

CONFORMER-BASED HYBRID ASR SYSTEM FOR SWITCHBOARD DATASET

Mohammad Zeineldeen^{1,2*}, Jingjing Xu^{1*}, Christoph Lüscher^{1,2}, Wilfried Michel^{1,2},
Alexander Gerstenberger¹, Ralf Schlüter^{1,2}, Hermann Ney^{1,2}

¹Human Language Technology and Pattern Recognition, Computer Science Department,
RWTH Aachen University, 52074 Aachen, Germany

²AppTek GmbH, 52062 Aachen, Germany

ABSTRACT

The recently proposed conformer architecture has been successfully used for end-to-end automatic speech recognition (ASR) architectures achieving state-of-the-art performance on different datasets. To our best knowledge, the impact of using conformer acoustic model for hybrid ASR is not investigated. In this paper, we present and evaluate a competitive conformer-based hybrid model training recipe. We study different training aspects and methods to improve word-error-rate as well as to increase training speed. We apply time downsampling methods for efficient training and use transposed convolutions to upsample the output sequence again. We conduct experiments on Switchboard 300h dataset and our conformer-based hybrid model achieves competitive results compared to other architectures. It generalizes very well on Hub5'01 test set and outperforms the BLSTM-based hybrid model significantly.

Index Terms— speech recognition, hybrid conformer-HMM, switchboard

1. INTRODUCTION & RELATED WORK

Hybrid neural network (NN)-hidden Markov model (HMM) automatic speech recognition (ASR) systems [1] have achieved state-of-the-art performance on different tasks [2, 3, 4]. Bi-directional long short-term memory (BLSTM) [5] has been widely used for acoustic modeling for conventional hybrid ASR systems. Other neural architectures such as time-delay neural networks [6] and convolutional neural networks (CNN) [7] were studied. The NN acoustic models (AMs) are often trained with cross-entropy using a frame-wise alignment generated by a Gaussian mixture model (GMM)-HMM system.

In the last few years, self-attention networks [8] have been shown to be better in terms of word-error-rate (WER) as well as training speed since the self-attention mechanism can be easily parallelized. Transformer-based hybrid models [9] have been investigated and shown to be very competitive. Recently, the conformer model [10], which is another self-attention-based model, was proposed and achieved state-of-the-art performance on Librispeech 960h dataset [11]. It uses

a convolution module to capture local context dependencies in addition to the long context captured by the self-attention module. The conformer architecture was investigated for different end-to-end systems such as attention encoder-decoder models [12, 13], and recurrent neural network transducer [10, 14]. Nevertheless, there has been no work investigating the impact of using a conformer AM for hybrid ASR systems.

In addition, the self-attention mechanism requires more memory resources and also, the time complexity grows quadratically with the sequence length. Thus, time downsampling techniques were introduced, mainly for end-to-end systems, such as pyramidal reduction [15], max-pooling layers [16], strided convolution [17], etc. When training hybrid ASR models with frame-wise alignment, upsampling the sequence at the output layer allows us to reuse the existing alignments. In this work, we utilize a simple but yet effective subsampling method by using strided convolution for downsampling and a transposed convolution [18] for upsampling. This method increases training speed and reduces memory consumption. This helps to do sequence discriminative training efficiently.

In this work, we propose and evaluate a conformer-based hybrid ASR system. To our best knowledge, there is no such training recipe for conformer-based hybrid models in the literature. We explore different aspects to improve the WER as well as training speed. We apply time subsampling techniques using strided convolution for downsampling and transposed convolution for upsampling. We also investigate the effect of parameter sharing between different components of the model. In addition, we use a method, called LongSkip, to connect the VGG [19] output directly to each conformer block which lead to moderate improvement. Moreover, due to time downsampling, it is possible to do sequence discriminative training efficiently as it reduces the memory usage requirements. We present improvements on top of the BLSTM hybrid model and competitive results on the Switchboard 300h Hub5'00 and Hub5'01 datasets.

