ACOUSTIC MODELING OF SPEECH WAVEFORM BASED ON MULTI-RESOLUTION, NEURAL NETWORK SIGNAL PROCESSING

Zoltán Tüske, Ralf Schlüter, Hermann Ney

Human Language Technology and Pattern Recognition, Computer Science Department,
RWTH Aachen University, 52056 Aachen, Germany

{tuske, schlueter, ney}@cs.rwth-aachen.de

ABSTRACT

Recently, several papers have demonstrated that neural networks (NN) are able to perform the feature extraction as part of the acoustic model. Motivated by the Gammatone feature extraction pipeline, in this paper we extend the waveform based NN model by a second level of time-convolutional element. The proposed extension generalizes the envelope extraction block, and allows the model to learn multi-resolutional representations. Automatic speech recognition (ASR) experiments show significant word error rate reduction over our previous best acoustic model trained in the signal domain directly. Although we use only 250 hours of speech, the data-driven NN based speech signal processing performs nearly equally to traditional handcrafted feature extractors. In additional experiments, we also test segment-level feature normalization techniques on NN derived features, which improve the results further. However, the porting of speech representations derived by a feed-forward NN to a LSTM back-end model indicates much less robustness of the NN front-end compared to the standard feature extractors. Analysis of the weights in the proposed new layer reveals that the NN prefers both multi-resolution and modulation spectrum representations.

Index Terms— ASR, neural network, time-signal, waveform, modulation spectrum, feature representation

1. INTRODUCTION

Before the recent advance of deep neural network (DNN) based models, manually defined feature extraction methods had a crucial role in developing high performing, noise-robust acoustic models (AM) for ASR, e.g. [1, 2, 3]. These features were designed according to the physiology of human hearing [4], psychoacoustical measurements [5], or to minimize errors of statistical models [6]. From the machine learning viewpoint, however, the feature extraction should be part of the model, and be derived according to the training criterion. There have been several attempts to let the model optimize some of the extraction steps, e.g. learning dynamical features [7], or Mel-like filters [8, 9]. As has also been shown in recent studies, powerful deep classifiers are indeed able to learn the complete feature extraction automatically [10]. But the model usually needs to be trained on several thousand of hours of data to perform as good as with cepstral features [11]. Imitating steps from the standard feature extraction, e.g. sharing the time-frequency decomposition layer over time, can clearly mitigate the data hunger of AM trained directly on speech signal [12, 13, 14]. Analyzing the learned weights, we showed previously that the NN learns band-pass filters and the first layer performs time-frequency decomposition [10]. It has also been revealed that the data-driven filters follow patterns which are similar to an audiological filterbank [15]. The bandwidths of the learned filters mostly varied between 100–800Hz. This result, however, indicates that a fixed rate sampling of the filter outputs might be sub-optimal. E.g. the convolutional filter outputs are usually downsampled to 10ms rate using maxout operation [11, 13, 16]. The improper downsampling of the output of wide-band filters can lead to serious aliasing which is not a reversible operation [17, 18]. Therefore, in this paper we substitute the usual max pooling layer – inserted after the time-frequency decomposition – by a second level time-convolution, which enables the network to exploit various sampling rate if necessary. We hypothesize that a representation based on multiple resolution levels can be more beneficial for NNs. The hypothesis is validated with broadcast news and conversation large vocabulary continuous speech recognition experiments. Analyzing the weights will reveal that the NN prefers to learn a modulation spectrum. To our best knowledge, this is the first time to show that a NN trained on waveforms learns such a speech representation. Segment-wise mean and variance normalization is a standard signal processing step for cepstral features. Experiments are also designed to investigate the effect of such techniques on a NN based feature extractor.

2. RELATION TO PRIOR WORK

This work is an extension of our previous work [13]. In [14] it was shown that training model on waveform can compete with standard feature extraction method using only 300 hours of speech. Our proposed trainable pooling layer can also be formulated in the network-in-network framework introduced for ASR in that paper. However, our proposed model allows easier analysis of the weights, and operates with longer overlap. Motivated by its success in computer vision field, in [19] multiscale convolutions were tested in an end-to-end approach. The method can be considered analogous to our solution. Because the authors used max pooling the size and sampling rate of the different poolings had to be predefined, but network learns such decisions automatically.

