Interspeech 2018
2-6 September 2018, Hyderabad

Twin Regularization for online speech recognition

Mirco Ravanelli, Dmitriy Serdyuk, Yoshua Bengio

Montreal Institute for Learning Algorithms (MILA)
Université de Montréal, Canada

mirco.ravanelli@gmail.com, serdyuk@iro.umontreal.ca

Abstract

Online speech recognition is crucial for developing natural human-machine interfaces. This modality, however, is significantly more challenging than off-line ASR, since real-time/low-latency constraints inevitably hinder the use of future information, that is known to be very helpful to perform robust predictions.

A popular solution to mitigate this issue consists of feeding neural acoustic models with context windows that gather some future frames. This introduces a latency which depends on the number of employed look-ahead features.

This paper explores a different approach, based on estimating the future rather than waiting for it. Our technique encourages the hidden representations of a unidirectional recurrent network to embed some useful information about the future. Inspired by a recently proposed technique called Twin Networks, we add a regularization term that forces forward hidden states to be as close as possible to cotemporal backward ones, computed by a "twin" neural network running backwards in time.

The experiments, conducted on a number of datasets, recurrent architectures, input features, and acoustic conditions, have shown the effectiveness of this approach. One important advantage is that our method does not introduce any additional computation at test time if compared to standard unidirectional recurrent networks.

Index Terms: online speech recognition, recurrent neural networks, regularization, deep learning.

1. Introduction

Automatic speech recognition (ASR) has made great strides in recent years [1]. Deep learning, in particular, has contributed to radical transformations in the field, allowing current technology to reach unprecedented performance levels [2, 3]. Despite the impressive achievements of the last years, many open challenges remain in the field [4].

One important issue is the performance drop observed when going from off-line to online speech recognition. The latter recognition modality is significantly more challenging than off-line ASR, due to the real-time/low-latency constraints which inevitably arise. To provide a speech transcription with low-latency, the speech decoding must start while acquiring the signal itself, forcing the acoustic model to perform predictions mostly based on current and past information. Future information plays an important role to perform robust predictions, due to both phoneme co-articulations and linguistic dependencies [5].

Despite its complexity, online speech recognition is a key component towards a more natural human-machine interaction, and extensive effort has been devoted in the last decade to improve this technology. Past online recognizers were based on the GMM-HMM framework [6], while current solutions rely on deep learning [2]. In particular, the use of feed-forward Deep Neural Networks (DNNs), including both fully-connected and convolutional architectures, has been largely investigated in the literature [7, 8], especially in the context of online ASR performed on small-footprint devices [9–13]. Attempts have also been made to develop robust online speech recognizers based on RNNs, exploiting both the traditional RNN-HMM framework [14–19] and, more recently, end-to-end ASR technology [20, 21].

A common aspect of past approaches is that they often employ asymmetric context windows [22–24], that embed more past than future information. Despite their effectiveness, context windows inevitably introduce a trade-off between latency (that depends on the number of look-ahead frames) and recognition accuracy. Moreover window-based approaches focus on short-term future dependencies only, while long-term information cannot be used without incurring an unacceptable latency.

In contrast to past work, this paper attempts to predict the future rather than waiting for it. Our technique encourages the hidden representations of a unidirectional recurrent network to embed some relevant features about the future, providing useful information on the upcoming phonetic and linguistic dependencies. Inspired by a recently proposed technique called *Twin Networks* [25], we add a regularization term that forces forward hidden states to be as close as possible to cotemporal backward ones, computed by a twin neural network running backwards in time. The twin backward network is employed at training time, when online constraints do not arise. At test time only the forward states are computed, leading to a model that (ideally) does not introduces any latency and does not add any computation compared to standard unidirectional recurrent networks.

The experiments, conducted on several datasets, recurrent architectures, input features, and acoustic conditions, have shown the effectiveness of our approach. To summarize, the contribution of this paper is two-fold. Firstly, we design a novel method for online ASR that predicts future states of the model and demonstrate that it is well motivated. Secondly, we evaluate the proposed method under a variety of experimental conditions, showing its effectiveness against strong baselines.