2. CONFORMER ARCHITECTURE

The standard conformer architecture [10] consists mainly of four modules: feed-forward (FFN) module, multi-head

*Equal contribution

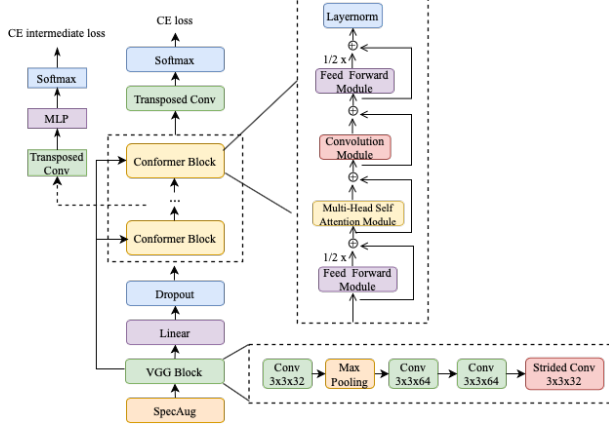


Fig. 1: Overview of the proposed conformer-based hybrid model with time downsampling, and other training methods. See Sections 3 and 4.1 for more description.

self-attention (MHSA) module, convolution (Conv) module, and another feed-forward module. Let x be the input sequence to conformer block i , then the equations can be defined as below:

$$\begin{aligned}
 x_{FFN_1} &= x + \frac{1}{2}FFN(x) \\
 x_{MHSA} &= x_{FFN_1} + MHSA(x_{FFN_1}) \\
 x_{Conv} &= x_{MHSA} + Conv(x_{MHSA}) \\
 x_{FFN_2} &= x_{Conv} + \frac{1}{2}FFN(x_{Conv}) \\
 \text{ConformerBlock}_i &= \text{LayerNorm}(x_{FFN_2})
 \end{aligned}$$

The illustration about the proposed conformer AM architecture can be seen in Figure 1.

3. TRAINING METHODS

To build and improve the training recipe for the conformer-based hybrid model, we investigate different training methods inspired from the literature. This helped to improve the WER as well as the training speed and memory efficiency.

Time down-/up-sampling: The self-attention mechanism requires allocating the whole input batch sequences into memory. Due to memory constraints, this can lead to batch size reduction making training slower. To handle this issue, time downsampling techniques can be applied. In this work, we use a strided convolution as part of the VGG network for downsampling. However, it is not straightforward to apply such downsampling methods for models trained with frame-wise target alignment. This will lead to a mismatch in the number of targets for computing the frame-wise loss objective function. To fix this issue, we use transposed convolution [18] inspired from computer vision in order to upsample again to the frame-wise target alignment length. For consistency, the filter size and the stride of the transposed convolution are set to the time reduction factor.

Intermediate loss: Training deeper networks requires careful tricks for convergence. Adding intermediate or auxiliary losses [9, 20] at different layers has been shown to

be effective for training stability. Note that when we do downsampling, a transposed convolution layer is needed for upsampling to compute the intermediate loss.

Parameter sharing can be used to reduce model size. We investigate the effect of sharing the parameters of the intermediate loss layers as well as the ones of the transposed convolution layers.

LongSkip is a simple residual connection [21] variant similar to densely connected convolution neural networks [22]. The motivation behind this is to connect a layer with previous layers to reuse learned features. In our case, we connect the output of the VGG network to the input of each conformer block which leads to some improvement.

Focal loss [23] is a method that reshapes the CE objective function to down-weight well-classified targets and thus makes the model focus on misclassified targets.

SpecAugment [24] is a data augmentation method that masks out blocks of frequency channels and blocks of time steps.

Sequence discriminative training [25] reduces the mismatch between training and recognition. We use that to further improve the performance of the conformer hybrid model.

4. EXPERIMENTAL SETUP

All acoustic models are trained on Switchboard 300h dataset [26] which consists of English telephony conversations. We use Hub5'00 as development set which consists of Switchboard (SWB) and CallHome (CH) parts. We use Hub5'01 as test set. We use RASR [27] for feature extraction and recognition. RETURNN [16] is used to train the acoustic models. All our config files and code to reproduce the results can be found online¹.

4.1. Baseline

For NN training, we use 40-dimensional Gammatone features [28]. The first block of the NN consists of a VGG network similar to [19]. We use 4 convolution layers each having 3×3 kernel size. The number of output filters for each layer are 32, 64, 64, 32 respectively. We apply Swish activation [29] between all convolution layers which we observe to be better than using ReLU [30]. Moreover, we apply max-pooling layer over feature dimension between first and second convolution layers. The last convolution layer is a strided convolution used for time downsampling by factor of 3. This is followed by 12 conformer blocks. For time upsampling, a transposed convolution is used. The attention dimension of each MHSA module is 512 with 8 attention heads. The dimension of the feed-forward module is 2048. We also use relative positional encoding. The output labels consists of state-tied triphones using CART [31]. The number of CART labels is 9001. We add two intermediate loss layers on the output of the 4th and 8th conformer blocks. These layers consist of a transposed convolution for upsampling followed by an MLP of one

¹<https://github.com/rwth-i6/returnn-experiments/tree/master/2021-swb-conformer-hybrid>

Table 1: WERs [%] of using different kernel sizes for depthwise convolution. 4-gram count-based LM is used.

