TIME AND FREQUENCY DOMAIN LONG SHORT-TERM MEMORY FOR NOISE ROBUST PITCH TRACKING

Yuzhou Liu[1] *and DeLiang Wang*[1,2]

[1]Department of Computer Science and Engineering, The Ohio State University, USA
[2]Center for Cognitive and Brain Sciences, The Ohio State University, USA
{liuyuz,dwang}@cse.ohio-state.edu

ABSTRACT

Pitch tracking in noisy speech is a challenging task as temporal and spectral patterns of the speech signal are both corrupted. This paper proposes long short-term memory (LSTM) based methods for pitch probability estimation. Two architectures are investigated. The first one is conventional LSTM that utilizes recurrent connections to model pitch dynamics. The second one is two-level time-frequency LSTM, with the first level scanning frequency bands and the second level connecting the first level through time. The Viterbi algorithm then takes the probabilistic output from LSTM to generate continuous pitch contours. Experiments show that both proposed models outperform a deep neural network (DNN) based model in most conditions. Time-frequency LSTM achieves the best performance at negative SNRs.

Index Terms— Pitch detection, long short-term memory, time and frequency modeling

1. INTRODUCTION

Pitch of human speech refers to the fundamental frequency of vocal fold vibrations. A reliable estimate of pitch is useful for various applications, including automatic speech recognition [3], speech separation [23] and emotion recognition [15].

Although many algorithms have been proposed for pitch tracking [2][20], they do not produce consistent results when speech is severely interfered by noise. The difficulty of pitch tracking in noise stems from the fact that both temporal continuities and harmonic patterns are corrupted. Recently, many studies try to address the noise-robustness issue for pitch tracking, and most of them consist of two stages. In the fist stage, pitch candidates or pitch probabilities are estimated for each time frame of speech using temporal, spectral, or spectrotemporal domain information [23]. Temporal domain methods analyze the periodic cue of speech; e. g., YIN [4] proposed a number of modifications to the autocorrelation method to improve pitch estimation accuracy. Spectral domain methods are based on harmonic modeling. For instance, PEFAC [7] used non-linear amplitude compression and a comb-filter to suppress noise in the spectrogram, and selected pitch candidates from harmonic peaks. Han and Wang [9] fed spectral features to a deep neural network (DNN) and a recurrent neural network (RNN) to predict frame-level probabilities of pitch states. Spectrotemporal methods first decompose the signal into a series of sub-bands, and then perform temporal analysis on each frequency channel. For example, Wang and Hansen [22] decomposed speech into overlapped time-frequency segments, and derived pitch candidates and likelihood scores for each segment. After the estimation of pitch candidates and probabilities, the second stage integrates local pitch clues into continuous pitch tracks using dynamic programming [7] or hidden Markov models (HMMs) [25].

Given that speech has long-term dependency in the time domain, it is natural to exploit temporal dynamics for pitch tracking. However, most pitch tracking algorithms only analyze speech signals within short-time windows in the first stage, resulting in inaccurate pitch estimates at noise-dominant frames. To address this problem, Han and Wang [9] proposed to use a standard RNN to estimate pitch probabilities over time. Such RNNs are designed to model sequential data, but they suffer from the vanishing and exploding gradient problem [1], and can not propagate information over a long span. Long short-term memory (LSTM) RNNs [11] use gates to stabilize gradient propagation, and are shown to be good at modeling long-term dependencies in many applications such as automatic speech recognition [8] [17] and machine translation [19].

