Acoustic Modeling with Deep Neural Networks Using Raw Time Signal for LVCSR

Zoltán Tüske[1], *Pavel Golik*[1], *Ralf Schlüter*[1], *Hermann Ney*[1,2]

[1]Human Language Technology and Pattern Recognition, Computer Science Department, RWTH Aachen University, 52056 Aachen, Germany
[2]Spoken Language Processing Group, LIMSI CNRS, Paris, France
{tuske,golik,schlueter,ney}@cs.rwth-aachen.de

Abstract

In this paper we investigate how much feature extraction is required by a deep neural network (DNN) based acoustic model for automatic speech recognition (ASR). We decompose the feature extraction pipeline of a state-of-the-art ASR system step by step and evaluate acoustic models trained on standard MFCC features, critical band energies (CRBE), FFT magnitude spectrum and even on the raw time signal. The focus is put on raw time signal as input features, i.e. as much as zero feature extraction prior to DNN training. Noteworthy, the gap in recognition accuracy between MFCC and raw time signal decreases strongly once we switch from sigmoid activation function to rectified linear units, offering a real alternative to standard signal processing. The analysis of the first layer weights reveals that the DNN can discover multiple band pass filters in time domain. Therefore we try to improve the raw time signal based system by initializing the first hidden layer weights with impulse responses of an audiologically motivated filter bank. Inspired by the multi-resolutional analysis layer learned automatically from raw time signal input, we train the DNN on a combination of multiple short-term features. This illustrates how the DNN can learn from the little differences between MFCC, PLP and Gammatone features, suggesting that it is useful to present the DNN with different views on the underlying audio.
Index Terms: acoustic modeling, raw signal, neural networks

1. Introduction

Since DNN based acoustic models have become a popular alternative to the Gaussian mixture models (GMMs), a lot of effort was put into feature engineering that aimed at finding a representation of input audio data that is most suitable for training of neural networks [1][2]. GMMs are quite sensitive to input features: the features need to be decorrelated so that a diagonal covariance matrix can be used for faster scoring and the dimension needs to be relatively low. These requirements have led to a large variety of feature extraction pipelines that build upon expert knowledge of speech production and perception. In contrast, hybrid DNN/HMM models [3] have none of these constraints and a DNN acoustic model can easily be trained on high dimensional features (several thousands) even with a large amount of correlation between components. Further, the universal approximator property of a neural network [4][5] with (multiple) hidden layers should allow the DNN to learn the necessary (non-linear) feature extraction steps from data. For this reason we investigated the question: how much feature extraction can be left for the DNN to discover?

Many groups have found logarithm of critical band energies (CRBE) extracted e.g. from a Mel filter bank to be most suitable for training DNNs. One of the reasons why CRBE often outperform MFCC or PLP might be the fact that they contain somewhat more high resolution information: while conventional MFCC and PLP range from 13 to 16 dimensions, the CRBEs used for DNN training are often 20- to 40-dimensional. Further, many steps of typical feature extraction pipelines boil down to linear projections, which should be easy to learn from data. Ultimately, to avoid a loss of information the acoustic model needs to be trained on full magnitude spectrum, e.g. [6], or even the raw audio samples of the waveform. While the cross entropy training is still performed on frame level, the latter case allows to present the DNN a sequence of audio samples without any notion of frame boundaries, thus allowing the neural network to discover all kind of non-stationary patterns. Such patterns correspond to various phonetic events that are described poorly with frame-based stationary processing such as FFT.

The cost for processing raw time signal is twofold. First, the high dimensionality of such feature spaces increases the number of free parameters. This issue can be counteracted by adjusting the network topology, e.g. introducing narrow matrix factorization layers [7]. Second, the raw time signal features discard the common assumption of most feature extraction pipelines that human perception resolution is non-linear along the frequency axis leaving it up to the DNN to discover. Our approach to tackle this is to initialize the weights of the first layer by impulse responses of Gammatone filters that follow the audiological spacing in the frequency domain.

Further, we investigate how the DNN can be presented with an increased time-frequency resolution without leaving the framework of conventional feature extraction. This is related to the concept of feature combination, where different short-term features describe more or less the same spectral properties of the signal [8]. However, the differences between the different feature streams are themselves an additional source of information about the underlying audio signal.

Previously, a dramatic degradation of recognition accuracy by training a DNN directly on raw speech signal was reported in [9]. Instead, the authors used convolutional neural networks [10] to obtain competitive results on the TIMIT task. Some works on processing of raw speech signal make use of other models such as linear predictive models [11][12] or SVMs [13]. They mostly evaluate on small classification tasks, leaving the question open, how much would the amount of training data compensate for the described difficulties.