The rest of the paper is organized as follows. Twin regularization for online ASR and the related work are outlined in Sec. 2. The experimental setup and the results are presented and discussed in Sec. 3 and Sec. 4 respectively. Finally, Sec. 6 draws our conclusions.

2. Twin Regularization for Online ASR

In the context of hybrid RNN-HMM speech recognition, the neural network processes the input speech sequence $X = \{x_1, ..., x_t, .., x_N\}$ and computes, at each time step t, a hidden state in the following way:

$$\overrightarrow{h_t} = f_{\overrightarrow{\theta}}(x_t, \overrightarrow{h}_{t-1}), \quad (1)$$

10.21437/Interspeech.2018-1407

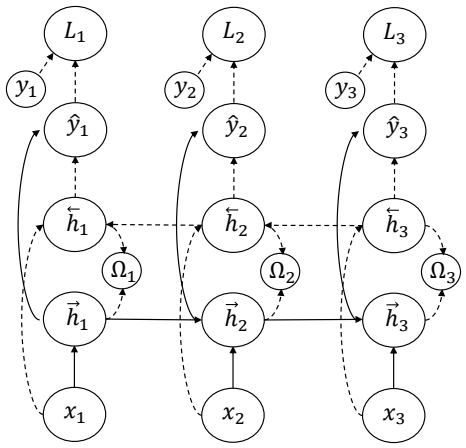

Figure 1: *An example of a recurrent acoustic model extended with a Twin Network. The regularization term Ω encourages forward and backward hidden states to be as close as possible. Dashed lines refer to computations done at training time only.*

where $f_{\overrightarrow{\theta}}$ is a function that depends on trainable parameters $\overrightarrow{\theta}$ (such as an LSTM or a GRU cell), and $\rightarrow$ highlights that the network scans the sequence in the forward direction. The forward states summarize the information about current and past elements of the sequence. A linear transformation, followed by a softmax classifier, is then employed to perform predictions $\widehat{Y} = \{\widehat{y}_1, ..., \widehat{y}_t, .., \widehat{y}_N\}$ over a set of phone states.

These posterior probabilities, after being normalized by their prior, feed a HMM-based decoder that integrates acoustic and linguistic information into a search graph and estimates the sequence of words uttered by the speaker. The decoding step is normally very computationally demanding, especially for large vocabulary speech recognition. For the unidirectional RNN models described above, however, the output $\widehat{y}_t$ at each time step t can be computed without waiting for the full speech utterance to finish. The decoding step can thus start while acquiring the speech signal from the user.

The RNN model is trained to optimize the *negative log-likelihood* (NLL) cost function:

$$L(X, Y; \overrightarrow{\theta}) = -\frac{1}{N}\sum_{t=1}^{N} \log P_{\overrightarrow{\theta}}(y_t|\{x_0, .., x_t\}),$$

where $Y = \{y_1, ..., y_t, .., y_N\}$ is the sequence of targeted phone labels and $P_{\overrightarrow{\theta}}(y_t|\{x_0, .., x_t\})$ is the output probability estimated by the neural network (that is given by reading the entry for y_t from the RNN output vector $\widehat{y}_t$).

When possible, it is very convenient to also process the speech sequence in the reverse time order, and compute backward states similarly to Eq. 1:

$$\overleftarrow{h}_t = f_{\overleftarrow{\theta}}(x_t, \overleftarrow{h}_{t+1}).$$

The backward states summarize the information about current and future elements of the sequence. In standard Bidirectional RNNs [26], forward and backward hidden states are combined to perform predictions based on the whole speech sequence. This leads to a substantial performance improvement in ASR [5]. Differently from unidirectional RNNs, bidirectional models cannot be used for online speech recognition, since each prediction $\widehat{y}_t$ depends on the full input sequence X.

