Interspeech 2018
2-6 September 2018, Hyderabad

Hierarchical Recurrent Neural Networks for Acoustic Modeling

Jinhwan Park, Iksoo Choi, Yoonho Boo, Wonyong Sung

Department of Electrical and Computer Engineering
Seoul National University

bnoo@snu.ac.kr, akacis@snu.ac.kr, dnsgh337@snu.ac.kr, wysung@snu.ac.kr

Abstract

Recurrent neural network (RNN)-based acoustic models are widely used in speech recognition, and end-to-end training with CTC (connectionist temporal classification) shows good performance. In order to improve the ability to keep temporarily distant information, we employ hierarchical recurrent neural networks (HRNNs) to the acoustic modeling in speech recognition. HRNN consists of multiple RNN layers that operate on different time-scales, and the frequency of operation at each layer is controlled by learned gates from training data. We employ gate activation regularization techniques to control the operation of the hierarchical layers. When tested with the WSJ eval92, our best model obtained the word error rate of 5.19% with beam search decoding using RNN based character-level language models. Compared to an LSTM based acoustic model with a similar parameter size, we achieved a relative word error rate improvement of 10.5%. Even though this model employs uni-directional RNN models, it showed the performance improvements over the previous bi-directional RNN based acoustic models.

Index Terms: speech recognition, recurrent neural network, acoustic modeling

1. Introduction

Recurrent neural network (RNN) based acoustic models are widely used in speech recognition systems, and they show very good performance especially in end-to-end models [1, 2, 3, 4, 5]. Connectionist temporal classification (CTC) [6] is a most widely used method to train the RNN for acoustic modeling, which generates text sequences directly from the input speech. More recently, other types of end-to-end structures are attracting attention, such as the encoder-decoder [7, 8] and RNN-transducer [9, 10].

For acoustic modeling, RNN runs typically with a 10 ms frame rate, which means that RNN acoustic models need to compute a lot of time steps compared to other tasks, such as language modeling. Learning long-term dependency in acoustic modeling can be more difficult because vanishing gradient problem is very severe when RNN is unrolled in many time steps [11]. Even if a gated structure like long-short term memory (LSTM) [12] is used, it is not easy to propagate information over 100 time steps, which corresponds to 1 second of input speech if 10 ms frame is employed.

There have been many studies to enable RNNs to maintain long-term context in their states. One of the approaches is employing hierarchical recurrent neural networks (HRNN). HRNN consists of multiple modules with different levels of abstraction. The module with higher-level abstraction focuses on the long-term context of the input by skipping update of the states for several time steps. It can improve the recognition performance because skipping state updates in RNN alleviates the vanishing gradient problem and the information can be propagated to the future states more easily.

The structure of HRNN has been studied actively in recent researches [13, 14, 15]. Clockwork RNN [13] has a fixed-rate operation frequency for each module. HRNN proposed in [14] use explicit boundary information such as word-boundary or end-of-sentence in the language model task. Hierarchical multiscale RNN (HM-RNN) [15] use the gates to control the operations of the modules, where the gates are trained from the data without explicit information. These HRNNs show a promising performance on sequential tasks such as language modeling and handwriting sequence generation while reducing the number of computations.

In this paper, we apply an HRNN model to acoustic modeling task by employing a trained hierarchical structure [15]. Learning the hierarchical structure from data is especially effective for acoustic modeling because it is not easy to obtain explicit boundary information from acoustic data in inference time. We expect high-level layers learn long-term relationships such as the correlation between phonemes or graphemes.

We trained HRNNs with CTC loss [6], which is a widely used training method in acoustic modeling. We compare the performance of HRNNs with various types of gates. We also introduce a regularization term to encourage the gate to learn meaningful boundary for speech recognition and lower the computational complexity. We expect that HRNN architecture can be applied to other types of end-to-end RNN speech recognition models, such as the encoder-decoder model.

