Compact Feedforward Sequential Memory Networks for Large Vocabulary Continuous Speech Recognition

Shiliang Zhang[1], *Hui Jiang*[2], *Shifu Xiong*[1], *Si Wei*[1], *Lirong Dai*[1]

[1]NELSLIP, University of Science and Technology of China, Hefei, Anhui, P. R. China
[2] Department of EECS, York University, 4700 Keele Street, Toronto, Ontario, M3J 1P3, Canada
zsl2008@mail.ustc.edu.cn, hj@cse.yorku.ca, {sfxiong,siwei}@iflytek.com, lrdai@ustc.edu.cn

Abstract

In acoustic modeling for large vocabulary continuous speech recognition, it is essential to model long term dependency within speech signals. Usually, recurrent neural network (RNN) architectures, especially the long short term memory (LSTM) models, are the most popular choice. Recently, a novel architecture, namely feedforward sequential memory networks (FSMN), provides a non-recurrent architecture to model long term dependency in sequential data and has achieved better performance over RNNs on acoustic modeling and language modeling tasks. In this work, we propose a compact feedforward sequential memory networks (cFSMN) by combining FSMN with low-rank matrix factorization. We also make a slight modification to the encoding method used in FSMNs in order to further simplify the network architecture. On the Switchboard task, the proposed new cFSMN structures can reduce the model size by 60% and speed up the learning by more than 7 times while the models still significantly outperform the popular bi-direction LSTMs for both frame-level cross-entropy (CE) criterion based training and MMI based sequence training.

Index Terms: feedforward sequential memory networks, compact FSMN, speech recognition, low rank factorization, sequence training

1. Introduction

Recently, deep neural networks have become the dominant acoustic models in large vocabulary continuous speech recognition (LVCSR) systems. Depending on how the networks are connected, there exist various types of neural network architectures, such as feedforward neural networks (FNN) and recurrent neural networks (RNN). Over the past few years, feedforward fully-connected neural networks [1, 2, 3, 4, 5] and convolutional neural networks (CNN) [6, 7, 8] are widely used in acoustic modeling and have achieved more than 30 % relative performance improvement than the traditional Gaussian Mixture Model (GMM) based acoustic models. More recently, researchers have paid more and more attention to recurrent neural networks.

For acoustic modeling, it is crucial to take advantage of the long term dependency within the speech signal. Recurrent neural networks (RNN) [9] are designed to capture long term dependency within the sequential data using a simple mechanism of recurrent feedback. RNNs can learn to model sequential data over an extended period of time and store the memory in the network weights, then carry out rather complicated transformations on the sequential data. As opposed to FNNs that can only learn to map a fixed-size input to a fixed-size output, RNNs can in principle learn to map from one variable-length sequence to another. Therefore, RNNs, especially the short term memory (LSTM) [10], have become more and more popular in acoustic modeling [11, 12] for speech recognition.

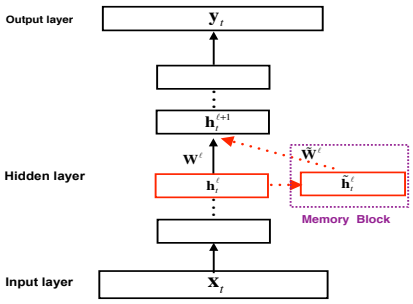

Figure 1: Illustration of the feedforward sequential memory networks (FSMN).

Compared with FNNs, an RNN is deep in time so that it is able to capture the long term dependency in sequences. Unfortunately, the learning of RNNs relies on the so-called back-propagation through time (BPTT) [13] due to the internal recurrent cycles that significantly increases the computational complexity of the learning. Because the learning of FNN is much easier and faster, it is somehow preferable to use a feedforward structure to learn the long-term dependency in sequences. A straightforward attempt is the so-called unfolded RNN [14], where an RNN is unfolded in time for a fixed number of time steps. The unfolded RNN only needs comparable training time as the standard FNNs while achieving better performance than FNNs. However, the context information learned by the unfolded RNNs is still very limited due to the limited number of unfolding steps in time. Moreover, it seems quite difficult to derive an unfolded version for more complex recurrent architectures, such as LSTM. Time delay neural network (TDNN) [15, 16, 17] is another popular feedforward architecture which can efficiently model the long temporal contexts.

