Single Channel auditory source separation with neural network

Zhuo Chen

Submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY
2017

(This page intentionally left blank)

(This page intentionally left blank)

Abstract

Single Channel auditory source separation with neural network

Zhuo Chen

Although distinguishing different sounds in noisy environment is a relative easy task for human, source separation has long been extremely difficult in audio signal processing. The problem is challenging for three reasons: the large variety of sound type, the abundant mixing conditions and the unclear mechanism to distinguish sources, especially for similar sounds.

In recent years, the neural network based methods achieved impressive successes in various problems, including the speech enhancement, where the task is to separate the clean speech out of the noise mixture. However, the current deep learning based source separator does not perform well on real recorded noisy speech, and more importantly, is not applicable in a more general source separation scenario such as overlapped speech.

In this thesis, we firstly propose extensions for the current mask learning network, for the problem of speech enhancement, to fix the scale mismatch problem which is usually occurred in real recording audio. We solve this problem by combining two additional restoration layers in the existing mask learning network. We also proposed a residual learning architecture for the speech enhancement, further improving the network generalization under different recording conditions. We evaluate the proposed speech enhancement models on CHiME 3 data. Without retraining the acoustic model, the best bi-direction LSTM with residue connections yields 25.13% relative WER reduction on real data and 34.03% WER on simulated data.

Then we propose a novel neural network based model called "deep clustering" for more general source separation tasks. We train a deep network to assign contrastive embedding vectors to each time-frequency region of the spectrogram in order to implicitly predict the segmentation labels of the target spectrogram from the input mixtures. This yields a deep network-based analogue to spectral clustering, in that the embeddings form a low-rank pairwise affinity matrix that approximates the ideal affinity matrix, while enabling much faster performance. At test time, the clustering step "decodes" the segmentation implicit in the embeddings by optimizing K-means with respect to the unknown assignments. Experiments on single-channel mixtures from multiple speakers show that a speaker-independent model trained on two-speaker and three speakers mixtures can improve signal quality for mixtures of held-out speakers by an average over 10dB.

We then propose an extension for deep clustering named "deep attractor" network that allows the system to perform efficient end-to-end training. In the proposed model, attractor points for each source are firstly created the acoustic signals which pull together the time-frequency bins corresponding to each source by finding the centroids of the sources in the embedding space, which are subsequently used to determine the similarity of each bin in the mixture to each source. The network is then trained to minimize the reconstruction error of each source by optimizing the embeddings. We showed that this frame work can achieve even better results.

Lastly, we introduce two applications of the proposed models, in singing voice separation and the smart hearing aid device. For the former, a multi-task architecture is proposed, which combines the deep clustering and the classification based network. And a new state of the art separation result was achieved, where the signal to noise ratio was improved by 11.1dB on music and 7.9dB on singing voice. In the application of smart hearing aid device, we combine the neural decoding with the separation network. The system firstly decodes the user's attention, which is further used to guide the separator for the targeting source. Both objective study and subjective study show the proposed system can accurately decode the attention and significantly improve the user experience.

Contents

(This page intentionally left blank)

List of Figures

(This page intentionally left blank)

List of Tables

(This page intentionally left blank)

List of Abbreviations

- FFT - Fast Fourier Transform
- DFT - Discrete Fourier Transform
- STFT - Short Time Fourier Transform
- CASA - Computational Auditory Scene Analysis
- NMF -Non Negative matrix factorization
- DNN - Deep Neural Network
- RNN - Recurrent Neural Network
- ASR - Automatic Speech Recognition
- SE - Speech Enhancement
- AE - Auto Encoder
- DC - Deep Clustering
- DAnet - Deep Attractor Network
- MIREX - Music Information Retrieval Evaluation eXchange
- LSTM - Long Short Term Memory
- BLSTM - Bi-directrional Long Short Term Memory
- SDR - Signal to Distortion Ratio
- SIR - Signal to Interference Ratio
- SAR - Signal to Artifact Ratio
- SNR - Signal to Noise Ratio
- MMSE - Minimum Mean Square Estimators
- PESQ - Perceptual Evaluation of Speech Quality
- RMS - Root Mean Squared
- SGD - Stochastic Gradient Descent

(This page intentionally left blank)

Acknowledgments

This thesis is dedicated to my wife Jie and my daughter Claire. Jie, I feel so lucky that I can have you accompanied for the journey of my PhD, I love you. Claire, having you is the best thing that happened to me.

