INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Gating Recurrent Enhanced Memory Neural Networks on Language Identification

Wang Geng, Yuanyan Zhao, Wenfu Wang, Xinyuan Cai and Bo Xu

Interactive Digital Media Technology Research Center,
Institute of Automation Chinese Academy of Sciences, Beijing 100190, China
{wang.geng, yyzhao5231, wangwenfu2013, xinyuan.cai, xubo}@ia.ac.cn

Abstract

This paper proposes a novel memory neural network structure, namely gating recurrent enhanced memory network (GREMN), to model long-range dependency in temporal series on language identification (LID) task at the acoustic frame level. The proposed GREMN is a stacking gating recurrent neural network (RNN) equipped with a learnable enhanced memory block near the classifier. It aims at capturing the long-span history and certain future contextual information of the sequential input. In addition, two optimization strategies of coherent SortaGrad-like training mechanism and a hard sample score acquisition approach are proposed. The proposed optimization policies drastically boost this memory network based LID system, especially on the large disparity training materials. It is confirmed by the experimental results that the proposed GREMN possesses strong ability of sequential modeling and generalization, where about 5% relative equal error rate (EER) reduction is obtained comparing with the approximate-sized gating RNNs and 38.5% performance improvements is observed compared to conventional i-Vector based LID system.

Index Terms: language identification, gating recurrent neural networks, learnable enhanced memory block, SortaGrad-like training approach, hard sample score acquisition

1. Introduction

Nowadays, the state-of-the-art language identification (LID) system has benefited a lot from the successful application of the deep neural networks (DNN). As shown in existing research achievements [1, 2, 3, 4, 5, 6], deep bottleneck feed forward neural network (DBN) which is used as a front-end feature extractor within the i-Vector framework remarkably boosts the LID system. Addtionally, a rectified i-Vector scheme based on a unified DBN is proposed to apply DNN to LID task at larger scale [2]. Different from the works mentioned above, this universal architecture covers both the front-end discriminative feature extraction and back-end acoustic modeling stages. It improves the transferability of the pre-trained DBN and largely boosts the generalization capability of the DBN based i-Vector representation LID system.

Even though the DBN based i-Vector framework achieves huge performance improvement, two limitations are obvious. First, its complex architecture detriments the expansibility of the LID system. Second, the DBNs applied to LID task are either shallow architectures or developed independently from the classification task. Motivated by the inherent discriminative nature of DNNs, the works apply the feedforward deep neural network (FDNN) directly to the LID task at the acoustic frame level [7, 8]. The powerful modeling capability of the FDNN can complement the discrimination insufficiency of the i-Vector framework based on sufficient training corpus.

However, a FDNN has its inherent limitation, that is, the sequential nature of one utterance is ignored. Complementing to feedforward DNN, RNN is able to capture long-span dependency across the input temporal sequences. Long short term memory (LSTM) [9, 10] and gated recurrent units (GRU) [11, 12] RNNs are two commonly used recurrent architectures. The elaborately designed gating mechanism of GRU and LSTM conduces to their great success in many pattern recognition fields, including neural machine translation [13], speech synthesis [14, 13] and speech recognition [15, 16]. Inspired by the powerful capability in temporal modeling in the acoustic signal, the LSTM RNN is adopted on long-span discriminative feature learning over the input acoustic sequence for automatic language identification [7]. Since the simplified structure and more intelligible working mode of GRU, more attention has been drawn from the community [17, 14] and similar performance to LSTM was reported. Both of the two gating units have demonstrated significant superiority over the conventional hyperbolic tangent/sigmoid activation function [18, 19, 20]. To further augment memory learning, a new research direction focusing on constructing neural computing models becomes popular. These neural computing models are composed of various forms of explicit longer-range memory units [21, 22, 23].

Inspired by the above memory learning network, this paper proposes a novel gating recurrent neural memory network which is equipped with external learnable memory resources for automatic language identification at the acoustic frame level. The proposed memory network which combines the GRU RNNs with an external memory enhancement block can capture the long-range history context information and certain future contextual information. Our motivation is that the inherent gating architecture of GRU in modeling temporal dependencies across the acoustic signal can learn long-span discriminative features over the acoustic sequential input for LID. To the best of our knowledge, this is the first time that a GRU coupled with an external memory block scheme is applied at large scale for automatic language identification.