Recently, in [18, 19], we have proposed a simple non-recurrent structure, namely feedforward sequential memory networks (FSMN), which can effectively model long term dependency in sequential data without using any recurrent feedback. Experimental results on acoustic modeling and language modeling tasks have shown that FSMN can significantly outperform the recurrent neural networks and these models can be learned much more reliably and faster. For a standard FSMN as shown in Figure 1, it is essentially a standard FNN with some memory blocks appended to the hidden layers and both the outputs from the hidden layer and memory block are fed

into the next hidden layer. Unfortunately, it may introduce a lot of additional parameters compared to FNN with the same architecture. If we want to reduce the model size, the straightforward method is to reduce the size of these hidden layers which are equipped with memory blocks. In this work, we propose a variant FSMN architecture, namely compact feedforward sequential memory networks (cFSMN), to simplify the FSMN architecture and speed up the learning. The proposed cFSMN is inspired by the previous works on low-rank weight matrix factorization in [20, 21] and LSTM with recurrent projection layer [12]. For cFSMN, we insert a separate smaller linear projection layer after the nonlinear hidden layer and add the memory block to the linear projection layer instead of the nonlinear hidden layer. Moreover, we also make a slight modification to the encoding method used in the previous FSMN work. As a result, we only need to feed the outputs of the memory block to the next hidden layer. We have evaluated the performance of FSMN and cFSMN under the frame-level cross-entropy (CE) criterion based fine-tuning and MMI based sequence training on the Switchboard (SWB) task. Experimental results have shown that cFSMN can make more effective use of model parameters than FSMN and BLSTM. For instance, cFSMN can reduce the model size by 60% and speed up the learning by more than 7 times while still achieving better performance than BLSTM. At last, we can achieve a WER of 12.0% without speaker-specific adaptation [22] and normalization by using cFSMN with MMI based sequence training.

2. Preliminaries: Feedforward Sequential Memory Networks

Feedforward sequential memory networks (FSMN) were proposed in [18, 19], which are essentially a standard feedforward fully connected neural network with some memory blocks appended to the hidden layers. For instance, Figure 1 shows a FSMN with one memory block added into its ℓ-th hidden layer. The memory block is used to encode N previous activities of the hidden layer into a fixed-size representation (called an N-th order FSMN), which is fed into the next hidden layer along with the current hidden activity. In [19], depending on the encoding method to be used, it has proposed two versions if FSMNs, namely scalar FSMNs (sFSMN) and vectorized FSMNs (vFSMN). Experimental results on the speech recognition task have shown that vFSMN can significantly outperform the sFSMN. Therefore, we only introduce vFSMNs in this paper.

Given an input sequence, denoted as $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$, where each $\mathbf{x}_t \in \mathcal{R}^{D\times 1}$ represents the input data at time instance t. We further denote the corresponding outputs of the ℓ-th hidden layer for the whole sequence as $\mathbf{H}^\ell = \{\mathbf{h}_1^\ell, \cdots, \mathbf{h}_T^\ell\}$, with $\mathbf{h}_t^\ell \in \mathcal{R}^{D_\ell \times 1}$. For an N-th order vFSMN, at each time instant t, we use a set of $N+1$ vector coefficients, $\{\mathbf{a}_i^\ell\}$, to encode $\mathbf{h}_t^\ell$ and its previous N terms at the ℓ-th hidden layer into a fixed-sized representation, $\tilde{\mathbf{h}}_t^\ell$, as the output from the memory block at time t:

$$\tilde{\mathbf{h}}_t^\ell = \sum_{i=0}^{N} \mathbf{a}_i^\ell \odot \mathbf{h}_{t-i}^\ell \tag{1}$$

Where $\odot$ denotes element-wise multiplication of two equally-sized vectors. In the above vFSMN definitions in eq.1, we call it unidirectional vFSMN since we only consider the past information in a sequence. It can be extended to bidirectional version

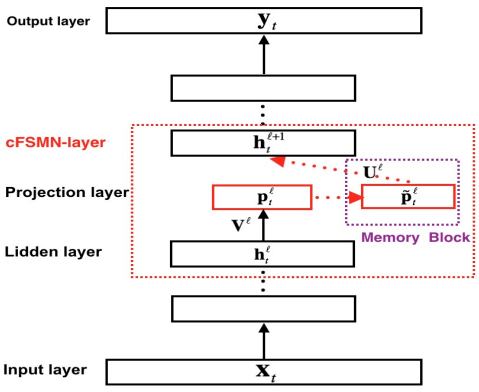

Figure 2: Illustration of the compact feedforward sequential memory networks (cFSMN).

as follows:

$$\tilde{\mathbf{h}}_t^\ell = \sum_{i=0}^{N_1} \mathbf{a}_i^\ell \odot \mathbf{h}_{t-i}^\ell + \sum_{j=1}^{N_2} \mathbf{c}_j^\ell \odot \mathbf{h}_{t+j}^\ell \tag{2}$$

Here, N_1 is called the look-back order, denoting the number of historical items looking back to the past, and N_2 the lookahead order, representing the size of the look-ahead window into the future.

