UNSUPERVISED SPEECH ENHANCEMENT WITH AUTO-ENCODER [REJECTED]

Ren-Chu Wang *Hung-Yi Lee*

College of Electrical Engineering and Computer Science, National Taiwan University
{b06901038, hungyilee}@ntu.edu.tw

ABSTRACT

Speech enhancement is a critical technique often used in preprocessing audio speech samples to improve its quality. Machine-learning based methods, while boasting much better performance, requires a lot of paired data which can be challenging to collect. In this work, we show the possibility of unsupervised speech enhancement with only clean or noisy audio by exploring the idea of taking advantage of the imperfection of auto-encoders. Experiments show that auto-encoders trained this way, although not as performant as supervisedly trained auto-encoders, works much better than expected.

Index Terms— Unsupervised, Speech Enhancement

1. INTRODUCTION

Speech enhancement aims to improve the quality of target speech, limiting the effect of unwanted background interference. Such an idea is at the very core of speech-related audio signal processing and is used in numerous practical applications, such as automatic speech recognition (ASR), hearing aids, and communication systems.

Before the prevalence of machine learning and deep learning, many classical algorithms, designed with such goal in mind, have been proposed. These algorithms include spectral subtraction [1], minimum mean-squared-error stort-time spectral amplitude (MMSE-STSA) estimator [2], a priori signal to noise estimation [3]. Most of the algorithms focus on the statistical difference between speech and noise, but they require some prior conditions to be satisfied, which may be unrealistic in practice. Hence, these algorithms often fail to generalize nicely to real-world-scenarios.

More recently, we see a huge improvement in the performance of such tasks with the aid of neural-network-based approaches [4, 5, 6]. These works take advantage of the nonlinearity characteristic of neural networks to better approximate the mapping from noisy speech to their clean representation. With a large enough number of pairs of clean and noisy inputs, neural networks are able to learn to reconstruct clean speech from the noisy input.

Notably, in addition to recurrent neural networks (RNNs) [7, 8], convolutional neural networks (CNNs) [9, 10, 8], generative adversarial networks (GANs) [11] are also introduced in this task [12, 13] to further tackle the problem that previously proved difficult to solve. RNNs (long short term memory (LSTM), or gated recurrent unit (GRU) in particular) model the dynamic, dependent nature of a time sequence, and have been established as the go-to approaches for time sequence modeling. Due to their stateful property, they can better avoid gradient vanishing and exploding, compared to vanilla RNN, and are rather gap-insensitive (compared to vanilla RN2N). CNNs model the relative, local relation between nearby tokens, and are introduced to this task a bit later. GANs are neural networks competing with one another in a game that reaches Nash equilibrium upon convergence. With supervised or Semi-supervised learning, they can learn a mapping from noisy speech to their clean version.

All of the approaches mentioned above require an enormous amount of paired clean and noisy data during the training phase. Only a handful of papers explore the idea of unsupervised machine-learning-based speech enhancement. With GAN, it is possible to achieve speech enhancement with non-parallel clean and noisy corpora, but audio data from both domains are still indispensable [14].

However, in practice, the clean data may not be available as it is tough to obtain completely clean data that is undisturbed by the environment noise. It is possible to train a speech enhancement model without clean speech, but though the noise distribution has to be known [15]. The deep prior network can achieve unsupervised speech enhancement with noisy data only, but it is computationally intensive during the inference time [16].

In this study, we further simplify and explore the idea of unsupervised machine-learning-based speech enhancement. We propose two new audio noise removal methods that only need either clean and noisy audio, and do not require any prior knowledge of input distribution. The proposed approach takes advantage of imperfect auto-encoders. We show that these methods, while profoundly simple, yield surprising results, being not to far off our benchmark, supervised learning, in most tasks, while superior in some tasks. This reveals some interesting properties of the neural-network-based approximated mapping.

2. METHODS

In this work, we focus on the methods to train a speech enhancement model with either clean and noisy audio. The method in section 2.2 only needs clean data, while in section 2.3 only noisy audio is required. We show the formal explanation of the two approaches in section 2.4.