This proposed memory network is composed of GRU RNNs equipped with a learnable memory block near the classifier. The GRU RNNs model sequential dependency across the traditional acoustic inputs. The learnable memory enhancement block encodes certain future time-step adjacent activities of the GRU hidden layer into a fixed-size feature representation, which is fed into the classifier along with the GRU output. This

The work is supported by 973 Program in China, grant No. 2013CB329302.

http://dx.doi.org/10.21437/Interspeech.2016-684

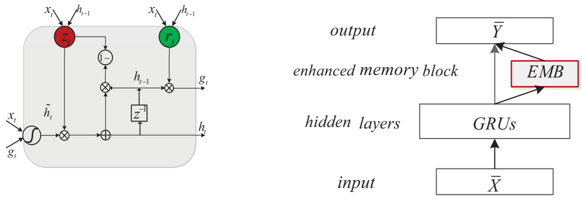
composition operation can integrate both the previous contextual information and certain future adjacent information within a look-ahead window from the present location.

Different from the work [7], this paper extends the gating RNNs model by integrating a memory enhancement block near the classifier. The learnable enhanced memory block which is a tapped-delay line structure augments the output of GRU RNNs by employing convolution-like operation within a look-ahead window into the future. It encodes certain future adjacent activities into a fixed-size feature representation. Two kinds of different external enhanced memory blocks are investigated: row shared convolution-like GREMN (rGREMN for short) and column shared convolution-like GREMN (cGREMN for short). These two convolution operation is similar to the encoding methods in feedforward sequential memory network (FSMN) [17, 21, 22]. The discriminative representation can be regarded as an integration of long surrounding context around current location . Another obvious difference is the model learning procedure, specifically, a SortaGrad-like training mechanism is explored in this paper. At evaluation stage, two methods on utterance level score acquisition are investigated, which are averaging the log of the classifier output of all the frames in an utterance (called soft average evaluation) and sampling the last representative frames within an utterance (called hard sample evaluation).

The remainder of this paper is organized as follows: Section 2 gives a brief description of the classical GREMN framework for LID. Experimental results and analysis are presented in Section 3, and our whole work is summarized in Section 4.

2. Gating recurrent enhanced memory networks

The proposed gating recurrent enhanced memory network (GREMN) will be described in this section. This new architecture is composed of a gating RNN and an external memory enhancement block near the classifier, see Figure 1(b).



(a) GRU (b) GREMN

Figure 1: *The architecture of GRU and GREMN.*

2.1. Gating recurrent neural networks

A simple RNN (SRNN) with tangent or sigmoid activation functions holds the potential to capture the long-range dependencies in time sequence. But, its learning process faces the challenge of the vanishing problem [24]. The LSTM and GRU which are equipped with various learnable gates are enhanced RNN architectures, these elaborately designed gating units ensure that the gradients can effectively flow back to the past.

The GRU which plays a role as the encoder-decoder in the machine translation [11] contains two gating units: update and reset gate. These two gates are used to modulate the flow of information inside the unit. Compared with LSTM, an candidate activation m_t is introduced to GRU, which is the accumulated vector representation of the network inputs and the learned histories. The output of GRU is controlled by update gate, it alternatively selects from the previous activation h_{t-1} and the candidate activation. Figure 1(a) presents the architecture of the GRU and the computation formulates are as follows:

$$r_t = \sigma\left(W_{rx}x_t + W_{rh}h_{t-1} + b_r\right) \tag{1}$$

$$m_t = g\left(W_{mx}x_t + W_{mh}\left(r_t \odot h_{t-1}\right)\right) \tag{2}$$

$$z_t = \sigma\left(W_{zx}x_t + W_{zh}h_{t-1} + b_z\right) \tag{3}$$

$$h_t = \left(1 - z_t\right) \odot h_{t-1} + z_t \odot m_t \tag{4}$$

where z, r, m and h respectively represents the activation vectors of update gate, reset gate, candidate and the unit output. σ denotes the logistic sigmoid function, W terms denote weight matrices and b terms are bias vectors.This simplified structure of GRU ensures its faster training process and lower divergence probability [7, 17].

