POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Enhancing a Stationary Noise Suppressor with Artificial Neural Networks for Non-Stationary Noise Removal

Laurea Magistrale in Music and Acoustic Engineering

Author: Michele Perrone

Advisor: Prof. Fabio Antonacci

Co-advisor: Christof Faller, Ph.D.

Academic year: 2025-2026

---

1. Introduction

1.1. Problem statement

Background noise can hinder the effectiveness of communication devices and impair the perceived quality of audio signals. With the rapid spread of high quality broadband telephony, voice over internet protocol (VoIP) devices, and multi-platform communication software, the problem of noise in speech signals has become even more pronounced and has introduced the need for noise reduction systems that can address wide-band noise. Another common aspect to most contemporary communication devices is that speech signals are usually acquired and transmitted with a single audio channel. While it is possible to achieve a substantial noise suppression for monaural speech signals, it is debatable whether this can lead to improved intelligibility. When dealing with communication devices such as mobile phones, webcams, or portable microphones, it is also important to keep in mind the computational cost of speech enhancement solutions, since the noise removal often takes place in low-resource and low-power microcontroller units (MCUs) that are embedded in the device.

1.2. State of the art

Over the years, many techniques have been developed for the task of noise reduction in speech signals. Most traditional noise suppressors are based on digital signal processing (DSP) algorithms that compute an estimate of the noise and then remove the noise from the signal with spectral subtraction [1]. A particularly effective technique for estimating the noise in a speech signal is minimum statistics [2]. More recently, data-driven techniques such as artificial neural networks (ANNs) have made significant progress in the task of noise reduction. By providing extensive examples of input and target data, ANNs can be trained to learn a generalized mapping between the two domains. A particularly interesting ANN architecture for denoising is an autoencoder (AE) [3], which creates a compressed representation of the input by discarding all non-essential data for reconstructing the output. Another useful family of architectures are recurrent neural networks (RNNs) [4], which are designed for treating sequential data and are therefore well suited for short and long-term structures in time-dependent features. The main limitations of most contemporary deep learning denoising approaches is that their design is not

suitable for real-time and causal processing, and that their computational and memory requirements are too high for low-resource applications such as microcontroller units, webcams, or mobile phones. Another issue that is often encountered in data-driven techniques is the possibility of unpredictable behavior when dealing with samples that are too different from the training dataset: this can be the case with different speakers, background noises, signal to noise ratios, reverberation, and microphone impulse responses. Some approaches have tried to take advantage of both expert knowledge and machine learning (ML) in order to reduce the overall computational cost. For example, [5] presents a hybrid DSP/ML that predicts a gain filter for speech denoising based on the power spectrogram of noisy speech and a combination of other input features. The system also detects voice activity and pitch, in order to attenuate the signal when speech is absent and to remove noise between the harmonics of speech. While achieving a relatively low computational and memory complexity, we argue that the voice activity detector (VAD) can be prone to cutting off speech in certain noise conditions, and that comb filtering can result in artificially sounding speech.

1.3. Approach

Given the limitations of traditional noise suppressors and purely data-driven methods, we approach the problem of noise suppression in speech signals by designing a system that seamlessly integrates a stationary noise suppressor with an artificial neural network (ANN). The goal of the ANN is to improve the performance of the noise suppressor by providing a further non-stationary noise reduction, removal of musical noise, and securing a more consistent performance of the system across different noise types and signal-to-noise ratios. Compared to state of the art deep learning techniques, we want to build a system that is more suitable for applications with constrained memory and computational resources. In order to achieve this, we build a pre-processing pipeline that implements a priori knowledge about the audio signal domain and about the nature of the noise. This enables us to drastically reduce the requirements of the ANN while achieving consistent results.