A special thanks to my parents, I can't even count how much help I received from you. It is your selfless love and help that led me to where I am. I'd like also to express my gratefulness to my parents-in-law. Without your help to take care of Claire, I would need at least another two years to finish my PhD, thank you so much!

Being very fortunate, I had a chance to work with Professor Dan Ellis during my master in 2011. From this opportunity, I started my career in audio research. Dan, words can't express my gratefulness to you. It is you who led me into this exciting field and I can always remember each guidance and help from you, Thank you so much!

After starting my PhD in 2012, I had the opportunities to get in touch with many professors. They are all very kind and from them I built a concrete background for my later research. Here I'd like to thank them all. Among them I'd like to specially express my gratefulness to Professor John Wright, who taught me in three different classes, where I extend the course project to my first publication.

In LabROSA, I am happy to meet and collaborate with many friends, who are all great researchers. Not only I learnt different perspective in research from them, more importantly, the time we spend together helped me to understand and adopt the entirely different culture in US. Colin, Dawen, Brian, Hélène, Byung Suk, Matt, Thierry, Courtenay, Rachel, Cyril, Diego, Minshu, James, Andy, and Douglas, thank you!

In 2014, I started an eight-month internship in Mitsubishi Electric Research Lab in Cambridge, which is definitely one of the most exciting experience in my life. Together with John, Shinji, Jonathan and Hakan, we made so many exciting works and I benefited so much from the collaborations, meetings and even casual chats. Here I would like to express my deepest appreciation to them. And I want to also thank my dear friends in MERL, Scott, Umut and Takuya, for the constructive discussions and the fun we had.

Then I had another three-month internship in Microsoft in Bellevue, which is also extremely beneficial. Although the time is short, with the help of my colleagues and friends there, we made full use of it. Those memory there is so nice that I decide to join Microsoft as my next stop after PhD. Jinyu, Yan, Yifan, Shixiong, Yong, Qiang, Min, Kaisheng, Yun, thank you for all the help and let's build something greater together!

In early 2016, I started the collaboration with Professor Nima Mesgarani, who is extremely talented and helpful. With this help, I start to actually unveil the mystery of deep network. Here I'd like to express my very special appreciation to Nima. Nima, thank you so much and I am sure we can keep making great work together even after I graduate. And I'd also like to express my thanks to my labmates in NAPlab. It is the first time in my life to study in a place where there are more girls than men. Tasha, Bahar, Laura, James, Rashida, Prachi, and Hassan, I always enjoyed the atmosphere the collaboration and the conversation we had, thank you so much! And a special thanks to Yi, who I have the honor to mentor since 2016, I am impressed by your altitude and efficiency.

And I'd like to thank all my friends in New York and China, thanks a lot for the support and the fun we had.

Finally, it might be a little strange, but I'd also like to express my appreciation to all the researchers in this field. I am always exciting to read the new and cool works you made. Thank you for all of your excellent works that opened this exciting era in artificial intelligence.

Chapter 1

Introduction

The world is filled with an extremely large variety of sound. Most of the audio that human percept daily contain more than one audio source. For example, the average ambient noise in quite rural area is 30dB. And in more noisy cases, such as car. The noise level can reach 77dB(acoustics, 2016). Fortunately, human is especially good at separating the mixed sound. It is a natural gift for human to separate target sound under even very challenging environment. For example, even a child could easily distinguish and understand the voice of their parents, in very noisy environment such as restaurant and street.

More impressively, the whole separating process in brain is usually performed unconsciously(Mesgarani and Chang, 2012). Most people rarely notice the separating process. And the study showed that the separation could be largely enhanced when the attention is paid on specific sources(O'Sullivan et al., 2015a).