2.2. Learnable enhanced memory blocks

Since the learnable enhanced memory block (EMB) can model certain long-range dependency in sequential data with a simplified structure, it can be used as a complementary modular to the GRU RNNs. The EMB adopts a tapped-delay line structure and employs a convolution-like mechanism. This EMB encodes future activations of the GRU RNNs output into a fixed-size representation, which is fed into the classifier along with the current activation of the GRU RNNs. Depending on the convolution mechanism, two different variants are adopted: i) row shared convolution-like EMB that elements in each row within a look-ahead window into the future share the same scalar encoding coefficient (rEMB for short); ii) column shared convolution-like learnable EMB that elements in each column within the encoding window share the same scalar encoding coefficience (cEMB for short).

For a T length acoustic input sequence, we denote the corresponding outputs of the GRU-RNN for the whole sequence as $\boldsymbol{H} = \{\boldsymbol{h}_1, ..., \boldsymbol{h}_T\}$. Suppose at time-step t, we use τ steps of future contexts. We now define a feature matrix $\boldsymbol{h}_{t:t+\tau} = [\boldsymbol{h}_{t+1}, \boldsymbol{h}_{t+2}, ..., \boldsymbol{h}_{t+\tau}]$ of size $D \times \tau$, so the scalar encoding coefficients of the row shared learnable EMB is D and the column shared one is τ. At each time instant t, the EMB encodes the future adjacent τ terms of $\boldsymbol{h}_t$ into a fixed-sized feature representation $\tilde{\boldsymbol{h}}_t$.

The computation process of rEMB which is the same with the scalar encoding method in works [17, 21] is actually a linear blend, specifically:

$$\tilde{\boldsymbol{h}}_t = \sum_{i=1}^{\tau} a_i{\cdot}\boldsymbol{h}_{t+i} \tag{5}$$

where $\boldsymbol{a} = \{a_1, a_1, ..., a_\tau\}$ denotes the row shared time-invariant coeffcients.

As for the column shared EMB, we use a parameter vector of size D to encode the future context information as follows:

$$\tilde{\boldsymbol{h}}_t = \sum_{i=1}^{\tau} \boldsymbol{a}{\odot}\boldsymbol{h}_{t+i} \tag{6}$$

where $\odot$ denotes element-wise multiplication, and the learnable parameter vector is defined as $\boldsymbol{a} = \{a_1, a_1, ..., a_D\}$. It is similar to the vectorized encoding method in works [17, 21] except that the encoding coefficience is a vector

Since both the two convolution-like encoding methods introduce very few parameters, they hold the intrinsic property of significantly faster convergence and can be extended to a much larger window size.

2.3. Gating recurrent enhanced memory networks

Figure 1(b) illustrates the GREMN architecture adopted in this paper. The EMB which is equipped between the GRU RNNs and classifier works as a complementary modular to the encoder, it encodes the activities of the GRU RNNs into a fixed-size feature representation and feeds it into the classifier along with the output of GRN RNNs. Therefore, a memory network based model is obtained to implement frame level classification to automatic language identification task.

With the composition mechanism of GRU and EMB, the frame level feature representation into the classifier can integrate both the previous information in the past as well certain future information within the look-ahead window block from the current location.

3. Experiments

3.1. Experimental setups

The NIST Language Recognition Evaluation (LRE) 2007 dataset is used for demonstrating the effectiveness of the proposed GREMN in this paper. The training dataset is composed of LRE05_OHSU, CALLFRIEND and LID05el and the experiment test corpus is a subset of the official NIST LRE 2007 3s condition evaluation set. 14 kinds of languages and 2158 segments are included in the 3s evaluation data. Both the training dataset and evaluation dataset come from Conversational Telephone Speech (CTS) audio source. There are three differences about the experiment corpus between the works [7, 8] and this paper. Firstly, only a subset of "Voice of America" news (VOA) audio source contained in LRE 2009 is adopted in the works [7, 8], while we evaluate our model on the CTS LRE 2007 dataset. Secondly, a subset of only 8 representative languages of which abundant training material (up to 200 hours) are selected for their experiment [7, 8], while we evaluate our memory network on all of the 14 languages to demonstrate the generalization capability of the proposed model despite of large disparity on training corpus for every language. Finally, the training utterances are split into random chunks of length between 2.5 and 3 seconds for better randomization and learning stability [7, 8], while we only split the much longer audio into about 30 seconds and keep the shorter ones in about 3 seconds to implement the proposed SortaGrad-like training mechanism in this paper.

