FULLY CONVOLUTIONAL SPEECH RECOGNITION

Neil Zeghidour[1,2,*], *Qiantong Xu*[1,*], *Vitaliy Liptchinsky*[1], *Nicolas Usunier*[1], *Gabriel Synnaeve*[1], *Ronan Collobert*[1]

[1] Facebook A.I. Research, Paris, France; New York & Menlo Park, USA
[2] CoML, ENS/CNRS/EHESS/INRIA/PSL Research University, Paris, France

ABSTRACT

Current state-of-the-art speech recognition systems build on recurrent neural networks for acoustic and/or language modeling, and rely on feature extraction pipelines to extract mel-filterbanks or cepstral coefficients. In this paper we present an alternative approach based solely on convolutional neural networks, leveraging recent advances in acoustic models from the raw waveform and language modeling. This fully convolutional approach is trained end-to-end to predict characters from the raw waveform, removing the feature extraction step altogether. An external convolutional language model is used to decode words. On Wall Street Journal, our model matches the current state-of-the-art. On Librispeech, we report state-of-the-art performance among end-to-end models, including Deep Speech 2 trained with 12 times more acoustic data and significantly more linguistic data.

***Index Terms*—** Speech recognition, end-to-end, convolutional, language model, waveform

1. INTRODUCTION

Recent work on convolutional neural network architectures showed that they are competitive with recurrent architectures even on tasks where modeling long-range dependencies is critical, such as language modeling [1], machine translation [2, 3] and speech synthesis [4]. In end-to-end speech recognition however, recurrent architectures are still prevalent for acoustic and/or language modeling [5, 6, 7, 8, 9].

There is a history of using convolutional networks in speech recognition, but only as part of an otherwise more traditional pipeline. They have been first introduced as TDNNs to predict phoneme classes [10], and later to generate HMM posteriorgrams [11]. They have more recently been used in end-to-end frameworks, but only in combination with recurrent layers [7], or n-gram language models [12], or for phone recognition [13, 14]. Nonetheless, convolutional architectures are prevalent when learning from the raw waveform [15, 16, 17, 14, 18], because they naturally model the computation of standard features such as mel-filterbanks. Given the evidence that they are also suitable on long-range dependency tasks, we expect convolutional neural networks to be competitive at all levels of the speech recognition pipeline.

In this paper, we present a fully convolutional approach to end-to-end speech recognition. Building on recent advances in convolutional learnable front-ends for speech [14, 18], convolutional acoustic models [12], and convolutional language models [1], the paper has four main contributions:

1. We present the first application of convolutional language models to speech recognition. They yield significant improvements over 4-gram language models on both Wall Street Journal (WSJ) and Librispeech datasets.

2. We show that fully convolutional approaches are competitive with approaches based on recurrent neural networks. In particular, on Librispeech, we improve by more than 1% absolute Word Error Rate the results of DeepSpeech 2 [7] and of the best sequence-to-sequence model [9].

3. We present the first state-of-the-art results (among end-to-end systems) on a large, publicly available dataset (Librispeech) that use end-to-end learning from the raw waveform. On WSJ, we improve on the best previous results based on learnable front-ends [18] and match the current state-of-the-art, an HMM-DNN system.

4. On Librispeech, learning the front-end has a larger impact in noisy than in clean recording conditions. These results corroborate previous observations on the VoiceSearch dataset [16], and give additional evidence that mel-filterbanks are suboptimal in the noisy setting.

2. MODEL

Our approach, described in this section, is illustrated in Fig. 1.

2.1. Convolutional Front end

Several proposals to learn the front-end of speech recognition systems have been made [16, 17, 14, 18]. Following the comparison in [18], we consider their best architecture, called