2. Proposed method

Figure 1 shows our proposed method. The primary aspect of our approach is the combination of a traditional stationary noise suppressor with an artificial neural network (ANN). Figure 1 provides a schematic representation of our approach. Given a noisy spectrogram, the stationary noise suppressor provides an estimate of the stationary noise. Based on this estimate, we compute a gain filter that could be multiplied element-wise with the noisy spectrogram in order to remove the estimated stationary noise. However, since our goal is to eliminate also non-stationary noise, we employ an ANN in order to enhance the aforementioned gain filter, resulting in the modified gain filter. This filter is then used to obtain the denoised spectrogram. In order to re-synthesize the denoised waveform, we combine the denoised power spectrogram with the phase spectrogram of the noisy waveform.
By integrating the ANN into a system that is designed with domain-specific knowledge, we are able to reduce the complexity of the network, simplify its training procedure, and reduce the risk of poor performance in situations that differ from the training dataset. There are key advantages in training the ANN to enhance a gain filter instead of the noisy spectrogram directly. First, gain filters contain values that are fixed in a well-defined numerical range that does not vary over time. Spectrograms, on the other hand, have to be normalized, which is not an easy task to achieve if the sound intensity varies over time, and it would involve the design of an automatic gain control (AGC). Second, we take advantage of the fact that we are able to remove a substantial amount of stationary noise with a computationally inexpensive noise suppressor. This way, we focus the attention of the ANN on the non-stationary noise components that are difficult to estimate and treat with a traditional digital signal processing approach, and we are able to reduce artifacts that are typical of stationary noise suppressors, such as musical noise. Third, by working with gain filters we make the feature space smaller and sparser in comparison to power spectrograms. We target a reasonable balance between noise reduction, residual artifacts, and rejection of sound that does resemble the time-frequency structure of speech.

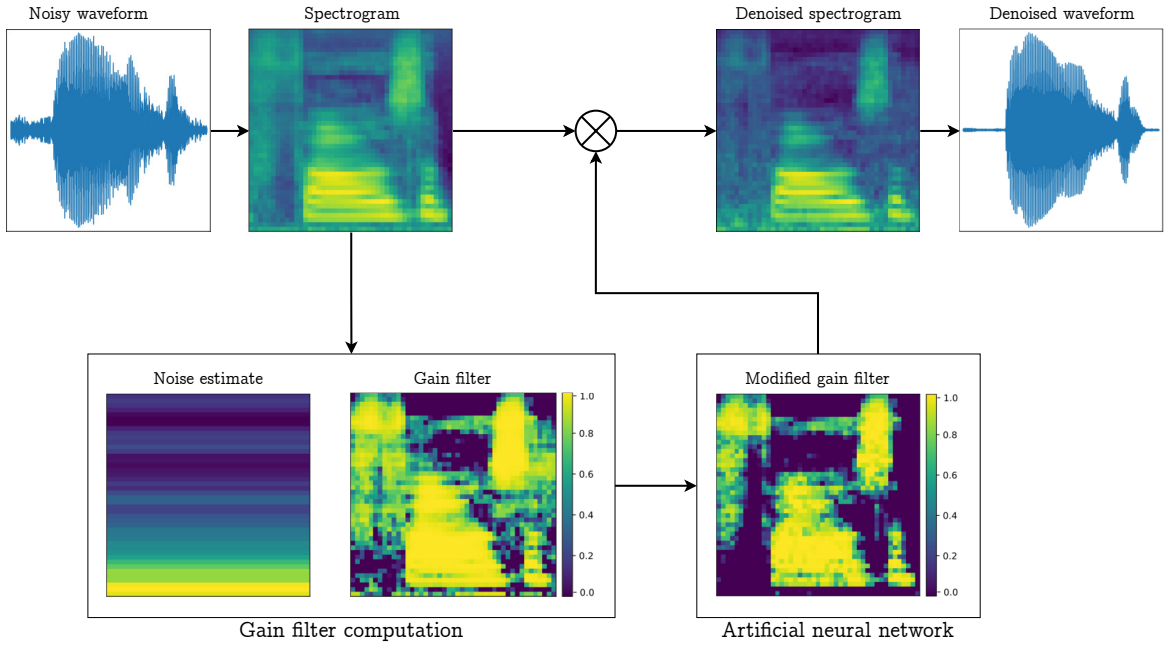


Figure 1: Schematic representation of the proposed noise suppressor.

