated from the DNN recognizer. The alignment mel-filter bank was then generated from the alignment and the state dictionary, which was the average noisy logarithmic mel-filter bank among the training data for each state. The final input feature for speech enhancement was the concatenation of noisy logarithmic mel-filter and alignment mel-filter bank, totaling 200 dimensions. And the training target for SE step was selected as the phase sensitive spectrum mask, as in [3]

We used the same network topology for the speech enhancement LSTM as in [3], in which two 384 BLSTM layers were firstly applied to the input layer, followed with 2 feedforward layer with activation function of tanh and logistic to get the final output. In the first iteration, the SE network was initialized with random Gaussian noise with mean 0 and standard deviation 0.1. After the first iteration, the trained SE network in the previous iteration was used as the initialization for each new iteration. The learning rate was set to 10.

For the recognition LSTM, the double delta logarithmic mel-filter bank from the enhanced speech was used as the input feature. To generate the feature, we used 26 coefficients for the filter bank, covering 20$\sim$ 8000Hz, with one extra coefficient for the frame energy. Then the double delta with the context window of 5 was applied and resulted in 81 dimension input feature. The recognition target was the clean alignment states, which is generated from the decoding process of GMM likelihood on clean data.

For the cross entropy LSTM, three different networks were evaluated, which were (1) two 300 BLSTM layers, (2) two 500

BLSTM layers, (3) three 300 BLSTM layers. Networks in (1) and (2) were randomly initialized, and the trained network in (1) was used to initialize the first two layers in (3), while the third BLSTM layer in 3) was initialized randomly. In the later iteration, since the performance were similar in the first iteration, only (1) was re-trained, using the trained net in the first iteration as initialization. The learning rate for all the cross entropy LSTM training was set to 50.

The stochastic gradient descent with momentum of 0.9 was adopted as the optimizer for all experiments. To alleviate the local optimum problem, Gaussian noise with 0 mean and 0.6 standard deviation was added to the inputs. For all the experiments, the training was stopped if there was no improved result for the development set for 20 epochs.

The sequence training LSTM was initialized with the trained cross entropy LSTM in the same iteration. Because of the well known overfitting problem of sequence training, the learning rate should be smaller than the CE training, which was set to 1 in the experiment. In the experiment, we observed that only the first epoch of sequence training gave the biggest boost in performance. Therefore, for all the experiments, the discriminative sequence training was performed for one epoch.

The sequence trained Kaldi DNN[16] was used as the comparison. The Kaldi DNN contains 8 feedforward layers, each of which has 4096 hidden nodes. Following the CHiME challenge, the word-error-rate (WER) was used as the evaluation criterion, lower meaning the better performance. Note that for the iterative architecture, since the result didn't improve after the second iteration, the results of first two iterations are reported.

5.3. Multitask Architecture

Similar to the pipeline experiment, the 100-dimensional noisy logarithmic mel-filter bank was used in the multitask experiment. However, in multitask architecture, the clean alignment was given as the objective, the noisy alignment was not used in the input,making the overall dimension of input feature 100.

As shown in Figure 1, the multitask network has two BLSTM layers with 300 hidden nodes at bottom, shared between the SE and ASR networks. Then separate feedforward output layers were used for the SE and ASR tasks correspondingly. The SE branch including the BLSTM layers were initialized with the separately trained SE BLSTM network. And the ASR feedforward layer was initialized randomly.

Four different α values were experimented with, which are 0 10^{-1}, 10^{-2} and 10^{-3}. Note that when $\alpha = 0$, the model would be reduced to the BLSTM for speech enhancement only. After the multitask training, a cross entropy LSTM of the same setting as in pipeline experiment was trained on the enhanced speech to obtain the WER results.

5.4. Results and discussion

The results are shown in Table 1$\sim$ 3. The WER-dev and WER-eval refer the word-error-rate for the development set and evaluation set.