* Equal contribution

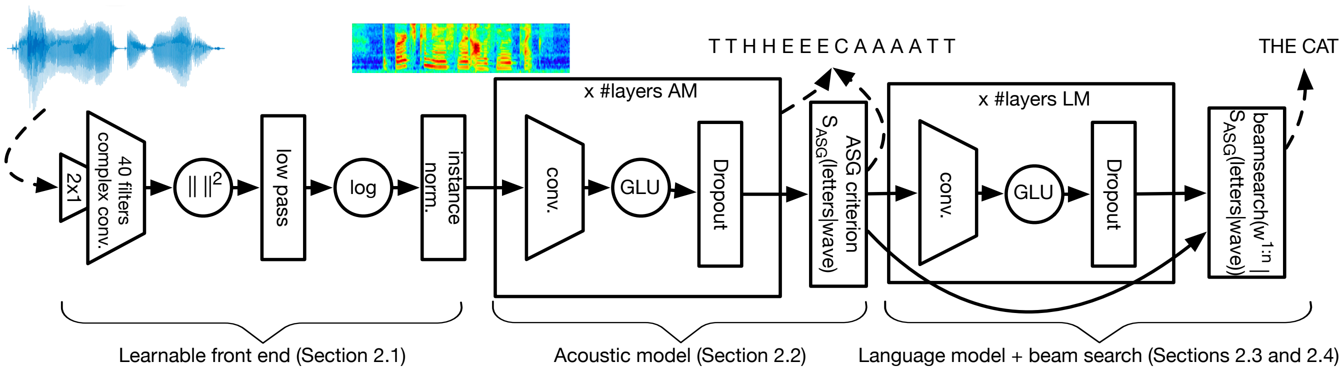

Fig. 1: Overview of the fully convolutional architecture.

"scattering based" (hereafter refered to as *learnable front-end*). The learnable front-end contains first a convolution of width 2 that emulates the pre-emphasis step used in mel-filterbanks. It is followed by a complex convolution of width 25ms and k filters. After taking the squared absolute value, a low-pass filter of width 25ms and stride 10ms performs decimation. The front-end finally applies a log-compression and a per-channel mean-variance normalization (equivalent to an instance normalization layer [19]). Following [18], the "pre-emphasis" convolution is initialized to $[-0.97; 1]$, and then trained with the rest of the network. The low-pass filter is kept constant to a squared Hanning window, and the complex convolutional layer is initialized randomly. In addition to the $k = 40$ filters used by [18], we experiment with $k = 80$ filters. Notice that since the stride is the same as for mel-filterbanks, acoustic models on top of the learnable front-ends can also be applied to mel-filterbanks (simply modifying the number of input channels if $k \neq 40$).

2.2. Convolutional Acoustic Model

The acoustic model is a convolutional neural network with gated linear units [1], which is fed with the output of the learnable front-end. Following [12], the networks uses a growing number of channels, and dropout [20] for regularization. These acoustic models are trained to predict letters directly with the Auto Segmentation Criterion (ASG) [21]. The only differences between the WSJ and Librispeech models are their depth, the number of feature maps per layer, the receptive field and the amount of dropout.

2.3. Convolutional Language Model

The convolutional language model (LM) is the GCNN-14B from [1], which achieved competitive results on several language modeling benchmarks. The network contains 14 convolutional residual blocks [22] with a growing number of channels, and uses gated linear units as activation function.

The language model is used to score candidate transcriptions in addition to the acoustic model in the beam search decoder described in the next section. Compared to n-gram LMs, convolutional LMs allow for much larger context sizes. Our detailed experiments study the effect of context size on the final speech recognition performance.

2.4. Beam-search decoder

We use the beam-search decoder presented in [12] to generate word sequences given the output from our acoustic model. The decoder finds the word transcription W to maximize:

$$AM(W) + \alpha \log P_{lm}(W) + \beta|W| - \gamma|\{i|\pi_i = \langle sil\rangle\}|,$$

where π_i is the value for the ith frame in the path leading to W and $AM(W)$ is the (unnormalized) acoustic model score of the transcription W. The hyperparameters $\alpha, \beta, \gamma \geq 0$ respectively control the weight of the language model, the word insertion reward, and the silence insertion penalty. The other parameters are the beam size and the beam score, a threshold under which candidates are discarded even if the beam is not full. These are chosen according to a trade-off between (near-)optimality of the search and computational cost.

3. EXPERIMENTS

We evaluate our approach on the large vocabulary task of the Wall Street Journal (WSJ) dataset [26], which contains 80 hours of clean read speech, and Librispeech [27], which contains 1000 hours with separate train/dev/test splits for clean and noisy speech. Each dataset comes with official textual data to train language models, which contain 37 million tokens for WSJ, 800 million tokens for Librispeech. Our language models are trained separately for each dataset on the official text data only. These datasets were chosen to study the impact of the different components of our system at different scales of training data and in different recording conditions.

