A Study of All-Convolutional Encoders for Connectionist Temporal Classification

Kalpesh Krishna[1], Liang Lu[2], Kevin Gimpel[3], Karen Livescu[3]

[1]IIT Bombay, India, [2]Microsoft AI and Research, USA, [3]TTI-Chicago, USA

Introduction

- End-to-end speech recognition primarily uses encoder-decoder or CTC models, mostly using LSTMs or a LSTM+CNN combination.
- We explore purely convolutional CTC models for lexicon-free conversational speech recognition, which are much faster than recurrent models.
- Unlike most previous work [1, 2] we focus on 1-D convolutions. TDNNs [3] are closely related to our work.

Model & Experimental Setup

Neural "encoders" map input sequences to hidden states h_t and a softmax layer maps h_t to a distribution over frame level CTC labels π_t.

Connectionist Temporal Classification (CTC)

- We use the standard CTC collapsing function $B(\pi)$ which removes the special blank symbols and consecutive repetitions.

$$p(\mathbf{z}|\mathbf{x}) = \sum_{\pi \in B^{-1}(\mathbf{z})} \prod_t p(\pi_t|h_t)$$

- We present results using a greedy decoding approach as well as beam search with a n-gram character LM. If $\mathbf{z} = B(\pi)$,

$$\hat{\pi} = \underset{\pi \in \mathbf{\Pi}}{\arg\max} \; p(\mathbf{z})^{\alpha} |\mathbf{z}|^{\beta} \prod_t p(\pi_t|h_t)$$

We use the decoding algorithm in [4].

Long Short-Term Memory (LSTM) Baseline

- 5-layer 320 hidden unit bidirectional LSTM network with dropout between consecutive layers. Every two consecutive input frames are concatenated to reduce time resolution.

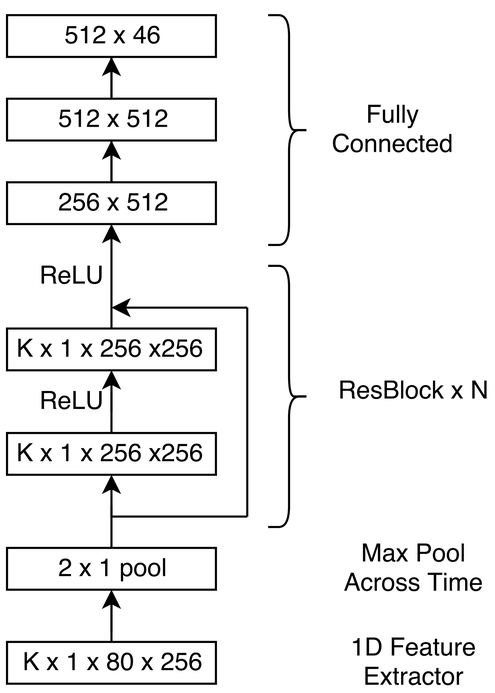
CNN Model

- 1-D convolutions across time only. Following [5], we use residual connections ("ResBlocks (RBs)") and batch normalization.

