S-SepGAN - GAN Based Music Source Separation

Mostafa ElAraby
mostafaelaraby.com

Kun Ni
University of Montreal

Stephan Anh Vu Tran
École de technologie supérieure

Abstract

Blind source separation (BSS) is the process of extracting source signals from a mixed signal without knowledge of the mixing process. It is a hard machine learning task as the input audio are correlated and some components information are missing in the mixture, so a model is needed to recover these missing data and recover the signal component. We demonstrate that an adversarial framework with one generator competing with one discriminator can separate music signals into its components. In the context of this work, we will mainly focus on the task of separating music instruments from a multiple instruments signal (mixture).

1 Introduction

1.1 Blind source separation

Separating an audio signal into multiple distinct signals have been an active field of research for a while. In effect, multiple uses can be derived from this task such as extracting voices from different persons of an audio recording into individual tracks, extracting a singing voice of a song for karaoke applications or even splitting every musical instruments of an audio track for music remastering and remixing purposes. Mainly, the task of separating audio sources is a subset of blind signal separation (BSS) which can be described as extracting source signals from a mixed signal without knowledge of the mixing process. Multiple sources s_i are being combined through a mixing process (microphone capturing multiple audio sources, music mixing process combining multiple instrumental tracks into a stereo or mono track, etc.) which produced a mixture m. A BSS process would then estimate the sources $\hat{s}_i$ as close as possible to s_i for all $i : 1...n$ where n is the number of sources. However, separating a music signal into its sources signals is an undetermined and challenging task as correlation between the mixed signals is highly common in these type of audio recordings. Also, high temporal resolution of audio signals could result in a high input dimension which could lead to a computationally heavy model to train. Furthermore, a couple of popular approaches define an posterior distributions over the sources estimates but by doing so, this could lead to decrease in performance due to bias with the true posterior distributions. Lastly, the publicly available dataset containing both the mixture signal and its sources $(m; s_1, ..., s_n)$ for supervised training are small which limits the capacity of the model.

With these challenges in mind, we are proposing a novel approach that uses an adversarial setting inspired by CycleGAN for each instrument we want to separate from the mixture. Thus, through this work, we would like to evaluate if this approach could solve the problems of music sources separation mentioned earlier and if it could outperform the current state of the art music BSS techniques.

The methodology used for this work is as follow:

1. Literature review: we will perform a review of papers treating audio synthesis and audio source separation with generative models.
2. Based on the challenge of BSS, the recent works and what we think could be improved from the literature review, we proposed a novel approach. We tested the approach to verify its feasibility.

3. Methods comparison: with the same dataset, we will compare the results of the recent works with our approach through different quantitative metrics and qualitative evaluation.

2 Related work

In this section, we will go over different works on to audio source separation with generative models and adversarial framework. Additionally, a review of audio synthesis techniques will be done since audio source separation and audio synthesis are closely related.

2.1 Audio synthesis

Current problem with audio synthesis methods includes the high dimensionality of data and the slow generation time. Furthermore, techniques using spectrogram (2D image) maybe results in poorer audio quality as the process is non-invertible. An estimation has to be made from the generated spectrogram which is a lossy process. Finally, audio data have correlation that span across a large window thus requiring a large receptive field. Donahue et al. [2018] proposed a technique that involves processing audio data as one dimensional image with generative adversarial network using Wasserstein distance with gradient penalty strategy. They based their architecture on DCGAN (Radford et al. [2015]) with transposed convolution operation (1-D conv.) for upsampling in the generator. The same is also applied in the discriminator but with downsampling convolutions. Since audio requires a larger receptive field, they modified the DCGAN architecture to have a larger stride.

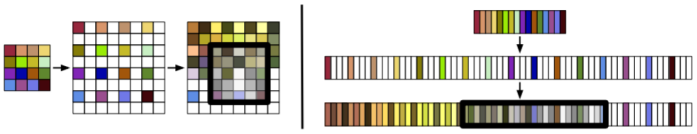

Figure 1: Transposed convolution for *left*) 2 dimensional input (image) *right*) 1 dimensional input (audio)

2.2 Compressed Sensing using Generative Models

Bora et al. [2017] provide an method to implement compressed sensing with generative models. Given pre-trained generator, we can find the latent variable to generate audio samples which are close to target samples. We apply this method on single track audio reconstruction and get acceptable result. We also notice that it is lack of inference method. we attempt to train a inference network for latest variable. The input of encoder network are audio mixture. We have to ensure the correlation between audio mixture and target single track of audio. Besides of applying it in source separation network, we can use it to evaluate the representation performance of generators.

