End-to-End Speech Recognition From the Raw Waveform

Neil Zeghidour\textsuperscript{1,2}, Nicolas Usunier\textsuperscript{1}, Gabriel Synnaeve\textsuperscript{1}, Ronan Collobert\textsuperscript{1}, Emmanuel Dupoux\textsuperscript{2}

\textsuperscript{1} Facebook A.I. Research, Paris, France; New York & Menlo Park, USA
\textsuperscript{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, handcrafted 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 (Zeghidou 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 \cite{hoshen2015end, sainath2015end, zeghidour2017scattering}. 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. \cite{hoshen2015end, sainath2015end} 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. \cite{zeghidour2017scattering} 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 \cite{zeghidour2017scattering};
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 \cite{zeghidour2017scattering} 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.
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 non-linearity 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 each 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.

In 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.

#### 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.

---

1The 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 non-linearity, 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) 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-250-500-8/C-175-400-9/C-200-450-10/C-225-500-11/C-250-500-12/C-250-600-13/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 + 1th convolution is half the size of the output of the n-th convolution because of the GLU. There are GLU layers with a dropout 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 with 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, 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 [15] 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.

### 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 (~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

<table>
<thead>
<tr>
<th>Model</th>
<th>NOV93-DEV</th>
<th>NOV92-EVAL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LER</strong></td>
<td><strong>WER</strong></td>
<td><strong>LER</strong></td>
</tr>
<tr>
<td>Deep Speech 2 [20]</td>
<td>−</td>
<td>−</td>
</tr>
<tr>
<td>RNN-WER - tri. LM [21]</td>
<td>−</td>
<td>−</td>
</tr>
<tr>
<td>RNN - WSFT decoding [22]</td>
<td>−</td>
<td>−</td>
</tr>
<tr>
<td>Seq2Seq + tri. LM [23]</td>
<td>−</td>
<td>9.7</td>
</tr>
<tr>
<td>Multi-task CTC/att [24]</td>
<td>11.3</td>
<td>−</td>
</tr>
<tr>
<td>Att + RL [25]</td>
<td>−</td>
<td>−</td>
</tr>
<tr>
<td><strong>SOTA – speech features</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Att Wav2Text (+transfer) [26] −</em></td>
<td>−</td>
<td>6.5</td>
</tr>
<tr>
<td>gamm (learnt)/gamm/max-pool</td>
<td>8.9</td>
<td>12.9</td>
</tr>
<tr>
<td>− (without inst. norm.)</td>
<td>8.8</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>FRONT FILTER LOW-PASS</strong></th>
<th><strong>NOV93-DEV</strong></th>
<th><strong>NOV92-EVAL</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LER</strong></td>
<td><strong>WER</strong></td>
<td><strong>LER</strong></td>
</tr>
<tr>
<td>mel-fbanks</td>
<td>6.9</td>
<td>9.5</td>
</tr>
<tr>
<td>gamm (learnt)</td>
<td>Han-fixed</td>
<td>6.9</td>
</tr>
<tr>
<td>gamm (rand)</td>
<td>max-pool</td>
<td>7.2</td>
</tr>
<tr>
<td>scatt (learnt)</td>
<td>Han-fixed</td>
<td>7</td>
</tr>
<tr>
<td>scatt (rand)</td>
<td>max-pool</td>
<td>7.2</td>
</tr>
<tr>
<td><strong>SOTA – waveform</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Att Wav2Text (+transfer) [26] −</em></td>
<td>−</td>
<td>6.5</td>
</tr>
<tr>
<td>gamm (learnt)/gamm/max-pool</td>
<td>8.9</td>
<td>12.9</td>
</tr>
<tr>
<td>− (without inst. norm.)</td>
<td>8.8</td>
<td></td>
</tr>
</tbody>
</table>

#### 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 filter-banks (with instance normalization) studied in this paper.
Our baseline model trained on mel-filterbanks as a strong base-

function is a fixed squared Hanning window. These comparisons validate

able pre-emphasis layer. All models are initialized either with

Table 3: Comparison of models trained with or without a learn-

able pre-emphasis layer. All models are initialized either with

from the waveform respectively. These comparisons validate

as a strong base-

Table 3: Comparison of models trained with or without a learn-

able pre-emphasis layer. All models are initialized either with


different numbers of filters, decimation or convolution width).

Comparing both trainable filterbanks with instance normaliza-

tion to the log mel-filterbanks baseline, we observe that the per-

ance of identical models (all using a fixed Hanning window,

show that adding an instance normalization layer on top of the

Trainable filterbanks vs mel-filterbanks

Comparing both trainable filterbanks with instance normaliza-

tion to the log mel-filterbanks baseline, we observe that the per-

ance of identical models (all using a fixed Hanning window,

instance pre-emp

max-pool in Table 2, we see a significant improvements of both LER

and WER due to instance normalization, with an absolute re-

duction in LER and WER of 1.5% and 2.8% respectively.

Impact of the low-pass filter

For low-pass filtering, we first compare the Han-fixed setting to

learning scattering-based 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.

Instance normalization

As described in Section 2, we compare the integration of

of kernel size 2 and stride 1, that can be plugged below time-
domain filterbanks, initialized with weights $[-0.97, 1]$, then

network. In Table 3, we compare the perform-

ance 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 (ex-

cept on nov93-dev WER/scatt) when using pre-emphasis.

Conclusion

This paper presents a systematic study of two approaches for

trainable filterbanks, which clarifies good practices and identi-

fies better architectures to learn from raw speech. Our results
display 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 archi-

tectures. Second, the use of a fixed squared Hanning window

as low-pass filter is critical to learn the scattering-based filter-

banks 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.

Acknowledgements

This research was partially funded by the European Research

Council (ERC-2011-AdG-295810 BOOPTHON), the Agence

Nationale pour la Recherche (ANR-10-LABX-0087 IEC, ANR-

10-IDEX-0001-02 PSL*).
7. References


