Removing Noise from Speech with Deep Learning

Glávits Balázs
Department of Measurement and Information Systems
Budapest University of Technology and Economics
glavits.balazs@gmail.com

Kiss Andor
Department of Automation and Applied Informatics
Budapest University of Technology and Economics
kissandor4@gmail.com

Konrád Márk
Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics
konrad0816@gmail.com

Abstract

In this paper we will present some already existing state of the art models for speech denoising and a detailed description of our own solution to the problem. After that we discuss a few alternative non-working solutions that we tried out, and some future plans to improve our proposed model.

Deep Learning in Practice with Python and LUA (2020)

1 Task summary

Our goal is to reduce (or in the best case, entirely remove) the noise from audio recordings containing noisy speech. The idea is to use the WaveNet[3] architecture to generate clean audio from the noisy one.

2 Different approaches

In the research phase of our task we found several different implementations of speech denoising deep neural networks. The best ones are:

- **Speech Enhancement Generative Adversarial Network (SEGAN)**[4], which is a GAN based approach where the generator receives the noisy data with a latent representation and the discriminator is just a binary classifier.
- **Speech Enhancement based on Denoising Autoencoder with Multi-branched Encoders** [7]
- **WaveNet**[3], which is a generative model aimed at creating raw audio waveforms. We experimented with several implementations. These can be seen in sections 5 and 6.

After considering these different solutions we decided to create an implementation based on the WaveNet architecture.

3 Data acquisition and exploration

For our training and testing data, we used a dataset called "Noisy speech database for training speech enhancement algorithms and TTS models"[6] by the University of Edinburgh. It consists of ∼23000 clean-noisy pairs (Figure 2) from 56 different speakers. The samples are stored in separate .wav files of varying length (Figure 1).

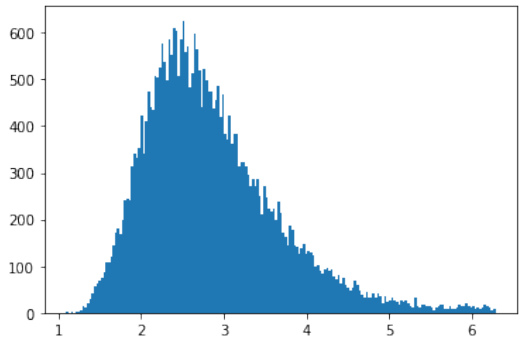
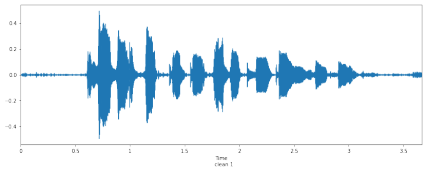
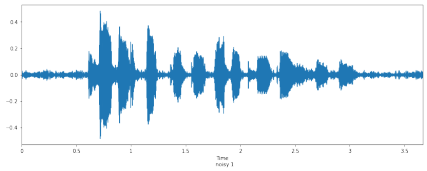


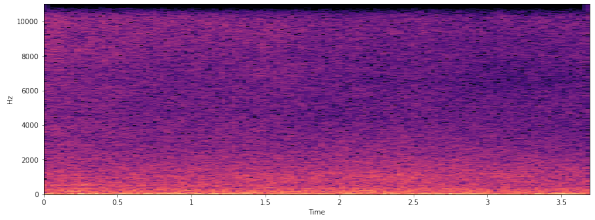
Figure 1: Duration distribution histogram of a subset of speeches



(a) clean



(b) noisy

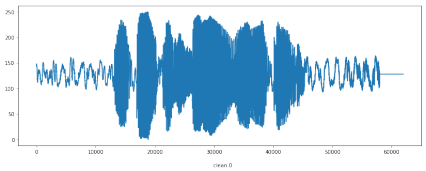


(c) noise spectogram

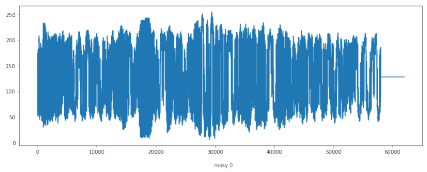
Figure 2: Noisy and clean samples and the spectogram of the noise from the audio in figures 2a and 2b

4 Data preprocessing

Our first approach was to select the n closest audio samples and zero-pad them to be the same duration, then reduce the samples to 8 bit with μ-law transformation (figure 3).