In this study, we extend the RNN based pitch tracking framework, and propose to use LSTM to model the posterior probability that a frequency bin (pitch state) is pitched given frame-level log-spectrogram features. Another important characteristic of voiced speech is that its harmonics are evenly spaced in frequency. When some frequency bands are contaminated by noise, we can still estimate the fundamental frequency from other reliable bands. To leverage this observation, we further apply a two-level LSTM structure. The first level is frequency domain LSTM (F-LSTM) that scans seg-

This research was supported in part by an AFOSR grant (FA9550-12-1-0130) and the Ohio Supercomputer Center.

ments of log-spectrogram along the frequency axis to detect harmonic patterns. The second level is time domain LSTM (T-LSTM), which takes the output of F-LSTM as input, and models pitch probabilities through time. The overall structure is denoted by time-frequency LSTM (TF-LSTM) in this study. Recently, a similar TF-LSTM network has been shown to outperform conventional LSTM in an automatic speech recognition task [17]. Once all frame-level pitch probabilities are derived, we use the Viterbi algorithm [5] to generate continuous pitch contours.

The rest of the paper is organized as follows. The proposed system is described in the next section. In Section 3, we present experimental results and comparisons. A conclusion is given in Section 4.

2. SYSTEM DESCRIPTION

The proposed pitch tracking algorithm consists of two stages: pitch probability estimation and Viterbi decoding.

In the first stage, we extract the log-spectrogram feature $\mathbf{y}_t$ from a noisy utterance sampled at 16 kHz, where t denotes the frame index. Neural networks then use $\mathbf{y}_t$ as input to estimate the posterior probability of pitch states $p(x_t|\mathbf{y}_t)$, where x_t denotes the pitch state at frame t. We quantize the frequency range 60 to 404 Hz into 67 bins $(s^1, s^2, ..., s^{67})$ using 24 bins per octave in a logarithmic scale [9]. Each s^i corresponds to a state in x_t. In addition, a non-pitched state s^0 is incorporated into x_t to represent unvoiced speech or silence. $p(x_t = s^i|\mathbf{y}_t)$ equals 1 if the groundtruth pitch is in the frequency bin of s^i, and 0 otherwise. We introduce a DNN as a baseline model in Section 2.2 . LSTM and TF-LSTM are described in Section 2.3 and Section 2.4. In the second stage, we use the Viterbi algorithm to connect frame-level probabilities and track pitch through time.

2.1. Feature extraction

The feature used in study is based on log-spectrogram. To get this feature, the signal is first divided into 32 ms frames with a 10 ms frame shift. We then apply a Hamming window to each frame and derive the spectrogram using 1024-point FFT. Lastly, we compute the logarithm of the amplitude spectrum, and pick bins 2 to 129 (corresponding to a frequency range up to 2000 Hz) as the 128-dimensional feature for each frame. We do not pick all bins in the spectrogram as the energy of high frequency harmonics is relatively low, and the frequency range up to 2000 Hz covers at least 5 harmonics of human speech, enough for continuous pitch tracking.

Since neighboring frames contain useful information, we splice a 15-frame window of features as the DNN's input. For LSTM and TF-LSTM, although the history of input is stored in their memory cells, it is still helpful to apply a context window so that they can receive richer input information at each time step. Taking the model size and computational cost into consideration, we splice a 7-frame window for the input of LSTM and TF-LSTM.

2.2. DNN based pitch probability estimation

We first utilize a DNN as a baseline model to estimate the posterior probability of pitch states when the frame-level feature vector is given, i. e., $p(x_t|\mathbf{y}_t)$. The DNN has four hidden layers, each with 1600 rectified linear units [6]. The output layer contains 68 soft-max units, corresponding to the number of pitch states. The cross-entropy cost function, mini-batch gradient descent, Adam optimization algorithm [14] and dropout regularization [10] are used during training. The initial learning rate is 0.001, and a learning rate decay of 0.7 each epoch is used. The training stops after 30 epochs.

2.3. LSTM based pitch probability estimation

To better encode the temporal dependency of human speech, we use LSTM for pitch probability estimation in this subsection. LSTM is composed of a series of recurrently connected memory blocks [11]. Each memory block has a memory cell which stores the temporal state of the network, an input gate which controls the amount of input activation added to the memory cell, a forget gate which adaptively resets the memory cell and an output gate which controls the amount of information passed from the memory cell to the output. In this work, we followed the LSTM architecture in [26]:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad (2)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (3)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \quad (4)$$

$$h_t = o_t \odot \tanh(c_t) \quad (5)$$

where i_t, f_t, c_t and o_t denote the input gate, forget gate, memory cell and output gate. x_t and h_t denote the input and output of the memory block. W terms and b terms denote different weight matrices and biases. σ is the logistic sigmoid function. $\odot$ represents element-wise multiplication.

