EXPLORING MULTIDIMENSIONAL LSTMS FOR LARGE VOCABULARY ASR

Jinyu Li, Abdelrahman Mohamed, Geoffrey Zweig, and Yifan Gong

Microsoft Corporation, One Microsoft Way, Redmond, WA 98052
{jinyli, asamir, gzweig, ygong}@microsoft.com

ABSTRACT

Long short-term memory (LSTM) recurrent neural networks (RNNs) have recently shown significant performance improvements over deep feed-forward neural networks. A key aspect of these models is the use of time recurrence, combined with a gating architecture that allows them to track the long-term dynamics of speech. Inspired by human spectrogram reading, we recently proposed the frequency LSTM (F-LSTM) that performs 1-D recurrence over the frequency axis and then performs 1-D recurrence over the time axis. In this study, we further improve the acoustic model by proposing a 2-D, time-frequency (TF) LSTM. The TF-LSTM jointly scans the input over the time and frequency axes to model spectro-temporal warping, and then uses the output activations as the input to a time LSTM (T-LSTM). The joint time-frequency modeling better normalizes the features for the upper layer T-LSTMs. Evaluated on a 375-hour short message dictation task, the proposed TF-LSTM obtained a 3.4% relative WER reduction over the best T-LSTM. The invariance property achieved by joint time-frequency analysis is demonstrated on a mismatched test set, where the TF-LSTM achieves a 14.2% relative WER reduction over the best T-LSTM.

Index Terms— *LSTM, RNN, time and frequency, multidimensional*

1. INTRODUCTION

Recently, significant progress has been made in automatic speech recognition (ASR) thanks to the application of deep neural networks (DNNs) [1][2][3][4][5][6]. DNNs, however, only consider information in a fixed-length sliding window of frames and thus cannot exploit long-range correlations in the signal. Recurrent neural networks (RNNs), on the other hand, can encode sequence history in their internal state, and thus have the potential to predict phonemes based on all the speech features observed up to the current frame. Unfortunately, simple RNNs, depending on the largest eigenvalue of the state-update matrix, may have gradients which either increase or decrease exponentially over time. Thus, the basic RNN is difficult to train, and in practice can only model short-range effects. Long short-term memory (LSTM) RNNs [7][8] were developed to overcome these problems. LSTM-RNNs use input, output and forget gates to achieve a network that can maintain state and propagate gradients in a stable fashion over long spans of time. These networks have been shown to outperform DNNs on a variety of ASR tasks [9][10][11][12][13][14]. All previously proposed LSTMs use a recurrence along the time axis to model the temporal patterns of speech signals, and we call them T-LSTMs in this paper.

In common practice, log-filter-bank features are often used as the input to the neural-network-based acoustic model [15]. In standard systems, the log-filter-bank features are independent of one-another, i.e. switching the positions of two filter-banks won't affect the performance of the DNN or LSTM. However, this is not the case when a human reads a spectrogram: a human relies on both patterns that evolve on time, and frequency, to predict phonemes. Switching the positions of two filter-banks will destroy the frequency-wise patterns. Meanwhile, switching the positions of two frames will destroy the time-wise patterns. Inspired by the way people read spectrograms, we recently proposed frequency LSTM (F-LSTM) in [16] which performs recurrence along the frequency axis to summarize the frequency involving patterns as the feature for the upper level T-LSTMs. All the LSTM operations in [16] are one-dimensional, either along the frequency axis or the time axis.

However, both time-wise and frequency-wise patterns are important to human spectrogram reading. Hence, it may be better to extract feature with both patterns. Further, the concept of multidimensional processing has been proved very successful in the handwriting recognition tasks [17][18] and the computer vision tasks [19], and it outperformed the traditional handwriting systems that use convolutional neural networks (CNNs) [20][21] as the feature extractor.

