End-to-End Speech Recognition From the Raw Waveform

Neil Zeghidour, Nicolas Usunier, Gabriel Synnaeve, Ronan Collobert, Emmanuel Dupoux

End-to-End Speech Recognition From the Raw Waveform

Neil Zeghidour[1,2]*, Nicolas Usunier*[1]*, Gabriel Synnaeve*[1]*, Ronan Collobert*[1]*, Emmanuel Dupoux*[2]

[1] Facebook A.I. Research, Paris, France; New York & Menlo Park, USA
[2] CoML, ENS/CNRS/EHESS/INRIA/PSL Research University, Paris, France
{neilz, usunier, gab, locronan}@fb.com, emmanuel.dupoux@gmail.com

Abstract

State-of-the-art speech recognition systems rely on fixed, hand-crafted features such as mel-filterbanks to preprocess the waveform before the training pipeline. In this paper, we study end-to-end systems trained directly from the raw waveform, building on two alternatives for trainable replacements of mel-filterbanks that use a convolutional architecture. The first one is inspired by gammatone filterbanks (Hoshen et al., 2015; Sainath et al, 2015), and the second one by the scattering transform (Zeghidour et al., 2017). We propose two modifications to these architectures and systematically compare them to mel-filterbanks, on the Wall Street Journal dataset. The first modification is the addition of an instance normalization layer, which greatly improves on the gammatone-based trainable filterbanks and speeds up the training of the scattering-based filterbanks. The second one relates to the low-pass filter used in these approaches. These modifications consistently improve performances for both approaches, and remove the need for a careful initialization in scattering-based trainable filterbanks. In particular, we show a consistent improvement in word error rate of the trainable filterbanks relatively to comparable mel-filterbanks. It is the first time end-to-end models trained from the raw signal significantly outperform mel-filterbanks on a large vocabulary task under clean recording conditions.

Index Terms: speech recognition, waveform, deep, end-to-end, scattering, gammatones

1. Introduction

State-of-the-art speech recognition systems rapidly shift from the paradigm of composite subsystems trained or designed independently to the paradigm of end-to-end training. While most of the work in this direction has been devoted to learning the acoustic model directly from sequences of phonemes or characters without intermediate alignment step or phone-state/senome induction, the other end of the pipeline model – namely, learning directly from the waveform rather than from speech features such as mel-filterbanks or MFCC – has recently received attention [1, 2, 3, 4, 5, 6, 7, 8], but the performances on the master task of speech recognition still seem to be lagging behind those of models trained on speech features [9, 10].

Yet, promising results have already been obtained by learning the front-end of speech recognition systems. We focus the discussion on trainable components that can be plugged in as replacement of mel-filterbanks without modification of the acoustic model. The approach inspired by gammatone filterbanks of Hoshen et al. and Sainath et al. [3, 4] achieved similar or better results than comparable mel-filterbanks on multichannel speech recognition and on far-field/noisy recording conditions. More recently, Zeghidour et al. [8] proposed an alternative learnable architecture based on a convolutional architecture that computes a scattering transform and can be initialized as an approximation of mel-filterbanks, and obtained promising results on end-to-end phone recognition on TIMIT. However, these approaches have not been proven to improve on speech features on large-scale, end-to-end speech recognition in clean recording conditions on English – admittedly one of the tasks for which mel-filterbanks have been the most extensively tuned.

We present a systematic comparison of the two previous architectures of learnable filterbanks, which we will (coarsely) refer to as gammatone-based and scattering-based, and evaluate them against mel-filterbanks within an end-to-end training pipeline on letter error rate and word error rate on the Wall Street Journal dataset. Our main contributions and results are the following:

1. A mean-variance normalization layer on top of the log non-linearity of learnable filterbanks appears to be critical for the efficient learning of the gammatone-based architecture, and makes the training of the scattering-based architecture faster;
2. The low-pass filter previously used in the scattering-based learnable filterbanks stabilizes the training of gammatone filterbanks, compared to the max-pooling that was originally proposed [3, 4];
3. For scattering-based trainable filterbanks, keeping the low-pass filter fixed during training allows to efficiently learn the filters from a random initialization, whereas the results of [8] with random initialization of both the filters and the low-pass filter showed poor performances compared to a suitable initialization;
4. Both trainable filterbanks improve against the mel-filterbanks baseline on word error rate on the Wall Street Journal dataset, in similar conditions (same number of filters, same end-to-end training convolutional architecture). This is the first time learnable filterbanks improve against a strong mel-filterbanks baseline on a large vocabulary, speech recognition task under clean recording conditions.