A natural question arisen, can this process be modeled with mathematical model? Solving the audio separation could not only help to build a better understanding for both audio signal and human brain, more importantly, it also has great practical value. The most obvious example is automatic speech recognition. Nowadays the best automatic speech recognition(ASR) system can reach human level performance when tested in matched conditions(Xiong et al., 2016b; Amodei et al., 2015). When tested under more complex environment, however, the recognition error rate increased greatly (Amodei et al., 2015). In recent years, as home intelligence becoming popular such as Amazon echo, further demand for robustness is required because usually the distance between the speaker and the sensor is much larger than the traditional ASR applications in cell phone. And the ability for noise and reverberation removal is even more important. In some other applications, the separation itself is the main feature, one such example is on music, where the task is to separate each instrument, which can be further used in applications such as audio resynthesis or karaoke.

1.1 background

Inspired by the observation on human and the practical demand, researchers started the odyssey for searching computational models for audio source separation in very early age of modern digital signal processing. In 1950s, the "cocktail party problem" was first introduced by Colin Cherry(Cherry, 1953), where the task was to separate each individual speaker in a cock-party, when all participant are talking simultaneously. Unfortunately, though the task seems easy for people, when researchers tried to build mathematical model to simulate the process, they found it was surprisingly difficult.

The problem of audio source separation is generally divided into two categories, multi-channel separation and the single channel source separation, where the number of the sensor(microphones) applied in the signal recording are different. Since the multi channel recording contains more than one sensor, the spatial information for each source can be inferred based the on time delay between microphones, i.e. the beamforming process(Fischer and Simmer, 1996; Anguera et al., 2007; Benesty et al., 2007; Kellermann, 1997). And this information provides extra clue for separation. In this thesis, we mainly focused on single channel recording, which is more challenging but common in real world scenarios.

In past six decades, different models were proposed, which can be roughly divided into signal processing based methods(Ephraim and Malah, 1985; Hu and Loizou, 2008, 2007), rule based methods(Brown and Cooke, 1994; Wang and Brown, 2006; Ellis, 1996b), and decomposition based methods(Raj et al., 2010; Chen and Ellis, 2013; Schuller et al., 2010). Unfortunately very few of them could achieve robust and high quality separation. In recent years, with increased data amount and stronger computational power, the deep neural network(DNN) based system brought revolution to this long stand problem. The DNN based systems significantly increased the separation performance for speech and noise. More impressively, by special design(Chen et al., 2016; Isik et al., 2016b; Hershey et al., 2016b; Yu et al., 2016), the DNN also demonstrates the ability to separate more challenging mixtures, such as unknown number of overlapped speakers. The successes with DNN model provided important steps towards eventually solving the cock-tail party problem.

1.2 Contribution

In this thesis, the problem of single channel audio source separation is analyzed and discussed in depth. The previous models proposed for this problem are systematically reviewed. We also introduce two extensions for deep neural network based speech enhancement system, which provided significantly better result for both separation and the recognition of the noisy speech.

We introduce two novel neural network - based models, deep clustering(DC) and deep attractor network(DAnet), which are designed for more general source separation problem, when the mixing sources are from similar sound families(within family separation), e.g. speech vs. speech, other than just speech vs. noise(cross family separation). In within family separation, we show that the separation is limited by two major difficulties, the permutation problem and the output mismatch problem. In deep clustering, the two problem was solved by using a clustering based objective function. This objective function also allowed the system to have many attractive features such as number of source invariance. We showed that DC outperformed the previous state of the art the by more than three times, and decreased the word error rate for recognition from 89.81% to 30.12%.

The deep attractor serves as the updated version of deep clustering, which enables several advantageous properties such as end-to-end optimization, flexible objective, reduced computational complexity etc. We discuss several variations of the DAnet, and show that the DAnet can provided even better result than DC.

We introduce two applications using the proposed model–the neural attention guided separation and music separation. In neural attention guided separation, the attention of the patient is firstly decoded from the neural signal collected with an invasive sensor, which controls the separation process for specific source. This application provide an important practice for the next generation of hearing aid devices. In music separation, we combine the DC and a classification objective, and achieve the state of the art performance in singing music separation. And we show that even under highly mismatched condition, the system can robustly separate the sining voice and background music with high quality.

Finally, we provide the code and a application program interface(API) for all the proposed system in this dissertation.

1.3 Overview

The thesis is organized as follows: background material about source separation is presented in Chapter 2. In Chapter 3, a brief introduction on feedforward and recurrent neural networks is given, with emphasize on the application in audio processing. The neural network based speech enhancement system is described in Chapter 4, where several aspects from modeling to implementation are discussed. In Chapter 5 and Chapter 6, the deep clustering and deep attractor network are introduced in detail. Chapter 7 and describes the applications of the proposed model. Finally the possible extensions and further works are discussed in the last chapter8 .