The main contribution of this paper is the proposal to use a multidimensional LSTM to model both time and frequency dynamics for speech recognition. We further propose a method for doing this joint time-frequency analysis in a highly efficient way. We term the proposed method the time-frequency LSTM or TF-LSTM. Evaluated on a 375-hour Microsoft short message dictation (SMD) task, the TF-LSTM consistently outperformed the F-LSTM and obtained 3.4% relative word error rate (WER) reduction from the T-LSTM on the SMD test set, and a 14.2% relative WER reduction on a mismatched test set.

The rest of the paper is organized as follows. In Section 2, we briefly introduce LSTMs and then we present the proposed time-frequency LSTM in Section 3. We differentiate the proposed method from the convolutional LSTM DNN (CLDNN) [14] and multi-dimensional RNN [17][18] in Section 4. Experimental evaluation of the algorithm is provided in Section 5. We summarize our study and draw conclusions in Section 6.

2. THE LSTM-RNN

An RNN is fundamentally different from the feed-forward DNN in that the RNN does not operate on a fixed window of frames; instead, it maintains a hidden state vector, which is recursively updated after seeing each time frame. This allows RNNs to be resilient to arbitrary input warping along the recurrence dimension leading to better generalization abilities. Stacking multiple layers of RNNs allows the network to discover relationships between frames on progressively higher levels of abstraction.

During learning, the simple RNN suffers from the vanishing/exploding gradient problem [22]. This problem is well handled in the LSTM-RNNs through the use of the following four components:

- *Memory units*: these store the temporal state of the network;
- *Input gates*: these modulate the input activations into the cells;
- *Output gates*: these modulate the output activations of the cells ;
- *Forget gates*: these adaptively reset the cell's memory.

Taken together as in Figure 1 below, these four components are termed a LSTM cell.

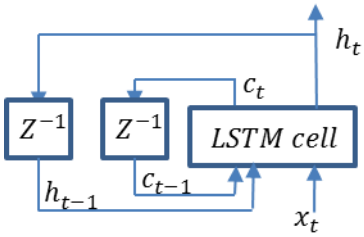

Figure 1: Architecture of LSTM-RNNs with one recurrent layer. Z^{-1} is a time-delay node.

Figure 1 depicts the architecture of an LSTM-RNN with one recurrent layer. In LSTM-RNNs, in addition to the past hidden-layer output $\boldsymbol{h}_{t-1}$, the past memory activation $\boldsymbol{c}_{t-1}$ is also an input to the LSTM cell.

This model can be described as:

$$\boldsymbol{i}_t^l = \sigma\left(\boldsymbol{W}_{xi}^l \boldsymbol{x}_t^l + \boldsymbol{W}_{hi}^l \boldsymbol{h}_{t-1}^l + \boldsymbol{W}_{ci}^l \boldsymbol{c}_{t-1}^l + \boldsymbol{b}_i^l\right), \quad (1)$$

$$\boldsymbol{f}_t^l = \sigma\left(\boldsymbol{W}_{xf}^l \boldsymbol{x}_t^l + \boldsymbol{W}_{hf}^l \boldsymbol{h}_{t-1}^l + \boldsymbol{W}_{cf}^l \boldsymbol{c}_{t-1}^l + \boldsymbol{b}_f^l\right), \quad (2)$$

$$\boldsymbol{c}_t^l = \boldsymbol{f}_t^l .* \boldsymbol{c}_{t-1}^l + \boldsymbol{i}_t^l .* \tanh\left(\boldsymbol{W}_{xc}^l \boldsymbol{x}_t^l + \boldsymbol{W}_{hc}^l \boldsymbol{h}_{t-1}^l + \boldsymbol{b}_c^l\right), \quad (3)$$

$$\boldsymbol{o}_t^l = \sigma\left(\boldsymbol{W}_{xo}^l \boldsymbol{x}_t^l + \boldsymbol{W}_{ho}^l \boldsymbol{h}_{t-1}^l + \boldsymbol{W}_{co}^l \boldsymbol{c}_t^l + \boldsymbol{b}_o^l\right), \quad (4)$$

