Use of trainable wavelet transform with adaptive threshold filtering for noise reduction of speech signals

A A Lependin¹, I D Ilyashenko¹ and R S Nasretdinov¹

¹Information Security Department, Altay State University, 61 Lenin ave., Barnaul, 656049, Russia
E-mail: andrey.lependin@gmail.com

Abstract. A method of noise reduction based on an adaptive threshold filtering over trainable wavelet transform was proposed. The restrictions on the trainable transform filter parameters were provided by a set of quadratic regularization terms. As an analog of the “hard” and “soft” threshold functions we used their smooth infinitely differentiable versions. The parameters of trainable wavelet transform, and threshold values were estimated by backpropagation and gradient-based optimization algorithm with an adaptive momentum estimation. Results of the proposed method were compared with an approach based on fixed discrete wavelet transform and non-adaptive global and level-dependent threshold algorithms on the model problem. We used signal-to-noise ratio between suppressed and clean signals to numerically estimate the efficiency of noise reduction. We showed that the best results were obtained when the proposed trainable method with Daubechies 4 wavelet filters fine-tuning and adaptive level-dependent thresholding were applied.

1. Introduction
Nowadays digital recording and speech processing methods became widespread. They are applied for voice biometrics, speech recognition and synthesis and phonoscopic examination in criminalistics. In many cases the quality of recorded audio signals is quite poor because they contain distortions [1]. The reasons of it are connected both with insufficiently high operational characteristics of the equipment used (narrow dynamic range, large coefficient of nonlinear distortion, etc.), and with external factors associated with a high level of ambient noise, reverberation in the room where the recording is made. Distortion of a speech signal leads to a significant decrease in the quality of biometric methods, reduces speech intelligibility when it is recognized by automatic systems or by a digital forensics specialist. Thus, eliminating excessive noise and highlighting useful information in the recorded speech signal is an extremely important problem.

One approach to cleaning and extracting the useful informative part of a digitized speech signal is threshold filtering based on a multilevel discrete wavelet transform [2]. It has become widespread and is currently one of the standard methods to noise reduction. However, it does not contain a sufficient degree of flexibility; its parameters are fixed and weakly depend on the characteristic of the processed audio recordings.

In this paper, we propose generalization of the standard approach, based on the replacement of all stages of the signal processing with their differentiable analogues, which made it possible to adjust parameters using methods similar to those used in machine learning. Due to this, it becomes possible to build more efficient noise reduction approaches focused on specific usage conditions.
2. Related work
In recent years a lot of methods where discrete wavelet transform was used for noise reduction were proposed. Threshold filtering technique was presented by Donoho in [3]. It is based on applying a shrinkage function on wavelet coefficients. The approach consists of decomposition of noisy signal to wavelet domain, applying “hard” or “soft” thresholding rule on the wavelet coefficients and reconstruction resulting features. In the end denoised signal is attained. Details about “soft” and “hard” thresholding functions will be discussed in Section 5 of this paper. In [4] the modified method of optimizing wavelet bases was proposed. The authors of this paper adjust wavelet filters coefficients by training them by gradient descent. Properties imposed on the filters were kept by projecting them into wavelet domain.

The authors of the work [5] presented an approach that uses nonlinear diffusion filtering with trainable shrinkage function. It was obtained by solving differential equation with wavelet gradient and its diffusivity function. Gradients for filtering were computed as rotationally invariant two-dimensional coupled Haar wavelet. Forward-and-Backward (FAB) diffusivity was used as a shrinkage function with two trainable parameters which control the amount of forward and backward diffusion.

In the work [6] an unusual noise reduction method in wavelet domain using wiener filter was proposed. The authors adjust wiener filter parameters to not remove noise completely. Remained background noise was transformed to another more acceptable for listeners noise. It can decry distortion caused by removing all of noise.

3. General outline of the proposed approach
The main idea of the adaptive threshold filtering approach with the trained wavelet-like transform was as follows (figure 1). Noisy signal \( \hat{x} \) is decomposed by \( m \)-level wavelet-like transform \( W(L, H) \) with the parameters \( L = \{ l_k, k = 1, ..., N \} \) and \( H = \{ h_k, k = 1, ..., N \} \), corresponding to coefficients of low-pass and high-pass discrete filters, respectively [7,8]. As a result, thresholding function \( \eta(T) \) was applied to the coefficients of the \( m \)-level decomposition \( z \), with the thresholding parameters vector \( T = \{ t_j, j = 1, ..., m \} \), which in general could depend on decomposition level \( j \). The denoised version \( y \) of the signal \( \hat{x} \) was obtained by using the inverse wavelet-like transform \( W^{-1}(L, H) \) to the modified coefficients \( z_T \).