2.1. Dataset

We use a publicly available dataset [6] in order to train, validate and test the proposed ANNs. This dataset contains pairs of noisy and clean recordings with a variety of speakers and background noises. The clean audio clips consist of sentences recorded by different speakers in a noise-free environment, while their noisy counterparts can be expressed by the additive noise signal model:

$$y[n] = x[n] + v[n] \tag{1}$$

where n is the current time index, $x[n]$ is the clean speech, $v[n]$ is the background noise, and $y[n]$ is the resulting noisy speech signal.

2.2. Pre-processing

The goal of the pre-processing is to build a representative feature space for the ANN. For each noisy and clean audio clip, we compute the stationary noise suppressor gain filter and the target gain filter. These gain filters are then post-processed in order to reduce their dynamic range and achieve better data sparsity. With the assumption of $x[n]$ and $y[n]$ being uncorrelated and zero-mean, we can express the relationship between their power spectrograms with the following equation:

$$P_X[j,m] = P_Y[j,m] - P_V[j,m] \tag{2}$$

where $P_X[j,m]$, $P_Y[j,m]$ and $P_V[j,m]$ are the power spectrograms of the clean, noisy and additive noise signal respectively, j is the time frame index of the windowed signal, and m is the frequency index. In order to obtain the gain filters $G[j,m]$, from Equation 2 we can derive the following:

$$G[j,m] = \frac{P_Y[j,m] - P_V[j,m]}{P_Y[j,m]} \tag{3}$$

Based on Equation 3, we compute the two sets of gain filters, the stationary noise suppressor gains G_{ns} and the ideal gains G_{id}. The gain filter G_{ns} is obtained by inserting the noise estimate of the stationary noise suppressor into $P_V[j,m]$, while for the ideal gain filter G_{id} the noise spectrogram is obtained by rewriting Equation 2 as $P_V[j,m] = P_Y[j,m] - P_X[j,m]$. The procedure for obtaining the stationary noise estimate is the following. We first start with a over-estimation of the stationary noise, in order to obtain an initial noise estimate. Then, for each $k-th$ sub-band, we compute the minimum between the current stationary noise estimate and the mean of the temporal analysis frame of each sub-band, which is comprised of $N = 6$ time-frames. The longer is the audio segment used, the higher is the accuracy of the stationary noise estimate. The advantage of this method is that we do not need a voice activity detector in order to separate the speech from the background noise.

To achieve better data sparsity, the ideal gains are post-processed by computing the element-wise minimum between the ideal gains G_{id} and the non-stationary noise removal gains G_{ns}, resulting in the target gains G_{tg}. As a last step, we limit the dynamic range of G_{ns} and G_{tg} by clamping each $j-th$ mel-frequency component with a value lower than $att_{lin} = 10^{\frac{att_{dB}}{20}}$ to att_{lin}, resulting in a range $att_{lin} \in [att_{lin}, 1]$. This range is then rescaled to $[0, 1]$ with the following equation:

$$D[j,m] = \frac{G[j,m] - att_{lin}}{1 - att_{lin}} \tag{4}$$

where $D[j,m]$ is the resulting data. The neural networks are therefore trained to find a mapping between the noise suppressor data D_{ns} and the target data D_{tg}.

2.3. Proposed networks

We test two main families of artificial neural networks: denoising convolutional autoencoders (DCAEs) and recurrent neural networks (RNNs). Autoencoders (AEs) [3] are neural architectures that introduce a bottleneck, which forces the network to create a compressed representation of the input, also called as latent vector. This way, the neural network discards most non-essential information when learning a mapping between its input and the target. Originally proven useful for unsupervised learning, autoencoders can be also employed for the task of noise removal. Unlike AEs, recurrent neural networks (RNNs) have the ability to model the evolution of data over time thanks to having an internal memory, which is composed of hidden units. The performance of the basic RNN, however, degrades quickly when dealing with long-term dependencies because of vanishing and exploding gradient problems, which has led to the development of long short-term memory recurrent networks (LSTMs) and gated recurrent units (GRUs). By introducing a forget gate, LSTMs and GRUs are able regulate what information should be kept or discarded in modeling a sequence and are able to achieve more efficient and predictable training results. Since GRUs are able to achieve a similar performance to LSTMs but require a smaller amount of parameters, we choose to test them GRUs in our proposed system.