$$\boldsymbol{h}_t^l = \boldsymbol{o}_t^l .* \tanh\left(\boldsymbol{c}_t^l\right), \quad (5)$$

where $\boldsymbol{i}_t^l$, $\boldsymbol{o}_t^l$, $\boldsymbol{f}_t^l$, and $\boldsymbol{c}_t^l$ denote the activation vectors of input gate, output gate, forget gate, and memory cell at the l-th layer and time t, respectively. $\boldsymbol{h}_t^l$ is the output of the LSTM cells at layer l and time t. $\boldsymbol{W}$ terms denote different weight matrices. For example, $\boldsymbol{W}_{xi}^l$ is the weight matrix from the cell input to the input gate at the l-th layer. $\boldsymbol{b}$ terms are the bias terms (e.g., $\boldsymbol{b}_i^l$ is the bias of input gate at layer l). ".*" denotes element wise multiplication.

In [11], a LSTM with an additional projection layer prior to the output was proposed to reduce the computational complexity of LSTM. A projection layer is applied to $\boldsymbol{h}_t^l$ as

$$\boldsymbol{r}_t^l = \boldsymbol{W}_{hr}^l \boldsymbol{h}_t^l$$

And then $\boldsymbol{h}_{t-1}^l$ in Eqs (1)--(4) is replaced by $\boldsymbol{r}_{t-1}^l$. In this study, we adopt this structure for T-LSTM modeling.

3. JOINT TIME-FREQUENCY ANALYSIS VIA MULTIDIMENSIONAL LSTM

In this section, we propose a time-frequency LSTM (TF-LSTM) as shown in Figure 2. In contrast to the frequency LSTM (F-LSTM) in our previous work [16] which scans the frequency bands so that frequency-evolving information is summarized by the output of the F-LSTM, the new method scans both the time and frequency axes jointly to perform the time-frequency analysis.

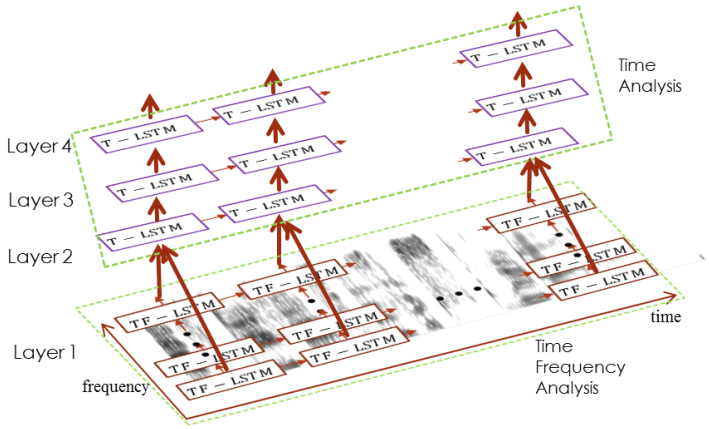

Figure 2: An example of time-frequency LSTM-RNN which scans both the time and frequency axis at the bottom layer using TF-LSTM, and then scans the time axis at the upper layers using T-LSTM. Note that the outputs of all TF-LSTM cells are fed into the upper layer T-LSTM.

The formulation of the TF-LSTM is as follows.

$$\boldsymbol{i}_{k,t}^l = \sigma\left(\boldsymbol{W}_{xi}^l \boldsymbol{x}_{k,t}^l + \boldsymbol{W}_{hi1}^l \boldsymbol{h}_{k,t-1}^l + \boldsymbol{W}_{hi2}^l \boldsymbol{h}_{k-1,t}^l + \boldsymbol{W}_{ci}^l \boldsymbol{c}_{k,t-1}^l + \boldsymbol{b}_i^l\right), \quad (6)$$