The output from the memory block, $\tilde{\mathbf{h}}_t^\ell$, may be regarded as a fixed-size representation of the long surrounding context at time instance t. As shown in Figure 1, $\tilde{\mathbf{h}}_t^\ell$ can be fed into the next hidden layer in the same way as $\mathbf{h}_t^\ell$. As a result, we can calculate the activation of the units in the next hidden layer as follows:

$$\mathbf{h}_t^{\ell+1} = f(\mathbf{W}^\ell \mathbf{h}_t^\ell + \tilde{\mathbf{W}}^\ell \tilde{\mathbf{h}}_t^\ell + \mathbf{b}^\ell) \tag{3}$$

where $f(\cdot)$ denotes the nonlinear activation function (sigmoid or ReLU), and $\mathbf{W}^\ell$ and $\mathbf{b}^\ell$ represent the standard weight matrix and bias vector for layer ℓ, and $\tilde{\mathbf{W}}^\ell$ denotes the weight matrix between the memory block and the next layer.

Therefore, compared with the standard feedforward neural networks (FNN) with a fixed-size context window in the input layer, FSMN can memorize a much longer term dependency using the appended memory blocks. Moreover, we can even add several memory blocks to multiple hidden layers of a deep neural network to capture more context information in various abstraction levels.

3. A New Compact FSMN Structure

In this section, we introduce the architecture of the proposed compact feedforward sequential memory networks (cFSMN). The proposed cFSMN is essentially a standard FNN with some special linear projection layers equipped with memory blocks, namely cFSMN-layers. As shown in Figure 2, it is an cFSMN with a single cFSMN-layer in the ℓ-th layer. The cFSMN-layer consists of three parts: a linear projection layer, a memory block and a weight connection from memory block to the next hidden layer. Compared to FSMN, the cFSMN can be viewed as inserting a smaller linear projection layer after the nonlinear hidden layers and add the memory block to the linear projection layers instead of the hidden layers.

The operations in the memory block of cFSMN remain the same as the standard FSMNs, using either scalar or vector based

encoding. In this paper, we choose the vector based encoding method for cFSMN. Moreover, as shown in Figure 2, we have proposed to further simplify the FSMN structure, i.e., only the outputs of the memory block in cFSMN are fed into the next hidden layer, which is different from the standard FSMN in Figure 1. In order to do this, we need to make a slight modification to the encoding formulation of the memory block as follows:

$$\tilde{\mathbf{p}}_t^\ell = \mathbf{p}_t^\ell + \sum_{i=0}^{N} \mathbf{a}_i^\ell \odot \mathbf{p}_{t-i}^\ell \tag{4}$$

$$\tilde{\mathbf{p}}_t^\ell = \mathbf{p}_t^\ell + \sum_{i=0}^{N_1} \mathbf{a}_i^\ell \odot \mathbf{p}_{t-i}^\ell + \sum_{j=1}^{N_2} \mathbf{c}_j^\ell \odot \mathbf{p}_{t+j}^\ell. \tag{5}$$

where, $\mathbf{p}_t^\ell = \mathbf{V}^\ell \mathbf{h}_t^\ell + \mathbf{b}^\ell$ denotes the linear output of the ℓ-th linear projection layer. Eq.4 and eq.5 are used in the unidirectional and bidirectional cFSMN respectively. In eq.4 and eq.5, we additionally add the hidden activities of current time instance, $\mathbf{p}_t^\ell$, to the representation of the memory block. This operation is essential to provide alignment information during the beginning of the learning.

Moreover, we can calculate the activation of the units in the next hidden layer as follows:

$$\mathbf{h}_t^{\ell+1} = f(\mathbf{U}^\ell \tilde{\mathbf{p}}_t^\ell + \mathbf{b}^{\ell+1}) \tag{6}$$

Obviously, this structure is equal to feed both the output of the memory block and the linear projection layer to the next layer using the same weight matrix, instead of two different matrices in FSMNs. Like FSMNs, cFSMNs can be efficiently learned using the standard back-propagation (BP) with mini-batch based stochastic gradient descent (SGD).

