UNSUPERVISED SINGLE CHANNEL SOURCE SEPARATION AUTOENCODERS

André Bergner

Native Instruments GmbH
Berlin, Germany
andre.bergner.0@gmail.com

Kevin N. Webster

FeedForward Ltd.
kevin@feedforwardai.com
Imperial College London
London, UK
kevin.webster@imperial.ac.uk

ABSTRACT

Deep learning models have greatly improved the performance of audio source separation models, and there is an emerging trend towards end-to-end learning for this task, dispensing with traditional STFT preprocessing of the audio signal. However, most of the developments have still focused on supervised training of neural networks, and require a considerable amount of training data. In this paper, we propose an alternative deep learning-based formulation that is completely unsupervised, and additionally requires no pre-training of the neural network. Instead, the source separation task is learned and executed at run time. We demonstrate a proof of concept for this architecture on both synthetic and audio data.

1. INTRODUCTION

Source separation is a very active and challenging area of research that has received considerable interest in recent years. There are numerous real-world applications from automatic speech recognition [1], speech enhancement for cochlear implant users [2], and music transcription [3]. With the growing prevalence of deep learning in areas of digital signal processing, several neural network architectures have been proposed for the purpose of audio source separation [4–6].

A particular problem case is the single channel blind source separation problem (SCBSS). In this setting, a single mixture channel should be separated into a (usually prescribed) number of sources. Blind source separation refers to the fact that no other information is used in the separation process; in particular, there is no pre-training phase that would tailor the algorithm to a certain data distribution. This is in contrast to most deep learning-based source separation models, that are trained for a specific task such as separating vocals from a mix.

A standard approach for multi-channel blind source separation is independent components analysis (ICA) [7] . In this approach, the underlying assumption is that the observed channels are linear mixtures of the statistically independent source signals. The goal is to find an un-mixing matrix such that the separated sources are statistically independent. We note that methods have been further developed to extend the ICA approach to nonlinear mixtures [8, 9] in the multi-channel setting. ICA has also been applied to the SCBSS problem [10], however, due to the nonlinear nature of the SCBSS problem the linear ICA is limited to non-overlapping spectra, besides other limitations and problems.

In this paper we propose the Source Separation Autoencoder (SSAE); an end-to-end deep learning-based single channel source separation model that furthermore learns the separation of sources at run time, i.e., in a fully unsupervised (blind) manner. All methods in this setting known to us either suffer from artefacts, require additional prior knowledge, or specify impractical constraints to the underlying sources such as non-overlapping frequency components. To our knowledge our proposed architecture is the first that has been proposed for SCBSS without prior assumptions or strong constraints on the mixture. Our work is carried out with audio applications in mind, but it is not limited to it. We demonstrate the performance of the model on both synthetic (non-audio) signals as well as preliminary audio examples as a proof of concept.

1.1. Related Prior Work

In the single channel setting, there have been several deep learning-based source separations models [11–15]. These models learn to encode the input signal to a latent representation that enables a decoder network to separate the sources. However, these autoencoder architectures are all trained in a supervised manner, and are therefore targeted at specific tasks.

In addition to neural networks, another standard approach is nonnegative matrix factorisation (NMF) [16, 17]. Both approaches are often based on spectral masking and involve some preprocessing. However, this approach frequently results in artefacts when transforming back into the time domain and has the additional problem of reconstructing the phase information. More recently, several network architectures have been developed as 'end-to-end' models that operate directly on the raw audio [14, 15, 18].

The remainder of the paper is organised as follows. In section 2 we describe the network architecture and training and prediction procedure. In section 3 we describe the experiments carried out with the model, and in section 4 we conclude with discussion on further development.

2. SOURCE SEPARATION AUTOENCODER

Our model is an autoencoder that encodes input audio directly in the time domain. Thus, there are no additional limitations or assumptions due to the pre-processing step in our approach. It does not require any knowledge about the underlying sources, except for the the necessary presence of time correlations (see section 2.3) and the fact that it is a linear mixture. Precisely, the assumption is that the observed signal is a sum of the underlying sources. However, supervised pre-training might help to improve or speed up separation in complicated cases.

The underlying intuition behind our model employs the same assumption as in ICA: a mixture observation is constructed from statistically independent sources. However, we do not make a linearity assumption for the function mapping the mixture to the sources (as is the case in the unmixing matrix), but instead use a flexible nonlinear function to model the separation of sources.