(This page intentionally left blank)

Chapter 2

Background

In this chapter, we introduce several basic aspects of signal channel audio source separation, including basic concepts and background methods.

2.1 Audio mixture

Since the sound is fundamentally the energy in medium, the mixture of the sound is the summation of the energies from individual sources. Mathematically, this relation can be represented as (2.1), where $y(t)$ is the audio mixture at time t and $x_i(t)$ refers the ith individual source.

$$y(t) = \sum_i x_i(t) \tag{2.1}$$

After recorded with microphone, the signal is usually discretized by sampling, while the additivity remains, which leads to (2.2), where n is used to represent each individual sample. And we usually refer this representation as time-domain representation.

$$y[n] = \sum_i x_i[n] \tag{2.2}$$

Though recently several new works(Li et al., 2016; Sainath et al., 2015b,a) showed that the model directly build in time domain could lead good performance in speech recognition and synthesis, the Fourier analysis is most commonly applied for audio processing. In Fourier analysis, the time domain signal is projected into a space that are support by the sinusoid functions with different frequencies, shown in (2.3), where X_i and Y are transformed single sources and the mixture.

$$Y(\omega) = \sum_i X(\omega) \tag{2.3}$$

When processing with long time series signal, such as audio. The time domain signal is often divided into overlapped segments, and then processed with Fourier transform separately.

This process is called short time Fourier transform(STFT). Since the Fourier transform is linear, the additivity remains, resulted in (2.4), where X and Y are the transformed time domain signal. The transformed representation is referred as spectrogram, and frame and bin are used to refer the time index t and frequency index f in spectrogram. Each number in spectrogram is known as T-F bin. We refer this spectrogram as frequency-domain representation.

$$Y(f,t) = \sum_i X_i(f,t) \tag{2.4}$$

Since the Fourier transformation often results in complex values, the magnitude, phase and power of spectrogram are referred as magnitude spectrogram, phase spectrogram and power spectrogram, accordingly. Study showed that phase is not very informative(Schluter and Ney, 2001), and since it is more difficult to process complex number, the phase information is usually discarded. Magnitude spectrogram or power spectrogram are usually selected for further processing. Such procedure would break the additivity, but when the mixing sources are not correlated, the additivity could be approximately preserved, as suggested in (2.5). Moreover, for applications in speech recognition, study showed that removing phase would not affect the recognition performance.

$$|Y(f,t)| \approx \sum_i |X_i(f,t)| \tag{2.5}$$

2.2 Mask

The mask is one of the most commonly used the representation in audio separation. As its name suggests, a mask $Min\mathbb{R}^{F\times T}$ is a matrix that can applied on the mixture spectrogram, by point-wise multiplication, to mask out the interference for the target source. Depend on the choice of the value, the mask can be roughly divided into three type: Binary mask, Ratio mask and Complex mask. In binary mask, each TF-bin in mask has the binary value, which means each TF bin can only belong to one source. Such mask is usually easier to optimize but has worse separation. In contrast, in ratio mask the elements have continuous value between 0 and 1, i.e. the soft assignment for each TF bin. In complex mask, all the elements have complex value, which could be directly applied on the complex mask. An example of mask is shown in fig 2.1. In fig 2.1, we can see that the ideal ratio mask can generate almost perfect separation.