4. Experiments

In this paper, we have evaluated the proposed compact feedforward sequential memory networks (cFSMN) on the Switchboard (SWB) database. The training data consists of 309-hour Switchboard-I training database and 20-hour Call Home English data. We divide the whole training data into two sets: training set and cross validation set. The training set contains 99.5% training data, and the cross-validation set contains the remaining 0.5%. Evaluation is performed in terms of word error rate (WER) on the NIST 2000 Hub5 evaluation set (containing 1831 utterances), denoted as Hub5e00.

4.1. Baseline Systems

As for the DNN-HMM baseline system, we follow the same training procedure as described in [23, 24] to train the conventional context dependent DNN-HMMs using the tied-state alignment obtained from the MLE trained GMM-HMMs baseline system. The total context-dependent tied-state is 8991. We have trained standard feedforward fully-connected deep neural networks (DNN) using either sigmoid or ReLU activation functions. The DNN consists of 6 hidden layers with 2,048 units per layer. The input to the DNN is the 120-dimensional log filter-bank (FBK) features concatenated from all consecutive frames within a long context window of 11 (5+1+5). The sigmoid DNN system is first pre-trained using the RBM-based layer-wise pre-training while the ReLU DNN is randomly initialized. In the fine-tuning, we use the mini-batch SGD algorithm to optimize the frame-level cross-entropy (CE) criterion. The mini-batch is set to 1024 and 4096 for sigmoid and ReLU DNNs respectively. And the initial learning rate is 0.2 and 0.08 for sigmoid and ReLU based DNNs. The performance of baseline DNN-HMMs systems is listed in Table 2 (denoted as Sigmoid-DNN and ReLU-DNN).

Furthermore, we rebuild the deep LSTM-HMM baseline systems by following the same configurations in [12]. The baseline LSTM-HMM contains three LSTM layers with 2048 memory cells per layer and each LSTM layer followed by a low-rank linear recurrent projection layer of 512 units. Each input to the LSTM is 120-dimensional filter-bank (FBK) features calculated from a 25ms speech segment. Since the information from the future frames is helpful for making a better decision for the current frame, we delay the output state label by 5 frames (equivalent to using a look-ahead window of 5 frames). The model is trained with the truncated BPTT algorithm [13] with a time step of 16 and a mini-batch size of 64 sequences using the frame-level cross-entropy (CE) criterion.

Moreover, we have also trained a deep bidirectional LSTM-HMMs baseline system. In our work, we have trained a deep BLSTM consisting of three hidden layers and 2048 memory cells per layer (1024 for forward layer and 1024 for backward layer). Similar to the unidirectional LSTM, each BLSTM layer is also followed by a low-rank linear recurrent projection layer of 512 units. The model is trained using the standard BPTT with a mini-batch of 16 sequences. The performance of the LSTM and BLSTM models is listed in the fourth and fifth rows of Table 2 respectively (denoted as LSTM and BLSTM).

For the FSMN, we have trained a bidirectional vFSMN in eq. (2) for this task. The vFSMN contains 6 hidden layer with 2048 units per layer and equipped with three bidirectional memory blocks in the first, third and fifth hidden layers respectively. The hidden units adopt the rectified linear (ReLU) activation function. We set both the look-back order and lookahead order to 40. The input is the 120-dimensional FBK features concatenated from three consecutive frames within a context window of 3 (1+1+1). In our work, we have found that it is enough to just concatenate three consecutive frames as input, which is different from DNNs. The learning schedule of vFSMN is the same as the baseline DNNs. The performance of the frame-level CE criterion trained vFSMN is as shown in Table 2.

4.2. cFSMN Results

For cFSMN, we have trained bidirectional cFSMNs in eq. 5 with various architectures: namely 360-N $\times$ [2048-P(N_1, N_2)]-M $\times$ 2048-P-8991. Here, N and M denotes the number of cFSMN-layers and fully-connected layers respectively, P is the size of the low rank linear projection layers, and N_1 and N_2 denotes the look-back order and lookahead order respectively. Here, we also apply the low-rank matrix factorization to the output layer. The input features and the learning schedule of cFSMN are the same as that of vFSMNs.