3. MULTI-RESOLUTION SIGNAL PROCESSING WITH NEURAL NETWORKS

Fig. 1 shows the proposed modification to a contemporary AM of direct waveform, which is explained in details below. The standard cepstral feature extraction start with time-frequency decomposition (TF) of the signal, using a predefined filterbank shared over time: e.g. Gammatone filterbank (GT) or short-time Fourier transformation [1, 3]. Assuming an input signal s_t the filterbank output is:

$$y_{k,t'} = s_t * h'_{k,t} \overset{\text{FIR}}{=} \sum_{\tau=0}^{N_{\text{TF}}-1} s_{t+\tau-N_{\text{TF}}+1} \cdot h_{k,\tau} \qquad (1)$$

Where we assumed that the filter output is sub-sampled, e.g. by a factor of 10, $t = 10 \cdot t'$. We also assumed that the filterbank has finite impulse response (FIR) of length N_{TF}. $h_{k,t}$ denotes the correlation filter, and corresponds to the mirrored and shifted version of $h'_{k,t}$. Sharing the filterbank over time is integrated into the NNs as a convolutional layer [12], also known as time-delay NN [20]. Sub-sampling has computational advantage, and can be applied together with FIR filters very efficiently. From the communication theory viewpoint, sampling under the Nyquist frequency is possible for bandpass signals. However, the absolute minimum sampling rate (two times the bandwidth) is only valid for specific center filter

Table 1. Baseline results with GT feature extraction pipeline in various configurations.

dim.	root	DCT	seg.-wise norm.		VTLN	WER	
			mean	var		dev	eval
50	×	×	×	×	×	13.2	17.9
	×	×	×	×		13.2	17.9
	×	×	×			**13.1**	**17.8**
	×	×				13.5	18.4
	×		×			13.3	18.1
	×					13.5	18.9
						14.2	19.6
70	×	×	×			**13.1**	**17.8**

frequency and bandwidth combinations [18]. Our choice of downsampling rate is based on both memory constraints and the observation that final TF filters can have 800Hz bandwidth. Oversampling of narrow bandpass output results in unnecessary computation but causes no information loss. It also leads to reliable envelope extraction from real-valued signal, and improves WER [19]. During the NN training the center frequencies and bandwidths are changing continuously. They can easier remain in a valid sampling region if the filters are oversampled [18]. In the following signal processing step the amplitude spectrum is extracted from the down-sampled TF filter outputs by envelope detection. Working with real-valued signals, this step applies either half- or full-wave rectification to the input, followed by low-pass filtering to smooth the final result:

$$x_{i,k,t''} \overset{\text{FIR}}{=} f_2\left(\sum_{\tau=0}^{N_{\text{ENV}}-1} f_1\left(y_{k,t'+\tau-N_{\text{ENV}}+1}\right)\cdot l_{i,\tau}\right) \qquad (2)$$

Here we assumed FIR low-pass filters ($l_{i,t'}$) with a response of N_{ENV} samples. The base-band signal can be downsampled according to the bandwidth of the TF filter, usually f_s = 100Hz. The rectification is noted by f_1, and is a rectified linear units (ReLU, [21]) or absolute value function in NNs. Acoustic models trained on direct waveforms integrate the envelope detector often as non-parameterized function, e.g. max pooling [22], average [11], p-norm [14]. Since hearing works on a non-linear scale, the envelope detection step can optionally be followed by a logarithmic or root compression [1, 2, 3], or other non-linearities used in NNs (f_2). The output $x_{i,k,t''}$ can be interpreted as the critical band energies (CRBE). In Eq. 2 we introduced the parameter i, which corresponds to learning various low-pass filters. These filters are shared not only in time but also between the TF filters. This simple modification allows multi-resolutional sampling of $f_1(y_{k,t'})$ if N_{ENV} and max(i) are chosen large enough, despite each $x_{i,k,t''}$ having the same sampling rate. For instance, to obtain the usual 100Hz-sampled GT CRBE, $l_{i,t'}$ should be initialized by the Hamming window of N_{ENV}=400 (25ms), and set e.g. $t = 10 \cdot t' = 160 \cdot t''$ if the input signal (s_t) is sampled at 16kHz. To sample the filter output $y_{k,t}$ four times faster, e.g. due to a wider bandwidth, four $l_{i,t'}$ should be allocated. Each of these envelope filters should contain a 6.25ms Hamming window, but at four different positions within the 25ms analysis window. Thus, the proposed model can learn wavelet like representations [23]. Although $x_{i,k,t''}$ would be extracted from a non-orthonormal basis, due to the exhaustive combination of envelope processing and TF filters, an orthogonal subvector can be selected. During model training we let the NN decide which elements of $x_{i,k,t''}$ contain useful information.