The models are evaluated in Word Error Rate (WER). Our experiments use the open source codes of wav2letter[1] for the

[1] https://github.com/facebookresearch/wav2letter

Model		dev-clean	dev-other	test-clean	test-other
CAPIO (Single) [23] *(speaker adapt., pronunciation lexicon)*		3.02	8.28	3.56	8.58
CAPIO (Ensemble) [23] *(Combination of 8 systems)*		2.68	7.56	3.19	7.64
DeepSpeech 2 [7] *(12k training hours AM, common crawl LM)*		-	-	5.83	12.69
Sequence-to-sequence [9]		3.54	11.52	3.82	12.76
Frontend	**LM**	**dev-clean**	**dev-other**	**test-clean**	**test-other**
mel-filterbanks	4-gram	4.26	13.80	4.82	14.54
mel-filterbanks	ConvLM	3.13	10.61	3.45	11.92
Learnable front-end (40 filters)	ConvLM	3.16	10.05	3.44	11.24

Table 1: WER (%) on Librispeech.

Model		nov93	nov92
E2E Lattice-free MMI [24] *(data augmentation)*		-	4.1
CNN-DNN-BLSTM-HMM [25] *(speaker adaptation, 3k acoustic states)*		6.6	3.5
DeepSpeech 2 [7] *(12k training hours AM, common crawl LM)*		5	3.6
Frontend	**LM**	**nov93**	**nov92**
mel-filterbanks	4-gram	9.5	5.6
mel-filterbanks	ConvLM	7.5	4.1
Learnable front-end (40 filters)	ConvLM	7.1	4.0
Learnable front-end (80 filters)	ConvLM	6.8	3.5

Table 2: WER (%) on the open vocabulary task of WSJ.

acoustic model, and fairseq[2] for the language model. More details on the experimental setup are given below.

Baseline Our baseline for each dataset follows [12]. It uses the same convolutional acoustic model as our approach but a mel-filterbanks front-end and a 4-gram language model.

Training/test splits On WSJ, models are trained on *si284*. *nov93dev* is used for validation and *nov92* for test. On Librispeech, we train on the concatenation of *train-clean* and *train-other*. The validation set is *dev-clean* when testing on *test-clean*, and *dev-other* when testing on *test-other*.

Acoustic model architecture The architecture for the convolutional acoustic model is the "high dropout" model from [12] for Librispeech, which has 19 layers in addition to the front-end (mel-filterbanks for the baseline, or the learnable front-end for our approach). On WSJ, we use the lighter version used in [18], which has 17 layers. Dropout is applied at each layer after the front-end, following [21]. The learnable front-end uses 40 or 80 filters.

[2]https://github.com/facebookresearch/fairseq

Language model architecture As described in Section 2.3, we use the GCNN-14B model of [1] with dropout at each convolutional and linear layer on both WSJ and Librispeech. We keep all the words (162K) in WSJ training corpus. For Librispeech, we only use the most frequent 200K tokens (out of 900K).

Hyperparameter tuning The acoustic models are trained following [12, 18], using SGD with a decreasing learning rate, weight normalization and gradient clipping at 0.2 and a momentum of 0.9. The language models are trained with Nesterov accelerated gradient [28]. Following [1], we also use weight normalization and gradient clipping.

The parameters of the beam search (see Section 2.4) α, β and γ are tuned on the validation set with a beam size of 2500 and a beam score of 26 for computational efficiency. Once α, β, γ are chosen, the test WER is computed with a beam size of 3000 and a beam score of 50.

4. RESULTS

4.1. Word Error Rate results

4.1.1. Wall Street Journal dataset

Table 2 shows Word Error Rates (WER) on WSJ for the current state-of-the-art and our models. The current best model trained on this dataset is an HMM-based system which uses a combination of convolutional, recurrent and fully connected layers, as well as speaker adaptation, and reaches 3.5% WER on *nov92*. DeepSpeech 2 shows a WER of 3.6% but uses 150 times more training data for the acoustic model and huge text datasets for LM training. Finally, the state-of-the-art among end-to-end systems trained only on WSJ, and hence the most comparable to our system, uses lattice-free MMI on augmented data (with speed perturbation) and gets 4.1% WER. Our baseline system, trained on mel-filterbanks, and decoded with a n-gram language model has a 5.6% WER. Replacing the n-gram LM by a convolutional one reduces the WER to 4.1% , and puts our model on par with the current best end-to-end system. Replacing the speech features by a learnable frontend finally reduces the WER to 4.0% and then to 3.5% when doubling the number of learnable filters, improving over DeepSpeech 2 and matching the performance of the best HMM-DNN system.