Kernel size	WER [%]		
	Hub5'00		
	SWB	CH	Total
6	8.4	17.1	12.8
8	8.1	16.8	12.5
16	8.2	17.6	12.9
32	8.4	18.0	13.2

linear projection layer with dimension 512×512 . For training, we use frame-wise cross-entropy (CE) criteria. The target frame-wise alignment is generated using a HMM-GMM system from this setup [32]. We use Adam optimizer with Nesterov momentum (Nadam) [33]. Newbob learning schedule [34] with decay factor of 0.9 is applied to control the learning rate based on CE development set scores. We apply dropout of 10% for all conformer modules as well as embedding and attention dropout. We apply weight decay [35] with a value of 0.01 to the transposed convolution layers. We observe that applying weight decay to other layers hurts the performance. To avoid overfitting, we use focal loss [23] with a factor of 2. We apply linear learning rate warmup from 0.0002 to 0.018 for 1.6 epochs. We use batch size of 10k frames. Batches are constructed with shuffled data based on sequence length distribution. The baseline is trained for 27 epochs.

4.2. Language Models

In recognition, we utilize 4-gram count-based language model (LM) and LSTM LM as first pass decoding [36]. The LSTM LM has 51.3 perplexity (PPL) on Hub5'00 dataset. We use a transformer (Trafo) LM for rescoring having PPL of 48.1 on Hub5'00.

5. EXPERIMENTAL RESULTS

In this section, our training recipe for the conformer-based hybrid model is investigated. We conduct various experiments to understand the impact of each training method and also to find the current optimum recipe.

5.1. Depthwise Convolution Kernel Size

In Table 1, we experiment with different kernel sizes for the depthwise convolution. We can observe that the kernel size has a significant effect on WER. The model performs much better with smaller kernel size and the best WER is achieved with kernel size 8. This is also consistent with downsampling effect as the local context is sampled.

5.2. Number of Conformer Blocks

We conduct experiments with different number of conformer blocks using the best training recipe. Results are shown in Table 2. We can observe that we gain performance as we use deeper network. Due to memory constraints, we use 12 conformer blocks as baseline for our experiments.

5.3. Time Downsampling Factors and Variants

In Table 3, we report results with different time downsampling factors showing tradeoff between speed and performance.

Table 2: WERs [%] of using different number of conformer blocks. L is the number of conformer blocks. 4-gram count-based LM is used.

L	Params. [M]	WER [%]		
		Hub5'00		
		SWB	CH	Total
6	42	8.5	18.0	13.3
8	59	8.1	17.3	12.7
12	88	8.1	16.8	12.5

Table 3: WERs [%] of applying different time downsampling factors. Training time is reported over 1/6 split of train data using a single GeForce GTX 1080 Ti GPU. 4-gram count-based LM is used.

Factor	Train time [h]	WER [%]		
		Hub5'00		
		SWB	CH	Total
2	1.28	8.3	16.4	12.4
3	0.92	8.1	16.8	12.5
4	0.86	8.4	17.9	13.2
5	0.73	8.7	18.6	13.7

We use downsampling by factor of 3 for the baseline as compromise. In addition to that, Table 4 shows results with different variants of downsampling. The BLSTM+maxpool variant consists of one BLSTM layer with 512 units in each direction followed by a time max-pooling layer. VGG-layerX means that we use a strided convolution as the X^{th} layer of the VGG network. Results show that strided convolution works better for downsampling. It is also better to apply downsampling at the end of the VGG network which allows it to encode more information.

5.4. Ablation Study of Training Methods

To better understand the importance of each training method described in Section 3, we switch off (-) or switch on (+) only one method each time without re-optimizing the model. The results are shown in Table 5 and they are sorted descendingly based on absolute WER on Hub5'00. SpecAugment is the most important method where it helps significantly to avoid overfitting and yields 20% relative improvement in WER. Using intermediate loss is also important for better convergence, which achieves 7% relative improvement in WER. Moreover, sharing parameters between transposed convolutions helps. Other training methods seem to have marginal improvements.