4. EXPERIMENTAL SETUP

We evaluated our multi-resolution, NN based signal processing on an English broadcast news and conversation speech recognition task. The training data consisted of 250 hours of speech, from which 10% was selected for cross-validation. The development and evaluation sets contain 3 hours of speech each, and the results are reported in word error rate (WER). For further details about the corpus we refer to our previous works [10, 24]. The experiments were carried out with the RASR toolkit [25]. As a back-end, a hybrid 12-layer (with 2000 nodes per layer) feed-forward ReLU network was used [26, 27]. The NNs were trained by the cross-entropy criterion, using stochastic gradient descent optimization with momentum, $l2$ regularization, and discriminative pretraining [28]. The NN front-end was jointly optimized with the back-end, and both were randomly initialized. It should be noted, that lot of computation and memory can be saved in the front-end if t' and t'' are integer multiple of t, and t'' is integer multiple of t'. We used the following, synchronized settings: $160 \cdot t'' = 10 \cdot t' = t$, for s_t sampled at 16kHz. The time-frequency decomposition was performed by up to 150 filters, each having a 512-sample (32ms) response. The filterbank was operated at 10-sample shift (0.625ms), and its output was processed with a maximum of 20 different envelope extractors. The $l_{i,t'}$ filters had a maximum length of 160 samples (0.1s). The back-end concatenated 17 neighboring frames, thus, the estimation of a single posterior vector is based on a waveform snippet of maximum 4662 samples. To perform the convolution by $l_{i,t'}$ and produce X'' in a single step the network worked with nearly 500k-dimensional vectors. Although the convolutional filters did not increase the number of parameters noticeably, compared to the size of the back-end, using 20 $h_{k,t}$, 150 $l_{i,t'}$ filters, and 17-frame window results in a 51000-dimensional input vector for the DNN. Therefore, the first layer of the DNN was always low-rank factorized by a 512-dimensional linear BN [29].

5. EXPERIMENTAL RESULTS

5.1. Baselines

In the first set of experiment we tested the GT features in several settings (Table 1). Switching off the signal processing steps incrementally, significant WER degradation can be observed. Most notably when the root compression is turned off. The DCT transformation (used without dimension reduction) together with segment-wise mean normalization resulted in the best setting. Vocal tract length (VTLN [30]) or segment-level variance normalization seem unnecessary for modern ASR. Increasing the dimension, or using different filter parameters, e.g. varying the quality factor, did not result in better WER. With acoustic models of waveforms we aimed at 18.4% WER on the evaluation set, because segment-level information was not provided to the NN based front-end. Fully-connected DNN ([10]) achieved 20.5%, and our initial convolutional net ([13]) only 21.2% WER on the evaluation set. Thus, the gain measured with the best convolutional processing on small task in [13], could not be carried over to a larger setup.

5.2. Results with neural network front-ends

The next experiment is comparable to [13], and convolutional raw time-signal processing was carried out with only a single envelope extractor (Table 2). The detector was either a trainable FIR filter or a maxout layer. We set the number of TF filter to 50, f_1 to absolute value function, and f_2 to absolute value function followed by

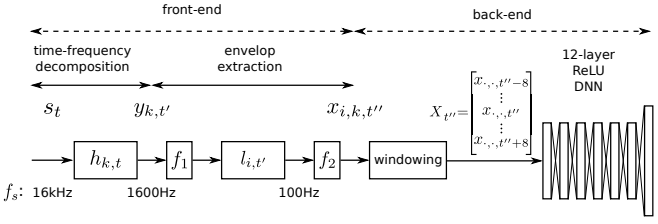

Fig. 1. *Multi-resolution and convolutional processing of speech signal with neural networks.*

Table 2. Effect of the type of a single envelope extractor.