Our LSTM has four hidden layers, each with 512 hidden units. The output layer is a soft-max layer with 68 units. The number of parameters in LSTM is close to that in the baseline DNN. To train LSTM, we use a backpropagation through time (BPTT) step of 100. The learning rate decay is set to 0.45 per epoch. Other training recipes follow the baseline DNN.

2.4. TF-LSTM based pitch probability estimation

In this subsection, we introduce time-frequency LSTM (TF-LSTM) which models temporal and spectral dynamics of speech simultaneously. A diagram of proposed TF-LSTM is shown in Fig. 1. The intention of this architecture is to first use F-LSTM to scan different frequency bands of the

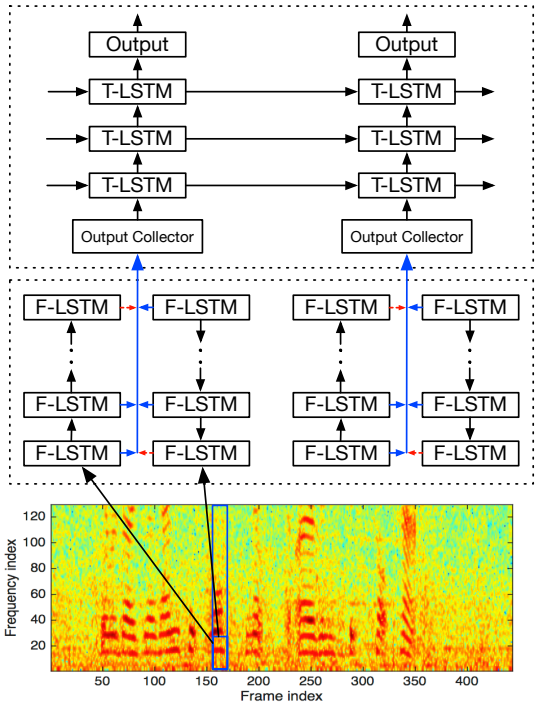

Fig. 1: Diagram of TF-LSTM.

log-spectrogram, where useful pitch information can be extracted from relatively clean bands, and will propagate along the frequency axis to further affect subsequent noisy bands. The output of F-LSTM is then collected and fed to T-LSTM to track pitch through time. A similar network has been applied to automatic speech recognition tasks and shown to outperform DNN based and conventional LSTM based neural networks [17].

To implement TF-LSTM, we first divide the 128×7-dimensional feature $\mathbf{y}_t$ along the frequency axis to get overlapped frequency segments. Each frequency segment contains 24×7 units, and has 16×7 overlapped units with each neighboring frequency segment. In other words, the stride of frequency segments is 8. F-LSTM, which is one layer bidirectional LSTM with 256 units per direction, takes all frequency segments in a frame as input. Therefore F-LSTM is unrolled on the frequency axis $(128 - 24)/8 + 1 = 14$ times at each frame. All parameters in F-LSTM are chosen from a development set. The output of F-LSTM is then fed into T-LSTM. Because we have 256×2×14 output units from F-LSTM at each frame, it's inefficient to directly use them in T-LSTM. We thus propose two methods to solve the problem. The first method only keeps the last outputs in the F-LSTM sequence (red dashed arrows in Fig. 1) for T-LSTM, denoted by TF-LSTM1. Here the last outputs are treated as an embedding of the log-spectrogram. The second method concatenates all output units in F-LSTM and uses a 512-unit linear transformation layer to reduce its dimensionality, denoted by TF-LSTM2. We will compare these two methods in Section 3. T-LSTM has three 512-unit hidden layers and a 68-unit soft-max output layer. Other details and training recipes follow conventional LSTM.