This paper is organized as follows. The different feature extraction pipelines, including the FFT and time signal features, are summarized in Section 2. The experimental setup is introduced in Section 3 and the results of our investigation are pre-

sented in Section 4. The conclusions are drawn in Section 6.

2. Feature extraction

This section gives a brief overview of the three cepstral features, the FFT based features and the raw signal features.

2.1. Waveform — "raw" time signal

Processing the audio sampled at 16 kHz with the same 10 ms steps as in case of typical cepstral features boils down to taking 160 samples from the PCM waveform. The windows are non-overlapping so that stacking neighboring vectors does not result in discontinuities. The samples quantized with 16 bit need to be normalized to a numerically robust range by performing the mean and variance normalization either globally over the complete training data or on the per-utterance level. This can be interpreted as DC bias removal and loudness equalization and at the same time it serves numerical purposes to stabilize the DNN training with gradient descent.

2.2. Amplitude spectrum — FFT

In contrast to raw time signal, the short-time Fourier transform (STFT) is performed on overlapping windows of 25 ms. The samples are zero-padded to a window of size 2^9 and weighted with a Hanning function, which exhibits smaller side lobes in the amplitude spectrum than a rectangular window. The 512-FFT results in a 257-dimensional vector due to the symmetry of the amplitude spectrum. The phase spectrum is discarded.

2.3. Mel-Frequency cepstral coefficients — MFCC

The feature extraction is based on the STFT of the pre-emphasized speech signal [14]. The amplitude spectrum is integrated by a filterbank with the *triangular* filters being equidistantly spaced on Mel-scale. The MFCC features are extracted from the logarithm filter outputs (also referred to as CRBE) by applying discrete cosine transform (DCT).

2.4. Gammatone features — GT

Instead of the STFT based analysis, the features are extracted from an audiologically motivated filterbank realized by time-domain *Gammatone* filters [15]. The auditory filters are placed equidistantly on Greenwood-scale. After spectral and temporal integration the 10th root is taken instead of the logarithm and the DCT is applied for decorrelation.

2.5. Perceptual linear predictive coefficients — PLP

These features are again based on the STFT of speech [16]. Simulating the critical band masking, the amplitude spectrum is integrated with *trapezoid* filters equally spaced on Bark-scale. The filterbank output is pre-emphasized according to equal-loudness curve. To simulate the relation between the intensity and perceived loudness of sound, cubic root amplitude compression is performed followed by all-pole model parameter estimation (linear predictive (LP) analysis). The autoregressive coefficients are directly transformed to cepstral coefficients.

3. Experimental setup

The acoustic model training is performed w.r.t. frame-wise cross entropy criterion on 50 hours of speech from the Quaero [17] English database *train11*, which amounts to ca. 16 million input vectors. The development and evaluation sets consist of ca. 3.5 hours of speech each, corresponding to about 1.2 million vectors. Some experiments are presented on a large 250 hours set from the same corpus *train11*. A 4-gram language model (LM) is used during the recognition.

Throughout all experiments we use 6 hidden layers with 2000 hidden units in each layer. The output layer with 4500 nodes corresponds to the generalized triphones tied by a phonetic classification and regression tree (CART). The number of trainable weights amounts to approx. 30M-35M depending on the features used. The input features always correspond to 17 stacked frames so that the overall amount of temporal context presented to the DNN at once is the same. The mini-batches of size 512 are drawn from the shuffled training set. The weights are initialized via discriminative pre-training (DPT) [1].

The ASR baseline system is a conventional GMM/HMM based model trained on the same database w.r.t. the maximum likelihood criterion. We applied linear discriminant analysis (LDA) to 9 consecutive MFCC frames to obtain the final 45-dimensional features. The GMM with a globally pooled diagonal covariance matrix consists of approx. 660k densities, which corresponds to about 30M trainable parameters. For acoustic training and recognition we used the RASR toolkit [18].

4. Results

In the first experiment we compared the baseline results obtained with the GMM and DNN acoustic models on MFCC features normalized for mean/variance and the vocal tract length (VTLN). The results are shown in Table 1. Unless stated otherwise, the training is performed on 50 hours of speech. The same DNN configuration was trained on the raw time signal as described in Section 2.1. As expected, the MFCC-based DNN model outperforms the GMM, but the WER of the system trained on raw time signal is still significantly higher.