The next section describes the learnable filterbanks architectures. Then, we present the end-to-end convolutional architecture used to perform the comparisons, and analyze the results of our comparative studies.

2. Learning filterbanks from raw speech

The two approaches that we consider for learning filterbanks from the raw waveform can be used as direct replacement for mel-filterbanks in any end-to-end learning pipeline: they are convolutional architectures that take the raw waveform as input and output 40 channels every 10ms. As such, they can directly be compared with standard mel-filterbanks, simply by changing the features stage of a neural-network-based acoustic model. The filters are then nothing more than an additional layer to the neural network and are learnt by backpropagation with the rest of the acoustic model.

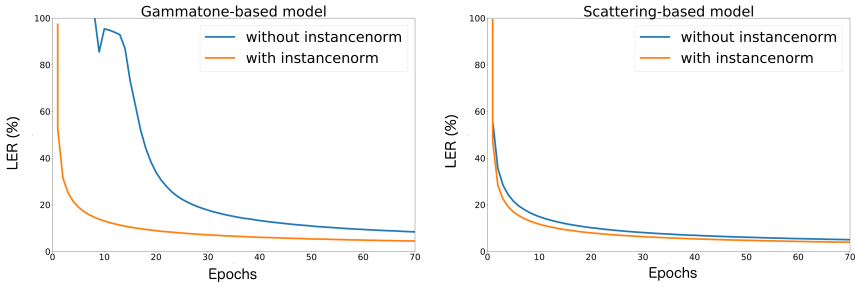

Figure 1: *Training Letter Error Rate (LER) for a gammatone-based architecture (left) and a scattering-based architecture (right), with and without instance normalization.*

The first architecture we consider is inspired by [3, 4], the second one is taken from [8]. They are described in Table 1.

In both architectures, a convolutional layer with window length 25ms (to match the standard frame size used in mel-filterbanks) is applied with a stride of 1 sample, and is followed by a nonlinearity to give 40 output channels for each sample. Then, a pooling operator of width 25ms with a stride of 10ms performs low-pass filtering and decimation. Finally, a log non-linearity reproduces the dynamic range compression of log mel-filterbanks. The parameters to be learnt are the convolution filters, and possibly the weights of the low-pass filters.

The two architectures differ by the choices of each layer of computation. Hoshen et al. and Sainath et al. use 40 real-valued filters with ReLU non-linearity, and rely on gammatones as filter values to approximate mel-filterbanks [3, 4]. In their work, they use a max-pooling operator for low-pass filtering. In contrast, Zeghidour et al. [8] use 40 complex-valued filters with a square modulus operator as non-linearity. Low-pass filtering is then performed by multiplying each output channel by a squared Hanning window so that, when using suitable Gabor wavelets as convolution filters, the architecture closely approximates mel-filterbanks computed on the power spectrum [11].

The number of filters (40), the convolution and pooling width of 25ms, as well as the decimation of 10ms are not necessarily the optimal parameters of either trainable architecture, but these are the standard settings of mel-filterbanks (and likely the best settings for these features on standard speech recognition datasets). We keep these values fixed for the trainable architectures, so that the comparison to mel-filterbanks is carried out in the setting most favorable for the non-learnable baseline.

In the next subsections, we describe the improvements we propose for these architectures: the low-pass filter and the addition of instance normalization.

2.1. Low-pass filtering

The original papers describing the gammatone-based trainable filterbanks used max-pooling as low-pass filter, whereas the scattering-based approach uses a squared Hanning window per channel. To make sure the low-pass filter is not responsible for notable differences between the two approaches we experiment with the squared Hanning window on both architectures. For both architectures, we also propose to keep this low-pass filter fixed while learning the convolution filter weights, a setting that was not explored by Zeghidour et al. [8], who learnt the low-pass filter weights when randomly initializing the convolutions.