Table 3: *Multitask architectue results*

α	WER-dev%	WER-eval%
0	**25.09**	20.60
10^{-3}	25.43	**20.11**
10^{-2}	25.20	20.39
10^{-1}	25.39	21.70

Table 1: *Result for different LSTM topologies*

Network Topology		WER-dev%	WER-eval%
Cross Entropy	Kaldi-DNN	22.42	17.13
	300-300	**21.28**	17.04
	500-500	21.78	17.08
	300-300-300	21.81	**16.99**
Sequence Training	Kaldi-DNN	21.54	16.58
	300-300	**20.11**	**16.04**
	500-500	20.26	16.12
	300-300-300	20.29	16.09

Table 2: *Iterative architectue results*

Iteration		WER-dev%	WER-eval%
Cross Entropy	1	21.28	**17.04**
	2	**21.03**	17.36
Sequence Training	1	20.11	**16.04**
	2	**19.91**	16.32

From Table 1, we can see that the proposed structure performed significantly better than the DNN baseline on both the development and evaluation data, and in both the cross entropy and sequence training conditions. We can observe that, compared with cross entropy training, the discriminative sequence training would lead to significantly better result. Also note that the best reported WER for the same task in [17] and [18] are 26.86% and 22.77%, which are much lower than the results for the proposed models. It further proves the effectiveness of the proposed model.

In Table 2, although the performance on the development set improved for the second iteration, the WER on the evaluation set did not benefit from the iterations. One possible explanation is that although the alignment for the second iteration was more accurate than the first iteration because of the better recognizer, the confusion between similar states, which is the main bottleneck for the recognition, did not increase much.

In Table 3, we can see that the WER of multi task network was lower than the baseline when $\alpha = 10^{-2}$ and 10^{-3} but was higher when $\alpha = 10^{-1}$. Since the recognition branch is mainly designed for increasing the discrimination among states in the enhanced speech, it is a compromise between the enhancement quality and discrimination. When α is too large, the worse enhancement quality may counter effect the better discrimination achieved for the recognition, which may lead to worse overall result. Compared with the results in Table 1, the WERs in Table 3 were higher. This shows that the noisy alignment is important for the speech enhancement. Also it suggests that by combining the input noisy alignment and the multitask architecture, it could have more potential to have better performance.

6. Conclusions

In this work, the problem of speech recognition under noisy conditions is investigated. Three different integration models between LSTM speech enhancement and LSTM speech recognition was proposed and evaluated. By combining the LSTM speech enhancement, cross entropy LSTM and discriminative sequence training LSTM, the WER of the proposed system was significantly lower than the state of the art. A novel multi-task LSTM architecture is also proposed and demonstrate potential in better ASR performance.

7. References

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] F. Weninger, J. Le Roux, J. R. Hershey, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *Proc. IEEE GlobalSIP 2014 Symposium on Machine Learning Applications in Speech Processing*, 2014.

[3] H. Erdogan, J. R. Hershey, J. Le Roux, and S. Watanabe, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *Proc. of ICASSP*, Apr. 2015.

[4] F. Weninger, S. Watanabe, J. Le Roux, J. Hershey, Y. Tachioka, J. Geiger, B. Schuller, and G. Rigoll, "The merl/melco/tum system for the reverb challenge using deep recurrent neural network feature enhancement," in *Proc. REVERB Workshop*, 2014.

[5] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602–610, 2005.

[6] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *arXiv preprint arXiv:1402.1128*, 2014.

[7] H. Sak, O. Vinyals, G. Heigold, A. Senior, E. McDermott, R. Monga, and M. Mao, "Sequence discriminative distributed training of long short-term memory recurrent neural networks," in *Interspeech*, 2014.

[8] J. T. Geiger, Z. Zhang, F. Weninger, B. Schuller, and G. Rigoll, "Robust speech recognition using long short-term memory recurrent neural networks for hybrid acoustic modelling," in *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2014.

[9] J. T. Geiger, F. Weninger, J. F. Gemmeke, M. Wöllmer, B. Schuller, and G. Rigoll, "Memory-enhanced neural networks and nmf for robust asr," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 6, pp. 1037–1046, 2014.

[10] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks." in *INTERSPEECH*, 2013, pp. 2345–2349.

[11] W. Macherey, L. Haferkamp, R. Schlüter, and H. Ney, "Investigations on error minimizing training criteria for discriminative training in automatic speech recognition." in *INTERSPEECH*, vol. 2005, 2005, pp. 2133–2136.

[12] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second chimespeech separation and recognition challenge: Datasets, tasks and baselines," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 126–130.

[13] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[14] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7398–7402.

[15] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable minimum bayes risk training of deep neural network acoustic models using distributed hessian-free optimization," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[16] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," 2011.

[17] Y. Tachioka, S. Watanabe, J. Le Roux, and J. R. Hershey, "Discriminative methods for noise robust speech recognition: A chime challenge benchmark," *Proc. CHiME-2013, Vancouver, Canada*, pp. 19–24, 2013.

[18] C. Weng, D. Yu, S. Watanabe, and B.-H. F. Juang, "Recurrent deep neural networks for robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5532–5536.