Nevertheless, even if the future is not accessible, we can try to roughly predict it, capturing some relevant features that help phone predictions. This principle can be implemented by means of a regularization term, as highlighted in Fig. 1. The idea is to penalize forward hidden representations $\overrightarrow{h}_t$ that are distant from the cotemporal backward ones $\overleftarrow{h}_t$ [25]. With this regard, one can add a regularization term that encourage the network to minimize the L_2 distance between forward and backward hidden states:

$$\Omega(\overrightarrow{\theta}, \overleftarrow{\theta}) = \frac{1}{N}\sum_{t=1}^{N} \|\overrightarrow{h}_t - \overleftarrow{h}_t\|^2. \quad (2)$$

The regularization term is averaged over all time steps. In general, multiple recurrent hidden layers are stacked together to perform more robust predictions. In this case, the regularization term can be simply averaged over all the recurrent layers. The total objective to be minimized thus becomes a weighted sum of the NLL costs plus the regularization term:

$$\widetilde{L}(X, Y; \overrightarrow{\theta}, \overleftarrow{\theta}) = L(X, Y; \overrightarrow{\theta}) + L(X, Y; \overleftarrow{\theta}) + \lambda\Omega(\overrightarrow{\theta}, \overleftarrow{\theta}),$$

where λ is an hyper-parameter controlling the importance of the penalty term, and $\widetilde{L}$ is the total loss that is averaged over all the sentences composing the mini-batch.

Note that the backward states are needed only at training time, when online constraints do not arise. During testing, the part of the model computing backward states can be omitted. This leads to an architecture particularly suitable for online ASR, since it requires exactly the same amount of computations needed for standard unidirectional RNNs (at inference time). Another remarkable aspect of this technique is that Eq. 2 is based on the backward states $\overleftarrow{h}_t$, that provide a summary of the full future part of the speech sequence. This means that our method could capture not only short-term future dependencies, but also long-term ones.

2.1. Related Work

Several methods have been proposed in the literature to approach online ASR with RNNs. A popular choice is to feed the RNN with a context window that embeds some future frames [14, 15]. Attempts have also been done to build low-latency bidirectional RNNs [16–19]. The latter solutions are based on chunking the speech signal into several overlapping or non-overlapping windows. Each chunk embeds both past and future speech frames and is processed by an bidirectional RNN to perform phone predictions. These approaches, however, only account for a limited fraction of the future information, inheriting the same issues discussed for feed-forward DNNs.

This paper proposes the use of twin regularization to improve the way online RNNs exploit the future information. Twin regularization has been recently proposed in [25]. Its effectiveness has been proved in sequence generation tasks, such as image captioning, language modeling, and monaural singing voice separation [27]. Some works [28, 29] take a similar approach to train stochastic recurrent models with a backward running RNN. These approaches have been applied to speech synthesis, language modeling, image generation, and demonstrate that the idea of predicting future states is well-motivated, practically sound, and worth exploring.

To the best of our knowledge, this paper is the first attempt to build a predictor of future states for an online ASR model.

Table 1: *PER(%) on the TIMIT dataset obtained with various RNN architectures and input features. Our approach* UniTwin *shows stable improvement in all cases.* BiDir *corresponds to the off-line recognition with bidirectional networks (reported here to provide a lower bound for the error rates of the online models).*

	MFCC			FBANK			fMLLR		
	UniDir	UniTwin	BiDir	UniDir	UniTwin	BiDir	UniDir	UniTwin	BiDir
LSTM	17.3	**17.1**	15.7	17.2	**17.0**	15.1	16.9	**16.3**	14.7
GRU	17.9	**17.4**	16.0	18.1	**18.0**	15.3	16.9	**16.6**	15.3
M-GRU	17.8	**17.5**	16.1	18.0	**17.6**	15.4	16.8	**16.4**	15.1
Li-GRU	17.5	**17.1**	15.5	17.3	**16.8**	14.6	16.7	**16.2**	14.6

3. Experimental Setup

In the following sub-sections, the corpora and the RNN-HMM setting adopted for the experimental activity are described.