$l_{i,t'}$ type	N_{ENV}	WER dev	WER eval
maxout	16	14.4	19.9
	25	14.3	19.8
	40	14.4	**19.7**
FIR	40	**14.1**	19.8

Table 3. Effect of envelope detector size, and non-linearities before and after the detector.

max(i)	N_{ENV}	f_1	f_2	WER dev	WER eval
5	40	ReLU	-	14.2	19.6
		ReLU	ReLU	14.2	19.5
		ReLU	ReLU+root	14.0	19.2
		Abs	-	14.2	19.6
		Abs	Abs	14.2	19.3
		Abs	Abs+root	**13.7**	**18.7**
		Abs+root	Abs	13.8	**18.7**
10	80	Abs	Abs	13.9	19.0
		Abs+root	Abs	13.8	19.0
20	160	Abs	Abs	14.3	19.3
		Abs	Abs+root	14.4	19.6

2.5th root. Using maxout envelope extractor the second rectification is superfluous. The choice of these non-linearities is based on the experimental results presented in Table 3. Taking the derivative of root function at 0 led to infinity, thus we applied gradient clipping when propagating the error signal through this layer. We also found that power of 0.4–0.6 fits better for training from scratch, than the 10th root compression of the GT pipeline [3]. As can be seen in Table 3, both types of envelope detector perform similarly. This systems already outperformed our fully-connected time-signal baseline, but only 4% relative. The improvement is mostly related to making the convolutional front-end more similar to GT pipeline, e.g. choosing root compression non-linearity.

In the following experiment we increased the number of TF filters up to 150. This decision decreased the WER on the evaluation set by about 0.5% absolute. As can be seen in Table 3, our current best result in waveform acoustic modeling is achieved by using the aforementioned non-linearities. Remarkably, our best NN front-end performs nearly equally well than the standard GT features without segment-level normalization. The relative performance difference is within 2%, and we used only 250 hours of speech data. Pushing the root function forward, we did not measure significant change in WER. Increasing the number of envelope detectors and their length (N_{ENV}), we obtained an unexpected degradation in WER, despite a significant improvement of the objective function on the validation set. This might be related to the huge dimension of $X_{t''}$ and the first weight matrix of the back-end, which we are going to address in the future by convolutional operation. We also note that with more and longer envelope detectors the effect of root compression is less pronounced. For comparison, we also trained a multi-resolution envelope detector using max pooling, similar to [19]. The first type of the detector operated on 25ms input, once per 10ms. The second detector used a 6.25ms window strided within the 25ms analysis window (N_{ENV}) in a non-overlapping way. Thus, the output ($x_{i,j,t''}$) of the multi-resolution maxout envelope extractor had the same dimension as our best trainable one. This system achieved 19.6% WER on the evaluation set.

In the third experiment, we investigated whether the back-end model could benefit from segment-level normalization even if NN front-end is used. We used the front-end of the best performing

Table 4. Effect of the segmentwise mean-and-variance normalization on matched and ported NN front-end features.

front-end type	front-end dim.	back-end	normalization mean	normalization variance	WER [%] dev	WER [%] eval
NN_1	512	MLP			13.7	18.7
			×	×	13.5	18.5
GT	70		×		**13.1**	**17.8**
NN_1	512	LSTM			14.5	18.7
			×		14.5	19.1
			×	×	13.0	16.8
NN_2	750		×		13.0	17.1
			×	×	13.9	18.1
GT	70		×	×	11.3	**14.5**
			×		11.6	14.9
					11.2	14.6

waveform model in Table 3. Features were extracted from the low rank factorized layer (NN_1). Optionally, we also extracted $x_{i,k,t''}$, (NN_2). Finally, we trained a new 12-layer feed-forward (or LSTM) back-end on these segment-level normalized features. The results are summarized in Table 4. As can be seen, segment-level signal processing technique improved the NN front-end results slightly, but overall the relative performance gap between the best handcrafted and NN derived features is 3–4%. We point out that in [14] research has been carried out to integrate long-term first and second order statistics into NN training. In the last experiment, the NN front-end features were fed into an LSTM model [31, 32]. We observed significant, 2.3% absolute WER difference on the evaluation set when standard GT and the ported data-driven features were compared. The results indicate the low-degree of robustness of transferring NN front-end between different models. Since the extraction of robust representation at the front-end level is not part of the criterion, the network seems to optimize the front-end by propagating higher level knowledge into it – like the type of the classifier – and so making it optimal only for feed-forward deep acoustic models. An additional fine-tuning or joint front-end training step might alleviate the mismatch.