In the first experiment, we have investigated the influence of the number of cFSMN-layers and fully-connected layers on the final speech recognition performance. We have trained cFSMN with three, four and five cFSMN-layers. Detailed architectures and experimental results are listed in Table 1 , it has shown that the proposed cFSMNs are not sensitive to the number of the cFSMN-layers. Since the architecture with four cFSMN-layers followed by two fully-connected layers achieves the best performance, we continues to investigate the influence of the look-back and lookahead orders on the performance using this architecture. Experimental results in Table 1 also show that cFSMN can achieve a WER of 12.8% when the look-back and lookahead orders are both set to be 30. This is a very strong

Table 1: Performance (WER in %) of various cFSMN acoustic models in the Switchboard task.

cFSMN architecture	WER (%)
360-3x[2048-512(40,40)]-3x2048-512-8991	13.0
360-5x[2048-512(24,24)]-2x2048-512-8991	12.9
360-4x[2048-512(30,30)]-2x2048-512-8991	12.8
360-4x[2048-512(20,20)]-2x2048-512-8991	13.0
360-4x[2048-512(10,10)]-2x2048-512-8991	13.1

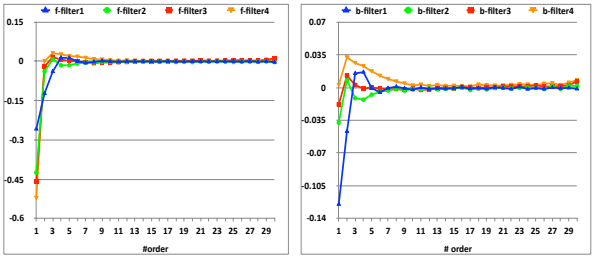

Figure 3: Illustration of the learned average filters in each memory blocks of cFSMN (order=30): left) the average coefficients of look-back filters; right) the average coefficients of lookahead filters.

performance reported on this task using the CE criterion without speaker-specific adaptation and model combination. In real-time speech recognition applications, we need to consider the decoding latency. In these cases, the bidirectional LSTMs are not suitable since the backward pass can not start until the full sequence is received, which normally cause an unacceptable time delay. However, the latency of bidirectional FSMNs can be easily adjusted by reducing the lookahead order. For instance, we can still achieve a very competitive performance (13.1% in WER) when setting the lookahead order to 10. In this case, the total latency per sequence is normally acceptable in the real-time speech recognition tasks.

In Figure 3, we have shown the learned average filters in each memory blocks of cFSMN when both look-back and lookahead orders are both set to be 30. From Figure 3, we can see that the energy of the learned filters is mainly concentrated in the first 10 orders. This explains why we can still achieve very promising result when setting the look-back and lookahead orders to 10. It indicates that the speech features of each phone are mostly influenced by several consecutive phones before and after, therefore there is no need to model the whole sequence as BLSTM does.

4.3. Model Comparison

In Table 2, we have summarized experimental results of various systems on the SWB task. Experimental results have shown that those models utilizing the long term dependency of speech signals, such as LSTM and FSMN, perform much better than DNN. The cFSMN in Table 2 consists of four cFSMN-layers followed by two fully-connected layers and the look-back and lookahead orders are both set to be 30. Form the results in Table 2 we can see that the proposed cFSMN can significantly outperform the vFSMN and BLSTM while being simpler in model structure and faster in learning speed. For instance, for one epoch of learning, BLSTMs take about 22.6 hours while the cFSMN only need about 3.1 hours, over 7 times speedup

Table 2: Comparison (model size in MB, training time per epoch in hour, recognition performance in WER) of various acoustic models in the Switchboard task. All models are trained with CE criterion using a single NVIDIA Tesla K20 GPU.

model	model size (MB)	time (hr)	WER (in %)
Sigmoid-DNN	160	5.0	15.6
ReLU-DNN	160	4.8	14.6
LSTM	110	9.4	14.2
BLSTM	180	22.6	13.5
vFSMN	203	6.9	13.4
cFSMN	**73**	**3.1**	**12.8**

Table 3: Performance (WER%) of various acoustic models trained with MMI criterion in the Switchboard task.

model	WER (in %)	
	CE	+ MMI sequence training
ReLU-DNN	14.6	13.4
LSTM	14.2	13.2
BLSTM	13.5	12.3
cFSMN	12.8	**12.0**

in training. Moreover, the total parameters of cFSMN is less than 40% of BLSTM. It indicates that cFSMN can make more effective use of the model parameters.