2.3 Composition and decomposition of GANs

Music audio is compositional in nature. Usually, a music audio is a mixture of different tracks. Harn et al. [2019] present a framework to learn the composition and decomposition function adversarially. Under this framework, given pre-trained generators and composition function, we should be able to learn the decomposition function. With this decomposition function, we can achieve the goal of separating unseen music mixture.

2.4 Source separation

2.4.1 Generative adversarial source separation

Subakan and Smaragdis [2017] wanted to solve the problem of specifying the output probability density shown in popular source separation methods using generative models such as the Non-Negative Matrix Factorization technique. By specifying an output distribution, the model will suffer from the generality of the approximated source distributions.

The main idea of this paper is to exploit the advantage of not having to specify a posterior by using an implicit generative model. Thus, in the paper's work, they chose the Generative Adversarial Network (Goodfellow et al. [2014]) to perform the task of source separation. This task is described as generating sources s_i of a mixture m by passing the latent variables h_i through a forward model:

$$h_1\ p_{latent}(h_1), s_1|h_1\ p_{forward1}(s_1|h1)$$
$$h_2\ p_{latent}(h_2), s_2|h_2\ p_{forward2}(s_2|h2)$$
$$x|s_1, s_2\ p_{mixture}(x|s_1 + s_2)$$

where the sources s_k are distributed as $p_{source_k}(s_k) = p_{source}(s_k) = \int p_{forward}(s_k|h_k)p(h_k)dh_k$. Finally, $p_{forward}$ is the forward model (the generator) of the sources, p_{latent} is the model of the latent variable which is from a Gaussian distribution and $p_{mixture}$ is the distribution of the mixture. As

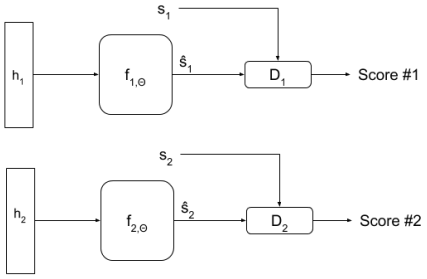

Figure 2: Architecture of the generative adversarial network framework for BSS

shown above, the architecture defined for source separation is composed of a generative/discriminator pair for each source signal. The forward models are two layers deep with soft-plus activation function after each layer. On the other side, the discriminators are also two-layers MLP but with tanh activation functions.

Then, during source separation phase (test time), to generate the estimates that will produce the best reconstruction loss, the algorithm will try to find the optimal latent variables that will satisfy:

$$\hat{h}_1, \hat{h}_2 = \arg\max_{h_1,h_2} \log p_{mixture}(x; f_{\hat{\theta}_1}(h_1) + f_{\hat{\theta}_2(h_2)})$$

The objective function that is used to find the optimal h is composed of the log likelihood of the reconstructed mixture (assuming Poisson distribution), a weighted critics score and a smoothing term which smooth out the spectrograms for a more natural result. The weighted discriminators score is an interesting add-on to the objective function because the generators might be able to generate two unrealistic source estimates which however summed up could still produce a good reconstruction of the mixture. With that objective function, optimization of that function with respect to the latent variables h_i is done to obtain the optimal h (backpropagation). Finally, the source estimates is obtained by passing h to the forward model (generator of each sources) and then by doing an inverse STFT.

In short, we think that the work proposed by Subakan and Smaragdis [2017]. is very interesting due to the very simple implementation of the technique. Furthermore, the suggested architecture of the generators and discriminators are very light which results in a very fast training and inference time. Also, we find that the two additional terms in the reconstruction objective function are efficient addons that allowed the technique to produce good source estimates. However, due to its lightness and simplicity, this method also have limitations: the generated audio source estimates are low resolutions. The audio quality is strongly degraded compared to the original audio.