![Figure 1. A proposed adaptive threshold filtering approach.](image)

During the training stage, when parameters \( L, H, T \) were adjusted, the denoised signal \( y \) was used to quantify the denoising quality by the function \( C(x, y) = \frac{1}{L} \sum_i (x_i - y_i)^2 \), which showed the degree of deviation of the cleaned signal \( y \) from the known clean signal \( x \). The quality function of the filtering approach was determined as the sum

\[
Q(x, y; L, H, T) = C(x, y) + \lambda R(L, H),
\]
where \( \lambda \) – constant determining the contribution of regularization \( R(L, H) \), which restricts coefficients of wavelet-like-transform filters \([9]\). Gradients of the function \( Q(x, y; L, H, T) \), with respect to the parameters \( L, H \) and \( T \) were calculated by backpropagation. Their adjustment was provided by modified gradient descent (details in the experiment description below).

4. Trainable wavelet-like transform
Wavelet-like transform \( W(L, H) \) in this paper was defined as a hierarchical application of low-pass and high-pass filters. Coefficients of “details” (high frequency component) and “approximation” (low frequency component) were sequentially computed for the input signal. In the standard discrete wavelet decomposition, a set of “details” and a final “approximation” of the highest level of decomposition are used to represent the signal. In this paper, successive levels of decomposition consisted of “details” and “approximation” coefficients calculated based on previous level “details” and “approximation”. A set of all “details” and “approximations” of the highest level \( m \) was used as a feature vector.

At the figure 2 an example of two-level wavelet-like decompositions is presented. We denoted the input signal to the transform as \( x \). The downsampling operation for some input vector \( v \) is denoted by \( d(v) = \{v_i, i = 0, 2, \ldots, M\} \), where \( M \) is the maximum even index in \( v \) and the convolution operation is denoted by \(*\). Low-pass and high-pass filters are denoted as \( L \) and \( H \). Then wavelet-like decomposition was obtained as follows. The vectors of the first level “details” and “approximation” coefficients are \( D_1 = d(x \ast H) \) and \( A_1 = d(x \ast L) \), respectively. The second level of decomposition consists of “details” vectors \( D_{21} = d(D_1 \ast H) \) and \( D_{22} = d(A_1 \ast H) \) and “approximation” vectors \( A_{21} = d(D_1 \ast L) \) and \( A_{22} = d(A_1 \ast L) \). The feature vector of the signal is a concatenation of four vectors of coefficients \( D_{21}, A_{21}, D_{22} \) and \( A_{22} \).

![Figure 2. Wavelet-like decomposition of input signal x.](image)

The filter coefficients of the wavelet-like transform were restricted by a regularization function \([9]\):

\[
R(L, H) = R_1(H) + R_2(L) + R_3(H) + R_4(L) + R_5(L, H),
\]

where \( R_1(H) \) and \( R_2(L) \) ensures fulfilling the orthonormality property of the wavelet-like basis, \( R_3(H) \) and \( R_4(L) \) limited the sum of high and low pass filter coefficients, \( R_5(L, H) \) was determined by the condition of orthogonality of the filters \( L \) and \( H \).

Initialization of the filters \( H \) and \( L \) was carried out by one of two methods. In the first initialization method, the high-pass filter \( H \) was filled with random values according to \([10]\), and then it was normalized by subtracting the mean and dividing by the standard deviation to ensure zero sum of the coefficients and filters energy limitation \( \sum_k h_k^2 = 1 \). The initial approximation of the low-pass filter
was calculated by the formula (3) obtained from the property of orthogonality of low-pass and high-pass filters [11]:

\[
l_k = (-1)^{k+1} \cdot h_{N-1-k}, \text{ for } k = 1, \ldots, N.
\]  

(3)

In the second initialization method, as the initial approximation of the filter coefficients, the parameter values of some standard discrete wavelet bases were used, for which, obviously, all the required restrictions on the orthogonality, normalization, and energy limitation were fulfilled. These coefficients were further adjusted during training process by backpropagation.

5. Thresholding function

Two types of thresholding functions over the coefficients of the wavelet-like transform [3, 12] were used in this work. “Hard” thresholding function with the threshold \( t \) is defined by:

\[
\eta_h(x, t) = \begin{cases} 
  x, & |x| > t \\
  0, & |x| \leq t
\end{cases}
\]  

(4)

A function of the form (5) is standardly used for “soft” thresholding calculation:

\[
\eta_s(x, t) = \text{sgn}(x)(|x| - t)_+ = \begin{cases} 
  x + t, & x < -t \\
  0, & |x| \leq t \\
  x - t, & x > t
\end{cases}
\]  

(5)