5.5. Comparison between Conformer and BLSTM AM

In Table 6, we compare the performance between using a BLSTM or conformer as AM architecture. The BLSTM-based model consists of 6 BLSTM layers following a well-optimized setup as here [4]. SpecAugment is used. We can observe that using a BLSTM-based model improves the WER as we increase the number of parameters but still worse than a less parameterized conformer-based model. In this case, with nearly comparable number of parameters, the conformer model outperforms the BLSTM model by around 9% relative

Table 4: WERs [%] for front-end and time downsampling variants. Downsampling factor of 3 is used. 4-gram count-based LM is used.

Method	WER [%]		
	Hub5'00		
	SWB	CH	Total
BLSTM+maxpool	8.2	17.0	12.7
VGG-layer2	8.4	17.7	13.1
VGG-layer4	8.1	16.8	12.5

Table 5: WERs [%] of ablation study on the best training recipe. 4-gram count-based LM is used.

Training method	WER [%]		
	Hub5'00		
	SWB	CH	Total
Baseline	8.1	16.8	12.5
- SpecAugment	9.8	21.5	15.7
- Intermediate loss	8.9	18.1	13.5
- Share transp. conv params.	8.5	17.3	12.9
- LongSkip	8.1	17.2	12.7
- Focal Loss	8.1	17.0	12.6
+ Share MLP params.	8.2	16.9	12.5

in terms of WER.

5.6. Sequence Discriminative Training (seq. train)

We use the lattice-based version of state-level minimum Bayes risk (sMBR) criterion [37]. The lattices are generated using the best conformer AM and a bigram LM. We observe that using a bigram LM is better than using a 4-gram LM. A small constant learning rate with value 1e-5 is used. At the final output layer, we use CE loss smoothing with a factor of 0.1. The sMBR loss scale is set to 0.9. Sequence discriminative training leads to 5% relative improvement as shown in Table 7. sMBR training requires the full input sequence without chunking. Thus, it is important to note that due to downsampling, it was possible to train with sMBR efficiently since the memory usage requirement is reduced and therefore we can use larger batch sizes.

6. OVERALL RESULTS & DISCUSSION

We summarize our results in Table 7 and compare with different modeling approaches and architectures from the literature. Overall, our conformer-based hybrid model yields competitive results. Compared to BLSTM hybrid, the conformer model with 4-gram LM is 14.4% relatively better in WER on Hub5'00. It is also 8.5% relatively better with LSTM LM. Moreover, our conformer model outperforms a well-trained RNN-T model with much fewer epochs. The model is also on par with a well-optimized BLSTM attention system [38] on Hub5'01 test set. However, the conformer hybrid model is still behind the state-of-the-art conformer attention-based system, yet it is trained with much smaller number of epochs and this seems to matter a lot. Note also that [13, 38] use cross-utterance LM [39] during recognition which boosts

Table 6: Comparison of WERs [%] between BLSTM and conformer AM architectures. All models are trained for 27 epochs. 4-gram count-based LM is used.

AM	LSTM dim.	Params. [M]	Hub5'00
BLSTM	500	41	14.2
	600	57	13.8
	700	76	13.8
	800	96	13.7
	1000	146	13.3
Conformer	-	88	12.5

Table 7: Overall WER [%] comparison with literature.

Work	#Epochs	Approach	AM	LM	seq. train	WER [%]	
						Hub 5'00	Hub 5'01
[4]	-	Hybrid	LSTM	4-gram LSTM	yes	13.9	-
						11.7	
[40]	100	RNN-T	LSTM	LSTM Trafo	no	11.5	11.5
						11.2	11.2
[38]	250	LAS	LSTM	LSTM	no	9.8	10.1
				-		9.9	10.1
[13]	250	LAS	Conf.	LSTM	no	8.6	8.5
				Trafo		8.4	8.5
ours	27	Hybrid	Conf.	4-gram LSTM	no	12.5	12.1
						11.3	10.5
				4-gram LSTM	yes	11.9	11.4
				Trafo		10.7	10.1
						10.3	9.7

their performance.

7. CONCLUSION

In this work, for the first time a training recipe for a conformer-based hybrid model is evaluated. We combined different training methods from the literature that boosted the word-error-rate. We successfully applied time downsampling using strided convolution to speedup training and used transposed convolution as a simple method to upsample again. We observe that SpecAugment and intermediate loss layers are necessary to achieve good performance. Sharing parameters between transposed convolution layers leads to moderate improvement. Our model generalizes very well on the Switchboard 300h test set Hub5'01 and outperforms the BLSTM-based hybrid model significantly. We believe further improvements are still possible if we do speed perturbation, speaker adaptation, and longer training epochs.