$$\boldsymbol{f}_{k,t}^l = \sigma\left(\boldsymbol{W}_{xf}^l \boldsymbol{x}_{k,t}^l + \boldsymbol{W}_{hf1}^l \boldsymbol{h}_{k,t-1}^l + \boldsymbol{W}_{hf2}^l \boldsymbol{h}_{k-1,t}^l + \boldsymbol{W}_{cf}^l \boldsymbol{c}_{k,t-1}^l + \boldsymbol{b}_f^l\right), \quad (7)$$

$$\boldsymbol{c}_{k,t}^l = \boldsymbol{f}_{k,t}^l .* \boldsymbol{c}_{k,t-1}^l + \boldsymbol{i}_{k,t}^l .* \tanh\left(\boldsymbol{W}_{xc}^l \boldsymbol{x}_{k,t}^l + \boldsymbol{W}_{hc1}^l \boldsymbol{h}_{k,t-1}^l + \boldsymbol{W}_{hc2}^l \boldsymbol{h}_{k-1,t}^l + \boldsymbol{b}_c^l\right), \quad (8)$$

$$\boldsymbol{o}_{k,t}^l = \sigma\left(\boldsymbol{W}_{xo}^l \boldsymbol{x}_{k,t}^l + \boldsymbol{W}_{ho1}^l \boldsymbol{h}_{k,t-1}^l + \boldsymbol{W}_{ho2}^l \boldsymbol{h}_{k-1,t}^l + \boldsymbol{W}_{co}^l \boldsymbol{c}_{k,t}^l + \boldsymbol{b}_o^l\right), \quad (9)$$

$$\boldsymbol{h}_{k,t}^l = \boldsymbol{o}_{k,t}^l .* \tanh\left(\boldsymbol{c}_{k,t}^l\right), \quad (10)$$

In this formulation, every gate now has three indices: layer l, frequency band k, and time t. For example, $\boldsymbol{f}_{k,t}^l$ denotes the activation vectors of forget gate at the layer l, frequency band k, and time t. Different from Eqs (1)--(4), now we have both time-delay input $\boldsymbol{h}_{k,t-1}^l$ and frequency-delay input $\boldsymbol{h}_{k-1,t}^l$. The $\boldsymbol{W}_{h.1}^l$ and $\boldsymbol{W}_{h.2}^l$ matrices denote the weight matrices connecting $\boldsymbol{h}_{k,t-1}^l$ and $\boldsymbol{h}_{k-1,t}^l$, respectively. The structure of a TF-LSTM cell is plotted in Figure 3, where ϕ denotes the tanh function.

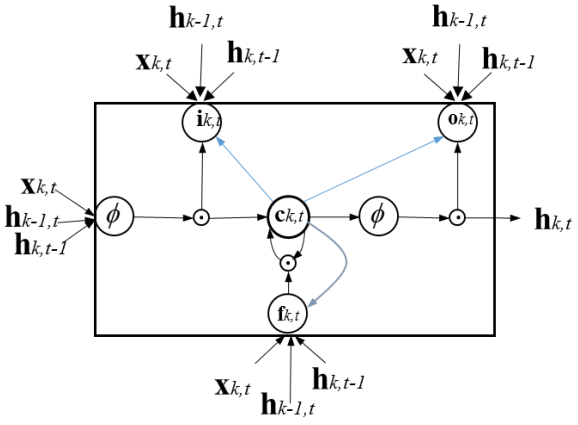

Figure 3: A TF-LSTM cell at frequency band k, and time t.

The proposed TF-LSTM in Eqs (6)--(10) is a general case of T-LSTM or F-LSTM. When all the time frequency bands are concatenated together as a single unit, frequency index k and all the items associated with $\boldsymbol{W}_{h.2}^{l}$ are removed. Then the TF-LSTM reduces to the T-LSTM of Eqs (1)--(5). In contrast, if all the items associated with $\boldsymbol{W}_{h.1}^{l}$ are removed, the TF-LSTM reduces to a F-LSTM, which can be viewed as removing the connections to $\boldsymbol{h}_{k,t-1}^{l}$ in Figure 3.

The detailed TF-LSTM processing is described as follows.