2.1. Background

In typical deep-learning-based audio speech enhancement, supervised learning is used, in which a function approximator of the form of a neural network, denoted J_Θ, Θ being the neural network's weight, is defined. The objective of the approximator is to minimize the objective function

$$\mathcal{L}_{\mathcal{N}\in\mathcal{C}} = \mathrm{E}[\mathcal{L}(T, J_\Theta(I))] \tag{1}$$

in which $\mathcal{L}_{\mathcal{N}\in\mathcal{C}}$ defines a distance between a clean target T and a transformed ouptut $J_\Theta(I)$, with I being the noisy input.

2.2. Clean To Clean

In the first case where only clean data is used, we define a mapping approximated by a function J_{Θ^C} with weight parameters C that takes clean input and reconstruct them. That is, we define an objective function

$$\mathcal{L}_{\mathcal{C}\in\mathcal{C}} = \mathrm{E}[\mathcal{L}(I, J_{\Theta^C}(I))] \tag{2}$$

This at first glance looks pointless, but experiments show that if we then substitute the input I that is clean, with a corrupted input I' that is noisy, the network is able to reconstruct an output $J_{\Theta^C}(I')$ that in terms of metrics, is cleaner than the original input I', without having seen the noise $I' - I$ that is added. The intuition here is, during the training, the model learns to generate clean data. Hence, when the input is noisy audio, the output would be the cleaner version of the input.

2.3. Noisy To Noisy

In the second case, only noisy input x is available. Here we train a new mapping approximated by a function J_{Θ^N} with weight parameters Θ^N that maps a noisy sample to its clean version y, but never seen y during the training phase. To obtain that, we define an objective function

$$\mathcal{L}_{\mathcal{N}\in\mathcal{N}} = \mathrm{E}[\mathcal{L}(I, J_{\Theta^N}(I))] \tag{3}$$

The networks are given a noisy distribution, and they are trained to directly reconstruct the noisy input. However, we show in experiments that with an imperfect auto-encoder, this is able to enhance audio quality with some assumptions. That is, $J_{\Theta^N}(I)$ is better in terms of metrics than the original input I. The intuitive explanation here is that the auto-encoders can only predict parts of the audio that is structured, and noises being difficult to predict means that it is not reconstructed.

2.4. Explanation

In this section we provide an explanation as to why our methods work, and how auto-encoders have the ability to remove noise even in cases where we have little information about the noises present. The arguments in this section applies to both section 2.2 and section 2.3. We make the assumption that the noise has zero-mean.

We now introduce some of our notations used in this section for clearer explanation. J_Θ defines a model that is parameterized by its weight Θ. We assume that noisy data can be decomposed to two components, C and N; that is, the noisy input is noted $C + N$. C denotes clean data, and N denotes pure noise. I denotes the input to the model. $I = C$ in *clean-to-clean* setup, and $I = C + N$ in *noisy-to-noisy* setup. T is our target; however, $T = I$ in both of our cases, $\mathcal{L}(x, y)$ is an arbitrary distance function evaluating the distance between x and y and acts as our objective function. By definition of a *distance function*, we have

$$\mathcal{L}(x, x) = 0, \forall x \in R^n \tag{4}$$

During training phase, the network minimizes the following objective loss term

$$\mathcal{L}_{\text{loss}} = \mathrm{E}[\mathcal{L}(T, J_\Theta(I))] \tag{5}$$

By doing so, we encourage the model output $J_\Theta(x)$ to be as close to the target output y.

If equation 5 is minimized; which is to say, the total distance between T and $J_\Theta(I)$ is minimized under the constraint of Θ, we get

$$\mathrm{E}[I] = \mathrm{E}[J_\Theta(I)] \tag{6}$$

However, since we assumed the noise has zero mean; that is, $E[N] = 0$, then we have for case 1:

$$\mathrm{E}[C] = \mathrm{E}[I] = \mathrm{E}[J_\Theta(I)] \tag{7}$$

and for case 2:

$$\mathrm{E}[C] = \mathrm{E}[C + N] = \mathrm{E}[I] = \mathrm{E}[J_\Theta(I)] \tag{8}$$

Though we are aware of the fact that the expectation of the output being the same to the true label, $E[C + N] = E[C]$, does not guarantee N to be zero, empirically it works as demonstrated in the experiments. We also show a trick that could improve performance for *noisy-to-noisy* models. Unfortunately, the same argument cannot be applied to *clean-to-clean* method since the reconstructed audio isn't entirely clean.

First we define a new notation $I' = J_{\Theta_i}(I)$, and $C' = J_{\Theta_{i+1}}(I')$. We pass the data through two mappings that are connected in a head-to-tail fashion.