6. LEARNED FILTERS

In the feed-forward structure, we could interpret the learned weights up to three layers. The analysis of the TF filterbank can be carried out similar to [10]. We observed that every frequency band was covered by various, narrow and wideband filters. This indicates the necessity of multi-resolution processing. Inspection of the envelope detector ($l_{i,t'}$) revealed that the position of the low-pass filters with diverse shapes varies within the 0.1s window (160 samples). This confirms that the NN indeed prefer multiple sub-sampling rate for the learned TF filters ($h_{k,t}$). After further analysis we also noticed a surprising fact. Figure 2 shows three examples of the twenty envelope filters learned by the network with $N_{\text{ENV}} = 160$. As can be

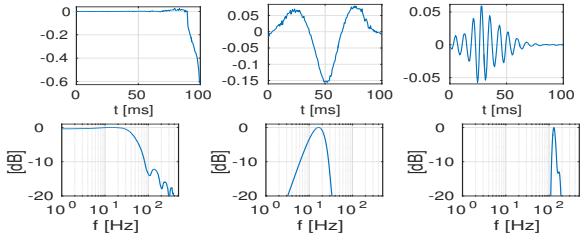

Fig. 2. *Examples of low-pass and modulation envelope filters learned from data: time-signal and its corresponding Bode magnitude plot.*

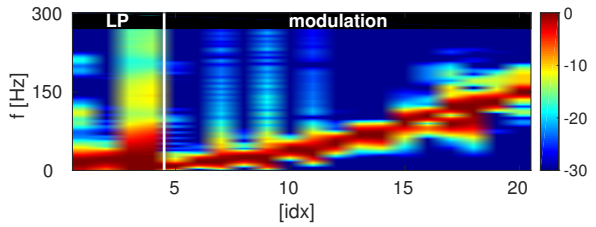

Fig. 3. *Sorted amplitude spectra of the learned low-pass (LP) and modulation filters.*

seen on the filter responses and the amplitude diagrams, the network learned not only low-pass, but also modulation filters. We separated manually the low-pass (LP) filters with noticeable (less than 6 dB attenuation) 0Hz component in the response. Then, the filters were sorted by the (highest) 3 dB cut-off frequencies in the amplitude characteristic. The magnitude spectrum of the sorted matrix of $l_{i,t'}$ is presented in Fig. 3. Previous experiments suggest that modulation spectrum between 1 and 50Hz covers the modulation content of speech signal [33]. As can be seen in Fig. 3, there is a clear frequency range of 0 to 200Hz, where all the data-driven modulation filters are placed. Thus, the network suppresses components that change too quickly to be speech, similar to RASTA filters. The higher range might be related to the rectification step (f_1). E.g. let us consider a narrow filter with 40Hz bandwidth and 1650Hz center frequency to analyze a 1650Hz signal amplitude modulated by a 10Hz sinusoid. After the sub-sampling from 16kHz by a factor of 10 the content of the filter can also be found around 50Hz. Passing through the full-wave rectification the modulation content appears around 0, 100, 200Hz, etc. Hence, the network could focus instead of the base-band also on replicas at higher frequencies. Direct access to the envelope by complex filter ([34, 35]) or quadratic non-linearity might mitigate this effect, but is out of the scope of this study. It should be also noted that an envelope detector length of 160 samples (0.1s) limits the resolution of the modulation spectrum to 10Hz. Larger window would be more comparable to the MRASTA setting of [33], and is part of our ongoing work.

Sorting the learned filterbanks $h_{k,t}$ and $l_{i,t'}$ by their estimated center frequencies, neural spectrogram and modulation spectrums can also be extracted from an input signal. Examples can be seen in Fig. 4. The first layer of the back-end can be also analyzed, simlar to [13]. From each row of the weight matrix we could select those weights which are associated to a specific envelope detector. The selected weights can be reshaped to a two-dimensional (2D) time-frequency patch. Plotting those patches, it is possible to gain some insight which patterns the NN prefers for the various spectral representations. Such plots can be seen in Fig. 5. On the data-driven filterbank output in 17-frame context, mostly complex 2D-correlation filters were learned (column four). However, we could also identify localized Gabor filters with many different shapes and directions (column two). Some patches show sensitivity only along the frequency axis (column one), and some others mostly along the time axis (column three). This is similar to cells in the auditory cortex which are also tuned to localize spectro-temporal modulations. These observations were roughly independent on which representations we analyzed (handcrafted GT CRBE, low-passed or band-passed NN spectrograms) However, we counted much more localized 2D filters on the GT spectrogram.