- At each time step, divide the N log-filter-banks at the current time into M overlapped chunks, shifting by C log-filter-banks between adjacent chunks. They are denoted as $\boldsymbol{x}_{k,t}^{1}, k = 1 \dots M$.
- Using the hidden activations at each frequency chunk from the previous time step $\boldsymbol{h}_{k,t-1}^{1}$, the hidden activations at each time step from the previous frequency chunk $\boldsymbol{h}_{k-1,t}^{1}$, and the input at the current frequency chunk and time step $\boldsymbol{x}_{k,t}^{1}$, go through Eqs (6)--(10) to generate the output of $\boldsymbol{h}_{k,t}^{1}, k = 1 \dots M$. Note that we use log-filter-banks as the input which means the time-frequency analysis is in the first layer, l is set as 1 in Eqs (6)--(10).
- Merge $\boldsymbol{h}_{k,t}^{1}, k = 1 \dots M$ into a super-vector $\boldsymbol{h}_{t}^{1}$ which can be considered as a trajectory of time-frequency patterns. Then use $\boldsymbol{h}_{t}^{1}$ as the input to the upper layer T-LSTM.

It is also worthwhile to investigate the stacking of multiple TF-LSTM layers. This can be easily done by replacing $\boldsymbol{x}_{k,t}^{l}$ with the hidden activations from the previous layer $\boldsymbol{h}_{k,t}^{l-1}$ in Eqs (6)--(9). Again, the output of the last TF-LSTM layer is merged into a super-vector as the input to the upper layer T-LSTM. A sample of stacked two TF-LSTM layers is shown in Figure 4.

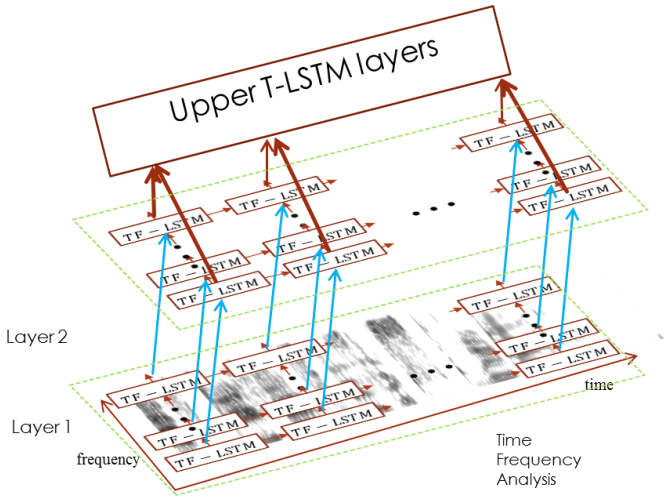

Figure 4: An example of stacked TF-LSTM layers.

4. RELATION TO PRIOR WORK

In this section, we first discuss the difference between our proposed TF-LSTM and the convolutional LSTM DNN (CLDNN) [14] which combines CNNs, LSTMs, and DNNs together. The CLDNN first uses a CNN to reduce the spectral variation, and then the output of the CNN layer is fed into a multi-layer LSTM to learn the temporal patterns. Finally, the output of the last LSTM layer is fed into several fully connected DNN layers for the purpose of classification.

The key difference between the TF-LSTM and the CLDNN is that the TF-LSTM uses joint time-frequency recurrence, whereas the CLDNN uses a sliding convolutional window for pattern detection. While the sliding window achieves some local invariance, it is not the same as a joint two-dimensional recurrent network which scans the whole time and frequency axis. The two approaches both aim to achieve invariance to input distortions, but the pattern detectors in the CNN maintain a constant dimensionality, while the TF-LSTM can perform a general time-frequency warping.