2.5. Viterbi decoding

After the estimation of $p(x_t|\mathbf{y}_t)$, we use the Viterbi algorithm [5] to connect all probabilities along time. The hidden state in the Viterbi algorithm corresponds to x_t, and the observation corresponds to $\mathbf{y}_t$. We use the training data to compute prior probabilities $p(x_t = s_i)$ and transition matrices. Emission probabilities can be computed using the estimated posterior probabilities divided by the prior $p(x_t = s_i)$. The Viterbi algorithm generates a sequence of most likely pitch states, which is then converted to mean frequencies of corresponding frequency bins. In the end, a three-frame moving average window is applied to smooth pitch estimates.

3. EVALUATION RESULTS AND COMPARISONS

We use the Mocha-TIMIT database [24] for experimental comparisons. This database consists of 460 utterances from both a male and a female speaker. Because the male speaker is less challenging for pitch tracking tasks, we use the female speaker in the following experiments for speaker-dependent learning. The training set is created by mixing 400 utterances from the female speaker with 10,000 noises from a sound-effect library (available at http://www.sound-ideas.com). Each clean utterance is mixed 100 times with a random segment of a random noise at a random SNR from -5 to 5 dB. The total duration of the training set is 44 hours. The test set includes 20 unseen utterances from the Mocha-TIMIT female speaker. Six noises, i. e., babble noise [12], factory noise [21], speech shape noise (SSN), cocktail-party noise [13], crowd playground noise [13] and crowd music noise [13], are used for test, and all of them are unseen during training. Each test utterance is mixed with the six test noises at -10, -5, 0, 5 and 10 dB, resulting in a total of 600 test mixtures. The groundtruth pitch is derived by applying the RAPT [20] algorithm on laryngograph signals. We manually remove erroneous pitch in unvoiced regions to further improve the quality of the groundtruth pitch.

We use two metrics to evaluate pitch estimates: detection rate (DR) and voicing decision error (VDE) [16]. DR indicates the percentage of correctly estimated voiced frames. VDE computes the percentage of frames that are misclassified in terms of pitched and unpitched decision:

$$\mathbf{DR} = \frac{N_{0.05}}{N_p}, \quad \mathbf{VDE} = \frac{N_{n\to p} + N_{p\to n}}{N} \tag{6}$$

Here $N_{0.05}$ is the number of frames whose estimated pitch deviates less than 5% from the groundtruth pitch. $N_{n\to p}$ and $N_{p\to n}$ are the number of frames misclassified as pitched and unpitched respectively. N_p is the number of pitched frames,

Table 1: Comparison of approaches in terms of DR.

SNR (dB)	-10	-5	0	5	10
PEFAC	0.373	0.555	0.657	0.696	0.714
Han and Wang DNN	0.434	0.635	0.728	0.755	0.756
Han and Wang RNN	0.406	0.633	0.727	0.755	0.763
Proposed DNN	0.664	0.861	0.934	0.953	0.958
Proposed LSTM	0.706	0.876	**0.938**	**0.956**	**0.959**
Proposed TF-LSTM1	**0.714**	**0.880**	0.937	0.954	0.958
Proposed TF-LSTM2	0.711	0.878	**0.938**	0.954	0.957

Table 2: Comparison of approaches in terms of VDE.

SNR (dB)	-10	-5	0	5	10
PEFAC	0.337	0.262	0.192	0.142	0.112
Han and Wang DNN	0.295	0.221	0.149	0.103	0.095
Han and Wang RNN	0.301	0.226	0.165	0.120	0.108
Proposed DNN	0.247	0.131	0.062	0.047	**0.041**
Proposed LSTM	0.228	0.119	0.063	0.048	0.043
Proposed TF-LSTM1	0.221	0.116	**0.059**	**0.046**	0.042
Proposed TF-LSTM2	**0.204**	**0.112**	0.061	0.047	0.042

and N is the total number of frames. Higher DR and lower VDE indicate better pitch estimates.