Acknowledgement

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement n° 694537, project "SEQCLAS"). The work reflects only the authors' views and the European Research Council Executive Agency (ERCEA) is not responsible for any use that may be made of the information it contains. This work was partially supported by the project HYKIST funded by the German Federal Ministry of Health on the basis of a decision of the German Federal Parliament (Bundestag) under funding ID ZMV11-2520DAT04A.



8. REFERENCES

- [1] H. A. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, Kluwer Academic Publishers, USA, 1993.
- [2] W. Zhou, W. Michel, K. Irie, M. Kitza, R. Schlüter, and H. Ney, “The RWTH ASR System for TED-LIUM Release 2: Improving Hybrid HMM with SpecAugment,” in *ICASSP*, Barcelona, Spain, May 2020, pp. 7839–7843.
- [3] C. Lüscher, E. Beck, K. Irie, M. Kitza, W. Michel, A. Zeyer, R. Schlüter, and H. Ney, “RWTH ASR Systems for LibriSpeech: Hybrid vs Attention,” in *INTERSPEECH*, Graz, Austria, Sept. 2019, pp. 231–235.
- [4] M. Kitza, P. Golik, R. Schlüter, and H. Ney, “Cumulative Adaptation for BLSTM Acoustic Models,” in *INTERSPEECH*, Graz, Austria, Sept. 2019, pp. 754–758.
- [5] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [6] V. Peddinti, D. Povey, and S. Khudanpur, “A Time Delay Neural Network Architecture for Efficient Modeling of Long Temporal Contexts,” in *INTERSPEECH*, Dresden, Germany, Sept. 2015, pp. 3214–3218.
- [7] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, “Convolutional Neural Networks for Speech Recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533–1545, 2014.
- [8] M. Sperber, J. Niehues, G. Neubig, S. Stüker, and A. Waibel, “Self-Attentional Acoustic Models,” in *INTERSPEECH*, Hyderabad, India, Sept. 2018, pp. 3723–3727.
- [9] Y. Wang, A. Mohamed, D. Le, C. Liu, A. Xiao, J. Mahadeokar, H. Huang, A. Tjandra, X. Zhang, F. Zhang, C. Fuegen, G. Zweig, and M.L. Seltzer, “Transformer-Based Acoustic Modeling for Hybrid Speech Recognition,” in *ICASSP*, Barcelona, Spain, May 2020, pp. 6874–6878.
- [10] A. Gulati, J. Qin, C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, “Conformer: Convolution-augmented Transformer for Speech Recognition,” in *INTERSPEECH*, Shanghai, China, Oct. 2020, pp. 5036–5040.
- [11] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “LibriSpeech: An ASR Corpus Based on Public Domain Audio Books,” in *ICASSP*, South Brisbane, Australia, Apr. 2015, pp. 5206–5210.
- [12] X. Wang, S. Sun, L. Xie, and L. Ma, “Efficient Conformer with Prob-Sparse Attention Mechanism for End-to-End Speech Recognition,” *CoRR*, vol. abs/2106.09236, June 2021.
- [13] Z. Tüske, G. Saon, and B. Kingsbury, “On the Limit of English Conversational Speech Recognition,” *CoRR*, vol. abs/2105.00982, May 2021.
- [14] M. Burchi and V. Vielzeuf, “Efficient Conformer: Progressive Downsampling and Grouped Attention for Automatic Speech Recognition,” *CoRR*, vol. abs/2109.01163, Sept. 2021.
- [15] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition,” in *ICASSP*, May 2016, pp. 4960–4964.
- [16] A. Zeyer, T. Alkhoul, and H. Ney, “RETURNN as a Generic Flexible Neural Toolkit with Application to Translation and Speech Recognition,” in *Annual Meeting of the Assoc. for Computational Linguistics*, Melbourne, Australia, July 2018.
- [17] W. Han, Z. Zhang, Y. Zhang, J. Yu, C. Chiu, J. Qin, A. Gulati, R. Pang, and Y. Wu, “ContextNet: Improving Convolutional Neural Networks for Automatic Speech Recognition with Global Context,” in *INTERSPEECH*, Shanghai, China, Sept. 2020, pp. 3610–3614.