Table 1: *Baseline results. WER in %.*

Features	model	dev	eval
MFCC	GMM	24.4	31.6
MFCC	DNN	19.4	25.3
time signal	DNN	29.4	36.8

In the next experiment we wanted to figure out, how the recognition accuracy depends on the various preprocessing steps. For this purpose we decomposed the MFCCs step by step and measured the performance. Table 2 shows the word error rate (WER) after each step. The results indicate that without mean/variance normalization and VTLN, the gap between MFCCs and FFT features decreases significantly.

4.1. Feature combination

From the results in Table 2 it is clear, that the presented features differ in the dimensionality by an order of magnitude. Still MFCC outperform the high-dimensional FFT and time signal features. How can we increase the amount of information within the framework of low-dimensional features? As described in Section 2, the different short-term feature extraction pipelines cover slightly different representations of the underlying audio. Hoping that the DNN can extract useful information from these differences we performed feature combination following the approach of [8]. The results in Table 3 confirm that a DNN being a powerful classifier can learn more from multiple feature streams than from every single feature set.

Table 2: *Feature preprocessing and normalization for DNN AM. Dimension of a single feature vector. WER in %.*

Features	dim.	dev	eval
MFCC	16		
+ global norm.		19.8	26.1
+ utterance norm.		19.7	25.5
+ VTLN		19.4	25.3
MFCC	20		
+ VTLN + utterance norm.		19.1	25.2
CRBE			
+ VTLN + utterance norm.	20	19.5	25.7
	40	19.7	26.2
|FFT|	257		
+ global norm.		21.3	27.8
+ 10th root		21.0	27.5
+ utterance norm.			
+ 10th root		20.6	26.8
time signal	160		
+ global norm.		29.4	36.8
+ utterance norm.		28.9	35.0

Table 3: *Feature combination. WER in %.*

Features	dev	eval
MFCC	19.1	25.2
PLP	19.2	24.8
GT	19.2	25.5
MFCC + PLP + GT	18.4	24.2

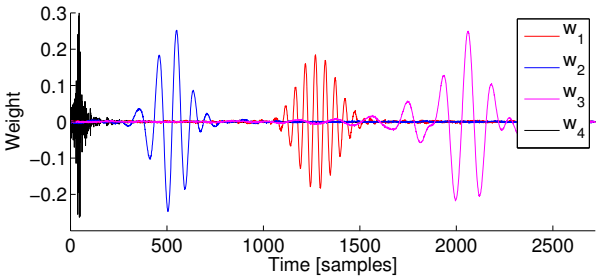

Figure 1: *Four rows from the first layer weight matrix trained on raw time signal. The time range corresponds to 17 frames of 10 ms* ($17 \cdot 10ms \cdot 16kHz = 2720$)

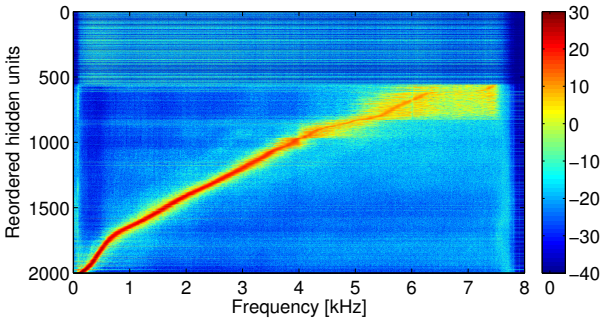

Figure 2: *Amplitude spectra of the reordered rows from the first layer weight matrix trained on time signal.*

4.2. Analysis of the input layer trained on time signal

Having obtained surprisingly reasonable results on the normalized raw time signal, we were curious what kind of patterns could have been learned by the neural network. Although the analysis of all parameters remains infeasible, we could detect clearly interpretable patterns within the first layer of the fully trained DNN. Figure 1 shows the weights learned by four of the hidden nodes of the first layer. Apparently, the DNN managed to learn some kind of impulse responses that correspond to band pass filters and other patterns (e.g. short bursts) purely from data. In order to illustrate the spectral properties of the discovered filters, we zero-padded every row in the weight matrix to 8000 entries, calculated the magnitude spectrum

$$W_i = |\text{FFT}\{w_{i,\cdot}\}| \in \mathbb{R}^{1\times 8000} \quad 1 \le i \le 2000 \qquad (1)$$

and sorted the rows by the location of the most prominent "blob". The position of the blob was calculated after smoothing the spectrum with a Gaussian kernel g as

$$\hat{W}_i = W_i * g \qquad (2)$$

$$f_c^i = \underset{1\le j\le 8000}{\text{argmax}} \{\hat{W}_{i,j}\} \qquad (3)$$

Assuming that every row can be interpreted as a band pass impulse response, the location of the blob corresponds to the center frequency of the learned transfer function. Figure 2 shows the obtained spectra as $20 \log_{10} W_i$. It can be seen that, without any prior knowledge, the DNN has discovered a large number of band pass like filters that exhibit roughly the audiological distribution. It means, the number of narrow band pass filters in the lower frequency region is quite high, while with increasing center frequency, the bandwidth of the filters becomes larger. Also the distribution of the center frequencies is non-linear. The bandwidth of the transfer function can be calculated as equivalent noise bandwidth by

$$f_b^i = \frac{\sum_j W_{i,j}^2}{(\max_j W_{i,j})^2} \qquad (4)$$

Figure 3 shows the scatter plot of the approximated parameters f_c and f_b of the learned filters.