	SCATTERING	GAMMATONES
Conv[1] *(#in-#out-width-stride)*	1-80-400-1	1-40-400-1
non-linearity	sq. L2-Pooling	ReLU
low-pass filter *(wdth=400, strd=160)*	sq. Hanning	max-pooling or sq. Hanning
log-compression[2]	$\log(1+\text{abs}(.))$	$\log(0.01+\text{abs}(.))$
normalization	mean-var. per-channel per-sentence	

Table 1: *Architectures of the two trainable filterbanks. Values of width and strides are given to match the standard settings of mel-filterbanks for waveform sampled at 16kHz.*

2.2. Instance normalization

More importantly, we noticed that a per-channel per-sentence mean-variance normalization after log-compression is important for the baseline mel-filterbanks. Consequently, we propose to add a mean-variance normalization layer on both trainable architectures, performed for each of the 40 channels independently on each sentence. Coincidently, this corresponds to an instance normalization layer [12], which has been shown to stabilize training in other deep learning contexts.

3. Experimental setup

The experiments compare different versions of the trainable architectures against log mel-filterbanks on a single deep convolutional network architecture for the acoustic model. The experiments are carried out on the open vocabulary task of the Wall Street Journal dataset [13], using the subset si284 for training, nov93-dev for validation, and nov92-eval for testing. Training is performed end-to-end on letters. We evaluate in both letter and word error rates. All our experiments use the open source code of wav2letter [14]. In the next subsections, we describe the model, the different variants we tested and the hyperparameters.

[1]The convolution for the scattering-based architecture uses 80-real valued output channels and squared L2-pooling on the feature dimension to emulate a complex-valued convolution with 40 filters followed by a squared modulus operator. Thus, after the nonlinearity, both architectures have 40 filters.

[2][8] use 1 to prevent $\log(0)$ and [3, 4] use 0.01. We kept the values initially used by the authors of the respective papers and did not try alternatives. We believe it has little impact on the final performance.

3.1. Acoustic model

Taking either log mel-filterbanks or trainable filterbanks, the acoustic model is a convolutional network with gated linear units (GLU) [15] trained to predict sequences of letters, following [16]. The model is a smaller version of the convolutional network used in [16] since they train on the larger LibriSpeech dataset. Using the syntax C-input channels-output channels-width, the architecture we use has the structure C-40-200-13/C-100-200-3/C-100-200-4/C-100-250-5/ C-125-250-6/C-125-300-7/C-150-350-8/C-175-400-9/ C-200-450-10/C-225-500-11/C-250-500-12/C-250-500-13/ C-250-600-14/C-300-600-15/C-300-750-21/C-375-1000-1. All convolutions have stride 1. The number of input channels of the $n+1$th convolution is half the size of the output of the n-th convolution because of the GLU. There are GLU layers with a dropout [17] of 0.25 after each convolution layer. There is an additional linear layer to predict the final letter probabilities. When predicting letters, the training and decoding are performed as in [16]. When predicting words, we use a 4-gram language model trained on the standard LM data of WSJ [13] and perform beam search decoding, as in [16].

3.2. Variants

We compare the two architectures of trainable filterbanks along different axes: how to initialize the convolutions of the trainable filterbanks, the low-pass filter, and instance normalization.

3.2.1. Gammatone-based architecture

Initialization of the convolution weights random (rand), or with gammatone filters (gamm) that match the impulse response of a reference open source implementation of gammatones [18];

Low-pass filter max-pooling as in [3], or the squared Hanning window (Han-fixed).

3.2.2. Scattering-based architecture

Initialization of the convolution weights random (rand), or Gabor filters (scatt) as described in Section 2.2 of [8];

Low-pass filter the squared Hanning window (Han-fixed), or a low-pass filter of same width and stride initialized with the weights of the squared Hanning window but the weights are then learnt by backpropagation (Han-learnt).