We rewrite equation 3 as

$$\mathrm{E}[C'] = \mathrm{E}[C] = \mathrm{E}[J_{\Theta_i^N}(I)] \tag{9}$$

This shows us that equation 3 can be applied recursively and so can our speech enhancement process; that is, we generalize equation 3 and minimize the equation below instead.

$$\begin{aligned}\mathcal{L}_{\mathcal{N}\in\mathcal{N}} &= \sum_{i=1}^{n} \mathrm{E}[\mathcal{L}(I_i, J_{\Theta^N}(I_i))] \\ &= \mathrm{E}[\sum_{i=1}^{n} (I_i, J_{\Theta^N}(I_i))]\end{aligned} \tag{10}$$

with i indicating the i^{th} model, and $I_1 = I$ being the original noisy input.

3. EXPERIMENTS

3.1. Implementation Details

The structure of the auto-encoder in use is described in table 1, originates from reference [17], with a slight modification. In the original model emb_t is a parameter in the decoder network, but here since we observed that emb_t is used across several layers, we modified it such that the whole structure mimics the structure of a Unet [18]. We trained the models with Adam optimizer with learning rate $1e-4$ on MAE loss. All waveform arrays are converted to spectrogram before training.

Encoder	
conv-bank block	Conv1d-bank-8, LReLU, IN
conv block $\times$ 3	C-512-5, LReLU C-512-5, stride=2, LReLU, IN, Res
dense block $\times$ 4	FC-512, IN, Res
recurrent layer	bi-directional GRU-512
combine layer	recurrent output + dense output

Decoder	
conv block $\times$ 3	emb_t, C-1024-3, LReLU, PS C-512-3, LReLU, IN, Res
dense block $\times$ 4	emb_t, FC-512, IN, Res
recurrent layer	emb_t, bi-directional GRU-256
combine layer	recurrent output + dense output

Table 1. Network architectures. C indicates convolution layer. FC indicates fully-connected layer. $Conv1d-bank-K$ indicates convolution layer with kernel size from 1 through K. $LReLU$ indicates $LeakyReLU$ activation. IN indicates instance normalization[19]. Res indicates residual connection. emb_t indicates linear transform of corresponding layers between encoder and decoder.

3.2. Model

The structure of the entire model, defined J_Θ, is a stack of n auto-encoders that is described in the section above. In other words, the k^{th} model's output is taken as the $(k+1)^{th}$ model's input, for $k \in [0, n), k \in N$. For the first case (to minimize objective function equation 2), $n = 1$. For the second case (to minimize objective function equation 3), $n \in N$. Experiments show that adding another auto-encoder helps improve the model in almost every metric, at the cost of more computation.

3.3. Data

TIMIT[20] is our choice of clean speech because of its focus on human speech, all 4620 training samples, and 1680 testing samples. We chose Nonspeech[21] as the source of noises. We used all 100 kinds of noise. We used Diverse Environments Multichannel Acoustic Noise Database (DEMAND)[22] as a cross-domain-noise-source, also all of them, resulting in 224 new kinds of noise. The sample rate we chose is 16K. All of the data is normalized and converted to their frequency spectrum representation before processing. The way new training and testing data is generated is as follows: first, we select a clean-speech-instance C_i, which will be added by the following two noises. We define two variables snr_n and snr_s as the *signal-to-noise-ratio* (in dB) for each of our noises. We generate *Gaussian noise* g by sampling a series of $i.i.d.$ Gaussian random variable $g \sim G$. We also generate *mixed noise*, sampling m from a random variable $m \sim M$, from our noise sources (Nonspeech or DEMAND) in the following fashion: each noise has a probability p to be selected. The selected noises are randomly shifted, then scaled according to a randomly decided list of weights W that are non-negative and sum to 1. Noises are scaled according to their signal-to-noise-ratio $snr_n, snr_s(dB)$, respectively. For comparison purposes, we focus on the case where $snr_n = 0, snr_s = 0$ in this work. The result $R_i = C_i + G + M$ is also randomly shifted at run-time to prevent over-fitting.

