E-005

Step-by-Step Speech Enhancement Using Stacked Hourglass Wave Networks

Aliaksei Khadanovich‡ Shuichi Arai‡

Abstract

We propose a novel method for speech enhancement, which is based on Stacked Hourglass Networks by Park Sungheon et al. [3]. The proposed method operates directly in the time domain and it enables to train to reduce different types of noise step-by-step rather than at once. This idea was achieved only through a stacked feature of the baseline model, which consists in connecting several encoder-decoder modules in series. The experimental results on Voice Bank corpus (VCTK) dataset evaluation show that our method achieves competitive results compared to the other methods in the speech enhancement task.

1. Introduction

Audio source separation is a complicated task which consists of separating one or more audio sources from a mixture of the sources [1, 2, 3]. One of the related tasks is speech enhancement [4, 5, 6].

Recently proposed methods [4, 5, 6] work directly on the time domain and process a denoising routine at once, from the noisy speech to the clean speech. We came to the idea to process the denoising routine not at once but step-by-step to reduce the noise gradually. To realize this idea we used the stacked feature of the Stacked Hourglass Networks by Sungheon et al [3].

2. Method

2.1 Model

Our model is based on Stacked Hourglass Networks [3], which was originally used for music source separation task. Baseline model uses spectrograms as an input and produces soft masks for an output. To overcome spectrogram based models' limitations, which were mentioned in the Wave-U-Net by Stoller et. al [1], the input and the output of the network were changed to the time-domain audio signal.

The original Stacked Hourglass Networks model is based on several encoder-decoder modules, which are stacked consistently, and an initialization block. In our experiments, we found that for the wave based model initialization block is unnecessary and it was removed. Also we found that using bottleneck encoder-decoder significantly decreases the number of training parameters, while the evaluation results have not changed compared to the baseline architecture. Our architecture of the one module is shown in Figure 1 and overall architecture of the model is shown in Figure 2.

Encoder contains *N*-downsampling blocks. Each downsampling block is based on a 1-dimensional convolutional layer (Conv1D) with kernel size equals to 11 and stride equals to one. After the convolutional layer comes a batch normalization layer and a downsampling layer. Convolutional layer with stride two and same kernel size equals to 11 was used for the downsampling layer. From the first downsampling block to the *N*-th downsampling block, channel size of the convolutional layer is growing from 32 with step 16.

Decoder contains *N*-upsampling blocks. Each upsampling block consists of an upsampling layer, the 1-dimensional convolutional layer with the same parameters as in the downsampling blocks and batch normalization layer. For the upsampling layer, bilinear interpolation was used.

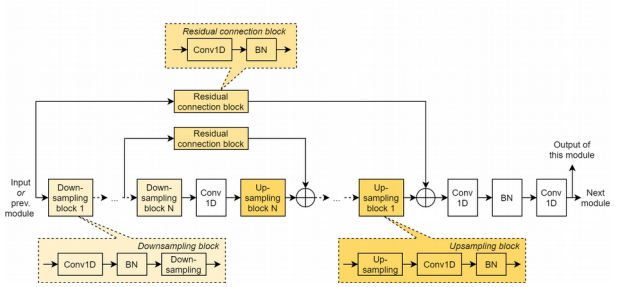

Figure 1: Architecture of the one encoder-decoder module.

For the residual connection, the 1-dimensional convolutional layer with the same parameters as in the downsampling and upsampling blocks and batch normalization layer were used. Addition was used to merge the upsampling block output and the residual connection.

Two 1-dimensional convolutional layers follow after the last upsampling block. The first convolutional layer has the same parameters as in the downsampling and upsampling blocks. The second convolutional layer has the same parameters as in the downsampling and upsampling blocks, but with channel size two to produce the output of the encoder-decoder module. Output contains target speech and target noise separately. In our experiments, we used four downsampling and upsampling blocks, because this setup shows the best results.

To stack several modules in series, we used a 1-dimensional cropping layer and cropped noise output of the modules.

2.2 Training process

After several experiments, we found that we had noise artifacts, which is produced by the model itself on the joining point between frames. To overcome this problem, we used triangle window and half frame overlap on our data in the training and evaluation stages.

Each module of the network was trained independently on its own data. Data was prepared for each module by the following rule: signal-to-noise ratio (SNR) of the output training data is five more than SNR of the input training data. For the last module, the output training data is clean speech.

After that, we stacked all modules in series and performed fine-tuning. Since each module outputs enhanced speech and noise and input for the next module is only noisy speech, we cropped noise output of all modules except the last one.

The overall loss of the model was calculated by the weighted summary of the all modules' losses as

$$L_{all}=\alpha_1 \cdot L_1+\alpha_2 \cdot L_2+...+\alpha_N \cdot L_N, \quad (1)$$

where L_{all} - overall model loss, $L_1 \sim L_N$ - loss of each modules, N - number of modules and $\alpha_1 \sim \alpha_N = 1.0$ - loss weights coefficients. In the fine-tuning stage same as in the module independent training stage, the output of each module is compared to the training data for this module.