7. CONCLUSIONS

We presented an extension to the convolutional waveform acoustic model. As has been shown, the introduced second level convolution can also be interpreted as a trainable envelope extraction step. We demonstrated that the proposed model is ideally suited to extract various speech representations: Gammatone features, wavelet like time-frequency decomposition, or modulation spectrum, etc. With properly chosen parameter settings, we could reduce the speech recognition errors significantly by 2% absolute (10% relative) compared to our previous best results. Without any signal processing and using only 250 hours of speech, our waveform model can perform almost as good as a comparable model of handcrafted cepstral features. Additional experiments demonstrated that the NN front-end shows low degree of robustness when ported between different back-end models. Analysis of the neural network parameters revealed that besides the multi-resolutional representation the network also learns modulation spectrum. In the future, we plan to carry out further optimization of the proposed structure, and to investigate whether our approach is also beneficial for the recurrent acoustic models.

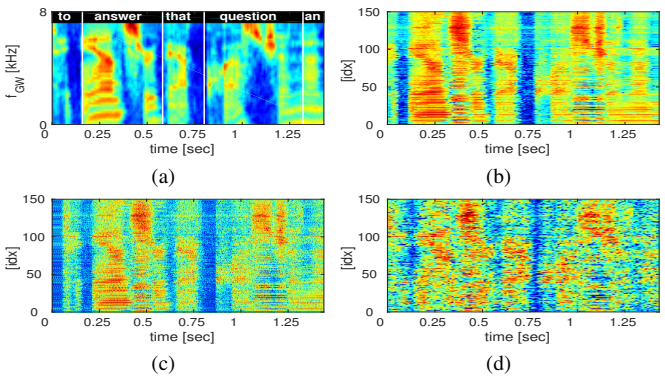

Fig. 4. *(a) GT spectrogram, f_s=100Hz; (b) NN spectrogram ($y_{k,t'}$), f_s=1600Hz; (c) low-passed NN spectrogram, f_s=100Hz; (d) 40Hz band-passed NN spectrogram spectrum, f_s=100Hz.*

8. ACKNOWLEDGEMENTS

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 694537). The work reflects only the authors' views and the European Research Council Executive Agency is not responsible for any use that may be made of the information it contains. This work has also been supported by European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 644283, and by Deutsche Forschungsgemeinschaft (DFG) under contracts no. Schl2043/11-1.

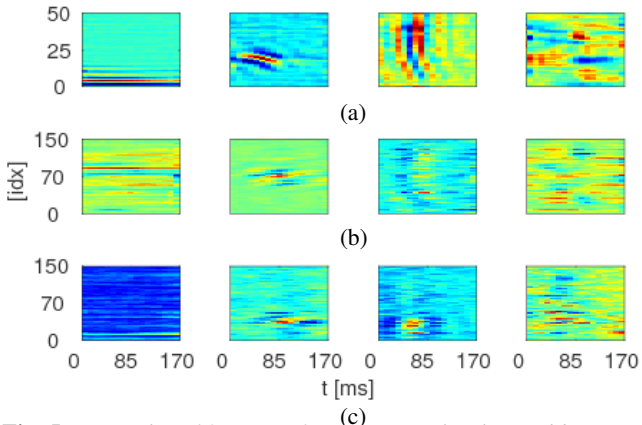

Fig. 5. *Examples of 2D, time-frequency patches learned by NN on various spectrograms: (a) GT spectrogram; (b), low-pass filtered NN spectrogram; (c) 40Hz filtered NN modulation spectrum*

9. REFERENCES

[1] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, Aug. 1980.

[2] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[3] R. Schlüter *et al.*, "Gammatone features and feature combination for large vocabulary speech recognition," in *ICASSP*, 2007, pp. 649–652.

[4] G. von Békésy, *Experiments in Hearing*. New York: McGraw-Hill, 1960.

[5] H. Fletcher and W. A. Munson, "Loudness, its definition, measurement and calculation," *The Journal of the Acoustical Society of America*, vol. 82, no. 5, pp. 82–108, Oct. 1933.

[6] S. Furui, "Comparison of speaker recognition methods using statistical features and dynamic features," *IEEE Trans. on Acoustic, Speech, and Signal Processing*, vol. 29, no. 3, pp. 342–350, Jun. 1981.