2.4.2 Adversarial semi-supervised audio source separation applied to singing

While the current methods require supervised training for separating audio sources from a mixture, it is an expensive process because of the the availability of labeled dataset for this kind of task. With that in mind, the problems Stoller et al. [2017] are trying to address with this paper are to mainly exploit the availability of unlabelled dataset and combine it with a supervised training for singing

voice extraction. Consequently, they came up with an architecture (Figure 3) which has a separator network that outputs source estimates and each of these estimates are then evaluated by their own discriminator. Referring to the Figure 3, the separator network f_θ will be trained in a supervised

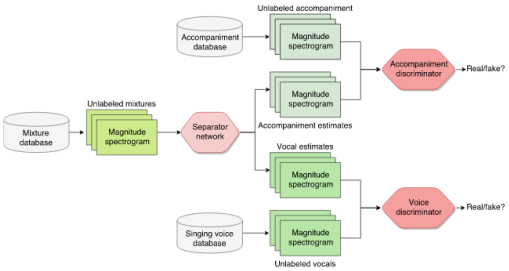

Figure 3: Architecture of adversarial semi-supervised framework for BSS

fashion so the source estimates $f_\theta(m_i)$ (where m is the mixture signal) at the output of the separator will try to minimize the reconstruction error with the real sources s_i. The Mean Squared Error (MSE) is used for the supervised loss function L_s over the M samples: $L_s = \frac{1}{M}\sum_{i=1}^{M} ||f_\phi(m_i) - s_i||_2$. Additionally, an unsupervised loss function is also derived for each sources from the output of the discriminators. Knowing that the Jensen-Shannon divergence optimization in the original GAN paper might lead to vanishing gradient, the authors chose to optimize the Wasserstein distance with gradient penalty (WGAN-GP) instead. Note the gradient penalty term has been modified to be one-sided because it led to better convergence time. Thus, the unsupervised loss for the model is

$$L_u = \mathbb{E}_{\hat{x}\sim P_g}[D(\hat{x})] - \mathbb{E}_{x\sim P_r}[D(x)] + \lambda\mathbb{E}_{\hat{x}\sim P_{\hat{x}}}[(||\nabla_{\hat{x}}D(\hat{x})||_2 - 1)^2]$$

Finally, a third loss called the additive loss $L_{add} = \frac{1}{U}\sum_{i=1}^{U} ||\sum_{k=1}^{K} f_\phi(m_i^u)_k - m_i^u||_2$ (where U is the number of samples and K is the number of audio sources) term is added to take into account the the reconstruction of the mixed signal audio from the estimates. The goal of this loss is to guide the model to produce source estimates that when combined together matches as much as possible the original mixture signal. The final loss is computed as sum of all the mentioned loss with possibility to adjust the weight of the unsupervised loss and the additive loss through hyper-parameters alpha and beta respectively: $L = L_s + \alpha * L_u + \beta * L_{add}$.

2.5 Analysis and overview

First, we've noticed that the current BSS techniques are preprocessing the audio data as spectrograms. Yet, Donahue et al. [2018] notes that spectrograms are non-invertible. To obtain audio samples from a spectrogram, it requires a lossy estimation process which will degrade the final reconstructed sources.

Also, these papers all uses the traditional generator/discriminator pairing where the output of one generator is connected to the input of one discriminator. However, we've recently seen improvement on the quality of generated image by modifying the framework of the generative/discriminator pairs. For example, the work of Neyshabur et al. [2017] showed some promising results by splitting the output of the generator into multiple low projections representation and then feeding each representation into their own discriminator.

3 Proposed method

In this section, we discuss our proposed adversarial approach inspired from recent works on audio source separation and CycleGAN framework (Zhu et al. [2017]).The input audio is treated as one-dimensional image and perform strided convolutions on it instead of using spectrograms. Our initial proposed approach was to implement CycleGAN framework (Zhu et al. [2017]) into our BSS solution. The idea is to extract a source signal from a mixture signal through a first generator/discriminator pair. Then, a second pair will attempt to reconstruct the input signal from the generated source signal while giving more weight in the loss function to the unmixing of input audio. After experimenting with the CycleGAN , we proposed a simpler framework with a single generator discriminator setup and the generator having similar archtecture to the one used in the CycleGAN.This proposed model

would take a mixed signal and output its single component.The only drawback of this approach is that it requires paired data.