In this work, these functions were replaced by their continuously differentiable analogues [9], which allowed calculating the derivatives with respect to the threshold values and the input values of the decomposition coefficients. It made it possible to use the backpropagation for training the parameters. The differentiable versions of the “hard” and “soft” threshold transformations are defined in the equations (6) and (7), respectively [13]:

\[
\eta_{hd}(x, t) = \left(1 + \exp\left(\frac{-x + t}{\mu_h}\right)\right)^{-1} - \left(1 + \exp\left(\frac{-x - t}{\mu_h}\right)\right)^{-1} x,
\]  

(6)

\[
\eta_{sd}(x, t) = x + \frac{1}{2} \left(\sqrt{(x - t)^2 + \mu_s} - \sqrt{(x + t)^2 + \mu_s}\right).
\]  

(7)

where constants \( \mu_h \) and \( \mu_s \) determined the proximity of these functions to \( \eta_h(x, t) \) и \( \eta_s(x, t) \). The limits of the functions (6) and (7) as \( \mu_h \) and \( \mu_s \) approaches zero equals to standard discontinuous versions of the thresholding functions (4) and (5).

We used two filtering approaches: global and level-dependent. In the global filtering the threshold vector consisted of the equal values \( t_j = t, j = 1, \ldots, m \) which were independent of the level of signal decomposition. In the case of level-dependent filtering, the values could differ and were adjusted independently of each other. When comparing with the non-adaptable version of the threshold transformation, the following fixed threshold estimates were used [12]:

\[
t = t_j = \sigma(\ln l)^{1/2},
\]  

(8)

\[
t_j = \sigma \left(2 \ln \frac{l}{2^j}\right)^{1/2},
\]  

(9)

for global and level-dependent values, respectively, where \( \sigma \) is the known standard deviation of noise added into the signal.

6. Experiments

The proposed approach was implemented on Python programming language. The trainable wavelet-like transform, thresholding functions (6) and (7) and the quality function (1) were implemented using the PyTorch machine learning library [14]. Due to this, the application of the backpropagation and gradient descent was greatly simplified, since in this library the construction of a computational graph
and the calculation of gradients were carried out automatically. To speed up the calculations, a specialized Nvidia GeForce 1080Ti GPU card was used. Minibatches of $B$ samples in each were fed to the input of the noise reduction model, for which the calculation of denoised signal, the estimation of gradients and parameters correction were simultaneously carried out. The filtering model was trained using the adaptive momentum estimation (Adam) [15]. The values of the exponential decay rates of this method were taken by default. The hyperparameters of the filtering approach are shown in the table 1.

| Hyperparameter          | Value |
|-------------------------|-------|
| Number of training epoch| 4     |
| Minibatch size $B$      | 100   |
| Learning rate           | $10^{-7}$ |
| Regularization $\lambda$| $10^{-3}$ |
| $\mu_h$                 | $10^{-1}$ |
| $\mu_s$                 | $10^{-2}$ |
| $M$                     | 6     |

We use Signal-to-Noise Ratio metrics (10) to quantify the denoising quality of the proposed approach:

$$SNR(x,y) = 10 \log \frac{\sum x_i^2}{\sum (x_i-y_i)^2},$$

(10)

where $x = \{x_i, i = 1, \ldots, L\}$ is the known cleansignal, $y = \{y_i, i = 1, \ldots, L\}$ is the denoised signal.

To evaluate performance of our method we used model problem based on a set of speech signal samples. Clean speech signals with a duration of 2.4 s and sampling frequency $f_s = 16$ kHz were cut out from the voice samples of the VCTK speech corpus [16]. They were normalized by energy and white Gaussian noise with a standard deviation of 0.3 was added to them. The signal-to-noise ratio of noised samples in average was equal to $SNR=5.8$ dB. Examples of spectra of the known clean signal and the corresponding noisy version are shown in figure 3.

![Figure 3](image_url)

**Figure 3.** Spectrograms of example signals: a) clean speech signal from VCTK corpus, b) signal with additive noise.

### 7. Results

Tables 2 and 3 show the quality of adaptive and non-adaptive methods of noise reduction using global (table 2) and level-dependent threshold filtering (table 3). As the basic transform for the multi-level decomposition of the signal, the discrete Daubechies wavelet transform db4 was chosen. This choice
was due to the fact that the speech signal was substantially non-smooth. Training of the proposed noise filtering approach was carried out in three scenarios:

1. the db4 wavelet coefficients that were not adapted by the gradient descent (non-adaptive db4).
2. random initialization of filter coefficients $L$ and $H$, which were then adapted to the given dataset by the gradient descent (random adaptive).
3. the db4 wavelet coefficients were used as the initial values of the “details” and “approximations” filter coefficients and then adapted by the gradient descent (adaptive db4).