4.1.2. Librispeech dataset

Table 1 reports WER on the Librispeech dataset. The CAPIO [23] ensemble model combines the lattices from 8 individual HMM-DNN systems (using both convolutional and LSTM layers), and is the current state-of-the-art on Librispeech. CAPIO (single) is the best individual system, selected either on dev-clean or dev-other. The sequence-to-sequence baseline is an encoder-decoder with attention and a BPE-level [29] LM, and currently the best end-to-end system on this dataset. We can observe that our fully convolutional model improves over CAPIO (Single) on the clean part, and is the current best end-to-end system on test-other with an improvement of 1.5% absolute. Our system also outperforms DeepSpeech 2 on both test sets by a significant margin. An interesting observation is the impact of each convolutional block. While replacing the 4-gram LM by a convolutional LM improves similarly on the clean and noisier parts, learning the speech frontend gives similar performance on the clean part but significantly improves the performance on noisier, harder utterances, a finding that is consistent with previous literature [16].

4.2. Analysis of the convolutional language model

Since this paper uses convolutional language models for speech recognition systems for the first time, we present additional studies of the language model in isolation. These experiments use our best language model on Librispeech, and evaluations in WER are carried out using the baseline system trained on mel-filterbanks. The decoder parameters are tuned using the grid search described in Section 3, a beam size is fixed to 2500 and a beam score to 30.

Correlation between perplexity and WER Figure 2 shows the correlation between perplexity and WER as the training progresses. As perplexity decreases, the WER on both *dev-clean* and *dev-other* also decreases following the same trend. It illustrates that perplexity on the linguistic data is a good surrogate of the final performance of the speech recognition pipeline. Architectural choices or hyper-parameter tuning can thus be carried out mostly using perplexity alone.

Influence of context size By limiting the context passed into the LM from the decoder, Table 3 reports WER obtained for context sizes ranging from 3 (comparable to the n-gram baseline) to 50 for our best language model. The WER decreases monotonically until a context size of about 20, and then almost stays still. We observe that the convolutional LM already improves on the n-gram model even with the same context size. Increasing the context gives a significant boost in performance, with the major gains obtained between a context of 3 to 9 (-1.9% absolute WER).

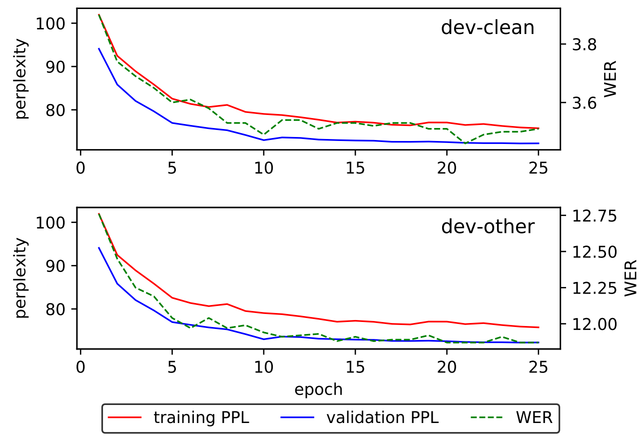

Fig. 2: Evolution of WER (%) on Librispeech with the perplexity of the language model.

Model Type	Context Length	WER dev-clean	WER dev-other
4-gram	3	4.26	13.80
ConvLM	3	4.11	13.17
ConvLM	9	3.34	11.29
ConvLM	19	3.27	11.06
ConvLM	29	3.25	11.09
ConvLM	39	3.24	11.07
ConvLM	49	3.24	11.08

Table 3: Evolution of WER (%) on Librispeech with the context size of the language model.