This paper is organized as follows. We review the hierarchical recurrent neural networks in Section 2. Section 3 describes the HRNN architectures for acoustic modeling. The word and character error rates of HRNN are evaluated and analyzed in Section 4. Concluding remarks are presented in Section 5.

2. Hierarchical Recurrent Neural Networks

Many RNN models including LSTM [12] or gated recurrent unit (GRU) [16] can be extended to HRNN architecture. General recurrent layers can be expressed with the function f and h as follows:

$$\begin{aligned}\mathbf{s}_t &= f(\mathbf{x}_t, \mathbf{s}_{t-1}),\\ \mathbf{y}_t &= h(\mathbf{s}_t),\end{aligned} \tag{1}$$

where $\mathbf{x}_t$, $\mathbf{y}_t$, $\mathbf{s}_t$ are the input, output, and state of RNN at the time step t. The above recurrent layer can be converted to *gated recurrent layer* with the gate $g_t \in \{0, 1\}$,

$$\begin{aligned}\mathbf{s}_t &= g_t f(\mathbf{x}_t, \mathbf{s}_{t-1}) + (1 - g_t)\mathbf{s}_{t-1},\\ \mathbf{y}_t &= h(\mathbf{s}_t).\end{aligned} \tag{2}$$

When $g_t = 1$, this is identical to the original recurrent layer. When $g_t = 0$, the state and the output of the recurrent layer

10.21437/Interspeech.2018-1797

keep the same values of the previous time step ($s_t = s_{t-1}, y_t = y_{t-1}$).

There are several methods to compute the gate value g_t. One approach is to use a periodical gate with a constant period T as [13]:

$$g_t = \begin{cases} 1 & \text{if } t \bmod T = 0, \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

This approach can be beneficial for implementation because it has a regular computational pattern, while it is not suitable for variable length modeling, such as speech recognition. Another approach is rule-based gating, where g_t is 1 only if the input satisfies the certain condition. This is an efficient method for character-level language modeling because the word boundary information can be easily obtained from the input feature.

The approach considered most promising is using trained parameters to generate g_t from the input $\mathbf{x}_t$ and the state $\mathbf{s}_{t-1}$ [15]:

$$g_t = round(\sigma(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{s}_{t-1} + b)), \quad (4)$$

with the weight matrices $\mathbf{W}$, $\mathbf{U}$, bias b, and activation function $\sigma : \mathbb{R} \mapsto [0, 1]$. The sigmoid function is typically used for the activation function σ. For the rounding function, stochastic rounding with Bernoulli sampling is used in training, while deterministic rounding with a threshold of 0.5 is used for inference. When this gate function is used, the loss is not differentiable due to the discontinuity of the rounding function. Straight-through estimator [17] can be used to train this model by approximating the $round$ function to the identity function in the backward path:

$$\frac{\partial round(x)}{\partial x} = 1. \quad (5)$$

This is a biased estimator with low variance. The slope annealing trick [15] can be used to reduce the bias of the straight-through estimator.

There can be also a number of ways to connect layers with different scales in HRNN. The hidden states within a layer can be divided into different time scales [13]. Otherwise, each layer in the RNN can be assigned to a different scale. In this case, the output of each layer can be summed or concatenated and passed to the following layer to consider both short-term and long-term information. In this paper, HRNN represents an RNN with one or more gated recurrent layers.