3.4. Metric

The metrics used in this works are:

1. mean-absolute-error (MAE) $\in [0, \infty)$
2. mean-squared-error (MSE) $\in [0, \infty)$
3. signal-to-noise-ratio (SNR) $\in [0, \infty)$
4. perceptual evaluation of speech quality (PESQ)[23] $\in [-0.5, 4.5]$
5. short-time objective intelligibility (STOI)[24] $\in [0, 1]$

For MAE and MSE lower value shows better results while for SNR, PESQ, and STOI, higher value indicates better performance. In the experiments, we show the peak performance during the training phase, evaluated on a new set of evaluation data.

		supervised		clean-to-clean		noisy-to-noisy							
N^{th} model		1		1		1		2		3		4	
domain		source	cross	source	cross	source	cross	source	cross	source	cross	source	cross
MAE	$best$	3.408	3.592	2.556	2.831	1.411	1.629	1.468	1.690	1.549	1.775	1.573	1.811
	avg	3.406	3.590	2.432	2.644	1.244	1.402	1.244	1.413	1.249	1.416	1.271	1.435
MSE	$best$	2.888	2.306	2.435	2.028	1.778	1.777	1.949	1.859	1.934	1.955	1.987	2.052
	avg	2.887	2.305	2.261	1.888	1.225	1.522	1.270	1.560	1.289	1.589	1.375	1.435
SNR	$best$	0.654	0.567	0.564	0.484	0.272	0.377	0.395	0.373	0.391	0.415	0.427	0.456
	avg	0.654	0.567	0.500	0.428	0.102	0.232	0.129	0.250	0.156	0.268	0.178	0.285
PESQ	$best$	-0.004	0.280	0.344	0.418	0.107	0.187	0.153	0.220	0.123	0.198	0.102	0.165
	avg	-0.378	-0.170	0.258	0.360	0.026	0.091	0.036	0.078	0.023	0.077	0.019	0.064
STOI	$best$	0.146	0.144	-0.137	-0.156	0.060	0.031	0.068	0.038	0.073	0.049	0.077	0.048
	avg	0.128	0.134	-0.144	-0.169	0.031	0.013	0.034	0.016	0.036	0.016	0.037	0.017

Table 2. The result for snr_n = snr_s = 0. N^{th} model means the result is the output of N^{th} model compared to input. Larger is better for all entries. $best$ is selected according to evaluation metric, avg is average over the training phase

In this work, we evaluate the model as follows:

For MAE and MSE, the number is the ratio of that metric applied on the input, divided by that metric evaluated on the output of N^{th} model. For example, for MAE, the result is $\mathbf{MAE}(input, clean)/\mathbf{MAE}(output_N, clean)$. The number shows the amount by which the metric is reduced.

For SNR, PESQ, and STOI, the number is the difference of N^{th} output over the input. For example, for SNR, the result is $\mathbf{SNR}(output_N, clean) - \mathbf{SNR}(input, clean)$. The number shows the score difference between input and output. We believe this is a better way to evaluate metrics that can easily approach ∞, as in SNR if the input is very clean, or is limited in a relatively small range, as in the case of PESQ and STOI.

4. RESULTS

Experimental results are shown in table 2. The results are displayed as follows: We train one full model for 1000 epochs, and through the whole training phase, we record the evaluation result every 50 epochs. For comparison purposes, we evaluated the **average** performance of the model over the whole training course, and the **best** performance of the model, that is selected on another set of evaluation data.

The values displayed in table 2 is calculated as follows. The reference is a set of clean data, which every metric is evaluated on, but is not available during training. Source domain refers to using Nonspeech[21] as the noise during the evaluation, which we also used in training, and cross-domain refers to using DEMAND[22] as the noise during evaluation.

In those metrics and under different domains, *supervised* approaches are much better in peak performance (best field), and much more consistent (average field), though the others aren't lagging too far behind with both *clean-to-clean*'s model and *noisy-to-noisy*'s 4^{th} model scoring around 12% less in its best performance under cross-domain with MSE metric.

5. DISCUSSION

The proposed models are not as performant, in many cases, as their *supervised* counterpart. Nonetheless, the experiments provide us with some crucial insight into some new ways of improving speech enhancement.

For the *clean-to-clean* model, as the experiment results suggest, it is not as good as supervised learning in terms of MAE, MSE, SNR, or STOI, and only out-scored the *supervised* model in PESQ metric. However, the result is nothing short of interesting. In terms of metrics, in most cases, the performance is not so far off *supervised* model, which is our benchmark, without having to collect paired clean and noisy data. This leaves us to believe that this could be a new way to train auto-encoders, as the result here shows that it is robust to cross-domain noises. Since we did not specify a certain kind of noise to be mixed into training data, in fact, we require no noise at all, so this can be a new way to train domain-robust auto-encoders.