2.1. Architecture

The proposed architecture uses an autoencoder framework, with N_{dec} decoders, one for each assumed source. The input signal

$$\mathbf{X} = \{x_1, x_2, \dots, x_T\} \tag{1}$$

is windowed to create $T - S(D-1)$ sample windows

$$\boldsymbol{\xi}_n := \{x_n, x_{n+S} \dots, x_{n+S(D-1)}\} \in \mathbb{R}^D, \tag{2}$$

where D is the window size in samples, and S a stride parameter.

We define the Source Separation Autoencoder (SSAE) as

$$\begin{aligned} \mathbf{z} &= e(\boldsymbol{\xi}_n) \\ \tilde{\boldsymbol{\xi}}^{(k)} &= d_k(\mathbf{z}_k) \\ \tilde{\boldsymbol{\xi}} &= \textstyle\sum_k^{N_{dec}} \tilde{\boldsymbol{\xi}}^{(k)} \end{aligned} \tag{3}$$

where $\mathbf{z} \in \mathcal{Z} = \mathbb{R}^Z$ is the latent code for a given window $\boldsymbol{\xi}_n$. The size of the latent space Z is a hyperparameter. The latent space $\mathcal{Z}$ is split into N_{dec} subspaces with $\mathcal{Z}_k = \mathbb{R}^{Z_k}$ and $\sum_{k=1}^{N_{dec}} Z_k = Z$. The decoders are independent from each other (no weight sharing).

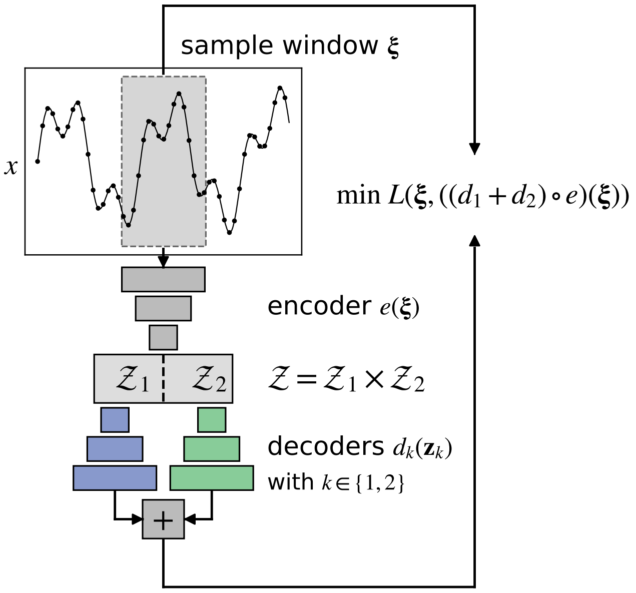
sample window ξ
x
$\min L(\xi, ((d_1+d_2)\circ e)(\xi))$
encoder $e(\xi)$
$\mathcal{Z}_1$
$\mathcal{Z}_2$
$\mathcal{Z} = \mathcal{Z}_1 \times \mathcal{Z}_2$
decoders $d_k(\mathbf{z}_k)$
with $k \in \{1,2\}$
+

Figure 1: *The SSAE architecture. The sample window is embedded into a latent space Z, which is partitioned and sent through separate decoders to reconstruct the sources.*

Henceforth in this paper and in our experiments we will assume $N_{dec} = 2$. The separate output sources $\tilde{\boldsymbol{\xi}}_1$, $\tilde{\boldsymbol{\xi}}_2$ should add together to reconstruct the original mixture. The source separation autoencoder is therefore trained by minimising a loss $L(\boldsymbol{\xi}, ((d_1 + d_2) \circ e)(\boldsymbol{\xi})) = L(\boldsymbol{\xi}, \tilde{\boldsymbol{\xi}}_1 + \tilde{\boldsymbol{\xi}}_2)$. In our experiments we use the least squares or mean absolute error loss functions. We found that absolute error losses tend to converge faster.

2.2. Model prediction and bagging

Assume for simplicity that $S = 1$. At test time, note that we will have many overlapping input windows $\{\boldsymbol{\xi}_n\}$, leading to overlapping prediction windows $\{\tilde{\boldsymbol{\xi}}_n^{(k)}\}$, $n = 0, 1 \dots, T - D$. Each sample (apart from samples towards the beginning or end of the input sequence) will be reconstructed D times. A simple choice would be to take say the first sample from each window $\tilde{\boldsymbol{\xi}}_n^{(k)}$ window to form the source signals. However, we choose to make use of all D reconstructions by averaging over all predictions for a given sample, which results in a kind of bagging algorithm, since each sample is reconstructed by different weights, or sub-parts, of the network. This is similar to the kind of bagging exploited in *dropout* regularization.