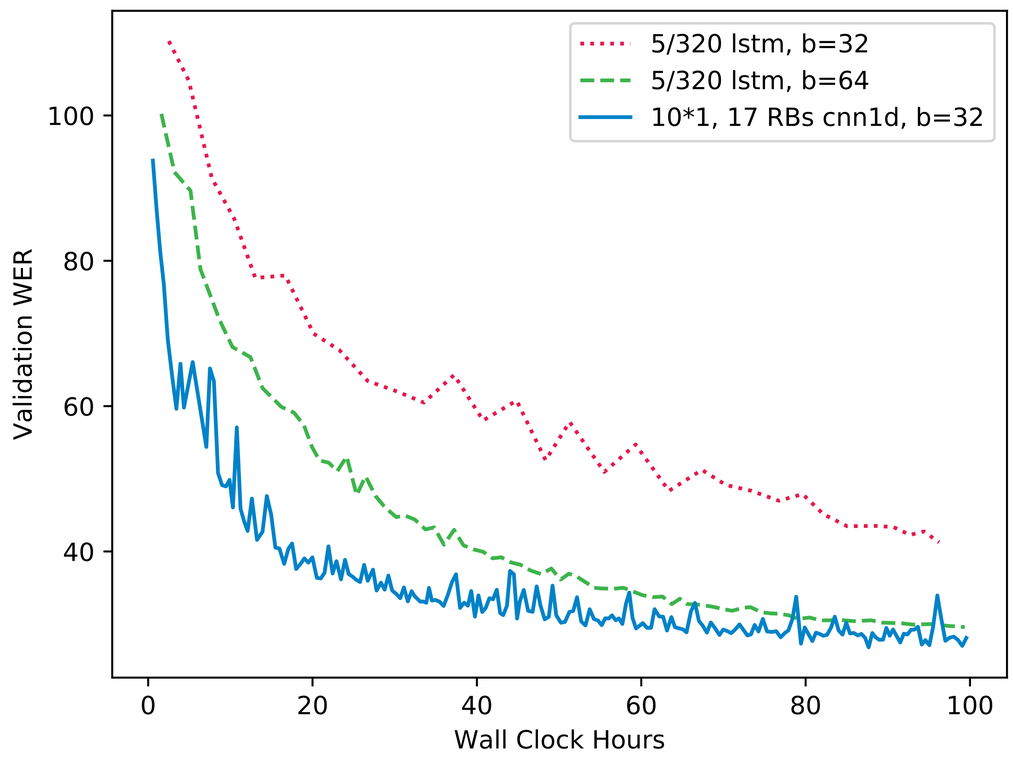
Experimental Setup

- We use the 300h Switchboard corpus for training, and report results on the 4k utterance Switchboard dev set, and Eval2000 setup consisting of Switchboard (SWB) and Callhome (CH) utterances.
- All models trained on a single Titan X GPU, with two CPU threads in TensorFlow `r1.1`.

CNN Architecture



Speed of Convergence



Main Results

Table: Final test set results on Eval2000.

Model	Switchboard	CallHome	Eval2000
5/320 LSTM + no LM	**27.7**	**47.5**	37.6
5/320 LSTM + 7-g	20.0	38.5	29.3
5/320 LSTM + 9-g	**19.7**	**38.2**	29.0
5*1 28 RBs, CNN + no LM	**27.9**	**48.6**	38.3
5*1 28 RBs, CNN + 7-g	21.7	40.4	31.1
5*1 28 RBs, CNN + 9-g	**21.3**	**40.0**	30.7
Maas [4] + no LM	38.0	56.1	47.1
Maas [4] + 7-g	27.8	43.8	35.9
Maas [4] + RNN	21.4	40.2	30.8
Zenkel [6] + no LM	30.4	44.0	37.2
Zenkel [6] + RNN	18.6	31.6	25.1
Zweig [7] + no LM	25.9	38.8	-
Zweig [7] + n-g	19.8	32.1	-

Table: Greedy decoding time on the Eval2000 in **seconds** averaged over three runs.

Model	# Weights	b	t_{wc} / t_{cpu} (s)
5/320 LSTM	11.1M	1	1813 / 3667
5/320 LSTM	11.1M	32	87 / 180
5/320 LSTM	11.1M	64	44 / 92
5*1, 28 RBs, CNN	19.0M	1	115 / 135
5*1, 28 RBs, CNN	19.0M	32	17 / 18
5*1, 28 RBs, CNN	19.0M	64	15 / 16

Table: Development set WER for 1-D CNNs vs. number of layers. b denotes batch-size. Each model is trained for 40 epochs with early stopping. t_{wc}/t_{cpu} are hours / epoch.