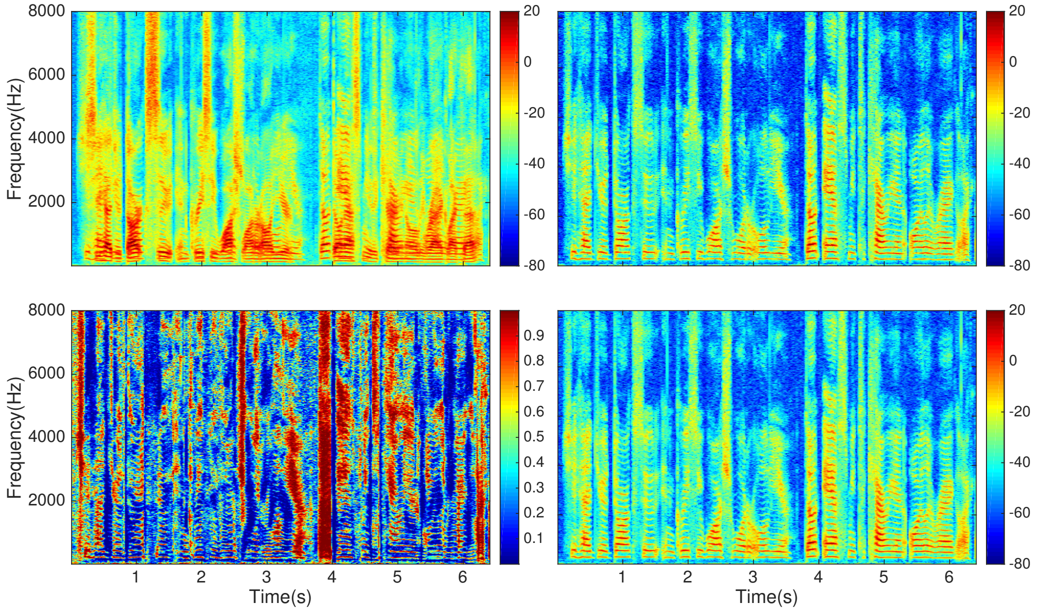


Figure 2.1: Upper left: The spectrogram of the the mixture between a male and a female speaker. Bottom left: The ideal ration mask for the female speaker. Upper right: The clean speech for the female speaker. Bottom right: The masked mixture.

2.3 Signal processing based source separation

In signal processing based source separation was mainly designed for speech enhancement, and was the first algorithm family proposed for this problem, in early 1970s. In this algorithm family, speech is usually assumed to follow specific distribution such as Gaussian or Laplacian. Then a maximum likelihood model is build based on this assumption. The noise is assumed to be stationary, whose statistical properties don't change through time. In practice, a voice activity detector is usually applied to noisy speech first, then the silent frames are collected to calculate the noise statistics, followed by the maximum likelihood optimization to get the speech.

Since most of the signal processing based model make over simplified assumption on speech, e.g. following gaussian distribution, and those method are not date driven, which means that the system could not learn from the actual data, the performance of signal processing based method are usually unsatisfiable and will not be further discussed in this thesis. We refer the reader who are interested in this family to (Ephraim and Malah, 1985; Hu and Loizou, 2008, 2007) for a more detailed description.

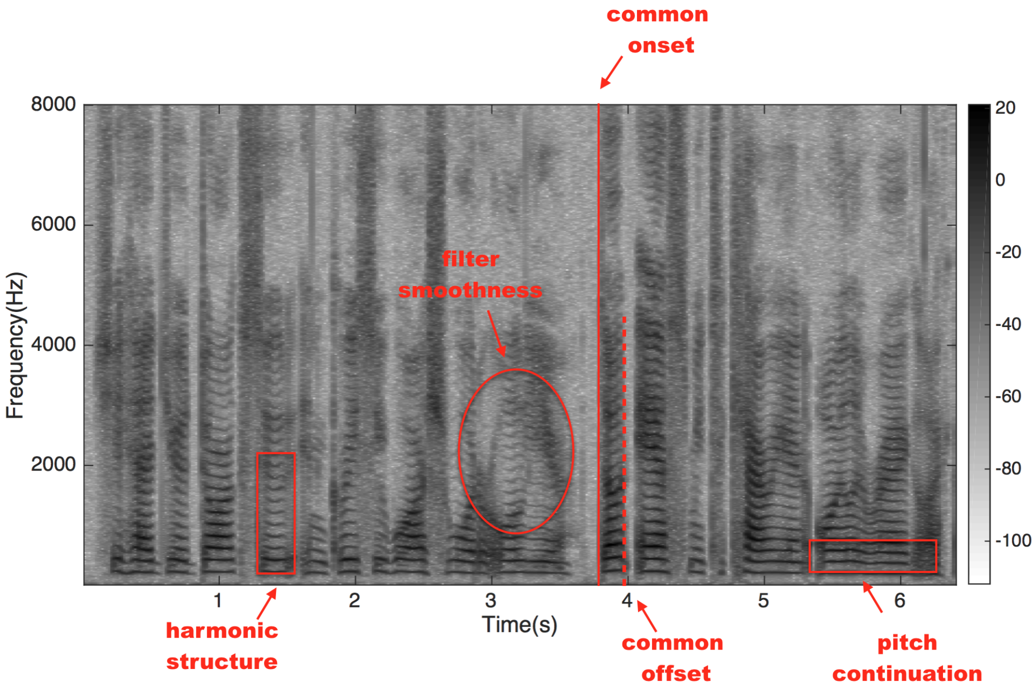


Figure 2.2: The rules formed from the observation.