Remarkably, the position of the filters in time is not restricted to the center of the stacked audio samples, but is scattered across left and right context approximately uniformly. These shifts (or time offsets) are expressed in the phase spectrum and are therefore not visible in Figure 2. This distribution indicates that the DNN was able to learn different filters for different parts of the presented audio context. Also, none of the learned narrow filters exhibits multiple passbands.

4.3. Rectified linear units and large scale experiments

In the following set of experiments we investigated how strong can we further reduce the gap in recognition accuracy between the various feature configurations by (a) switching the activation function and (b) increasing the amount of training data. First we compared sigmoid activation function with the rectified linear units (ReLU) [19]. From the previous experience

Table 4: *Feature and activation function comparison, training on **50h**. WER in %.*

Features	dev		eval	
	sigmoid	ReLU	sigmoid	ReLU
MFCC	19.1	18.0	25.2	23.8
MFCC + PLP + GT	18.4	16.6	24.2	21.7
|FFT|	20.6	18.4	26.8	24.7
time signal	28.9	22.6	35.0	28.5

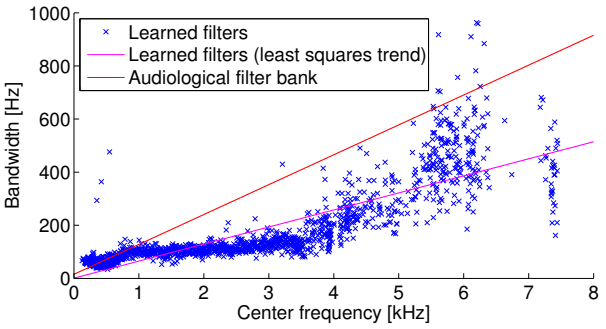

Figure 3: *Scatter plot of approximated parameters of the learned filter bank.*

Table 5: *Feature and activation function comparison, training on **250h**. WER in %.*

Features	dev		eval	
	sigmoid	ReLU	sigmoid	ReLU
MFCC	15.2	15.9	20.4	21.1
MFCC + PLP + GT	14.8	14.0	19.8	18.9
|FFT|	16.1	15.8	21.6	21.5
time signal	19.2	17.6	25.6	23.5

we know that ReLUs are sensitive to regularization so we used L_2-regularization with a value of 0.0001. In contrast, sigmoid non-linearities perform best with no regularization at all. The results shown in Table 4 suggest that the ReLUs have a stronger effect on the systems with high error rates, which is presumably due to a more difficult optimization problem. In addition, we repeated these experiments with DNNs trained on 250 hours of speech. Table 5 shows the obtained results. Further large scale experiments revealed that increasing the number of hidden layers up to 12 narrowed the performance gap between MFCC and raw time signal achieving 20.9% WER on the evaluation corpus.

4.4. Manual weight initialization with audiological filters

After we observed the filter shapes that have been learned from raw time signal, we investigated, whether we can initialize the weights of the first hidden layer in a way that makes it easier for the DNN to discover further meaningful filters during training with gradient descent. For this purpose we calculated the real part of impulse responses of a stationary Gammatone filter bank that follows the audiological filter distribution [20]. The parameters of the 32 filters were defined as follows (with $l = 24.7$ and $q = 9.265$):

$$f_c^i = l \cdot q \cdot (e^{i/q} - 1) \quad (5)$$

$$f_b^i = l + f_c^i/q \quad (6)$$

In order to account for different positions in time we created multiple shifted copies of each filter's impulse response to obtain a weight matrix of the same size as the randomly initialized first layer weights in the previous experiments. Table 6 shows the comparison of three different approaches: random initialization (as in Table 4), initialization by a Gammatone filter bank with regular weight update through backpropagation, and a fixed Gammatone filter bank layer with no update throughout the training. The latter case corresponds to a fixed "feature extraction layer" where only layers above the first one are trained, so that we can compare whether the DNN can improve the weights by backpropagation upon the initialization.