(a) clean



(b) noisy

Figure 3: An example clean-noisy pair after padding and μ-law transformation

With this implementation we ran into the problem of limited hardware resources so, we had to come up with a less resource-intensive preprocessing pipeline.

Our solution was to load the raw data, normalize it between -1 and 1 and downsample it to 16kHz. After these preprocessing steps, we feed the audio to the model with a data generator which can generate batches of smaller same sized pieces of data. With this approach, we eliminated the memory problem but sacrificed some of the continuity in our speech samples. The other upside of this solution is that we don't have to feed the meaningless data to the model in the form of zeroes.

The original implementation of the WaveNet architecture used one-hot encoded μ-law transformed data, but after experimenting with these, we could not generate audio with acceptable quality.

5 The WaveNet architecture

For our network architecture, we choose to use a modified version of the WaveNet[3] architecture developed by DeepMind.

WaveNet is a deep neural network capable of generating raw audio waveforms. This can be achieved with the use of a dilated causal convolutional layers. Causal means that the network is only conditioned on past and current inputs. With this approach, we can make sure that the ordering of the data is not violated. With dilation, we can achieve a large receptive field with the preservation of the input resolution. The structure of a stack created with these kinds of layers can be seen in Figure 4.

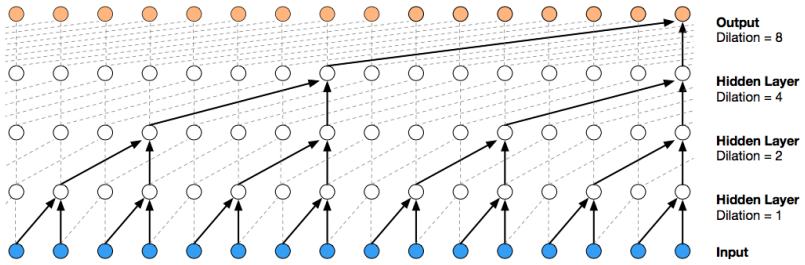


Figure 4: A dilated causal convolutional layer stack

The model uses residual blocks and skip connections to speed up the convergence and enable the use of deeper networks.

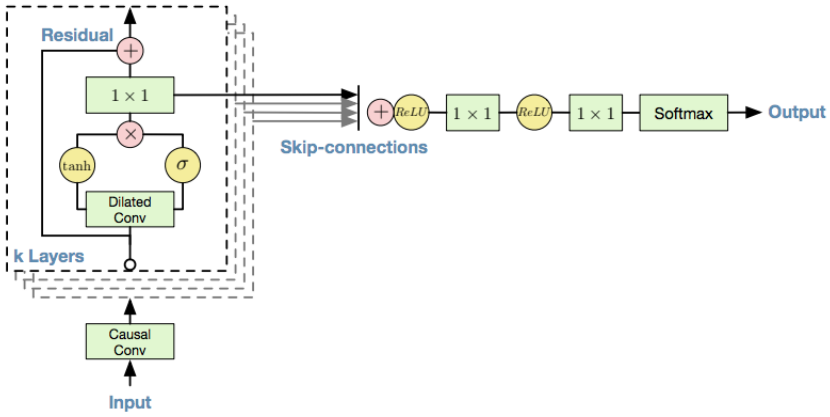


Figure 5: Residual blocks with skip connections in WaveNet

To avoid making assumptions of the output shape the model uses discrete softmax distribution. The overview of the architecture can be seen in Figure 6.