3. Acoustic Modeling with HRNN

3.1. Additional loss for gate training

When we trained HRNN with CTC loss from scratch, we found that the gate was often converged to an output value of 1. This is not the desired behavior because we want to capture the long-term information by skipping state updates. Thus, we use regularization term to control the gate value. One approach is to use L2 or L1 regularization on the gate to encourage fewer state updates [18]:

$$L_{train} = L_{CTC} + \lambda \sum ||g_t||^2, \quad (6)$$

where λ is a hyperparameter and L_{CTC} is CTC loss. However, it was not easy to determine the value of λ in our preliminary experiments because a large λ makes the gate always have a value of zero. Instead, we add the L2 loss between the gate g_t and the phoneme boundary label $l_t \in \{0, 1\}$ to the original CTC loss L_{CTC}:

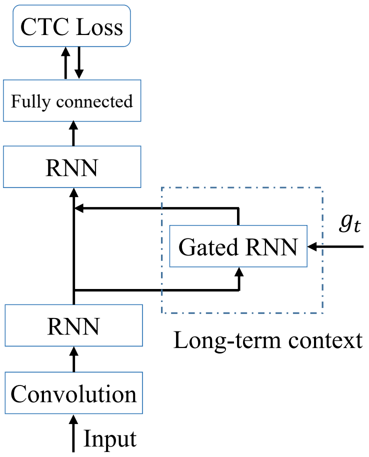


Figure 1: *Hierarchical RNN structure for acoustic modeling.*

$$L_{train} = L_{CTC} + \lambda \sum ||g_t - l_t||^2. \quad (7)$$

This is partly inspired by the research in computer vision area [19], which shows recognition performance improvement when the model is trained using multi-task loss with the object boundary information. Unfortunately, most speech datasets do not have the frame-wise phoneme annotation. We generate phoneme boundary labels from training data using Montreal Forced Alignment tool with the pretrained GMM/HMM acoustic model [20].

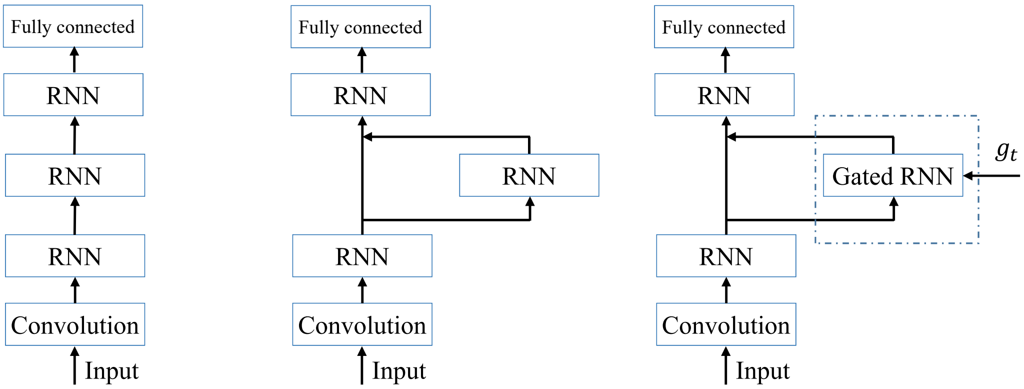
In order to enable HRNN to learn more flexible hierarchical structure besides the phoneme, we used the loss in Eq. 7 for the initial 10 epochs. Only CTC loss is used after then. By adopting this method, the gate was trained stably. We used the value of 0.01 for λ in our experiments.

3.2. HRNN architecture for acoustic modeling

We used CNN-HRNN models described in Figure 1. We used three layers of 512-dimensional LSTM for HRNN. LSTM can be easily replaced with other types of RNN models. The first LSTM layer is computed every time step as described in Eq. (1). The second LSTM layer is a gated LSTM that updates the states only if $g_t = 1$. The second layer is expected to learn the long-term context in this model. The output of the first and the second layers are concatenated and used as the input of the third LSTM layer. By concatenating two layers, both long-term and short-term contexts can be considered simultaneously. The output of the third layer is connected to the softmax layer.

We have trained the following models to compare and analyze HRNN structures:

- LSTM: three LSTM layers are stacked as the conventional RNN model.
- LSTM-skip: there is a skip connection between the first and the third layers. This is to keep the same topology as HRNN.
- HRNN with periodic gate: g_t is computed as in Eq. (3). $T = 4$ is used.