CAE The convolutional autoencoder network (CAE) is designed as a fully symmetrical encoder-decoder. The encoder is comprised of three convolutional layers with a decreasing output size and an increasing number of channels, followed by two fully connected layers of decreasing output size. Conversely, the decoder starts with two fully connected layers with increasing output size, followed by three convolutional layers. The output size of the last convolutional layer of the decoder is the exact same size as the input layer of the encoder.

CAESkip The architecture of this network is similar to CAE, but it introduces a skip connection between the second convolutional layer of the encoder and the second convolutional layer of the decoder. This reduces the risk vanishing gradients and degradation of high-level features.

CAESkipRT This neural network is an asymmetrical autoencoder. Similarly to CAESkip, the encoder is composed of three convolutional layers, two fully connected layers, and a skip connection that taps in between the second and third convolutional layers. The decoder, however, is designed to output only the gain filter corresponding to the latest temporal frame.

GRUNet We implement a lightweight recurrent network based on five stacked gated recurrent unit (GRU) layers. Each GRU layer computes the function defined in the following equations:

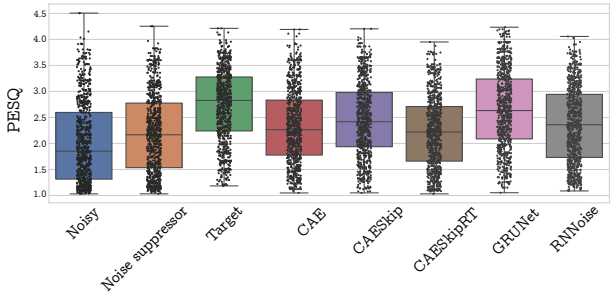
$$\begin{aligned} r_t &= \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{(t-1)} + b_{hr}) \\ z_t &= \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_{(t-1)} + b_{hz}) \\ n_t &= \tanh(W_{in}x_t + b_{in} + r_t \odot (W_{hn}h_{(t-1)} + b_{hn})) \\ h_t &= (1 - z_t) \odot n_t + z_t \odot h_{(t-1)} \end{aligned}$$

where $\odot$ is the Hadamard (element-wise) product, σ and tanh are the sigmoid activation and hyperbolic tangent activation functions respectively, r_t, z_t and n_t are the reset, update and new gates respectively, h_t is the hidden state of the layer at the time instant t, x_t is the input at the time instant t, W are the weight matrices, and b are the bias values. Compared to convolutional layers, GRUs focus on capturing the evolution of gain filters across time, instead of finding time-frequency independent noise patterns in the 2D

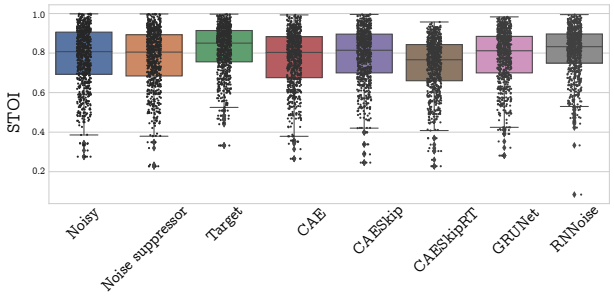
spectrogram image that span over a significant number of time frames. This makes it possible to reduce the input dimensionality and the memory requirements of our system, because the input layer of GRUNet is supplied with gain filters that correspond to a single time frame only. Additionally, the lower dimensionality of the GRU layers that compose GRUNet imply a significant reduction in computational complexity.