The input of the memory network is the 42-dimensional acoustic feature vectors that composed of 13-dimensional perceptual linear prediction coefficients (PLP) and pitch coefficient along with their first and second delta. All experiments are carried on the open toolkit KALDI [25]. For experimental comparison, three types of architectures, which are LSTM RNNs, GRU RNNs and GREMN respectively, are established. Two kinds of LSTM RNNs models with different depth are explored. Each hidden layer of the LSTM RNNs contains 800 memory cells with 512 recurrent projection units while the GRU RNNs models with 800 memory units per layer. The GREMN are composed of 3 hidden layer GRU and a EMB near the classifier. All models are optimized with the famous truncated backpropagation through time (BPTT) learning algorithm [7, 8]. Additionally, the proposed SortaGraid-like training method is adopted to make full use of our training materials [17].

For test scoring, two types of utterance level score acquisition approaches are investigated, which are soft average evaluation and hard sample evaluation.

Table 1: *Performance (EER %) of the sortaGrad-like training method on LRE 2007 (3s segments).*

model	Hard Evaluation
GRU_h3_HB	15.75
GRU_h3_SL	13.16
LSTM_h3_HB	15.02
LSTM_h3_SL	12.24

3.2. Experimental results and analysis

We evaluate all memory network based frame level LID systems in this section. The proposed SortaGrad-like training mechanism will be discussed and two types of computing an utterance level score methods will be investigated.

3.2.1. SortaGrad-like training mechanism

Some algorithmic challenges exist when training RNNs on meterials of varying length. The work in [7, 8] tackled this issue by spliting the training examples into random chunks of duration 3 seconds using the BPTT training algorithm [26]. However, this harms the ability to learn longer-range correlations. Some works have found it a gradual process for learning long sequence modeling about recurrent neural networks and that presenting training materials in order of difficulty contributes a lot to online learning [27, 28]. Sequence learning in LID task faces the same challenge of tackling longer term dependencies as the automatic speech recognition [17].

In the GRU-RNN modeling learning procedure, the ability of learning longer span correlations partly relates to the length of the training examples. The hidden states in GRU-RNN depend on the previous ones implicitly, but, this dependency shrinks with the input sequence length increases. Inspired by the curriculum learning strategy algorithm: SortaGrad [17], we treat the length of the examples as a heuristic motivation for memory learning augmentation, since longer examples can further arouse potential of learning longer term dependencies for the GRU-RNN than short ones. Rather than splitting all the training corpus into chunks of duration 3 seconds, this work splits the much longer audio into about 30 seconds and keep the shorter ones in about 3 seconds. The implementation of the proposed SortaGrad-like method is as follows: At the early training stage, only the shorter examples in the training set are utilized for pre-training, which is an effective way to quickly bring the model parameters in a better range. Then the longer examples are used to further augment the ability to learn longer term dependencies of the GRU-RNNs. During training process, rather than setting the same alignment sparsely: 1 in every 5 frames for a chunk in the work [7, 8], we set the same target language id for each training example, so the errors are calculated from every frame in a trunk.

Table 1 shows a comparison of training precedure with and without SortaGrad-like. GRU_h3_SL and LSTM_h3_SL are the 3 hidden layers GRU and LSTM models learned by the SortaGrad-like training method while the GRU_h3_HB and LSTM_h3_HB are the models trained without it for contrast.

Experiment result shows that the SortaGrad-like training method is more effective to exploit the potential of the RNNs to model long-range sequence. Huge performance improvement is observed that about 16.4% and 18.5% relative reduction in EER are obtained on the 3 hidden layers GRU and LSTM model. We suspect that this benefit occurs primarily because the

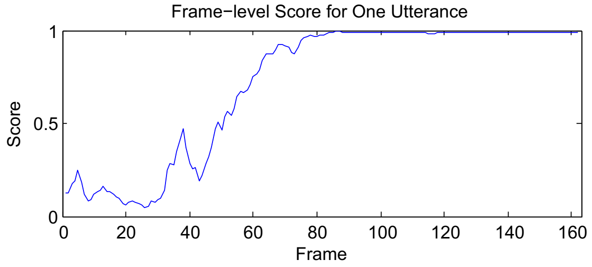


Figure 2: *Frame level score on target language.*

learning longer term dependencies is a process of gradual enhancement for the RNNs. The proposed SortaGrad-like training method best fits to this enhancement process: the shorter utterances used in the earlier training stage can ensure the model stability and the longer utterances further augment the sequential modeling capability and accelerate online learning.