3.1. Corpora and Tasks

The first set of experiments was performed with the TIMIT corpus [30], considering the standard phoneme recognition task (aligned with the Kaldi s5 recipe [31]).

To validate our model in a more challenging scenario, experiments were also conducted in distant-talking conditions with the DIRHA-English dataset [32]. Training was based on the original WSJ-5k corpus (consisting of 7138 sentences uttered by 83 speakers) that was contaminated with a set of impulse responses measured in a real apartment [33]. The test phase was carried out with the real-part of the dataset, consisting of 409 WSJ sentences uttered in the aforementioned apartment by six native American speakers.

Additional experiments were conducted with the CHiME 4 dataset [34], that is based on speech data recorded in four noisy environments (on a bus, cafe, pedestrian area, and street junction). The training set is composed of 43690 noisy WSJ sentences recorded by five microphones (arranged on a tablet) and uttered by a total of 87 speakers. The test set *ET-real* considered in this work is based on 1320 real sentences uttered by four speakers, while the subset *DT-real* has been used for hyperparameter tuning. The CHiME experiments were based on the single channel setting [34].

Finally, experiments were performed with LibriSpeech [35] dataset. We used the training subset composed of 100 hours and the *dev-clean* set for the hyperparameter search. Test results are reported on the *test-clean* part.

3.2. RNN-HMM setting

The experiments are set up considering different acoustic features, i.e., 39 MFCCs (13 static+Δ+$\Delta\Delta$), 40 log-mel filterbank features (FBANKS), as well as 40 fMLLR features (extracted as reported in the s5 recipe of Kaldi [31]), that were computed using windows of 25 ms with an overlap of 10 ms.

Neural acoustic models consisted of multiple recurrent layers, that were stacked together prior to the final softmax classifier. These recurrent layers were unidirectional or bidirectional RNNs. Beyond standard LSTM [36] and GRU [37], we also considered recently proposed architectural variations [38]: M-GRU [39] is the minimal GRU architecture based on replacing reset gate with updated gate activations, while light GRU (Li-GRU) [40] directly avoids the reset gate and exploits ReLU activations for the hidden activations.

The feed-forward connections of the architecture were initialized according to the *Glorot*'s scheme [41], while recurrent weights were initialized with orthogonal matrices [42]. Recurrent dropout was used as regularization technique [43, 44]. Batch normalization was adopted for feed-forward connections only, as proposed in [38, 45].

The optimization was done using the RMSprop algorithm running for 24 epochs. The performance on the development set was monitored after each epoch, and the learning rate was halved when the relative performance improvement went below 0.1%. Back-propagation through time was not truncated, allowing the system to learn arbitrarily long time dependencies.

The main hyperparameters of the model (i.e., learning rate, number of hidden layers, hidden neurons per layer, dropout factor, as well as the twin regularization term λ) were optimized on the development datasets. In particular, we guessed some initial values according to our experience, and starting from them we performed a grid search to progressively explore better configurations. As a result, we adopted $\lambda = 0.6$ for TIMIT experiments, and $\lambda = 0.1$ for the other datasets. Please refer to the github repository referenced in the footnote below for more details about the considered hyperparameters.

The labels were derived by performing a GMM-based forced alignment on the original training datasets (see the standard s5 recipe of Kaldi for more details [31]). During test, the posterior probabilities generated by the RNN were normalized by their prior probabilities. The obtained likelihoods were processed by an HMM-based decoder, that estimated the sequence of words uttered by the speaker.

The RNN part of the system was implemented with Pytorch [46], that was coupled with the Kaldi decoder [31] to form a context-dependent RNN-HMM speech recognizer.[1]

4. Results

In the following sub-sections, we report the experimental results obtained with TIMIT, DIRHA, CHiME, and LibriSpeech datasets.