3. Results

To evaluate the performance of our setup, we use the testing set provided in [6]. The speakers, the types of noise and the SNR levels are different than those contained in the training set, which is particularly suitable for evaluating the generalization properties of the trained ANNs and the robustness of the pre-processing procedure. To perform a comparison between the clean and the different denoised version of each audio file, we use two objective metrics: the perceptual evaluation of speech quality (PESQ) and the short-term objective intellegibility (STOI). These metrics are computed across the entire testing set, and we analyze their distributions in order to evaluate the outputs of our stationary noise suppressor, target gain filters, and the proposed ANN models. Our comparison also includes RNNoise[1], which is presented in [5]. In terms of PESQ, we can observe from Figure 2a that our stationary noise suppressor improves the overall quality of the testing set, and that our target gain filters provide another significant improvement step, which validates our problem formulation. Compared to the other ANNs, we can clearly see that GRUNet is able to outperform all three convolutional autoencoders (CAE, CAESkip, and CAESkipRT) and RNNoise, achieving a mean PESQ score of 2.657, which is significantly closer to that of the target (2.734) rather than the stationary noise suppressor (2.193). STOI, on the other hand, shows no improvement in the signals outputted the stationary noise suppressor, but still assigns the highest score to the signals that are denoised with the proposed target gain filters, which is followed by RNNoise and GRUNet. Since GRUNet and RNNoise achieve the best score in terms of PESQ and STOI respectively, we compare their computational complexity in Table 1, which shows that GRUNet has lower requirements in terms of model weights and a significantly lower computational cost. Compared to RNNoise, the number of parameters of GRUNet is 32% lower, and the decrease of multiply-accumulate operations for one prediction amounts to 187%.



(a) PESQ results.



(b) STOI results.

Figure 2: Evaluation results on the testing set across all noise types and SNR values.

ANN model	Parameters	MACs
CAE	369'577	6'988'704
CAESkip	133'513	6'785'760
CAESkipRT	68'644	1'337'056
GRUNet	**59'400**	**60'940**
RNNoise	87'503	175'000

Table 1: Memory requirements (number of parameters) and computational complexity (MACs/inference) comparison for the tested neural networks.

4. Conclusion

This manuscript presents a low-complexity system for real-time noise reduction of speech signals that combines a stationary noise suppressor with an artificial neural network (ANN). The task of the stationary noise suppressor is to com-

[1] Compiled from `https://gitlab.xiph.org/xiph/rnnoise`, commit hash `7f449bf8`

pute gain filters based on a statistical noise estimate, while the ANN enhances the aforementioned filters in order to achieve a higher level of noise reduction. The enhanced filters can also remove residual artifacts that are potentially caused by gain filtering. We implement several ANN architectures and our results show that a multi-layer recurrent network based on gated recurrent units (GRUs) is able to outperform convolutional autoencoders with a small fraction of their computational complexity. Additionally, we compare our proposed approach against RNNoise [5], a denoising system that has been designed with similar goals in mind, i.e. low complexity, causality, and real-time processing. The comparison is performed with two objective metrics, the perceptual evaluation of speech quality (PESQ) and the short-time objective intelligibility (STOI), which assess the degradation of a signal relatively to the original signal. Compared to RNNoise, our approach achieves a higher PESQ score and a slightly lower STOI score with a fraction of the computational cost and significantly lower memory requirements. Future research will mainly focus on two goals. First, we will explore the possibility of enhancing the extracted features, with the aim of computing target filters with higher perceptual quality and stronger noise suppression. Second, we will endeavor in further reducing the overall complexity of the proposed system.

References

[1] Steven F. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(2):113–120, 1979. doi: 10.1109/TASSP.1979.1163209.

[2] Rainer Martin. Noise power spectral density estimation based on optimal smoothing and minimum statistics. *IEEE Transactions on Speech and Audio Processing*, 9(5):504–512, 2001. doi: 10.1109/89.928915.

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*, chapter 14. Autoencoders, pages 507–512. MIT Press, 2016. doi: 10.1007/s10710-017-9314-z.

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*, chapter 10. Sequence Modeling: Recurrent and Recursive Nets, pages 367–415. MIT Press, 2016. doi: 10.1007/s10710-017-9314-z.

[5] Jean-Marc Valin. A hybrid dsp/deep learning approach to real-time full-band speech enhancement. In *2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–5, 2018. doi: 10.1109/MMSP.2018.8547084.

[6] Cassia Valentini Botinhao. Noisy speech database for training speech enhancement algorithms and tts models, 2017. doi: 10.7488/ds/2117.