3.2.2. Two types of utterance level score acquisition

At the evaluation stage, two methods on computing an utterance level score will be investigated for sequential input data. The first one is identical to the method adopted in works [7, 8] that averaging the log of the classifier output for the target language of all the frames in an utterance (soft average evaluation), the second one is carrying out a sampling on the last representative frames within an utterance (hard sample evaluation). The motivation of hard sample evaluation approach is that the inherent capability of sequential modeling of the memory networks and the ubiquitous phenomenon of increasing frame level score with time advancing showed in Figure 2. Theoretically, the output representative fixed-size features of the last few frames from the memory networks are more discriminative. So, this paper samples the representative features of the last few frames in an utterance and then take the average the sampled frames for evaluation. What's more, the heuristic observation on the test score distribution trend chart demonstrates the effectiveness of this hard sample evaluation method. Figure 2 illustrates the score distribution of one example on its target language. As it shows, the score on the target output of each frame increases with the temporal lasting in one utterance no matter whether this utterance gets the correct classification result. Depending on the proposed hard sample evaluation approach, about 18.8% and 19.5% relative EER reduction are observed on the 3 hidden layer GRU and LSTM models.

As shown in Table2, GRU_h2_SL, GRU_h3_SL, LSTM_h2_SL, LSTM_h3_SL are the 2 and 3 hidden layers GRU and LSTM models learned by the SortaGrad-like operation. The hard sample evaluation method performs much better than the traditional average operation based on the RNN models. This result confirms the ability of the RNN architecture to model longer-range context information. With temporal lasting in one test audio, the future adjacent frames which integrate the complex long-range correlations are more discriminative and get higher scores on its target language. Therefore, the proposed hard sample evaluation method is more effective and this evaluation operation is kept for the remainder of the experiments.

3.2.3. Evaluation of the GREMB in modeling sequential data

This section gives a detailed analysis on the augmentation of enhanced memory block to the GRU RNNs model. Two kinds of EMB: column shared enhanced memory block (cEMB) and row shared enhanced memory block (rEMB) are equipped to the 3 hidden layers GRU model to get the cGREMN and rGREMN models. In the experiment, two different sized windows are explored, conv_11 and conv_21 separately denote the future adjacentitems looking ahead into the future.

Table 2: *Performance (EER %) comparison between the hard sample evaluation and soft average evaluation approach.*

model	Soft Evaluation	Hard Evaluation
GRU_h2_SL	16.53	14.13
GRU_h3_SL	16.21	13.16
LSTM_h2_SL	16.03	12.92
LSTM_h3_SL	15.20	12.24

Table 3: *Performance (EER %) of the GREMNs.*

model	Hard Sample Evaluation
GRU_h3_SL	13.16
cGREMN_conv11	12.88
cGREMN_conv21	13.02
rGREMN_conv11	12.88
rGREMN_conv21	12.55
i-Vector baseline	20.39

As it shows in Tabel 3, both of the two kinds GREMN perform better than the 3 hidden layer GRU architecture. Obviously, the proposed rGREMN architecture with 21 time-step look-ahead window achieves the best performance that nearly 5% relative reduction compared with 3 hidden layer GRU and 38.5% relative reduction compared with traditional i-Vector approach in EER are obtained on the 3s test condition. Experimental result confirms the complement of the enhanced memory block to the GRU RNNs and further demonstrates that weight sharing in the convolution operation contribute to more effective modeling. To our knowledge, this is the best result of applying memory neural networks to LID tasks at large scale on the huge disparity of training materials. Distinguished from elaborately selecting a subset of only 8 representative languages with up to 200 hours training materials [7, 8], this paper evaluates the proposed GREMN on the all of the languages to demonstrate the generalization capability of the proposed model despite of large disparity on training corpus.

4. Conclusions

In his paper, a novel gating recurrent enhanced memory network (GREMN) is applied to the automatic language identification task to implement a frame level classification. The proposed GREMN is a stacking GRU RNNs equipped with a learnable enhanced memory block near the classifier. It is able to model the long-term history and certain future contextual information which best fits the frame level classification. Additionally, the two optimization strategies: the coherent SortaGrad-like training mechanism and a hard sample score acquisition approach, drastically boost the memory network based LID system, especially on the large disparity training materials. Excellent experimental results are observed that 5% relative EER reduction is obtained comparing with the GRU RNNs and 38.5% performance improvements is observed comparing with the conventional i-Vector based LID system.