The proposed method is similar to the multidimensional LSTM [17][18] which is used for handwriting recognition. Multidimensional LSTM has been used in [23] on a very small phone recognition task, TIMIT [24], using connectionist temporal classification (CTC) [25] as the training criterion. However, there is no accuracy comparison with T-LSTM in [23]. In contrast, we will show the advantage of our proposed TF-LSTM over T-LSTM with the cross-entropy training criterion on a large scale speech recognition task in next section. Although using similar concepts, the proposed TF-LSTM has a different formulation from the multidimensional LSTM in [17][18]. The proposed TF-LSTM has only a single memory unit and a single forget gate while the multidimensional LSTM in [17][18] has multiple forget gates, each handling one dimensional information. Thus we achieve a significant reduction in complexity.

We are currently building a strong CLDNNs baseline to compare with, and it will be reported in the future. We will also implement the multidimensional LSTM with multiple forget gates [17][18] and compare with our proposed method.

5. EXPERIMENTS AND DISCUSSIONS

The proposed methods are evaluated on a Microsoft Windows phone short message dictation task. The transcribed training data contain 375 hours of US-English audio. The test set is from the same Windows Phone task, and has 125k words. This large test set guarantees the significance of reported improvement.

The 87-dimentional feature used in the DNN and T-LSTM experiments consists of the 29-dimensional static log-filter-bank outputs and their first- and second-order derivatives [26]. For the F-LSTM and TF-LSTM experiments, we only use the static log-filter-banks as the feature. All models evaluated in this study use 5976 tied-triphone states (senones), determined by a baseline CD-GMM-HMM system, and were trained to minimize the frame-level cross-entropy criterion. All experiments were conducted using the Computational Network Toolkit (CNTK) [27], which allows us to build and evaluate various network structures efficiently without deriving and implementing complicated training algorithms.

To build the baseline DNN, we augment the 87-dimensional feature vectors with 5 frames of context on either side (5-1-5). The DNN has 5 hidden layers, each with 2048 sigmoid units. The baseline T-LSTM is modeled after that in [11]. Each T-LSTM layer has 1024 hidden units and the output size of each T-LSTM layer is reduced to 512 using a linear projection layer. There is no frame stacking, and the output HMM state label is delayed by 5 frames as in [11]. When training T-LSTM, the backpropagation through time (BPTT) [28] step is 20. We use a 4-layer T-LSTM as our baseline.

This has 15.35% WER. It outperforms the baseline DNN with 10.39% relative WER reduction. This setup is better than the model with three or five T-LSTM layers as shown in Table 1. There is a 4.3% relative WER reduction when increasing one additional layer from 3-layer T-LSTM to 4-layer T-LSTM. However, a 5-layer LSTM does not outperform a 4-layer T-LSTM.

Table 1: WER and model size comparison of DNN and T-LSTM. M denotes million in the column of number of parameters.

Model	WER (%)	Number of parameters
DNN	17.13	30.2 M
3-layer T-LSTM	16.04	15.2 M
4-layer T-LSTM	15.35	19.8 M
5-layer T-LSTM	15.44	24.4 M

In Table 2, we compare the performance of the F-LSTM and TF-LSTM models. The F-LSTM model uses a single LSTM to scan the log-filter-banks while the TF-LSTM uses a single LSTM to scan both the time and log-filter-banks. The generated time-frequency evolving summary or the frequency evolving summary will then be passed into 3 or 4 layers of T-LSTMs.

At each time step, the 29 log-filter-bank channels are divided into 22 overlapped chunks with each chunk containing 8 log-filter-banks, which means the frequency shift is 1 log-filter-bank. This log-filter-bank grouping strategy follows our previous wisdom in CNN [29]. Then these 22 chunks are fed into F-LSTM. The input to the TF-LSTM cells includes not only the previous frequency chunks but also the output of this TF-LSTM cell in the previous time frame. Both the F-LSTM and TF-LSTM have 24 memory cells, introducing small computational cost. The upper layer T-LSTMs have the same structure as the baseline T-LSTMs, with 1024 hidden units in each layer, and the output size is reduced to 512 using a projection.