4.1. Phoneme recognition on TIMIT

To provide a thorough assessment of our methodology, several RNNs models and features are considered. Table 1 shows the results obtained with TIMIT. The results with off-line bidirectional models (*BiDir* columns) are reported only to provide a lower bound for the error rates that can be achieved with an online model. Moreover, to ensure a more accurate comparison, five experiments varying the initialization seeds were conducted for each RNN model and input feature. Results of Table 1 are thus reported as the average *phone error rates* (PER)[2].

[1] The code is available at `http://github.com/mravanelli/pytorch-kaldi/`.

[2] Standard deviations σ range between 0.1 and 0.2 for all the experiments.

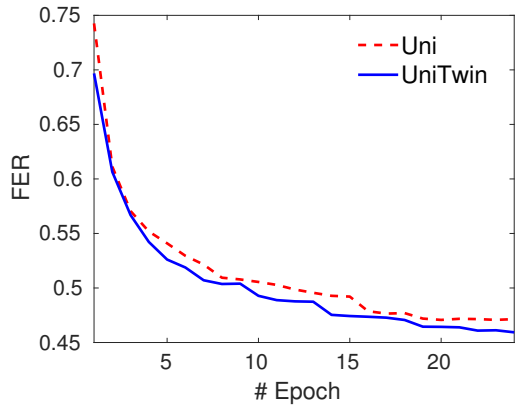

Figure 2: *Learning curves for unidirectional and twin RNN models (fbank features, Li-GRU model).*

Table 2: *PER(%) on TIMIT obtained with a context windows that embeds some future frames (Li-GRU model, fbank feats).*

# Future Frames	UniDir	UniTwin
0 frames	17.3	**16.8**
5 frames	16.5	**16.1**
10 frames	16.8	**16.5**
15 frames	17.5	**16.9**

Twin regularization (*UniTwin* columns) helps to improve the recognition performance, consistently outperforming standard unidirectional models (*UniDir* columns) in all the considered experimental conditions. Although our method is still far from bridging the gap with off-line bidirectional models, we observe an average relative improvement of 2.1%, which is obtained with a simple technique that does not introduce any additional computation at test time.

Li-GRU consistently outperforms the other RNN models, as previously observed in [38]. A remarkable achievement is the average PER of 14.6% obtained with Li-GRUs using fMLLR and fbank features. To the best of our knowledge, this result yields one of the highest published performance on the TIMIT test-set. We plot the learning curves for the *frame-level error rates* (FER) obtained on the development set over the duration of training in Fig. 2. Our experiments show that twin regularization converges to a better solution.

The results presented above are obtained with RNNs fed with the current frame only. Similarly to the window-based approaches described in Sec. 2, Tab. 2 extends our previous results by adding a small context window that concatenates some future frames. The table shows that a small look-ahead context window embedding 5 or 10 future frames is helpful to improve the ASR performance. Interestingly, our method outperforms standard unidirectional RNNs even under this experimental condition. This achievement confirms that twin regularization can focus on long-term future dependencies, providing useful information also when some look-ahead frames embed a short-term future context. This allows our method to be used in conjunction with previous window-based approaches.

4.2. Word recognition on other datasets

As a last experiment, we extend our previous achievements to more realistic ASR tasks. To test our technique into a complex acoustic scenario, Tab. 3 reports the *word error rate* (WER) obtained with the DIRHA dataset. For the sake of compactness, only the results with MFCCs are reported. It is worth mentioning that we obtained a similar experimental evidence using fbank and fMLLR features.

Table 3: *WER(%) for the DIRHA dataset (MFCC feats).*

	UniDir	UniTwin	BiDir
LSTM	32.9	**32.5**	27.8
GRU	30.2	**29.6**	27.2
Li-GRU	29.2	**28.7**	26.9

Table 4: *WER(%) for CHiME (ET-Real) and Librispeech (Test-Clean) using Li-GRU models and MFCC feats.*

	UniDir	UniTwin	BiDir
CHiME	23.7	**23.0**	19.2
LibriSpeech	10.4	**10.2**	9.2

We conclude from this experiment that twin regularization is also effective in challenging acoustic conditions characterized by the presence of both noise and reverberation.