(a) (b) (c)

Figure 2: *The structure of the RNN models trained for comparison: (a) baseline LSTM, (b) LSTM-skip, (c) HRNN. In (c), g_t can be computed using Eq. (3) or Eq. (4).*

- HRNN with trained gate: g_t is computed as in Eq. (4). No additional gate loss is used.
- HRNN with λ = 0.01: g_t is computed as in Eq. (4). Additional gate loss in Eq. (7) is used with λ of 0.01 during the entire training process.
- HRNN with λ scheduling: g_t is computed as in Eq. (4). Additional gate loss in Eq. (7) is used with phoneme boundary information during the initial 10 epochs.

The baseline LSTM model has the fewest parameters because the concatenated input of the third LSTM layer increases the number of parameters in other models. Other models have almost the same number of parameters, with negligible difference because of the parameters for the gate. Figure 2 describes the structure of these models.

4. Experimental Results

4.1. Experimental setup

We used the WSJ SI-ALL training set, which includes all the speaker-independent training utterances in the WSJ corpus. This corresponds to 167 hours of speech data. A 40-dimensional log mel frequency filterbank feature is extracted from raw speech data. The feature vectors are extracted every 10 ms with 25 ms Hamming window. Two convolutional layers are used on the input side of the HRNN. The input of the convolution layer is two-dimensional feature maps with time and frequency axes. Three feature maps are generated by using the filterbank with its delta and double-delta. Each convolutional layer has a filter size of 3, and generates 32 output feature maps. The first convolutional layer down-samples the input frames with the stride of 2. Because of the down-sampling, the input feature of HRNN corresponds to 20 ms of speech data.

We applied batch normalization [21] to every output of convolutional layers, and variational dropout [22] to every LSTM layer for regularization. Adam optimizer [23] is used for training. We used initial learning rate of 3e-4, and the learning rate is reduced to half if the validation error is not lowered for consecutive 8 epochs. Gradient clipping with 4.0 is applied. When the gate is trained, linear slope annealing from 1 to 5 is applied for initial 40 epochs. Hyperparameters are kept the same for all the models.

Table 1: *WER and CER on WSJ eval92 test set with greedy decoding.*

Model	CER	WER
LSTM	**5.93%**	**20.16%**
LSTM-skip	6.79%	23.32%
HRNN with periodic gate	7.83%	27.66%
HRNN with trained gate	6.01%	20.50%
HRNN with $\lambda = 0.01$	6.92%	23.39%
HRNN with λ scheduling	6.09%	21.74%

Table 2: *WER and CER on WSJ eval92 test set when decoding is conducted with trigram word LM.*

Beam size	Model	CER	WER
128	LSTM	4.98%	13.69%
	LSTM-skip	5.82%	15.91%
	HRNN with periodic gate	6.38%	17.52%
	HRNN with trained gate	4.34%	11.95%
	HRNN with $\lambda = 0.01$	4.99%	13.55%
	HRNN with λ scheduling	**4.30%**	**11.93%**
512	LSTM	4.79%	12.85%
	LSTM-skip	5.38%	14.47%
	HRNN with periodic gate	5.89%	15.88%
	HRNN with trained gate	4.15%	11.16%
	HRNN with $\lambda = 0.01$	4.75%	12.55%
	HRNN with λ scheduling	**3.96%**	**10.69%**

4.2. Results on WSJ eval92

We reported the character error rate (CER) and word error rate (WER) with three different decoding algorithms: greedy decoding, beam search decoding with a trigram word-level LM, and beam search decoding with an RNN character-level LM. Greedy decoding does not use any external information except the RNN acoustic model. The trigram word LM was generated with the IRSTLM toolkit [24] included in the KALDI speech recognition tool. We used the WSJ non-verbalized punctuation text corpus that contains 37M words to build the LM. For the character-level language model (CLM), the HRNN based CLM is employed [14]. In HRNN CLM, the gate is activated when the input is word-boundary or the end-of-sentence. HRNN CLM is