2.4 Rule based source separation

Because of the physical structure, the speech contains several properties. Based on this observation, researchers proposed a serious of rule based separation(Brown and Cooke, 1994; Wang and Brown, 2006; Ellis, 1996b; Bregman, 1994), also referred as computational auditory scene analysis(CASA). The rules can be roughly summarized as follows.

- The common onset and/or offset
- Harmonic
- Pitch continuity
- Vocal tract continuity
- Pitch exclusiveness
- Common special location
- Common temporal modulation

Figure 2.2 shows one example of mixture spectrogram, with different rule annotation.

Among all rules, pitch is usually the most powerful ones. Because of the physical structure of human's vocal track, the changing of pitch involves several muscles, which decides that the change of pitch must be continuous. Meanwhile, human can only produce one pitch at a time. Therefore, for regions that contain two or more pitches, there must be more than one sources. The harmonic and onset can also provide useful supplement since the harmonic of the sound must be the multiples of the fundamental frequency, and all the harmonic start simultaneously.

Based the those clues, in a typical CASA system, for each sample, a feature by choice is firstly calculated. Then based on hand designed rules, the TF bins are grouping into sources. Finally, a binary mask is formed based on the assignment of each TF bin, to segment the mixture. A example set of rules is cited (Shao et al., 2010) as follows:

- Low-frequency signals are grouped based on periodicity and temporal continuity.
- High-frequency signals are grouped based on amplitude modulation and temporal continuity.
- For unvoiced sounds, use a auditory segmentation and segment classification.

For CASA based system, the feature extraction step are usually the most important one, since the feature has to clearly demonstrate the grouping effect, in order to guarantee the further clustering step to generate meaningful result. Therefore, the choice of pitch tracker (Wang and Seneff, 2000; Huang and Seide, 2000; Lee and Ellis, 2012) is essential in all CASA systems.

Though the idea in CASA is very intuitive, it suffers from many drawbacks. The most obvious one is that it only works on speech, where there is only one pitch per time, no pitch jump and has clear continuous structure. However, for a broader perspective of audio source separation, such assumptions usually don't hold. Another very important limitation is that all the rules are hand designed, based on the simple observation on few samples, which is clearly sub-optimum since no guaranteed that the formed rules can generalize, especially on the unvoiced sound. And since the final segmentation is based on the assignment, the best possible result is to form a oracle binary mask, which has been showed to be suboptimal in different scenarios (Wang, 2005; Kjems et al., 2009). Finally, the entire system largely depend on the accuracy of pitch tracker, however, the robustness of pitch tracker can usually not guaranteed under complex acoustic condition, which leads to less robustness in separation as well.

2.5 Decomposition based source separation

In rule based system(CASA), the rules are formed based the observation on spectrogram. A natural extension is to build a system that can automatically discover the rules from data. The decomposition based model is an early attempt for this direction. The basic assumption for the decomposition based model is that the audio spectrogram has low rank structure, which can be represented with a small number of basis, as shown in (2.6).

$$Y = WH \tag{2.6}$$

In (2.6), the spectrogram $Y \in \mathbb{R}^{F\times T}$ is decomposed into the matrix product of two matrix $W \in \mathbb{R}^{F\times K}$ and $H \in \mathbb{R}^{K\times T}$, where K is the hyper-parameter, usually much smaller than F and T, which resulted in the low rank approximation of Y.

With different constraint, the decomposition can resulted in different specific representation. For example, the additional orthogonal constraint changes 2.6 into principle component analysis(PCA)(Jolliffe, 2002), the sparsity constraint would lead to the sparse coding. In audio processing, the most popular decomposition is non-negative matrix factorization(NMF)(Lee and Seung, 2001), where W and H is constrained to be non-negative.