3.1 Background: Generative adversarial network and Cyclic GAN

The generative adversarial network is composed of a generator which produce samples from a latent variable and its goal is to generated sample that matches the real data distribution. On the other end, the discriminator will be fed with real and generated samples. Its goal is to try to evaluate if its input is a real sample or generated sample. In particular, the work of Zhu et al. [2017] (CycleGAN) caught our attention by its ability to transfer a given input of domain A to an equivalent output of domain B and vice versa. Also, since this framework is under-constrained, the authors added a inverse mapping stage where the generated output on domain B is transferred back to domain A. We then force the difference between the input X and $\hat{X} = G_1(G_2(X))$ to be smallest as possible (cycle consistency) through optimization where $\hat{X}$ is the reconstructed input, G_1 is the generator transferring samples from domain A to B and G_2 is the generator transferring samples from domain B to A.

3.2 CyclicGan

Applying the CycleGAN framework to a BSS task can be described as: given an mixture signal m (domain B), the first generator (B2A) will attempt to produce a sample which contains a single audio source $\hat{s}$ (domain A). Then, a second generator (A2B) will reconstruct the mixture signal $\hat{m}$ (domain B) (Figure 4).

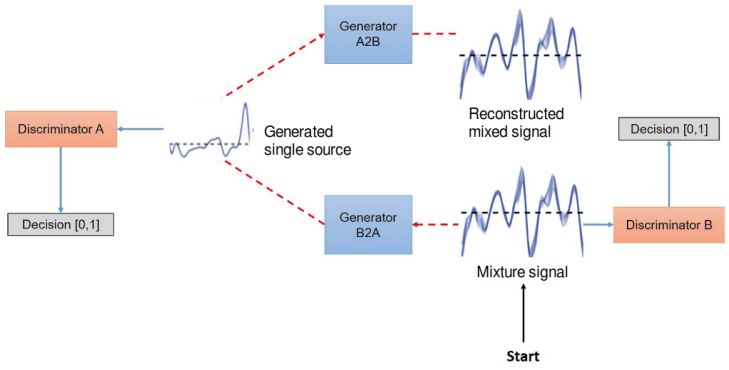

Figure 4: Architecture of CycleGAN framework for BSS

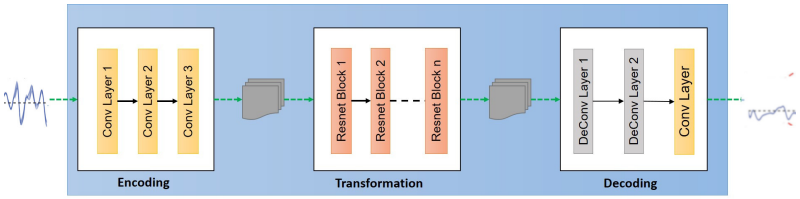

Figure 5: Architecture of CycleGAN generator

During training, multiple losses will be computed. Firstly, for the generators parameter updates, we will use the identity loss $\mathcal{L}_i = ||s - \hat{s}||_1$ which will guide the B2A generator to produce generated single source similar to the real single instrument audio source. Secondly, a cyclical loss $\mathcal{L}_{cyc} = ||m - \hat{m}||_1$ will ensure that the network is capable of recreating the input mixture from the generated single audio source i.e. cycle consistent. Thirdly, for both generator/discriminator pair, we uses the adversarial loss originally introduced in the generative adversarial network framework:

$$\mathcal{L}_{B2A} = \mathbb{E}_{s\sim p_{data}(s)}[\log D_A(s)] + \mathbb{E}_{m\sim p_{data}(m)}[1 - \log D_A(G_{B2A}(m))]$$

$$\mathcal{L}_{A2B} = \mathbb{E}_{m\sim p_{data}(m)}[\log D_B(m)] + \mathbb{E}_{\hat{s}\sim p_{data}(\hat{s})}[1 - \log D_B(G_{A2B}(\hat{s}))]$$

Note that these losses are weighted by hyper-parameters and that we've noticed that putting higher weight on the cyclical loss results in better performance. Finally, we trained the network using Adam optimization for each discriminator and generator of the framework.