2.3. Motivation and intuition

The design of the Source Separation Autoencoder (SSAE) is motivated by Takens' embedding theorem [19], which–in short–states that there exists a diffeomorphism that embeds the attractor of a dynamical system into a delay coordinates embedding space with a sufficiently high dimension. Further, if we assume independence of the signal generating dynamical systems, their common state space is the product space of the individual state spaces. In our setting, we can think of the SSAE as trying to find an embedding space $\mathcal{Z}$ that is the product of two embedding spaces $\mathcal{Z}_1$ and $\mathcal{Z}_2$, where the separate attractors corresponding to the independent sources are embedded. The decoders are then acting as the inverse of the embedding diffeomorphism for each source.

Autoencoders are natural candidates for this task as they try to find a low-dimensional manifold the represents the training data as good as possible. Thus, the autoencoder tries to learn the diffeomorphism between the embedding (the window) and the state space (latent space). By splitting the latent space into disjoint parts which feed distinct decoders we enforce the multiple autoencoders (with the one shared encoder) to learn the factorised representations which is assumed to exist in the case of independent sources.

3. EXPERIMENTS

In this section, we demonstrate the wide applicability of the SSAE by testing on synthetic harmonic signals, time series generated from a chaotic dynamical system, and real audio data. In the audio case, we compare the performance against a NMF algorithm, as it can also be used completely unsupervised (no pre-training). The NMF approach however makes use of a spectral representation of the time series, whereas the SSAE works in the time domain. As a result, the NMF algorithm frequently suffers from artefacts due to the phase reconstruction. We used the `nussl` library implementation of the NMF algorithm [20].

3.1. Model parameters

In our experiments, we tested the use of densely connected or 1D convolutional layers for both encoder and decoder, and found that convolutional layers are most effective and converge more reliably. The number of chosen layers are directly connected to the chosen window size D. We have chosen D to be a power of two. Each one-dimensional convolutional layer in the encoder is downsampled (strided) by a factor of two, we stack layers until we reach a single output feature frame, i.e. layer sizes are $D, D/2, D/4, \dots, 1$. We experimented with different number of features throughout the network. However, to limit the amount of hyper-parameters to tune we fixed the number of features throughout the whole network, whereas the precise number varies between experiments. Typical values we used are 16, 24, or 32, depending of the complexity and *sharpness* of the sources. We used a kernel size of 3, and `tanh` activation functions. For the decoders we reverse this setup and upsample with zeros (often referred to as deconvolution) to get from smaller to bigger layer sizes. The gradient descent is done using the *Adam* optimizer with default parameters. Convergence time in our experiments varied from a few minutes to some hours running on a 13" Macbook Pro (2015) depending on the length of the signals and their complexity.

Throughout our experiments we discovered that adding noise helps to guide the separation process in particular for simple signal with low entropy that can easily be reconstructed jointly by just one of the autoencoders. To combat that we injected noise into all decoder layers and the latent space. Further we added an annealing parameter to the noise that reduces the noise throughout the training process and leaves more capacity to improve reconstruction after some stable separation regime was found with strong initial noise. Our interpretation is that the noise acts as regulariser that reduces the channel (in the Shannon sense) capacity. Consequently the autoencoder needs to factorise the distribution, i.e. separate the signals, in order to utilise the given capacity as much as possible.

3.2. Modulated sinusoidal and square wave

In the first experiment we demonstrate separating the sum of a square wave with a fixed frequency and a sinusoidal signal which is slightly frequency modulated around the same frequency, i.e. both signal cover a similar frequency range. The SSAE parameters are $D = 64$ resulting in 6 convolutional layers each with 16 channels. The latent space dimensions are $Z_1 = Z_2 = 3$. The SSAE is able to perfectly separate both signal despite of the complete frequency overlap. The results are shown in Fig. 2.

3.3. Signals with broad power spectra

In order to test the SSAE with a more realistic yet synthetic signal, we constructed a mixture of a numerically integrated chaotic system, the Lorenz system, and a strongly phase modulated sinusoid. Both systems have a broad power spectrum with significant overlap (see Fig. 3). The Lorenz system is defined as $\dot{\mathbf{u}} = \big(10(u_2 - u_1), u_1(28 - u_3) - u_2, u_1 u_2 - (8./3.)u_3\big)$ where $\mathbf{u} = (u_1, u_2, u_3)$. It is a paradigmatic system for chaos and is intensively studied in the literature. We numerically integrated it using a fourth order Runge-Kutta Scheme with step size $h = 0.01$. We normalised the resulting time series in order to have both signals in a similar range.