All the setups in Table 2 outperform the baseline 4-layer T-LSTM. With a 3-layer T-LSTM on top of it, the F-LSTM and TF-LSTM perform almost the same. However, with a 4-layer T-LSTM on top it, the TF-LSTM is much better than the F-LSTM, and gets 14.83% WER – a 3.4% relative WER reduction from the baseline 4-layer T-LSTM. The joint time-frequency modeling provides a better feature for the upper layer T-LSTMs to consume. As shown in Table 1, simply increasing number of layers from 4 to 5 doesn't give any gain.

Table 2: Comparison of F-LSTM or TF-LSTM

Model	WER (%)	Number of parameters
F-LSTM + 3-layer T-LSTM	15.11	17.0 M
F-LSTM + 4-layer T-LSTM	15.23	21.6 M
TF-LSTM + 3-layer T-LSTM	15.09	17.0 M
TF-LSTM + 4-layer T-LSTM	14.83	21.6 M

We further investigate the performance of stacked F-LSTM and TF-LSTM in Table 3. To have the same number of layers as the "TF-LSTM + 4-layer T-LSTM" setup in Table 2, we tried to use either 2-layer F-LSTM or 2-layer TF-LSTM, followed by 3-layer T-LSTM. Again, the setup using TF-LSTM outperformed the setup with F-LSTM. However, none outperformed the "TF-LSTM + 4-layer T-LSTM" setup. Note that it only introduces 0.1M additional parameters from the "TF-LSTM + 3-layer T-LSTM" setup in Table 2 to the "2-layer F-LSTM + 3-layer T-LSTM" setup in Table 3 and this brings very slight WER improvement. This is because the TF-LSTM itself has very small number of parameter because the cell size is only 24. In the future, we can have 2-layer TF-LSTM followed by 4-layer T-LSTM to get some further gains.

Table 3: The stacking of F-LSTM and TF-LSTM

Model	WER (%)	Number of parameters
2-layer F-LSTM + 3-layer T-LSTM	15.29	17.1 M
2-layer TF-LSTM + 3-layer T-LSTM	15.00	17.1 M

In a final set of experiments, we evaluated the invariance properties of the TF-LSTM model by testing the models trained with Windows phone data on the Aurora 4 [30] test sets. Two clean evaluation sets (A and C) are recorded with the Sennheiser microphone and the secondary microphone, respectively. The remaining two groups (B and D), are recorded with two types of microphone respectively, and 6 types of noise are added with randomly chosen SNRs between 5 and 15 dB for each of the microphone types. Therefore, these test sets have totally mismatched acoustic environments from the Windows phone training set. We used the baseline 4-layer T-LSTM model in Table 1 and the TF-LSTM model in Table 2 for the evaluation. The language model is a bigram provided by Aurora 4. As shown in Table 4, the TF-LSTM performs much better than the T-LSTM in all test conditions, and reduced the average WER from 17.46% to 15.01%, a 14.2% relative WER reduction. This confirms the robustness [31] of the joint time-frequency analysis of the TF-LSTM.

Table 4: The WER comparison of T-LSTM and TF-LSTM models on the mismatched Aurora 4 test sets. Models are trained with Windows phone short message dictation data.

Model	A	B	C	D	Avg.
4-layer T-LSTM	6.37	14.25	9.14	23.90	17.46
TF-LSTM + 4-layer T-LSTM	5.45	12.07	8.07	20.69	15.01

6. CONCLUSIONS

In this paper, we have presented a two-dimensional TF-LSTM architecture that scans both the time and frequency axes to model the evolving patterns of the spectrogram. The TF-LSTM uses a LSTM to perform a joint time-frequency recurrence that summarizes spectro-temporal patterns. The summarized patterns are then fed into upper level T-LSTMs. The proposed TF-LSTM obtained a 3.4% relative WER reduction over the traditional T-LSTM on a 375-hour short message dictation task. We further investigated the effectiveness of stacking multiple TF-LSTM layers, and found that the additional accuracy gain is marginal. This indicates that a one layer TF-LSTM is good enough to extract the patterns relevant to speech recognition. When evaluated with a totally mismatched Aurora 4 test set, the TF-LSTM demonstrates much better resistance to the distortion, giving 14.2% relative WER reduction over a T-LSTM.