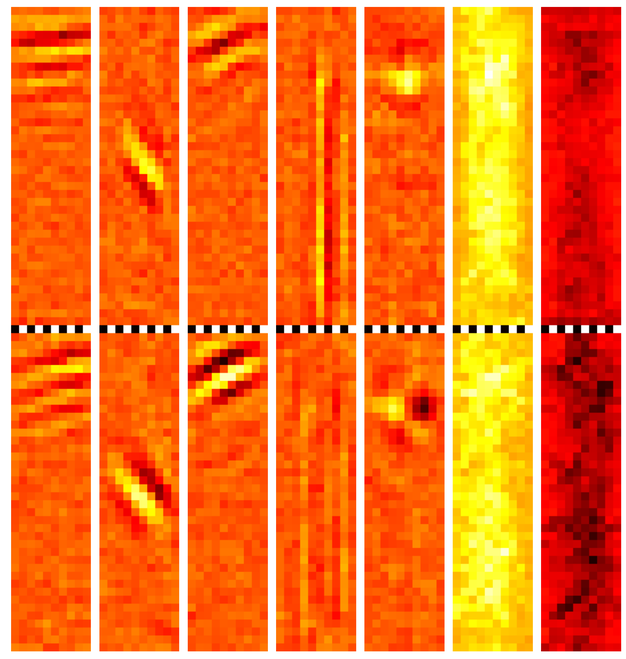
Model	# Weights	WER %	b	t_{wc} / t_{cpu}
5/320 LSTM	11.1M	28.54	64	3.3 / 5.8
10*1, 8 RBs	11.1M	36.71	32	0.9 / 2.2
10*1, 11 RBs	15.1M	32.67	32	1.0 / 2.5
10*1, 14 RBs	19.0M	30.92	32	1.1 / 2.8
10*1, 17 RBs	22.9M	**29.82**	32	1.5 / 3.5

Table: Development set WER for 1-D CNNs vs. filter size, each trained for 40 epochs with early stopping. We vary filter size / depth at a constant number of weights.

Model	# Weights	WER %	t_{wc} / t_{cpu}
5*1, 16 RBs	11.1M	**33.26**	1.0 / 2.3
10*1, 8 RBs	11.1M	36.71	0.9 / 2.2
15*1, 6 RBs	12.4M	39.83	0.9 / 2.4
5*1, 28 RBs	19.0M	**29.65**	1.4 / 3.5
10*1, 14 RBs	19.0M	30.92	1.1 / 2.8
15*1, 10 RBs	20.3M	33.94	1.1 / 3.0

First Layer Filters

Figure: Visualization of first layer CNN filters. The static and delta channels separated by a checkerboard pattern.



Key Results

- 1-D CNNs train and decode significantly faster than LSTMs for speech recognition with CTC.
- For the same number of weights, deeper networks with smaller filters perform best.
- CNNs are only 0.2% behind LSTMs on the Switchboard test set, but are a larger 1.1% behind on CallHome, indicating over-fitting.
- CNNs respond less to language model based beam-search decoding.
- Very deep ResNet-style CNNs [5] (50+ layers) are needed to match LSTM performance.

Future Work

- Better regularization techniques for CNN architectures to prevent over-fitting
- Analysis of larger all-CNN systems on a word level CTC architecture
- Response of all-convolutional systems to non-CTC architectures, and different decoding schemes [6]

References

[1] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. Laurent, Y. Bengio, and A. C. Courville, "Towards end-to-end speech recognition with deep convolutional neural networks," in *Interspeech*, 2016.

[2] Y. Wang, X. Deng, S. Pu, and Z. Huang, "Residual convolutional CTC networks for automatic speech recognition," *CoRR*, vol. abs/1702.07793, 2017.

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

[4] A. L. Maas, Z. Xie, D. Jurafsky, and A. Y. Ng, "Lexicon-free conversational speech recognition with neural networks," in *HLT-NAACL*, 2015.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[6] T. Zenkel, R. Sanabria, F. Metze, J. Niehues, M. Sperber, S. Stüker, and A. Waibel, "Comparison of decoding strategies for CTC acoustic models," in *Interspeech*, 2017.

[7] G. Zweig, C. Yu, J. Droppo, and A. Stolcke, "Advances in all-neural speech recognition," in *ICASSP*, 2017.