Using the SSAE we are able to separate a mixture of these signals almost perfectly as it is shown in Fig. 3. For this experiment the SSAE has been set up as follows. The window spans 256 samples with a stride of 2, the latent space dimensions have been set to 4 for both decoders. We use 32 features for the seven convolutional layers. The initial noise variance was set to 0.2.

3.4. Audio mixture

Finally, we tested the SSAE on three mixtures of audio clips downloaded from `freesound.org` and compared against the NMF algorithm. As we can see in Table 1, the NMF algorithm fails to separate the audio sources in a fully unsupervised manner, however the SSAE successfully accomplishes the task and achieves far higher metric scores in most cases.

The SSAE parameters are: $D = 256$ and $D = 128$ (for mixture 2), 24 convolutional features, and latent space dimension are set both to 5. The initial noise variance was set to 0.1. We only separated two seconds of audio mixture due to the demanding computational effort in the current setup.

4. CONCLUSIONS

In this paper we have presented a new method for source separation that is entirely unsupervised (requires no pre-training), and makes no prior assumptions about the nature of the sources, other than the number of sources and the existence of time correlations in the signal. In this sense the method is very general. For audio signals, the method avoids artefacts commonly associated with pre-processing the data into spectral representations of the data by using raw audio directly.

4.1. Future development

The main architectural experiments we have carried out include testing of different kinds of layers (convolutional or dense) and regularisation techniques through the addition of noise variables into the network. The performance of the SSAE is relatively insensitive to high-level architecture changes, however we believe that a more thorough ablation study of the design of the network will be helpful in increasing the model robustness. A secondary aim will be to accelerate the convergence speed of the method, possibly by making the model more compact.

This includes studies on the capacity of the network and the corresponding effect on artefacts introduced into the separated signals (see the square wave example), the size of the latent space and form of regularisation. Recent work makes use of a discriminator network to ensure statistical independence of separated components [9], and it may be interesting to investigate the use of adversarial training within our context. Similarly, one might consider constraining the distribution within the latent space with a specified prior, and train the SSAE within the framework of variational autoencoders.

5. REFERENCES

[1] Andrew Maas, Quoc V. Le, Tyler M. O'Neil, Oriol Vinyals, Patrick Nguyen, and Andrew Y. Ng, "Recurrent neural networks for noise reduction in robust asr," in *INTERSPEECH*, 2012.

[2] K. Kokkinakis and P. C. Loizou, "Using blind source separation techniques to improve speech recognition in bilateral cochlear implant patients," *J. Acoust. Soc. Amer.*, vol. 123, no. 4, pp. 2379–2390, 2008.

[3] Mark D. Plumbley, Samer A. Abdallah, Juan Pablo Bello, Mike E. Davies, Giuliano Monti, and Mark B. Sandler, "Automatic music transcription and audio source separation," *Cybernetics and Systems*, vol. 33, pp. 603–627, 2002.

[4] Emad M. Grais, Mehmet Umut Sen, and Hakan Erdogan, "Deep neural networks for single channel source separation," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3734–3738, 2014.

[5] Aditya Nugraha, Antoine Liutkus, and Emmanuel Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, 06 2015.

[6] S. Venkataramani, J. Casebeer, and P. Smaragdis, "End-to-end Source Separation with Adaptive Front-Ends," *`https://arxiv.org/abs/1705.02514`ArXiv e-prints*, May 2017.

[7] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja, *Independent Component Analysis*, Wiley, 2001.

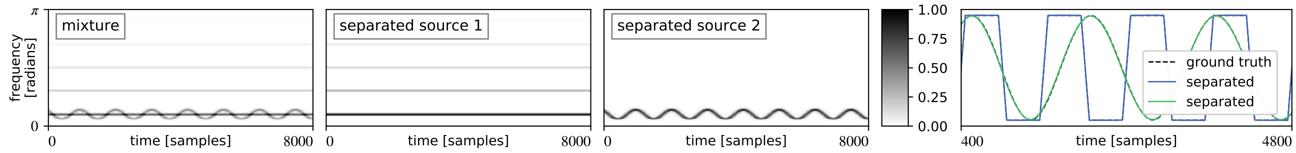

Figure 2: *A square wave and a frequency modulated sinusoidal signal, both with the same center frequency. The three left spectrograms show the frequency overlap of the mixture and the sources. The right figure shows the near perfect separation of the sources in the time domain.*

Table 1: *Evaluation on audio source separation. Freesound IDs are given in the Mixture column. The NMF evaluation scores were highly variable so we gathered statistics for the scores, and below we present the mean and standard deviation for each of the evaluation metrics. Higher numbers are better.*