4.4. MMI based Sequence Training

In this experiment, we investigate the performance of various acoustic models using the MMI based full sequence training [25]. We use the best CE trained cFSMN (12.8% in WER) to generate the lattices which are used to train all the models listed in Table 3. For sequence training we use the MMI criterion to update the CE trained model for one epoch. From the results in Table 3, we can see that all the models can achieve the similar performance improvement after sequence training. Finally, we can achieve a WER of 12.0% by using cFSMN, which is still better than the BLSTM.

5. Conclusions

In conclusion, we have proposed a variant FSMN architecture, namely compact feedforward sequential memory networks (cFSMN), to simplify the FSMN architecture and speed up the learning. The cFSMNs can make more effective use of the model parameters. Experimental results on SWB task shown that cFSMNs can significantly outperform the FSMN and BLSTM while being simpler in model structure and faster in training speed. Overall, we can a WER of 12.0% by using cFSMNs with MMI based sequence training. This is a very competitive performance on this task without using speaker-specific adaptation and model combination.

6. Acknowledgements

We acknowledge the support of the following organizations or programs for research funding: National Nature Science Foundation of China (Grant No. 61273264), Science and Technology Department of Anhui Province (Grant No. 15CZZ02007), Chinese Academy of Sciences (Grant No. XDB02070006), National Key Technology Support Program (2014BAK15B05).

7. References

[1] A. R. Mohamed, G. Dahl, and G. Hinton, "Deep belief networks for phone recognition," in *NIPS workshop on deep learning for speech recognition and related applications*, vol. 1, no. 9, 2009.

[2] A. R. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 14–22, 2012.

[3] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012.

[4] J. Pan, C. Liu, Z. Wang, Y. Hu, and H. Jiang, "Investigation of deep neural networks (DNN) for large vocabulary continuous speech recognition: Why DNN surpasses GMMs in acoustic modeling," in *Chinese Spoken Language Processing (ISCSLP), 2012 8th International Symposium on*. IEEE, 2012, pp. 301–305.

[5] Y. Bao, H. Jiang, C. Liu, Y. Hu, and L. Dai, "Investigation on dimensionality reduction of concatenated features with deep neural network for LVCSR systems," in *Signal Processing (ICSP), 2012 IEEE 11th International Conference on*, vol. 1. IEEE, 2012, pp. 562–566.

[6] O. Abdel-Hamid, A. R. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, 2012, pp. 4277–4280.

[7] T. N. Sainath, A. R. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for lvcsr," in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*, 2013, pp. 8614–8618.

[8] O. Abdel Hamid, A. R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 10, pp. 1533–1545, 2014.

[9] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

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

[11] A. Graves, A. R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*, 2013, pp. 6645–6649.

[12] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling." in *Proceedings of Interspeech*, 2014, pp. 338–342.

[13] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[14] G. Saon, H. Soltau, A. Emami, and M. Picheny, "Unfolded recurrent neural networks for speech recognition," in *Proceedings of Interspeech*, 2014.

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

[16] V. Peddinti, G. Chen, D. Povey, and S. Khudanpur, "Reverberation robust acoustic modeling using i-vectors with time delay neural networks," *Proceedings of Interspeech*, 2015.

[17] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 37, no. 3, pp. 328–339, 1989.

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

[19] 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.

[20] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*, 2013, pp. 6655–6659.

[21] J. Xue, J. Li, and Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition." in *Proceedings of Interspeech*, 2013, pp. 2365–2369.

[22] S. Xue, O. Abdel-Hamid, H. Jiang, L. Dai, and Q. Liu, "Fast adaptation of deep neural network based on discriminant codes for speech recognition," *IEEE/ACM Trans. on Audio, Speech and Language Processing*, vol. 22, no. 12, pp. 1713–1725, 2014.

[23] S. Zhang, Y. Bao, P. Zhou, H. Jiang, and L. Dai, "Improving deep neural networks for LVCSR using dropout and shrinking structure," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 6849–6853.

[24] S. Zhang, H. Jiang, S. Wei, and L.-R. Dai, "Rectified linear neural networks with tied-scalar regularization for LVCSR," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[25] A. R. Mohamed, D. Yu, and L. Deng, "Investigation of full-sequence training of deep belief networks for speech recognition." in *Proceedings of Interspeech*, 2010, pp. 2846–2849.