Table 3: *WER and CER on WSJ eval92 test set when decoding is conducted with RNN CLM.*

Beam size	Model	CER	WER
128	LSTM	2.71%	6.56%
	LSTM-skip	2.96%	6.86%
	HRNN with periodic gate	2.96%	6.89%
	HRNN with trained gate	2.74%	6.43%
	HRNN with $\lambda = 0.01$	3.10%	7.21%
	HRNN with λ scheduling	**2.37%**	**5.79%**
512	LSTM	2.24%	5.80%
	LSTM-skip	2.78%	6.39%
	HRNN with periodic gate	2.66%	6.16%
	HRNN with trained gate	2.54%	6.01%
	HRNN with $\lambda = 0.01$	2.81%	6.58%
	HRNN with λ scheduling	**2.16%**	**5.19%**

Table 4: *The average value of the gates when tested on WSJ eval92.*

Model	Average Value
HRNN with periodic gate	0.25
HRNN with trained gate	1
HRNN with $\lambda = 0.01$	0.17
HRNN with λ scheduling	0.74

trained with the same text corpus as that used for trigram word LM training. The tree-structure based beam search decoding algorithm [25] is used for evaluation.

The WER and CER of the models are shown in Table 1-3. The models are evaluated on WSJ eval92 set, which is 42-minute speech data. Table 4 shows the average value of gates when HRNN models are tested on WSJ eval92 test set. The average value of the gates is related to the computational complexity of the models because the gated layer operates only when the gate value is 1. Using CLM improves the CER and WER significantly in all the models. When any language model is not applied, LSTM shows the lowest CER and WER, which is slightly better than HRNN trained without phoneme boundary information. However, when a language model is applied, HRNN with λ scheduling shows the best performance regardless of the language model.

Among the HRNN models, HRNN without any gate loss term shows the second-best performance in most cases. According to Table 4, the gate in the model always shows an output of 1. Its computational complexity is equal to that of LSTM-skip, but HRNN shows much lower word and character error rate. We consider this is because skipping state updates before the gate converges to 1 has a similar effect to Zoneout [26], which works as a regularizer.

When training HRNN with the gate loss in the whole training process, it shows worse recognition performance than HRNN without the gate loss regardless of the decoding methods. Training HRNN with predefined phoneme boundary-based hierarchy can degrade the recognition performance of the model because the units of recognition, characters, differ from phonemes. By training with the phoneme boundary labels only in the first few epochs, which corresponds to HRNN with λ scheduling, the HRNN can find improved hierarchical structure.

HRNN with the periodic gate shows the worst performance when greedy decoding and word-level language model are used. Interestingly, HRNN with the periodic gate shows better performance than HRNN with $\lambda = 0.01$ when character-level language model is used. It can be interpreted that HRNN with the periodic gate lacks the character-level modeling capacity. With a period of 4, the gated layer in HRNN considers 80 ms of speech. Because some of the characters are pronounced in less than 80 ms, they cannot be captured by the gated layer which is updated every 80 ms.

Table 5 shows the published WER on WSJ eval92 test set. When decoding is conducted with RNN CLM, we obtained the recognition accuracy comparable to that of a human [1] with only 6.4M parameters. Note that our HRNN model contains the least number of parameters. Moreover, all other models except HRNN are bidirectional RNN. Using a bidirectional RNN for speech recognition causes additional delay because decoding must be delayed until the end of speech. Considering this, HRNN model is suitable for real-time speech recognition.