We compare our methods with two state of the art pitch tracking algorithms: PEFAC [7] and Han and Wang [9]. PEFAC is a representative unsupervised approach that performs relatively well in low SNR conditions. Han and Wang's approach used the same DNN/RNN-HMM framework as ours, and was trained on a speaker-independent dataset. Two of Han and Wang's training noises are seen in our test set.

Table 1 and Table 2 list the DR and VDE of different approaches, where all values are averaged across six noise types. As shown in the tables, all supervised learning approaches produce better results than PEFAC across all SNRs. A standard RNN was used by Han and Wang to model temporal dynamics, but it does not outperform their DNN based approach in most cases, which is due to the fact that such RNNs are more difficult to train and can not model long-term effects. By virtue of the large training set, speaker-dependent training [18] and better training recipes, our proposed methods show significant improvement over Han and Wang's approach. When the SNR is non-negative, all proposed methods generate exceptionally accurate pitch estimates while making few voicing decision mistakes. When it comes to negative SNRs, the LSTM based method produces clearly higher DRs and lower VDEs than the DNN based method, indicating that the capacity of sequence modeling makes LSTM better at processing very noisy speech. TF-LSTM1 and TF-LSTM2 both outperform LSTM at negative SNRs. They yield comparable results in terms of DR, and TF-LSTM2 achieves a VDE of 0.204 at -10 dB, significantly lower than all other approaches. Such improvement is contributed to the frequency scanning module in TF-LSTM, which makes the model better at distinguishing unpitched signals from voiced speech.

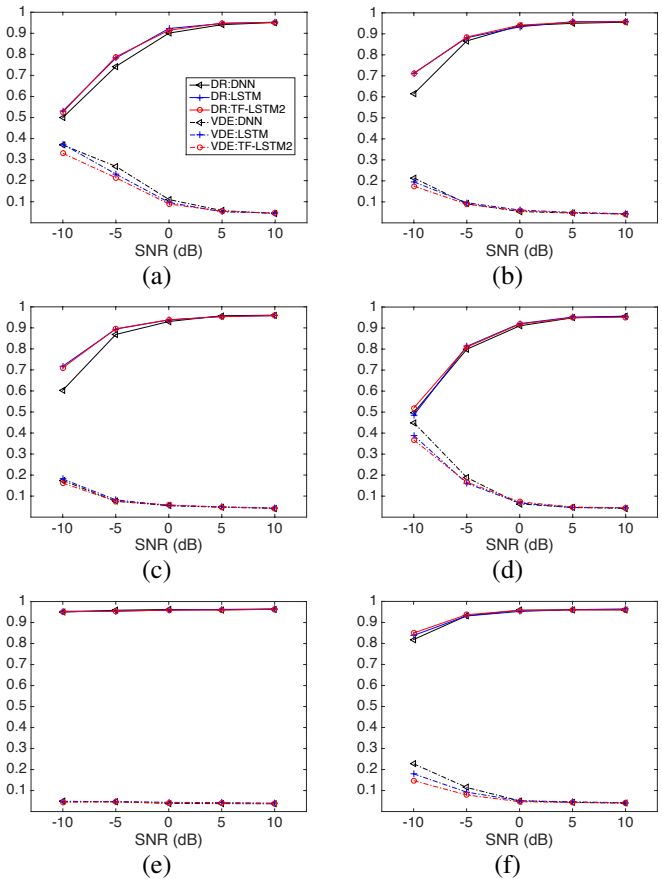

Fig. 2: DR and VDE for (a) babble noise, (b) factory noise, (c) SSN, (d) cocktail-party noise, (e) crowd playground noise, (f) crowd music noise.

Fig. 2 compares the performance of the proposed DNN, LSTM and TF-LSTM2 in different noises, which is pretty consistent with the results in Table 1 and 2. The major improvement of LSTM and TF-LSTM2 comes from negative SNRs. LSTM outperforms the DNN in most negative SNR conditions. TF-LSTM2 generates similar DR as LSTM, but consistently lower VDE in all cases.