The results of the global threshold filtering in Table 2 show that using random initialization of the wavelet-like transform coefficients did not allow getting noise reduction quality comparable to the two remaining scenarios. An additional adaptation of the Daubechies conversion coefficients led to a slight increase in quality. The adaptation of the threshold value had a more significant effect.

| Scenario      | Thresholding | SNR, dB | Threshold adaptation |
|---------------|--------------|---------|----------------------|
|               |              |         | OFF      | ON      |
| non-adaptive  | hard         | 12.5    | 12.8     |
| db4           | soft         | 10.4    | 11.8     |
| randomadaptive| hard         | 9.2     | 12.0     |
|               | soft         | 10.9    | 11.4     |
| adaptive      | hard         | 13      | 13.1     |
| db4           | soft         | 12.3    | 12.9     |

The use of a more flexible approach of level-dependent threshold filtering allowed improving the quality of noise reduction. The best efficiency was shown by the combination of a hard threshold filtering with an adaptation of the threshold value and db4 initialization of wavelet coefficients. It should be noted that in this case, random initialization proved to be not worse than applying the basic non-adaptive case. In bold, Table 3 indicates the best SNR for each scenario.

| Scenario      | Thresholding | SNR, dB | Threshold adaptation |
|---------------|--------------|---------|----------------------|
|               |              |         | OFF      | ON      |
| non-adaptive  | hard         | 9.3     | 14.7     |
| db4           | soft         | 11.0    | 14.2     |
| randomadaptive| hard         | 11.3    | 12.7     |
|               | soft         | 7.6     | 13.0     |
| adaptive      | hard         | 11.8    | 14.9     |
| db4           | soft         | 12.4    | 14.7     |

Figure 4 shows spectrograms of denoised signals for each scenario. The spectrogram of the signal denoised by the adaptive db4 transform (figure 4b) is qualitatively different from the remaining cases. It can be seen that this version of the noise reduction algorithm made it possible to better highlight the low-pass component of the input signal.

Figure 5 shows the spectrograms of low-pass and high-pass filters corresponding to each scenario. It should be noted that the differences between the adaptive and non-adaptive versions of db4 filters are extremely small. The trained version has narrower filter bandwidths. Training from random initialization has led to wider band pass filters with fewer slopes in the decline area.
a)

Figure 4. Spectrograms of denoised signals (level-dependent threshold): a) db4 without training, b) db4 with training, c) random initialization with training.

b)

c)

Figure 5. Spectral characteristics of the wavelet-like transform filters: a) db4 without training, b) db4 with training, c) random initialization with training.

8. Conclusion

In this paper we proposed a new approach to cleaning an audio signal from noise and distortion, based on the use of differentiable trainable transforms. On the model problem of speech denoising we showed that it allows achieving a better quality in comparison with the standard technique with fixed transformations and thresholds. The developed approach can find practical application in a number of tasks related to speech recognition, information security and digital forensic science.

References

[1] Loizou P 2017 *Speech Enhancement. Theory and Practice, 2nd Edition* (Miami: CRC Press) p 711
[2] Singh J and Aggarwal R 2011 *Int. Journal of Computer Applications* 20 14-9
[3] Donoho D and Johnstone I 1994 *Biometrika* (Oxford: Oxford University Press) 81 pp 425-55
[4] Grandits T, Pock T 2018 *Energy Minimization Methods in Computer Vision and Pattern Recognition* (Cham: Springer) 249-62
[5] Tobias A and Joachim W 2019 Learning a Generic Adaptive Wavelet Shrinkage Function for Denoising *Preprint* cs.NE/1910.09234
[6] Mohammadi M et al 2008 *INTERSPEECH 2008, 9th Annual Conference of the International Speech Communication Association* 561-564
[7] Hurley N 2007 *15th Int. Conf. on Digital Signal Processing* 15 pp 631-634
[8] Søgaard A 2017 Learning optimal wavelet bases using a neural network approach *Preprint* cs.NE/1706.03041v2
[9] Ilyashenko I et al 2019 *J. Phys.: Conf. Ser.* 1333 032029
[10] Bengio Y and Glorot X 2010 *Proc. of the 13th Int. Conf. on Artificial Intelligence and Statistics* 9 pp 249-56
[11] Lee Fugal D. 2009 Conceptual Wavelets in Digital Signal Processing (San Diego: Space & Signals Technologies) p 302
[12] Shemi P and Ali M 2015 Proc. of 2015 IEEE Int. Conf. on Power, Instrumentation, Control and Computing pp 1-6
[13] Zhang X-P 2001 IEEE Transactions on Neural Networks 12 pp 567-84
[14] Paszke A et al 2019 Advances in Neural Information Processing Systems pp 8026-37
[15] Kingma D and Ba J 2014 Adam: A Method for Stochastic OptimizationProc. of Int. Conf. on Learning Representations(Preprint cs.LG/1412.6980)
[16] Veaux C et al 2017 CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit Proc. of CSTR