Table 5: *Published WER on WSJ eval92 in the literature. Our best models are also shown.*

Model	LM	WER	Params
Miao *et al.* [3]	Trigram	7.34%	8.5M
Chorowski and Jaitly [4]	Trigram	6.7%	6.6M
Hannun *et al.* [5]	Bigram	14.1%	20.9M
Wu *et al.* [27]	Trigram	8.2%	6.5M
Deep Speech2 [1]	5-gram	3.60%	100M
Human [1]		5.03%	–
HRNN	Trigram	10.69%	6.4M
HRNN	RNN CLM	5.19%	6.4M

5. Concluding Remarks

In this paper, we used hierarchical recurrent neural networks (HRNNs) for acoustic modeling in speech recognition. We trained LSTM and HRNN with different gating algorithms. We analyzed and compared the performance of each type of HRNN. We also proposed an effective HRNN training algorithm for acoustic modeling. By applying the proposed algorithm, WER of 5.19% is obtained on WSJ eval92 with 6.4M parameters. In our future work, we plan to apply HRNNs for other end-to-end speech recognition models, such as the encoder-decoder [7] and the RNN-transducer [9].

6. Acknowledgements

This work was supported in part by the Brain Korea 21 Plus Project and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2015R1A2A1A10056051). This work was also supported by Samsung Advanced Institute of Technology through Neural Processing Research Center (NPRC) in Seoul National University.

7. References

[1] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *International Conference on Machine Learning*, 2016, pp. 173–182.

[2] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, K. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models,"

in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*. IEEE, 2018, pp. 4774–4778.

[3] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 167–174.

[4] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," *Proc. Interspeech 2017*, pp. 523–527, 2017.

[5] A. Y. Hannun, A. L. Maas, D. Jurafsky, and A. Y. Ng, "First-pass large vocabulary continuous speech recognition using bi-directional recurrent DNNs," *arXiv preprint arXiv:1408.2873*, 2014.

[6] A. Graves, S. Fernández, 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, 2006, pp. 369–376.

[7] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4960–4964.

[8] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4945–4949.

[9] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[10] E. Battenberg, J. Chen, R. Child, A. Coates, Y. G. Y. Li, H. Liu, S. Satheesh, A. Sriram, and Z. Zhu, "Exploring neural transducers for end-to-end speech recognition," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, Dec 2017, pp. 206–213.

[11] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310–1318.

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

[13] J. Koutnik, K. Greff, F. Gomez, and J. Schmidhuber, "A clockwork RNN," in *International Conference on Machine Learning*, 2014, pp. 1863–1871.

[14] K. Hwang and W. Sung, "Character-level language modeling with hierarchical recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5720–5724.

[15] J. Chung, S. Ahn, and Y. Bengio, "Hierarchical multiscale recurrent neural networks," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[16] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," *Syntax, Semantics and Structure in Statistical Translation*, p. 103, 2014.

[17] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[18] V. Campos, B. Jou, X. Giró-i Nieto, J. Torres, and S.-F. Chang, "Skip RNN: Learning to skip state updates in recurrent neural networks," *arXiv preprint arXiv:1708.06834*, 2017.

[19] R. Girshick, "Fast R-CNN," in *Computer Vision (ICCV), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1440–1448.

[20] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: trainable text-speech alignment using Kaldi," in *Proceedings of interspeech*, 2017.

[21] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[22] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Advances in neural information processing systems*, 2016, pp. 1019–1027.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] M. Federico, N. Bertoldi, and M. Cettolo, "IRSTLM: an open source toolkit for handling large scale language models," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[25] K. Hwang and W. Sung, "Character-level incremental speech recognition with recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5335–5339.

[26] D. Krueger, T. Maharaj, J. Kramár, M. Pezeshki, N. Ballas, N. R. Ke, A. Goyal, Y. Bengio, A. Courville, and C. Pal, "Zoneout: Regularizing RNNs by randomly preserving hidden activations," *Proceedings of the International Conference on Learning Representations*, 2017.

[27] Y. Wu, S. Zhang, Y. Zhang, Y. Bengio, and R. R. Salakhutdinov, "On multiplicative integration with recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 2856–2864.