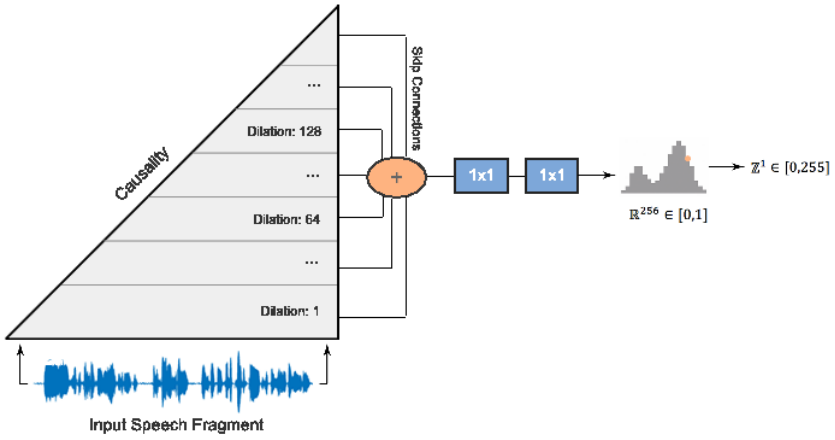


Figure 6: A simplified overview of the WaveNet architecture

It must be noted that this model is primarily designed to be used in Text-to-Speech (TTS) applications.

5.1 WaveNet for speech denoising

While the WaveNet architecture proposed by DeepMind is a state of the art implementation for TTS applications its current form is not suitable for speech denoising. Rethage et al. [5] proposed a WaveNet adaptation for the purpose of removing noise from speech samples (Figure 7).

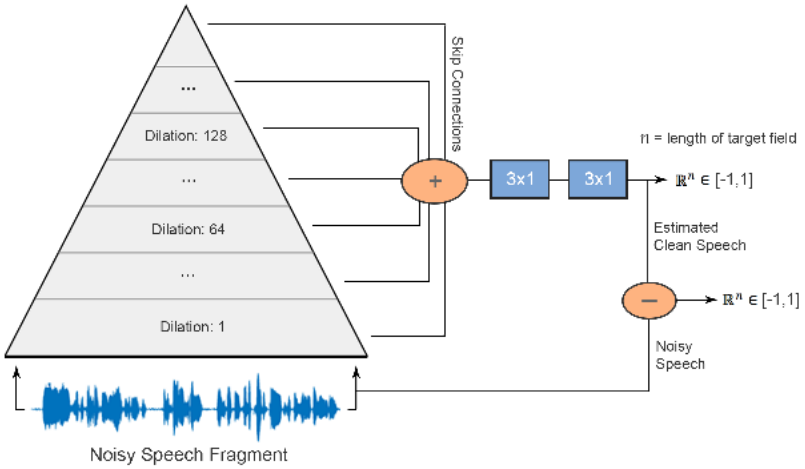


Figure 7: A simplified overview of the WaveNet speech denoising architecture

The original WaveNet architecture uses causal convolutional layers for keeping the ordering of the data, but in speech denoising a substantially more accurate prediction can be achieved with non-causal convolutions. This change essentially doubled the context available to the model (Figure 8).

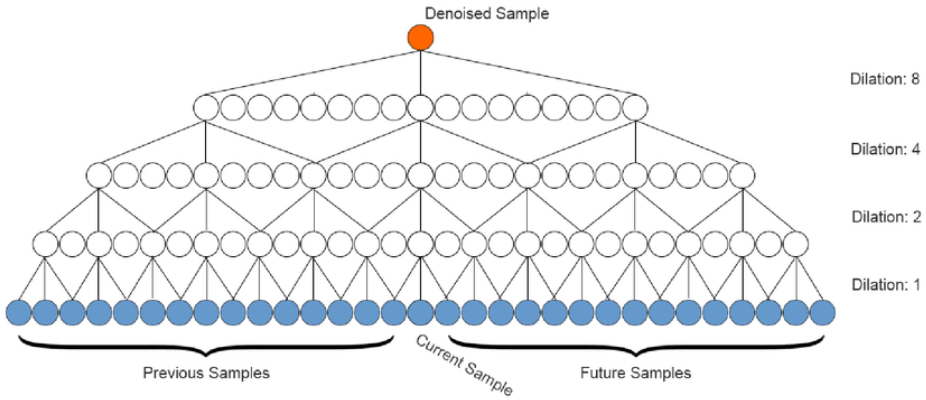


Figure 8: A dilated non-causal convolutional layer stack

5.2 Our denoising WaveNet implementation

To solve the problem proposed in section 1, we created our denoising WaveNet implementation. It consists of a stack of dilated non-causal convolutional layers with skip connections and three additional one-dimensional convolutional layers (Conv1D) to incorporate features extracted at every hierarchical level. The first two Conv1D layers have 2048 and 256 filters with the kernel size of three and, the last one has one filter with the size of one. This structure can be seen in figure 9. The skip connections (Figure 5) use the ReLU activation function. We tried out the HeNormal kernel initialization proposed by He et al. [1] but, it could not provide any improvement so, we stayed with the previously used Glorot uniform initialization. We tried to use different optimizers like Adam and