3.3. Hyperparameters and training

For models trained on the raw waveform, the signal was first normalized with mean/variance normalization by sequence. The network is trained with stochastic gradient descent and weight normalization [19] for all convolutional layers except the front-ends. First, 80 epochs are performed with a learning rate of 1.4, then training is resumed for 80 additional epochs with a learning rate of 0.1. These hyperparameters were chosen from preliminary experiments as they seemed to work well for all architectures. Additional hyperparameters are the momentum and the learning rate for the training criterion, respectively chosen in $\{0, 0.9\}$ and $\{0.001, 0.0001\}$ [14, 16].

For Letter Error Rate (LER) evaluations, the hyperparameters are selected using the LER on the validation set, validating every epoch. For Word Error Rate (WER) evaluations, the hyperparameters are chosen on the validation set using the WER, validating every 10 epochs. The model selected on LER is also included for validation. The additional hyperparameters are the weight of the language model and the weight of word insertion penalty (see [16] for details). We set them between 5 and 8 by steps of 0.5, and between -2 and 0.5 by steps of 0.1, respectively. For hyperparameter selection, the beam size of the decoder is set to $2,500$; the final performances are computed with the selected hyperparameters but using a beam size of $25,000$.

MODEL			NOV93-DEV LER	NOV93-DEV WER	NOV92-EVAL LER	NOV92-EVAL WER
SOTA – speech features						
Deep Speech 2 [20]			–	–	–	3.1
– (+ additional data)						
RNN-WER - tri. LM [21]			–	–	–	8.2
RNN - WSFT decoding [22]			–	–	–	7.3
Seq2Seq + tri. LM [23]			–	9.7	–	6.7
Multi-task CTC/att [24]			11.3	–	7.3	–
Att + RL [25]			–	–	6.1	
SOTA – waveform						
Att Wav2Text (+transfer) [26]			–	–	6.5	–
gamm (learnt)/gamm/max-pool			8.9	12.9	6.4	8.8
– (without inst. norm.)						
FRONT END	FILTER INIT	LOW-PASS	NOV93-DEV LER	NOV93-DEV WER	NOV92-EVAL LER	NOV92-EVAL WER
mel-fbanks			6.9	9.5	4.9	6.6
gamm (learnt)	gamm	Han-fixed	6.9	9.1	4.9	5.9
		max-pool	7.2	9.3	4.9	6.0
	rand	Han-fixed	7	8.9	4.9	5.9
		max-pool	7.2	9.2	5.1	6.3
scatt (learnt)	scatt	Han-fixed	6.7	8.3	4.6	6.1
		Han-learnt	6.7	8.9	4.5	6.3
	rand	Han-fixed	6.8	8.5	4.7	5.7
		Han-learnt	6.9	8.9	4.9	5.8

Table 2: *Results on the open vocabulary task of the WSJ dataset.* (i) *SOTA – speech features: for state-of-the-art and representative baselines using speech features (mel-filterbanks, spectrograms or MFCC),* (ii) *SOTA-waveform: state-of-the-art from the raw waveform, including our own implementation of vanilla gammatone filterbanks without instance normalization, and* (iii) *our baseline and the different variants of the trainable filterbanks (with instance normalization) studied in this paper.*

4. Experiments

4.1. Baseline results

Table 2 contains our results together with end-to-end baselines from the literature. [20] is the current state-of-the-art on the WSJ dataset; it is given as a topline but uses much more training data ($\sim 12,000h$ of speech) so the results are not comparable. [21, 22, 23, 24] are representative results in terms of WER and LER from the literature of end-to-end models trained on speech features from 2014-2017, in chronological order. [25] and [26] are the current state-of-the-art in LER on speech features and

MODEL	PRE-EMP	NOV93-DEV		NOV92-EVAL	
		LER	WER	LER	WER
gamm (learnt)	no pre-emp	6.9	9.1	4.9	5.9
	pre-emp	6.8	9	4.7	5.7
scatt (learnt)	no pre-emp	6.7	8.3	4.6	6.1
	pre-emp	6.5	8.7	4.5	5.7

Table 3: *Comparison of models trained with or without a learnable pre-emphasis layer. All models are initialized either with the scattering or gammatone initialization, and the pooling function is a fixed squared Hanning window.*

from the waveform respectively. These comparisons validate our baseline model trained on mel-filterbanks as a strong baseline in light of recent results, as it outperforms the state-of-the-art in LER by a significant margin (4.9% vs 6.1% for [25]), and achieves a test WER of 6.6%, better than all other end-to-end baselines ([27] and [7] report WER that are below our 6.6% but are on easier closed vocabulary tasks).