3.3 Single generator discriminator framework

Instead of training two generators and two discriminators as we need only the unmixing direction of the audio, we propose using a single generator discriminator setup. The generator model tries to generate a single component from the input mixed audio and the discriminator is trying to check whether the generator's output is the real single component close to the real distribution or it is a variant of the mixed signal or a single audio not related to the main component distribution.

The generator loss function consists of an identity loss which compares the distance between the input and the output in case the model takes the main component to avoid having unnecessary changes to the input audio if it's our component of interest, the second loss is the generative loss trying to fool the discriminator and the cyclic loss having more weight than the first two which is trying to measure the distance between the generated component and the real single component from our paired dataset.

The discriminator's loss function is trying to discriminate between real single main components and generated signals. In order to give meaningful information to the generator we treated mixed audio from our dataset as fake samples to penalize the generator if it started to generate the same input mixed data.

$$\mathcal{L}_D = \mathbb{E}_{s\sim p_{data}(s)}[\log D(s)] + \mathbb{E}_{m\sim p_{data}(m)}[1 - \log D(G(m))] + \mathbb{E}_{m\sim p_{data}(m)}[1 - \log D(m)]$$

The network has been trained on a long length piano track (Mancini piano track) that has been split into several $\sim$1 minute audio files which serves as ground truth. To generate the mixture signals, these short piano audio files have been mixed with various musical instrument audio samples from the IRMAS dataset Bosch et al. [2012]. Finally, when using these samples in our framework, each audio files are deconstructed into multiple one dimensional vector of signal amplitude with a window length of approximately one second.

4 Discussion

With the current state of the work, we have succeeded in extracting the piano audio source from a given mixture signal in both CycleGAN and the proposed approach but the CycleGAN output is more of domain transfer not the single component and the samples from our proposed method outperformed the CycleGAN framework. Thus, at this point, we know that it is possible to apply blind source separation task to a cycle-consistent adversarial framework with amplitude waveform instead of spectrograms.

5 Conclusion

We have introduced a novel method for blind source separation using an adversarial framework . Also, unlike other current BSS techniques, we have chosen to manage audio data as one dimensional image representing signal amplitude over time to avoid having lossy estimates with spectrogram due to its non-invertible characteristic. Current work progression have shown promising results from the generated samples of the extracted source. Thus, for the next part of the work, we will start getting quantitative metrics such as source to noise ratio (SNR), source to interference ratio (SIR), source to distortion ratio (SDR) and source to artifacts ratio (SAR). This will provide a mean of comparison with other BSS techniques. Also, we will attempt to improve the audio preprocessing step by including improvement from Engel et al. [2019] where instantaneous frequency and information is passed in the input data.

References

A. Bora, A. Jalal, E. Price, and A. G. Dimakis. Compressed sensing using generative models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 537–546. JMLR. org, 2017.

J. J. Bosch, J. Janer, F. Fuhrmann, and P. Herrera. A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals. *Proc. ISMIR (pp. 559-564)*, 2012.

C. Donahue, J. McAuley, and M. Puckette. Adversarial audio synthesis. *arXiv:1802.04208 [cs.SD]*, 2018.

J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts. Gansynth: Adversarial neural audio synthesis. *arXiv:1902.08710 [cs.SD]*, 2019.

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *arXiv:1406.2661 [stat.ML]*, 2014.

Y. Harn, Z. Chen, and V. Jojic. Composition and decomposition of gans. *CoRR*, abs/1901.07667, 2019. URL `http://arxiv.org/abs/1901.07667`.

B. Neyshabur, S. Bhojanapalli, and A. Chakrabarti. Stabilizing gan training with multiple random projections. *arXiv:1705.07831 [cs.LG]*, 2017.

A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv:1511.06434 [cs.LG]*, 2015.

D. Stoller, S. Ewert, and S. Dixon. Adversarial semi-supervised audio source separation applied to singing voice extraction. *arXiv:1711.00048 [cs.LG]*, 2017.

C. Subakan and P. Smaragdis. Generative adversarial source separation. *arXiv:1710.10779 [cs.SD]*, 2017.

J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv:1703.10593 [cs.CV]*, 2017.