For the *noisy-to-noisy* model, we have found something interesting. For the second model on-wards, each auto-encoder doesn't directly have access to the input, rather, it further transforms the output of the previous model. The experiments suggest that this transformation is getting closer and closer to the clean version of our noisy input. This shows that probably in general, auto-encoders are robust to noises. Also note that the structure of the model allows this model to be further improved, which is that the auto-encoders in the model have access to the data in sequential order. We could simply stack another *noisy-to-noisy* model as the output, and treat the output of the current model as the input of the extended model. We can then "concatenate" the two and further augment the performance of our model. Source code and more results here. [1]

[1] https://github.com/MutatedFlood/rm-noise

6. REFERENCES

[1] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE TASLP*, vol. 27, no. 2, pp. 113–120, April 1979.

[2] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 6, pp. 1109–1121, December 1984.

[3] P. Scalart and J. V. Filho, "Speech enhancement based on a priori signal to noise estimation," in *IEEE ICASSP, 1996 - Volume 02*, Washington, DC, USA, 1996, ICASSP '96, pp. 629–632, IEEE Computer Society.

[4] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori, "Speech enhancement based on deep denoising autoencoder.," in *Interspeech*, 2013, pp. 436–440.

[5] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM TASLP*, vol. 23, no. 1, pp. 7–19, 2015.

[6] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal processing letters*, vol. 21, no. 1, pp. 65–68, 2013.

[7] Felix Weninger, Hakan Erdogan, Shinji Watanabe, Emmanuel Vincent, Jonathan Le Roux, John Hershey, and Björn Schuller, "Speech enhancement with lstm recurrent neural networks and its application to noise-robust asr," 08 2015, vol. 9237.

[8] Han Zhao, Shuayb Zarar, Ivan Tashev, and Chin-Hui Lee, "Convolutional-recurrent neural networks for speech enhancement," in *2018 IEEE ICASSP*. IEEE, 2018, pp. 2401–2405.

[9] Emad M Grais and Mark D Plumbley, "Single channel audio source separation using convolutional denoising autoencoders," in *2017 IEEE GlobalSIP*. IEEE, 2017, pp. 1265–1269.

[10] Szu-Wei Fu, Yu Tsao, and Xugang Lu, "Snr-aware convolutional neural network modeling for speech enhancement.," in *Interspeech*, 2016, pp. 3768–3772.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[12] Santiago Pascual, Antonio Bonafonte, and Joan Serra, "Segan: Speech enhancement generative adversarial network," *arXiv preprint arXiv:1703.09452*, 2017.

[13] Daniel Michelsanti and Zheng-Hua Tan, "Conditional generative adversarial networks for speech enhancement and noise-robust speaker verification," *arXiv preprint arXiv:1709.01703*, 2017.

[14] M. Mimura, S. Sakai, and T. Kawahara, "Cross-domain speech recognition using nonparallel corpora with cycle-consistent adversarial networks," in *2017 IEEE ASRU*, Dec 2017, pp. 134–140.

[15] Dan Stowell and Richard E. Turner, "Denoising without access to clean data using a partitioned autoencoder," 2015.

[16] Michael Michelashvili and Lior Wolf, "Audio denoising with deep network priors," 2019.

[17] Ju chieh Chou, Cheng chieh Yeh, Hung yi Lee, and Lin shan Lee, "Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations," 2018.

[18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[19] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky, "Instance normalization: The missing ingredient for fast stylization," 2016.

[20] Garofolo and John S., "Timit acoustic-phonetic continuous speech corpus," Linguistic Data Consortium, 1993.

[21] Hu G. and Wang D.L., "A tandem algorithm for pitch estimation and voiced speech segregation.," 2010, dataset collected during this work.

[22] Joachim Thiemann, Nobutaka Ito, and Emmanuel Vincent, "DEMAND: a collection of multi-channel recordings of acoustic noise in diverse environments," June 2013, Supported by Inria under the Associate Team Program VERSAMUS.

[23] Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *2001 IEEE ICASSP. Proceedings (Cat. No. 01CH37221)*. IEEE, 2001, vol. 2, pp. 749–752.

[24] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *2010 IEEE ICASSP*, March 2010, pp. 4214–4217.