REFERENCES

[1] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. Interspeech*, pp. 437-440, 2011.

[2] N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke, "An application of pretrained deep neural networks to large vocabulary conversational speech recognition," in *Proc. Interspeech*, 2012.

[3] T. N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, A.-R. Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *Proc. ASRU*, pp. 30-35, 2011.

[4] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Large vocabulary continuous speech recognition with context-dependent DBN-HMMs," in *Proc. ICASSP*, pp. 4688-4691, 2011.

[5] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio Speech and Language Process.*, vol. 20, no. 1, pp. 14-22, Jan. 2012.

[6] L. Deng, J. Li, J.-T. Huang et. al. "Recent advances in deep learning for speech research at Microsoft," in *Proc. ICASSP*, 2013.

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

[8] A. Gers, J. Schmidhuber, and F. Cummins. "Learning to forget: Continual prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451-2471, 2000.

[9] A. Graves, A. Mohamed, G. Hinton. "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*, 2013.

[10] A. Graves, N. Jaitly, A. Mohamed. "Hybrid speech recognition with deep bidirectional LSTM," in *Proc. ASRU*, 2013.

[11] H. Sak, A. Senior, F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. Interspeech*, 2014.

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

[13] X. Li and X. Wu, "Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition," in *Proc. ICASSP*, 2015.

[14] T. N. Sainath, O. Vinyals, A. Senior and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. ICASSP*, 2015.

[15] A. Mohamed, G. Hinton, and G. Penn, "Understanding how deep belief networks perform acoustic modelling," in *Proc. ICASSP*, pp. 4273–4276, 2012.

[16] J. Li, A. Mohamed, G. Zweig, and Yifan Gong, "LSTM time and frequency recurrence for automatic speech recognition," in *Proc. ASRU*, 2015.

[17] A. Graves, S. Fernández, J. Schmidhuber, "Multi-dimensional recurrent neural networks," in *ICANN*, pp. 549-558, 2007.

[18] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," *Advances in Neural Information Processing Systems*, pp. 545-552, 2009.

[19] W. Byeon, T. M. Breuel, F. Raue, and M. Liwicki, "Scene labeling with LSTM recurrent neural networks," In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3547-3555, 2015.

[20] T. N. Sainath, A. Mohamed, B. Kingsbury and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proc. ICASSP*, 2013.

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

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

[23] A. Graves, "Practical variational inference for neural networks." In *Advances in Neural Information Processing Systems*, pp. 2348-2356, 2011.

[24] 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," U.S. Dept. of Commerce, NIST, Gaithersburg, MD, February 1993.

[25] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning. ACM*, pp. 369–376, 2006.

[26] J. Li, D. Yu, J. T. Huang, and Y. Gong. "Improving wideband speech recognition using mixed-bandwidth training data in CD-DNN-HMM," in *Proc. IEEE Spoken Language Technology Workshop*, pp. 131–136, 2012.

[27] D. Yu, A. Eversole, M. Seltzer, et. al., "An introduction to computational networks and the computational network toolkit," *Microsoft Technical Report MSR-TR-2014-112*, 2014.

[28] H. Jaeger, "Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the echo state network approach," *GMD Report 159*, GMD—German National Research Institute for Computer Science, 2002.

[29] J.-T. Huang, J. Li, and Y. Gong, An analysis of convolutional neural networks for speech recognition, in *Proc. ICASSP*, 2015.

[30] N. Parihar and J. Picone, "Aurora working group: DSR front end LVCSR evaluation AU/384/02," *Tech. Rep.*, Institute for Signal and Information Processing, Mississippi State Univ., 2002.

[31] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, *Robust Automatic Speech Recognition: A Bridge to Practical Applications*, Elsevier Press, 2015.