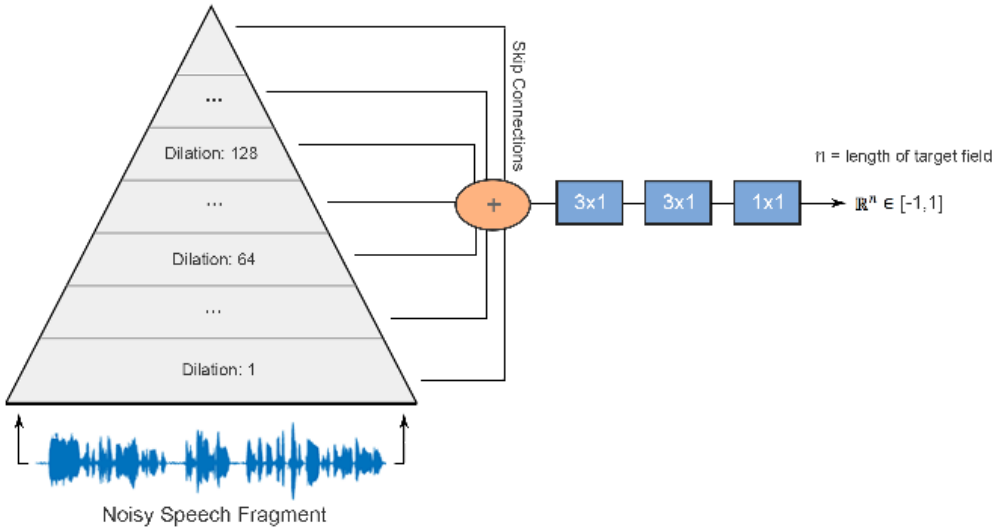
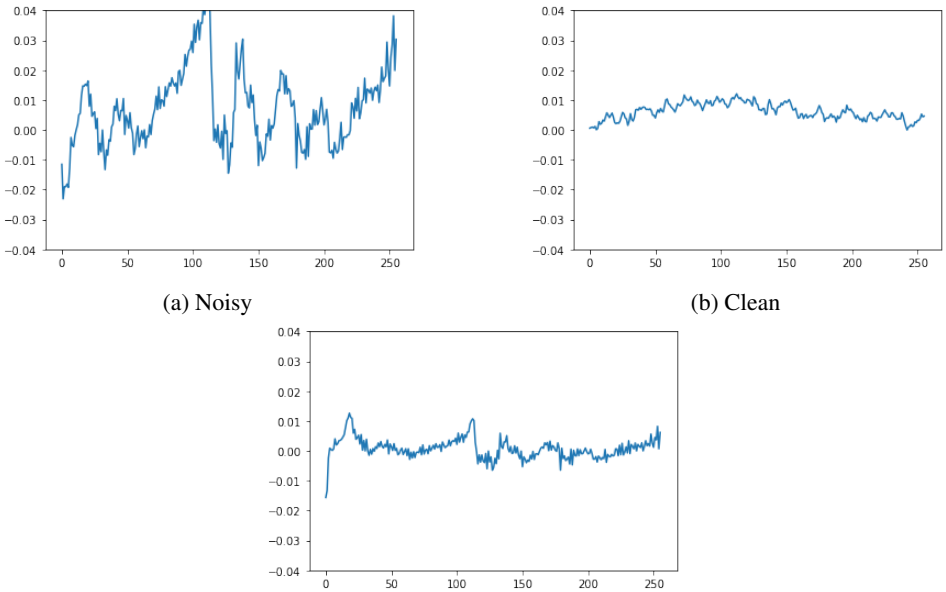


Figure 9: Overview of the denoising WaveNet architecture

SGD. For a smaller volume of data the Adam optimizer, for larger the SGD optimizer was the better choice. We also tried the AdamW[2] optimizer, but it could not improve the output.