To further test its robustness in noisy environments, we also performed some experiments with the CHiME dataset (see first row of Tab. 4). Moreover, to provide evidence on a larger vocabulary task, the second row of Tab. 4 reports the results achieved with LibriSpeech. These results are obtained with the 100 hours subset decoded with the *tgsmall* language model (see Kaldi s5 recipe [31]). Our experiments target the online ASR scenario, therefore the results reported in the table do not consider complex techniques as multi-microphone processing, data-augmentation. Neither system combination nor lattice rescoring are used here. It is however worth noting that the effectiveness of the proposed approach is one more time confirmed.

5. Conclusions

This paper explored the use of twin regularization for predicting the future states of an online RNN-HMM speech recognition. The proposed technique, that encourages forward hidden representations to be predictive of the future, has shown to be effective in several experimental conditions.

An average relative performance improvement of 2% is obtained over a standard unidirectional RNNs. The improvement is consistent across datasets, architectures, and input features. Furthermore, our proposed technique is simple and does not add any additional computational cost at test time.

A noteworthy aspect of our method is that it also accounts for long-term future dependencies, which differs from current dominant approaches based on short-term context windows. This offers the possibility of using twin regularization in conjunction with existing techniques.

6. Acknowledgment

We would like to thank Titouan Parcollet (Université d'Avignon), Kyle Kastner (Université de Montréal) and Maurizio Omologo (Fondazione Bruno Kessler) for their helpful comments. This research was enabled in part by support provided by Calcul Québec and Compute Canada (www.computecanada.ca).

7. References

[1] D. Yu and L. Deng, *Automatic Speech Recognition – A Deep Learning Approach*. Springer, 2015.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[3] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, no. 1, 2012.

[4] F. M. S. Watanabe, M. Delcroix and J. R. Hershey, *New Era for Robust Speech Recognition - Exploiting Deep Learning*. Springer, 2017.

[5] A. Graves, N. Jaitly, and A. Mohamed, "Hybrid speech recognition with Deep Bidirectional LSTM," in *Proc of ASRU*, 2013.

[6] X. Huang, A. Acero, and H. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall PTR, 2001.

[7] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *Proc. of ICASSP*, 2014.

[8] M. Bacchiani, A. W. Senior, and G. Heigold, "Asynchronous, online, GMM-free training of a context dependent acoustic model for speech recognition," in *Proc. of Interspeech*, 2014.

[9] Y. Wang, J. Li, and Y. Gong, "Small-footprint high-performance deep neural network-based speech recognition using split-VQ," in *Proc. of ICASSP*, 2015.

[10] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *Proc. of ICASSP*, 2014.

[11] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Proc. of Interspeech*, 2015.

[12] X. Lei, A. Senior, A. Gruenstein, and J. Sorensen, "Accurate and compact large vocabulary speech recognition on mobile devices." in *Proc. of Interspeech*, 2013.

[13] L. Lu and S. Renals, "Small-footprint highway deep neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, no. 7, 2017.

[14] X. Tian, J. Zhang, Z. Ma, Y. He, J. Wei, P. Wu, W. Situ, S. Li, and Y. Zhang, "Deep LSTM for large vocabulary continuous speech recognition," *arXiv:1703.07090*, 2017.

[15] V. Peddinti, Y. Wang, D. Povey, and S. Khudanpur, "Low Latency Acoustic Modeling Using Temporal Convolution and LSTMs," *IEEE Signal Processing Letters*, no. 3, 2018.

[16] S. Xue and Z. Yan, "Improving latency-controlled BLSTM acoustic models for online speech recognition," in *Proc. of ICASSP*, 2017.

[17] A. Zeyer, R. Schlüter, and H. Ney, "Towards online-recognition with deep bidirectional LSTM acoustic models," in *Proc. of Interspeech*, 2016.