Table 6: *Weight initialization for learning from raw time signal. WER in %.*

Weight initialization	update allowed	dev	eval
random	yes	22.6	28.5
GT	yes	22.4	28.7
	no	24.9	31.1

It can be seen that the manually designed filter bank does not help the DNN much to discover better features compared with fully random initialization. Also, keeping the first layer weights fixed throughout the training rather hurts the recognition performance. This indicates that the initial filter bank configuration is suboptimal, presumably because of a too low frequency resolution and the lack of non-band-pass patterns.

5. Conclusions

In this paper we have shown that using hybrid DNN/HMM acoustic models allows to obtain reasonable recognition results even without any processing of the raw time signal. The performance gap between raw time signal and conventional MFCC features could be reduced strongly by switching from sigmoid activation function to rectified linear units. The amount of training data further reduced the gap.

Our analysis of the learned weights suggests that without any prior knowledge, the DNN is able to learn a set of band pass filters in time domain purely from the raw time signal. We presented a way to interpret the learned parameters: by reordering the rows within the input layer weight matrix, it is possible to see the approximately audiological distribution of the filters. This again nicely confirms the result of many years of research on feature extraction. Further, this result shows a real alternative to the otherwise (mostly) stationary feature extraction pipelines: presenting the DNN with data on sampling frequency level allows the acoustic model to learn non-stationary patterns, localized in time across frame boundaries. Also, the loss of information can be reduced by processing time domain data.

Finally we presented a trade-off between feature dimensionality and level of detail of the underlying audio. By training the DNN on a combination of MFCC, PLP and Gammatone features, the resulting acoustic model outperformed all other systems, even with a large amount of training data. This suggests that the differences in these feature extraction pipelines allow the DNN to gain additional knowledge about the input data.

6. Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 287755 (transLectures). This work has received funding from the Quaero Programme funded by OSEO, French State agency for innovation. H. Ney was partially supported by a senior chair award from DIGITEO, a French research cluster in Île-de-France. Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD / ARL) contract no. W911NF-12-C-0012. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

7. References

[1] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Hawaii, USA, Dec. 2011, pp. 24–29.

[2] D. Yu, M. L. Seltzer, J. Li, J.-T. Huang, and F. Seide, "Feature learning in deep neural networks - a study on speech recognition tasks," in *International Conference on Learning Representations*, Scottsdale, AZ, USA, May 2013.

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

[4] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989.

[5] K. Hornik, M. B. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, Jul. 1989.

[6] T. N. Sainath, B. Kingsbury, A.-r. Mohamed, and B. Ramabhadran, "Learning filter banks within a deep neural network framework," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Olomouc, Czech Republic, Dec. 2013, pp. 297–302.

[7] S. Wiesler, A. Richard, R. Schlüter, and H. Ney, "Mean-normalized stochastic gradient for large-scale deep learning," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Florence, Italy, May 2014, pp. 180–184.

[8] C. Plahl, R. Schlüter, and H. Ney, "Improved acoustic feature combination for LVCSR by neural networks," in *Proc. Interspeech*, Florence, Italy, Aug. 2011, pp. 1237–1240.

[9] D. Palaz, R. Collobert, and M. Magimai.-Doss, "Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks," in *Proc. Interspeech*, Lyon, France, Aug. 2013, pp. 1766–1770.

[10] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Kyoto, Japan, Mar. 2012, pp. 4277–4280.

[11] A. B. Poritz, "Linear predictive hidden Markov models and the speech signal," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, vol. 7, Paris, France, May 1982, pp. 1291–1294.

[12] Y. Ephraim and W. J. J. Roberts, "Revisiting autoregressive hidden Markov modeling of speech signals," *IEEE Signal Processing Letters*, vol. 12, no. 2, pp. 166–169, Feb. 2005.

[13] J. Yousafzai, Z. Cvetković, and P. Sollich, "Subband acoustic waveform front-end for robust speech recognition using support vector machines," in *Proc. Interspeech*, Brighton, UK, Sep. 2009, pp. 2679–2682.

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

[15] R. Schlüter, I. Bezrukov, H. Wagner, and H. Ney, "Gammatone features and feature combination for large vocabulary speech recognition," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Honolulu, Hawaii, USA, Apr. 2007, pp. 649–652.

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

[17] Quaero Programme. http://www.quaero.org.

[18] D. Rybach, S. Hahn, P. Lehnen, D. Nolden, M. Sundermeyer, Z. Tüske, S. Wiesler, R. Schlüter, and H. Ney, "RASR - the RWTH Aachen university open source speech recognition toolkit," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Hawaii, USA, Dec. 2011.

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

[20] 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.