While training the network we reduced the learning rate if there was no improvements for several epochs. We saved the model after every epoch because in this case validation loss is not necessarily the best metric to track improvement and we did not want to lose a potentially good model.

The model's input is a batch of a 2048 float long (the sample rate of the data is 16kHz so, the input is 128 milliseconds long) noisy slices from the dataset provided by a data generator, and the output should be the same 2048 long sample without noise. A zoomed-in test run can be seen in Figure 10.



(a) Noisy

(b) Clean

(c) Output

Figure 10: An example noise reduction

6 Experiments

- **WaveNet with non-causal convolutions, μ-law transform and softmax distribution**
  At first, we tried to create a denoising model with dilated non-causal convolutional layers and, μ-law transformed one-hot encoded input data. After some test runs, we decided to try another approach because, with this attempt, we could not reduce the noise on the samples at all.
- **WaveNet with non-causal convolutions, regression, and dense output layer**
  After the first failed experiment, we tried to skip the μ-law transform step and move towards a regression-based model. In this attempt, we used dilated non-causal convolutions with a dense layer for output. The resulting quality was questionable because the network removed part of the clean speech along with some of the noise.
- **WaveNet with non-causal convolutions, regression, and flatten + dense output layers**
  This model was the first that could provide reasonably good output. The thought process behind the addition of a flatten layer was that with it there are be more connections in the network, and the output weights will be conditioned on the output of the convolutional layers and will eventually generate clean speech. It could remove most of the noise, but the quality of the speech was much worse than the input. The other downside of this model was that the training was extremely slow because of the size of the network.
- **WaveNet + auto-encoders**
  We tried to stack a denoising auto-encoder on top of a non-causal WaveNet because we thought that by giving the latent space from the WaveNet's output to an auto-encoder, we could decrease the noise even more. Unfortunately, this attempt could not even replicate the input speech.
- **Auto-encoder surrounded with WaveNets**
  Another attempt consisting of a combination of WaveNets and an auto-encoder was the idea of surrounding an auto-encoder with two identical WaveNets. We experienced a steady decrease in the model's loss, but unfortunately, we did not have the resources and time to test this model more.
- **WaveNet with non-causal convolutions, regression, and extra one dimensional convolutional layers on the output**
  Our final model is a WaveNet with non-causal convolutions followed by three one dimensional convolutional layers with filter sizes three, three, one. The output is promising because it can eliminate most of the noise and, the deterioration of the sound quality is much better than any of our past attempts. With more computation time on a machine with specialized resources, there is a chance that the sound quality will improve more. For more information about this model, refer to section 5.2.

7 Future plans

Since hyperparameter optimization is usually a never-ending task, obviously one of our plans for the future is to optimize our best-performing model, to see if we can denoise speech even better, and potentially to reduce the loss of sound quality when denoising. We tried out some other architectures as well, which have shown promise, especially the auto-encoder surrounded with wavenets. In the future, if we have more time and computing power in our hands, we can train that model for a long time, if it shows some better results, we can also start our never-ending journey of optimizing the hyperparameters of it as well.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: (2015). arXiv: 1502.01852 [cs.CV].

[2] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2019. arXiv: 1711.05101 [cs.LG].

[3] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. "WaveNet: A Generative Model for Raw Audio". In: (2016). arXiv: 1609.03499 [cs.SD].

[4] Santiago Pascual, Antonio Bonafonte, and Joan Serrà. "SEGAN: Speech Enhancement Generative Adversarial Network". In: (2017). arXiv: 1703.09452 [cs.LG].

[5] Dario Rethage, Jordi Pons, and Xavier Serra. "A Wavenet for Speech Denoising". In: (2018). arXiv: 1706.07162 [cs.SD].

[6] Cassia Valentini-Botinhao. "Noisy speech database for training speech enhancement algorithms and TTS models". In: (2017). DOI: 10.7488/DS/2117.

[7] Cheng Yu, Ryandhimas E. Zezario, Jonathan Sherman, Yi-Yen Hsieh, Xugang Lu, Hsin-Min Wang, and Yu Tsao. "Speech Enhancement based on Denoising Autoencoder with Multi-branched Encoders". In: (2020). arXiv: 2001.01538 [eess.AS].