4.2. Instance normalization

As described in Section 2.2, we evaluate the integration of instance normalization after the log-compression in the trainable filterbanks, which was not used in previous work [3, 4, 7, 8] but is used in our baseline. Figure 1 shows training LER as a function of the number of epochs for scattering-based and gammatone-based filterbanks models, with and without instance normalization. We can see that this normalization drastically improves the training stability of the gammatone-based model, while it moderately improves the scattering-based model. We observed a positive impact of instance normalization in all settings, and so only report as a reference the results of our implementation of a vanilla gammatone-based trainable filterbanks following [3, 4]. Comparing gammatone (learnt)/gamm/max-pool without instance norm (under SOTA – waveform) to the results of gammatone (learnt)/gamm/max-pool in Table 2, we see a significant improvements of both LER and WER due to instance normalization, with an absolute reduction in LER and WER of 1.5% and 2.8% respectively.

4.3. Impact of the low-pass filter

For low-pass filtering, we first compare the Han-fixed setting to max-pooling for gammatone-based filterbanks (as max-pooling was previously used in [3, 4]), and to Han-learnt for scattering, all with instance normalization. The tendency is that the Han-fixed setting consistently improves the results in LER and WER of both trainable filterbanks. More importantly, using either an Han-fixed or Han-learnt filter when learning scattering-based filterbanks from a random initialization removes the gap in performance with the Gabor wavelet initialization that was observed in [8] where the lowpass filter was also initialized randomly. This is an important result since carefully initializing the convolutional filters is both technically non-trivial, and also relies on the prior knowledge of mel-filterbanks. We believe the ability to use random initialization is an important first step for more extensive tuning of trainable filterbanks (e.g., trying different numbers of filters, decimation or convolution width).

Compared to the literature, replacing the max-pooling by a low-pass filter and adding an instance normalization layer leads to a 23% relative improvement in LER and a 33% relative improvement in WER on nov92-eval on the gammatone-based trainable filterbanks, a significant improvement compared to the existing approach [3, 4]. Our models trained on the waveform also exhibit a gain in performance in LER of $22 - 31\%$ relative compared to the state-of-the-art end-to-end model trained on the waveform with its first 6 layers being pre-trained for mel-filterbanks reconstruction [26], and outperform various end-to-end models trained on speech features, both in LER [24, 25] and WER [21, 22, 23].

4.4. Trainable filterbanks vs mel-filterbanks

Comparing both trainable filterbanks with instance normalization to the log mel-filterbanks baseline, we observe that the performances of the Han-fixed settings and of the mel-filterbanks are comparable in terms of LER. However, we observe a consistent improvement in terms of WER of all trainable filterbanks. To the best of our knowledge, this is the first time a significant improvement in terms of WER relatively to comparable mel-filterbanks has been shown on a large vocabulary task under clean recording conditions. Some improvements on the clean test of the Switchboard dataset have previously been observed by [7], but their comparison point is MFCC rather than mel-filterbanks and the number of filters of the trainable architecture differs from their MFCC baseline.

4.5. Adding a learnable pre-emphasis layer

The first step in the computation of mel-filterbanks is typically the application of a pre-emphasis layer to the raw signal. Pre-emphasis is a convolution with a first-order high-pass filter of the form $y[n] = x[n] - \alpha x[n-1]$, with α typically equal to 0.97. This operation can be performed by a convolutional layer of kernel size 2 and stride 1, that can be plugged below time-domain filterbanks, initialized with weights $[-0.97 \quad 1]$, then learned with the network. In Table 3, we compare the performance of identical models (all using a fixed Hanning window, and a gammatone or scattering initialization) with and without pre-emphasis. We observe a gain on both LER and WER (except on nov93-dev WER/scatt) when using pre-emphasis.