3. Experiments

3.1 Experimental setup

Mean square error (MSE) was chosen as a loss function same as in the [6]. RMSprop was chosen as an optimizer. Default

‡ 東京都市大学大学院 Tokyo City University Graduate School

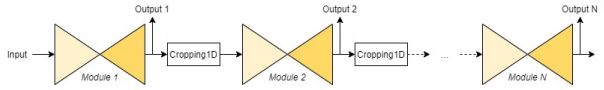

Figure 2: Architecture of the in series stacked modules.

parameters of Keras framework were left unchanged: learning rate – 0.001, gradient moving average – 0.9. Learning rate was reduced by 0.1 factor every two epochs without improving in validation loss till 0.0000001 value. Training was stopped after five epochs without improving in validation loss and the best validation loss model was restored. We used frame size equal to 8192. Batch size of 64 was used to train modules independently and batch size of 16 was used for fine-tuning process.

3.2 Evaluation method

824 files from the test set were used to evaluate our models. Evaluation was done based on Perceptual Evaluation of Speech Quality (PESQ) metric [8] same as in Wave-U-Net for Speech Enhancement [6] and SEGAN [5]. PESQ metric value lies within -0.5 and 4.5.

3.3 Dataset

We used the same Voice Bank corpus (VCTK) dataset [7] as in Wave-U-Net for Speech Enhancement [6] and SEGAN [5]. This dataset contains records of 28 English-speakers for the training set and records of two English-speakers for the test set.

Train set contains mixes of the clean speech and 40 different types of noises (11,572 items at all). These noises contain 10 types of noise in 4 different signal-to-noise ratios (SNR) (0, 5, 10, 15 dB).

Test set contains mixes of the clean speech and 20 different types of noises (824 items at all). These noises contain 5 types of noise (different from train set) in 4 different signal-to-noise ratios (SNR) (2.5, 7.5, 12.5, 17.5 dB).

4. Results and Conclusion

4.1 Experiment 1

In order to check whether the architecture of our module is competitive with Wave-U-Net [6] architecture, we trained only one module using noisy speech as input and clean speech and noise as output. Table 2 shows the results of the PESQ metric of the dataset itself (Noisy), competitors (SEGAN [5], Wave-U-Net [6]) and our own experiments (Experiment 1 and Experiment 2).

Refer to the results of Experiment 1, our one module architecture is effective and competitive to Wave-U-Net [6] architecture.

4.2 Experiment 2

Since we received promising results in Experiment 1, we trained five modules based on the same module architecture as in Experiment 1. These modules were trained independently on our own “step” dataset. Detailed SNR levels for each module of input data and output data are indicated in Table 1. After that, all modules were stacked in series and fine-tuning was performed. Experimental results are shown in Table 2.

Experiment 2 showed that our five modules architecture is less effective compared to our one module architecture (Experiment 1) and Wave-U-Net [6] nevertheless the last module’s validation loss of the five modules model is lower than the validation loss of the one module model.

We think that this is due to the fact that we crop the noise after each module and use only an enhanced speech as an input for the next module. Therefore, backpropagation loses its effectiveness and still a lot of noise remains in the output signal. Also, we came to the conclusion that MSE loss is not suitable for training time-domain based networks, since the validation loss of the last output of the five modules model is lower than the validation loss of the last output of the one module model. It has to be investigated in future works.

Table 1: Input and output SNR levels of Modules

Module	Input SNR, dB	Output SNR, dB
Module 1	0, 5, 10, 15	5, 10, 15, 20
Module 2	5, 10, 15, 20	10, 15, 20, 25
Module 3	10, 15, 20, 25	15, 20, 25, 30
Module 4	15, 20, 25, 30	20, 25, 30, 35
Module 5	20, 25, 30, 35	clean

Table 2: PESQ comparison table.

Model name	PESQ
Noisy	1.97
SEGAN	2.16
Wave-U-Net	2.40
Experiment 1 (our)	**2.44**
Experiment 2 (our)	2.37

5. Future works

The results show that the effectiveness of our model still has ability to grow by adjusting step parameters and modules number. We are thinking of changing the overall architecture and the training process, so that each module is able to use not only the enhanced output speech of the previous module but also the output noise from the previous module. Therefore, the backpropagation process will be simplified. Also, it is necessary to investigate possible replacement for MSE loss, which will be more suitable for training time-domain models.

References

[1] Stoller, Daniel et al. “Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation.” ArXiv abs/1806.03185 (2018): n. pag.

[2] Jansson, Andreas et al. “Singing Voice Separation with Deep U-Net Convolutional Networks.” ISMIR (2017).

[3] Park, Sungheon et al. “Music Source Separation Using Stacked Hourglass Networks.” ISMIR (2018).

[4] Rethage, Dario et al. “A Wavenet for Speech Denoising.” 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2018): 5069-5073.

[5] Pascual, Santiago et al. “SEGAN: Speech Enhancement Generative Adversarial Network.” ArXiv abs/1703.09452 (2017): n. pag.

[6] Macartney, Craig and Tillman Weyde. “Improved Speech Enhancement with the Wave-U-Net.” ArXiv abs/1811.11307 (2018): n. pag.

[7] Valentini-Botinhao, C. (2017). Noisy speech database for training speech enhancement algorithms and TTS models, 2016 [sound]. University of Edinburgh. School of Informatics. Centre for Speech Technology Research (CSTR). https://doi.org/10.7488/ds/2117

[8] P.862.2: Wideband extension to Recommendation P.862 for the assessment of wideband telephone networks and speech codecs, ITU-T Std. P.862.2, 2007.