- [18] J. Long, E. Shelhamer, and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation,” in *CVPR*, Boston, USA, June 2015, pp. 3431–3440.
- [19] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” in *ICLR*, San Diego, USA, May 2015.
- [20] A. Tjandra, C. Liu, F. Zhang, X. Zhang, Y. Wang, G. Synnaeve, S. Nakamura, and G. Zweig, “DEJA-VU: Double Feature Presentation and Iterated Loss in Deep Transformer Networks,” in *ICASSP*, Barcelona, Spain, May 2020, pp. 6899–6903.
- [21] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *CVPR*, Las Vegas, USA, June 2016, pp. 770–778.
- [22] G. Huang, Z. Liu, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” in *CVPR*, Los Alamitos, CA, USA, Jul 2017, pp. 2261–2269.
- [23] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal Loss for Dense Object Detection,” in *ICCV*, Venice, Italy, Oct. 2017, pp. 2999–3007.
- [24] D. S. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition,” in *INTERSPEECH*, Graz, Austria, 2019, pp. 2613–2617.
- [25] K. Vesely, A. Ghoshal, L. Burget, and D. Povey, “Sequence-discriminative Training of Deep Neural Networks,” in *INTERSPEECH*, Lyon, France, Aug. 2013, pp. 2345–2349.
- [26] J.J. Godfrey, E.C. Holliman, and J. McDaniel, “Switchboard: Telephone Speech Corpus for Research and Development,” in *ICASSP*, San Francisco, USA, Mar. 1992, vol. 1, pp. 517–520.
- [27] S. Wiesler, A. Richard, P. Golik, R. Schlüter, and H. Ney, “RASR/NN: The RWTH Neural Network Toolkit for Speech Recognition,” in *ICASSP*, Florence, Italy, May 2014, pp. 3313–3317.
- [28] R. Schluter, I. Bezrukov, H. Wagner, and H. Ney, “Gamma-tone Features and Feature Combination for Large Vocabulary Speech Recognition,” in *ICASSP*, Honolulu, USA, Apr. 2007, pp. 649–652.
- [29] P. Ramachandran, B. Zoph, and Q. V. Le, “Searching for Activation Functions,” in *ICLR*, Vancouver, Canada, Apr. 2018.
- [30] A. Agarap, “Deep Learning using Rectified Linear Units (ReLU),” *CoRR*, vol. abs/1803.08375, 2018.
- [31] S. J. Young, J. J. Odell, and P. C. Woodland, “Tree-Based State Tying for High Accuracy Acoustic Modelling,” in *HLT*, Plainsboro, USA, 1994, p. 307–312.
- [32] Z. Tuske, P. Golik, R. Schluter, and H. Ney, “Speaker Adaptive Joint Training of Gaussian Mixture Models and Bottleneck Features,” in *ASRU*, Scottsdale, USA, Dec. 2015, pp. 596–603.
- [33] T. Dozat, “Incorporating Nesterov Momentum into Adam,” in *ICLR workshop*, 2016.
- [34] A. Zeyer, P. Doetsch, P. Voigtlaender, R. Schluter, and H. Ney, “A Comprehensive Study of Deep Bidirectional LSTM RNNs for Acoustic Modeling in Speech Recognition,” in *ICASSP*, New Orleans, LA, USA, Mar. 2017, pp. 2462–2466.
- [35] A. Krogh and J. Hertz, “A Simple Weight Decay Can Improve Generalization,” in *NIPS*, Colorado, USA, Dec. 1991.
- [36] E. Beck, W. Zhou, R. Schlüter, and H. Ney, “LSTM Language Models for LVCSR in First-Pass Decoding and Lattice-Rescoring,” *CoRR*, vol. abs/1907.01030, 2019.
- [37] M. Gibson and T. Hain, “Hypothesis Spaces for Minimum Bayes Risk Training in Large Vocabulary Speech Recognition,” in *INTERSPEECH*, Pittsburgh, USA, Sept. 2006.
- [38] Z. Tüske, G. Saon, K. Audhkhasi, and B. Kingsbury, “Single Headed Attention Based Sequence-to-Sequence Model for State-of-the-Art Results on Switchboard,” in *INTERSPEECH*, Shanghai, China, Sept. 2020, pp. 551–555.
- [39] K. Irie, A. Zeyer, R. Schlüter, and H. Ney, “Training Language Models for Long-Span Cross-Sentence Evaluation,” in *ASRU*, Sentosa, Singapore, Dec. 2019, pp. 419–426.
- [40] W. Zhou, S. Berger, R. Schlüter, and H. Ney, “Phoneme Based Neural Transducer for Large Vocabulary Speech Recognition,” in *ICASSP*, June 2021, pp. 5644–5648.