5. References

[1] W. Geng, J. Li, S. Zhang, X. Cai, and B. Xu, “Multilingual tandem bottleneck feature for language identification,” in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[2] Y. Song, X. Hong, B. Jiang, R. Cui, I. V. McLoughlin, and L. Dai, “Deep bottleneck network based i-vector representation for language identification,” 2015.

[3] Y. Song, B. Jiang, Y. Bao, S. Wei, and L.-R. Dai, “I-vector representation based on bottleneck features for language identification,” *Electronics Letters*, vol. 49, no. 24, pp. 1569–1570, 2013.

[4] B. Jiang, Y. Song, S. Wei, I. V. McLoughlin, and L.-R. Dai, “Task-aware deep bottleneck features for spoken language identification.” in *INTERSPEECH*, 2014, pp. 3012–3016.

[5] Y. Song, R. Cui, X. Hong, I. McLoughlin, J. Shi, and L. Dai, “Improved language identification using deep bottleneck network,” in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4200–4204.

[6] T. Fu, Y. Qian, Y. Liu, and K. Yu, “Tandem deep features for text-dependent speaker verification.” in *INTERSPEECH*, 2014, pp. 1327–1331.

[7] J. Gonzalez-Dominguez, I. Lopez-Moreno, H. Sak, J. Gonzalez-Rodriguez, and P. J. Moreno, “Automatic language identification using long short-term memory recurrent neural networks.” in *INTERSPEECH*, 2014, pp. 2155–2159.

[8] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, and P. Moreno, “Automatic language identification using deep neural networks,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5337–5341.

[9] A. Graves, “Generating sequences with recurrent neural networks,” *arXiv preprint arXiv:1308.0850*, 2013.

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

[11] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.

[12] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.

[13] Y. Fan, Y. Qian, F.-L. Xie, and F. K. Soong, “Tts synthesis with bidirectional lstm based recurrent neural networks.” in *Interspeech*, 2014, pp. 1964–1968.

[14] W. Wang, S. Xu, and B. Xu, “Gating recurrent mixture density networks for acoustic modeling in statistical parametric speech synthesis,” in *Proc. ICASSP*. IEEE, 2016.

[15] Y. Miao, M. Gowayyed, and F. Metze, “Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding,” *arXiv preprint arXiv:1507.08240*, 2015.

[16] A. Graves and N. Jaitly, “Towards end-to-end speech recognition with recurrent neural networks,” in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1764–1772.

[17] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos *et al.*, “Deep speech 2: End-to-end speech recognition in english and mandarin,” *arXiv preprint arXiv:1512.02595*, 2015.

[18] Y. Zhao, J. Li, J. Xue, and Y. Gong, “Investigating online low-footprint speaker adaptation using generalized linear regression and click-through data,” pp. 4310–4314, 2015.

[19] P. Swietojanski and S. Renals, “Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models,” pp. 171–176, 2014.

[20] S. M. Siniscalchi, J. Li, and C.-H. Lee, “Hermitian polynomial for speaker adaptation of connectionist speech recognition systems,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2152–2161, 2013.

[21] S. Zhang, C. Liu, H. Jiang, S. Wei, L. Dai, and Y. Hu, “Feedforward sequential memory networks: A new structure to learn long-term dependency,” *arXiv preprint arXiv:1512.08301*, 2015.

[22] S. Zhang, H. Jiang, S. Wei, and L. Dai, “Feedforward sequential memory neural networks without recurrent feedback,” *arXiv preprint arXiv:1510.02693*, 2015.

[23] A. Graves, G. Wayne, and I. Danihelka, “Neural turing machines,” *arXiv preprint arXiv:1410.5401*, 2014.

[24] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *Neural Networks, IEEE Transactions on*, vol. 5, no. 2, pp. 157–166, 1994.

[25] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, “The kaldi speech recognition toolkit,” in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[26] R. J. Williams and J. Peng, “An efficient gradient-based algorithm for on-line training of recurrent network trajectories,” *Neural computation*, vol. 2, no. 4, pp. 490–501, 1990.

[27] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 41–48.

[28] W. Zaremba and I. Sutskever, “Learning to execute,” *arXiv preprint arXiv:1410.4615*, 2014.