5. CONCLUSION

We introduced the first fully convolutional pipeline for speech recognition, that can directly process the raw waveform and shows state-of-the art performance on Wall Street Journal and on Librispeech among end-to-end systems. This first attempt at exploiting convolutional language models in speech recognition shows significant improvement over a 4-gram language model on both datasets. Replacing mel-filterbanks by a learnable front-end gives additional gains in performance, that appear to be more prevalent on noisy data. This suggests learning the front-end is a promising avenue for speech recognition with challenging recording conditions.

6. REFERENCES

[1] Yann Dauphin, Angela Fan, Michael Auli, and David Grangier, "Language modeling with gated convolutional networks," in *ICML*, 2017.

[2] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann Dauphin, "Convolutional sequence to sequence learning," in *ICML*, 2017.

[3] Jonas Gehring, Michael Auli, David Grangier, and Yann Dauphin, "A convolutional encoder model for neural machine translation," in *ACL*, 2017.

[4] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," in *SSW*, 2016.

[5] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *ICML*, 2014.

[6] Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH*, 2010.

[7] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang 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.

[8] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals, "Listen, attend and spell," *CoRR*, vol. abs/1508.01211, 2015.

[9] Albert Zeyer, Kazuki Irie, Ralf Schlüter, and Hermann Ney, "Improved training of end-to-end attention models for speech recognition," *arXiv preprint arXiv:1805.03294*, 2018.

[10] Alexander H. Waibel, Toshiyuki Hanazawa, Geoffrey E. Hinton, Kiyohiro Shikano, and Kevin J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 37, pp. 328–339, 1989.

[11] Ossama Abdel-Hamid, Abdel rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, pp. 1533–1545, 2014.

[12] Vitaliy Liptchinsky, Gabriel Synnaeve, and Ronan Collobert, "Letter-based speech recognition with gated convnets," *CoRR*, vol. abs/1712.09444, 2017.

[13] Ying Zhang, Mohammad Pezeshki, Philemon Brakel, Saizheng Zhang, César Laurent, Yoshua Bengio, and Aaron C. Courville, "Towards end-to-end speech recognition with deep convolutional neural networks," in *INTERSPEECH*, 2016.

[14] Neil Zeghidour, Nicolas Usunier, Iasonas Kokkinos, Thomas Schatz, Gabriel Synnaeve, and Emmanuel Dupoux, "Learning filterbanks from raw speech for phone recognition," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5509–5513, 2018.

[15] Dimitri Palaz, Mathew Magimai Doss, and Ronan Collobert, "Convolutional neural networks-based continuous speech recognition using raw speech signal," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4295–4299.

[16] Yedid Hoshen, Ron J Weiss, and Kevin W Wilson, "Speech acoustic modeling from raw multichannel waveforms," in *Proceedings of ICASSP*. IEEE, 2015.

[17] Tara N Sainath, Ron J Weiss, Andrew Senior, Kevin W Wilson, and Oriol Vinyals, "Learning the speech front-end with raw waveform cldnns," in *Interspeech*, 2015.

[18] Neil Zeghidour, Nicolas Usunier, Gabriel Synnaeve, Ronan Collobert, and Emmanuel Dupoux, "End-to-end speech recognition from the raw waveform," in *Interspeech*, 2018.

[19] D Ulyanov, A Vedaldi, and V Lempitsky, "Instance normalization: the missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2017.

[20] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," *Journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[21] Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve, "Wav2letter: an end-to-end convnet-based speech recognition system," *arXiv preprint arXiv:1609.03193*, 2016.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[23] Kyu J Han, Akshay Chandrashekaran, Jungsuk Kim, and Ian Lane, "The capio 2017 conversational speech recognition system," 2017.

[24] Hossein Hadian, Hossein Sameti, Daniel Povey, and Sanjeev Khudanpur, "End-to-end speech recognition using lattice-free mmi," in *Interspeech*, 2018.

[25] William Chan and Ian Lane, "Deep recurrent neural networks for acoustic modelling," *arXiv preprint arXiv:1504.01482*, 2015.

[26] Douglas B Paul and Janet M Baker, "The design for the wall street journal-based csr corpus," in *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1992, pp. 357–362.

[27] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5206–5210.

[28] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*, 2013, pp. 1139–1147.

[29] Rico Sennrich, Barry Haddow, and Alexandra Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.