4. CONCLUSION

In this study, we have introduced LSTM for robust pitch tracking in noisy speech. Both conventional LSTM and two-level TF-LSTM are utilized to estimate probabilistic pitch states. TF-LSTM first uses F-LSTM to scan harmonic patterns, and then uses T-LSTM to connect frequency-domain activations. Due to the capacity of sequence modeling, both LSTM based models outperform a DNN based model. TF-LSTM further reduces the VDE of conventional LSTM by 10% at -10 dB. In the future, we will incorporate sub-band features into TF-LSTM. We will also perform speaker-independent training to evaluate how well TF-LSTM generalizes to unseen speakers.

5. REFERENCES

[1] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, pp. 157–166, 1994.

[2] P. Boersma and D. Weenink, "Praat, a system for doing phonetics by computer," in *Glot Int.*, vol. 5, 2001, pp. 341–345.

[3] C. Chen, R. Gopinath, M. Monkowski, M. Picheny, and K. Shen, "New methods in continuous mandarin speech recognition," in *Proceedings of Eurospeech*, 1997, pp. 1543–1546.

[4] A. D. Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *J. Acoust. Soc. Amer.*, vol. 111, pp. 1917–1930, 2002.

[5] G. D. Forney Jr, "The viterbi algorithm," in *Proc. of the IEEE*, vol. 61, 1973, pp. 268–278.

[6] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *AISTATS*, 2011, pp. 315–323.

[7] S. Gonzalez and M. Brookes, "PEFAC-A pitch estimation algorithm robust to high levels of noise," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 22, pp. 518–530, 2014.

[8] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*, 2013, pp. 6645–6649.

[9] K. Han and D. L. Wang, "Neural network based pitch tracking in very noisy speech," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, pp. 2158–2168, 2014.

[10] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–1780, 1997.

[12] G. Hu. 100 nonspeech sounds, 2006. [Online]. Available: http://www.cse. ohio-state.edu/pnl/corpus/HuCorpus.html

[13] ——, "Monaural speech organization and segregation," Ph.D. dissertation, The Ohio State University, Columbus, OH, 2006.

[14] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[15] S. G. Koolagudi and K. S. Rao, "Emotion recognition from speech: a review," *International Journal of Speech Technology*, vol. 15, pp. 99–117, 2012.

[16] B. S. Lee and D. P. W. Ellis, "Noise robust pitch tracking by subband autocorrelation classification," in *Proceedings of Interspeech*, 2012.

[17] J. Li, A. Mohamed, G. Zweig, and Y. Gong, "LSTM time and frequency recurrence for automatic speech recognition," in *Proc. ASRU*, 2015, pp. 187–191.

[18] Y. Liu and D. L. Wang, "Robust pitch tracking in noisy speech using speaker-dependent deep neural networks," in *Proc. ICASSP*, 2016, pp. 5255–5259.

[19] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. NIPS*, 2014, pp. 3104–3112.

[20] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," *Speech Coding Synth.*, pp. 495–518, 1995.

[21] A. Varga and H. J. M. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Communication*, vol. 12, pp. 247–251, 1993.

[22] D. Wang and J. H. L. Hansen, "F0 estimation for noisy speech by exploring temporal harmonic structures in local time frequency spectrum segment," in *Proc. ICASSP*, 2016, pp. 6510–6514.

[23] D. L. Wang and G. Brown, Eds., *Computational Auditory Scene Analysis: Principles, Algorithms and Applications*. Wiley-IEEE Press, 2006.

[24] A. Wrench, "A multichannel/multispeaker articulatory database for continuous speech recognition research," *Phonus*, vol. 5, pp. 3–17, 2000.

[25] M. Wu, D. L. Wang, and G. Brown, "A multipitch tracking algorithm for noisy speech," *IEEE Trans. Speech Audio Process.*, vol. 11, pp. 229–241, 2003.

[26] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.