5. Conclusion

This paper presents a systematic study of two approaches for trainable filterbanks, which clarifies good practices and identifies better architectures to learn from raw speech. Our results show that adding an instance normalization layer on top of the trainable filterbanks is critical for learning gammatone-based architectures, and speeds up learning of scattering-based architectures. Second, the use of a fixed squared Hanning window as low-pass filter is critical to learn the scattering-based filterbanks from random initialization of the filters, and improves on max-pooling for gammatone-based filterbanks. With these two improvements, we observe a consistent reduction of WER against comparable mel-filterbanks on the open vocabulary task of the WSJ dataset, in the setting of speech recognition under clean recording condition – most likely the setting on which mel-filterbanks have been the most heavily tuned.

6. Acknowledgements

This research was partially funded by the European Research Council (ERC-2011-AdG-295810 BOOTPHON), the Agence Nationale pour la Recherche (ANR-10-LABX-0087 IEC, ANR-10-IDEX-0001-02 PSL*).

7. References

[1] D. Palaz, R. Collobert, and M. M. Doss, "End-to-end phoneme sequence recognition using convolutional neural networks," *arXiv preprint arXiv:1312.2137*, 2013.

[2] Z. Tüske, P. Golik, R. Schlüter, and H. Ney, "Acoustic modeling with deep neural networks using raw time signal for lvcsr," in *Interspeech*, 2014.

[3] Y. Hoshen, R. J. Weiss, and K. W. Wilson, "Speech acoustic modeling from raw multichannel waveforms," in *Proceedings of ICASSP*. IEEE, 2015.

[4] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform cldnns," in *Interspeech*, 2015.

[5] Z. Zhu, J. H. Engel, and A. Hannun, "Learning multiscale features directly from waveforms," *arXiv preprint arXiv:1603.09509*, 2016.

[6] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[7] P. Ghahremani, V. Manohar, D. Povey, and S. Khudanpur, "Acoustic modelling from the signal domain using cnns." 2016.

[8] N. Zeghidour, N. Usunier, I. Kokkinos, T. Schatz, G. Synnaeve, and E. Dupoux, "Learning filterbanks from raw speech for phone recognition," *arXiv preprint arXiv:1711.01161*, 2017.

[9] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, "The microsoft 2017 conversational speech recognition system," *CoRR*, vol. abs/1708.06073, 2017. [Online]. Available: http://arxiv.org/abs/1708.06073

[10] K. J. Han, A. Chandrashekaran, J. Kim, and I. Lane, "The capio 2017 conversational speech recognition system," *arXiv preprint arXiv:1801.00059*, 2017.

[11] J. Andén and S. Mallat, "Deep scattering spectrum," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4114–4128, 2014.

[12] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[13] D. B. Paul and J. 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.

[14] R. Collobert, C. Puhrsch, and G. Synnaeve, "Wav2letter: an end-to-end convnet-based speech recognition system," *arXiv preprint arXiv:1609.03193*, 2016.

[15] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," *arXiv preprint arXiv:1612.08083*, 2016.

[16] V. Liptchinsky, G. Synnaeve, and R. Collobert, "Letter-based speech recognition with gated convnets," *CoRR*, vol. abs/1712.09444, 2017. [Online]. Available: http://arxiv.org/abs/1712.09444

[17] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[18] "Gammatone-based spectrograms, using gammatone filterbanks or fourier transform weightings." https://github.com/detly/gammatone, accessed: 2018-03-19.

[19] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 901–909.

[20] 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.

[21] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International Conference on Machine Learning*, 2014, pp. 1764–1772.

[22] 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 Workshop (ASRU)*, 2015.

[23] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," *arXiv:1612.02695*, 2016.

[24] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4835–4839.

[25] A. Tjandra, S. Sakti, and S. Nakamura, "Sequence-to-sequence asr optimization via reinforcement learning," *arXiv preprint arXiv:1710.10774*, 2017.

[26] ——, "Attention-based wav2text with feature transfer learning," *arXiv preprint arXiv:1709.07814*, 2017.

[27] Y. Zhou, C. Xiong, and R. Socher, "Improving end-to-end speech recognition with policy learning," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.