[7] R. Haeb-Umbach *et al.*, "Improvements in connected digit recognition using linear discriminant analysis and mixture densities," in *ICASSP*, vol. II, 1993, pp. 239–242.

[8] A. Biem *et al.*, "A discriminative filter bank model for speech recognition," in *Eurospeech*, 1995, pp. 545–548.

[9] T. N. Sainath *et al.*, "Learning filter banks within a deep neural network framework," in *ASRU*, 2013, pp. 297–302.

[10] Z. Tüske *et al.*, "Acoustic modeling with deep neural networks using raw time signal for LVCSR," in *Interspeech*, 2014, pp. 890–894.

[11] T. N. Sainath *et al.*, "Learning the speech front-end with raw waveform CLDNNs," in *Interspeech*, 2015, pp. 1–5.

[12] D. Palaz *et al.*, "Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks," in *Interspeech*, 2013, pp. 1766–1770.

[13] P. Golik *et al.*, "Convolutional neural networks for acoustic modeling of raw time signal in LVCSR," in *Interspeech*, 2015, pp. 26–30.

[14] P. Ghahremani *et al.*, "Acoustic modelling from the signal domain using CNNs," in *Interspeech*, 2016, pp. 3434–3438.

[15] B. R. Glasberg and B. C. J. Moore, "Derivation of auditory filter shapes from notched-noise data," *Hearing Research*, vol. 47, no. 1-2, pp. 103–138, Aug. 1990.

[16] I. J. Goodfellow *et al.*, "Maxout networks," in *ICML*, 2013, pp. 1319–1327.

[17] C. E. Shannon, "Communication in the presence of noise," in *Proc. of Institute of Radio Engineers*, vol. 37, Jan. 1949, pp. 10–21.

[18] R. Vaughan *et al.*, "The theory of bandpass sampling," *IEEE Trans. on Signal Processing*, vol. 39, no. 9, pp. 1973–1984, Sep. 1991.

[19] Z. Zhu *et al.*, "Learning multiscale features directly from waveforms," in *Interspeech*, 2016, pp. 1305–1309.

[20] A. Waibel *et al.*, "Phoneme recognition using time-delay neural networks," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 37, no. 3, pp. 328–339, 1989.

[21] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *the 27th Int. Conf. on Machine Learning*, 2010, pp. 807–814.

[22] Y. Hoshen *et al.*, "Speech acoustic modeling from raw multichannel waveforms," in *ICASSP*, 2015, pp. 4624–4628.

[23] A. Haar, "Zur theorie der orthogonalen funktionen systeme," *Mathematische Annalen*, vol. 69, no. 3, pp. 331–371, 1910.

[24] M. Nußbaum-Thom *et al.*, "The RWTH 2009 QUAERO ASR evaluation system for English and German," in *Interspeech*, 2010, pp. 1517–1520.

[25] D. Rybach *et al.*, "RASR - the RWTH Aachen University open source speech recognition toolkit," in *ASRU*, 2011.

[26] D. E. Rumelhart *et al.*, *Learning Internal Representations by Error Propagation*. Cambridge, MA, USA: MIT Press, 1986, pp. 318–362.

[27] H. A. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*. Norwell, MA, USA: Kluwer Academic Publishers, 1993.

[28] F. Seide *et al.*, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *ASRU*, Hawaii, USA, 2011, pp. 24–29.

[29] K. Veselý *et al.*, "Convolutive bottleneck network features for LVCSR," in *ASRU*, Hawaii, USA, 2011, pp. 42–47.

[30] L. Welling *et al.*, "Speaker adaptive modeling by vocal tract normalization," *IEEE Trans. on Speech and Audio Processing*, vol. 10, no. 6, pp. 415–426, Sep. 2002.

[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[32] P. Doetsch *et al.*, "RETURNN: The RWTH extensible training framework for universal recurrent neural networks," in *ICASSP*, 2017, pp. 5345–5349.

[33] H. Hermansky and P. Fousek, "Multi-resolution RASTA filtering for TANDEM-based ASR," in *Interspeech*, 2005, pp. 361–364.

[34] T. L. Clarke, "Generalization of neural networks to the complex-plane," in *International Joint Conference on Neural Networks*, vol. 2, 1990, pp. 435–440.

[35] E. Variani *et al.*, "Complex linear projection (CLP): A discriminative approach to joint feature extraction and acoustic modeling," in *Interspeech*, 2016, pp. 808–812.