[18] K. Chen and Q. Huo, "Training deep bidirectional LSTM acoustic model for LVCSR by a context-sensitive-chunk BPTT approach," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, no. 7, 2016.

[19] A. Mohamed, F. Seide, J. Yu, D.and Droppo, A. Stolcke, G. Zweig, and G. Penn, "Deep bi-directional recurrent networks over spectral windows," in *Proc. ASRU*, 2015.

[20] N. Jaitly, Q. V. Le, O. Vinyals, I. Sutskever, D. Sussillo, and S. Bengio, "An online sequence-to-sequence model using partial conditioning," in *Proc of NIPS*, 2016.

[21] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *ICML*, 2016.

[22] M. Ravanelli and M. Omologo, "Automatic context window composition for distant speech recognition," *Speech Communication*, 2018.

[23] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc of Interspeech*, 2015.

[24] M. Ravanelli and M. Omologo, "Contaminated speech training methods for robust DNN-HMM distant speech recognition," in *Proc. of Interspeech*, 2015.

[25] D. Serdyuk, N. R. Ke, A. Sordoni, A. Trischler, C. Pal, and Y. Bengio, "Twin networks: Matching the future for sequence generation," in *ICLR*, 2018.

[26] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, no. 11, 1997.

[27] K. Drossos, S. I. Mimilakis, D. Serdyuk, G. Schuller, T. Virtanen, and Y. Bengio, "MaD TwinNet: Masker-denoiser architecture with Twin Networks for monaural sound source separation," *IJCNN*, 2018.

[28] A. Goyal, A. Sordoni, M.-A. Côté, N. Ke, and Y. Bengio, "Z-Forcing: Training stochastic recurrent networks," in *Proc. of NIPS*, 2017.

[29] S. Shabanian, D. Arpit, A. Trischler, and Y. Bengio, "Variational Bi-LSTMs," *arXiv preprint arXiv:1711.05717*, 2017.

[30] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "DARPA TIMIT Acoustic Phonetic Continuous Speech Corpus CDROM," 1993.

[31] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *Proc. of ASRU*, 2011.

[32] M. Ravanelli, L. Cristoforetti, R. Gretter, M. Pellin, A. Sosi, and M. Omologo, "The DIRHA-English corpus and related tasks for distant-speech recognition in domestic environments," in *Proc. of ASRU*, 2015.

[33] M. Ravanelli, P. Svaizer, and M. Omologo, "Realistic multi-microphone data simulation for distant speech recognition," in *Proc. of Interspeech*, 2016.

[34] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third CHiME Speech Separation and Recognition Challenge: Dataset, task and baselines," in *Proc. of ASRU*, 2015.

[35] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. of ICASSP*, 2015.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, no. 8, Nov. 1997.

[37] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," in *Proc. of SSST*, 2014.

[38] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "Improving speech recognition by revising gated recurrent units," in *Proc. of Interspeech*, 2017.

[39] G. Zhou, J. Wu, C. Zhang, and Z. Zhou, "Minimal gated unit for recurrent neural networks," *arXiv:1603.09420*, 2016.

[40] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "Light gated recurrent units for speech recognition," *IEEE Transactions on Emerging Topics in Computational Intelligence*, no. 2, April 2018.

[41] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. of AISTATS*, 2010.

[42] Q. Le, N. Jaitly, and G. Hinton, "A simple way to initialize recurrent networks of rectified linear units," *arXiv:1504.00941*, 2015.

[43] T. Moon, H. Choi, H. Lee, and I. Song, "RNNDROP: A novel dropout for RNNS in ASR," in *Proc. of ASRU*, 2015.

[44] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Proc. of NIPS*, 2016.

[45] C. Laurent, G. Pereyra, P. Brakel, Y. Zhang, and Y. Bengio, "Batch normalized recurrent neural networks," in *Proc. of ICASSP*, 2016.

[46] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proc. of NIPS*, 2017.