Mixture	SSAE			NMF		
	SDR	SIR	SAR	SDR	SIR	SAR
39914 *(singing voice)*	**2.32**	**12.67**	**17.06**	-3.93 (±7.57)	-3.47 (±16.78)	4.42 (±1.44)
195138 *(cello)*	-5.59	-5.54	**20.01**	**-1.12** (±6.98)	**2.39** (±12.80)	4.25 (±1.37)
328727 *(flute)*	**1.46**	**1.54**	**22.81**	-0.14 (±1.85)	0.34 (±5.77)	2.86 (±4.39)
248355 *(clarinet)*	**8.98**	**41.16**	**26.56**	1.71 (±4.33)	1.88 (±7.14)	14.57 (±7.80)
145513 *(glass)*	**10.56**	**12.07**	**17.30**	1.18 (±3.19)	-1.93 (±11.47)	8.47 (±1.68)
203742 *(bass bow)*	**12.88**	**42.09**	**19.67**	0.44 (±4.83)	-5.27 (±22.81)	4.58 (±4.45)

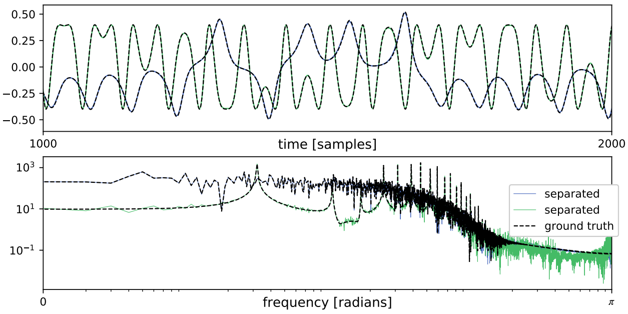

Figure 3: A separated mixture of a chaotic time series and frequency modulated sinusoid. The left plot shows the original and estimated sources, respectively. The right plot shows the respective power spectra from shows how the frequency content completely overlaps.

[8] Aapo Hyvarinen and Hiroshi Morioka, “Unsupervised feature extraction by time-contrastive learning and nonlinear ica,” in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., pp. 3765–3773. Curran Associates, Inc., 2016.

[9] Philemon Brakel and Yoshua Bengio, “Learning independent features with adversarial nets for non-linear ica,” *arXiv e-prints*, vol. 1710.05050, Oct. 2017.

[10] Mike E Davies and Christopher J James, “Source separation using single channel ica,” *Signal Processing*, vol. 87, no. 8, pp. 1819–1832, 2007.

[11] Pritish Chandna, Marius Miron, Jordi Janer, and Emilia Gómez, “Monoaural audio source separation using deep convolutional neural networks,” in *LVA/ICA*, 2017.

[12] E. M. Grais and M. D. Plumbley, “Single Channel Audio Source Separation using Convolutional Denoising Autoencoders,” `https://arxiv.org/abs/1703.08019`*ArXiv e-prints*, Mar. 2017.

[13] Laxmi Pandey, Anurendra Kumar, and Vinay Namboodiri, “Monoaural audio source separation using variational autoencoders,” in *Interspeech*, 2018.

[14] Y. Luo and N. Mesgarani, “TasNet: time-domain audio separation network for real-time, single-channel speech separation,” `https://arxiv.org/abs/1711.00541`*ArXiv e-prints*, Nov. 2017.

[15] D. Stoller, S. Ewert, and S. Dixon, “Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation,” `https://arxiv.org/abs/1806.03185`*ArXiv e-prints*, June 2018.

[16] Cédric Févotte, Nancy Bertin, and Jean-Louis Durrieu, “Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis.,” *Neural Computation*, vol. 21, no. 3, pp. 793–830, 2009.

[17] Paris Smaragdis and Judith C. Brown, “Non-negative matrix factorization for polyphonic music transcription,” in *In IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2003, pp. 177–180.

[18] S. Venkataramani and P. Smaragdis, “End-to-end Networks for Supervised Single-channel Speech Separation,” `https://arxiv.org/abs/1810.02568`*ArXiv e-prints*, Oct. 2018.

[19] Floris Takens, “Detecting Strange Attractors in Turbulence,” in *Dynamical Systems and Turbulence, Warwick 1980*, David Rand and Lai-Sang Young, Eds., vol. 898 of *Lecture Notes in Mathematics*, chapter 21, pp. 366–381. Springer, Berlin, 1981.

[20] Ethan Manilow, Prem Seetharaman, and Bryan Pardo, “The northwestern university source separation library,” in *Proceedings of the 19th ISMIR Conference*. Sept. 2018, ACM.