2.5.1 Non-negative matrix factorization

The basic formulation for NMF is shown in equation 2.7, where c is the index for each source. In 2.7, each source is firstly modeled by the low rank approximation then sum to the mixture. Since both W and H are non-negative, in reconstruction of Y, there is no cancellation between sources, which models the additivity between sources in mixture, as discussed above.

$$\begin{aligned} Y &= \sum_c W_c H_c \\ W_c &\geq 0 \\ H_c &\geq 0 \end{aligned} \tag{2.7}$$

$$\begin{aligned} &\min_{W,H} D(Y||WH) \\ &\text{s.t. } W \geq 0, H \geq 0 \end{aligned} \tag{2.8}$$

However, since the decomposition process is not convex, and the whole system is underdeterminated, simply apply the decomposition on the mixture would most likely not lead to any meaningful results. In practice, the NMF based system usually has the following pipeline, as shown in fig 2.3.

In training stage, the source model is usually leant by applying the decomposition on the clean sources, e.g. speech, noise, music etc. Through the decomposition, each clean source is mapped into a set of basis and activations. During the testing stage, the learnt bases for each source are fixed and only optimize the activation for each source, which makes sure that optimization is convex and a global optimum can be achieved. And finally each source is reconstructed by the pre-learnt bases and the corresponding activation. The basic NMF algorithm is given as follows. An example of NMF decomposition is given in fig 2.4.

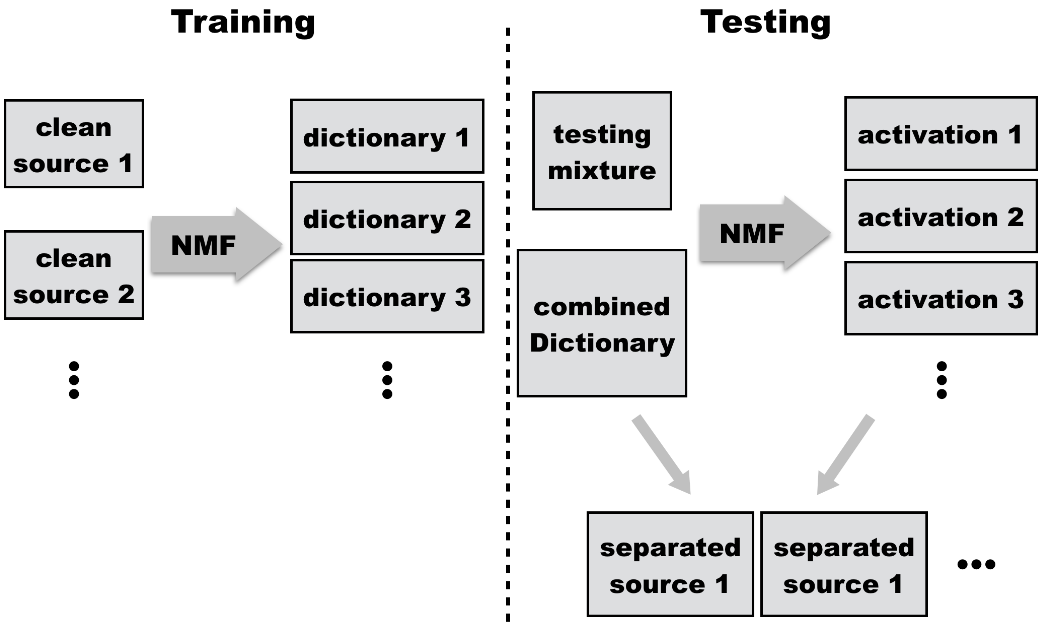


Figure 2.3: Left: The training process, where a set of dictionary is learnt for each individual source. Right: The testing process, where the dictionary is fixed. After the decomposition, the reconstructed source is synthesized by each source dictionary and corresponding activation.

2.5.2 Variation of NMF

Different extensions on NMF based system were proposed based on different observation on audio signal, which is briefly summarized below.

2.5.3 Sparse NMF

Sparse NMF(Hoyer, 2004; Schmidt and Olsson, 2006; Virtanen, 2007) further constraint the activation H to be sparse. Since the the dictionaries for each source usually shares a large amount of common pattern, e.g. pitch, the sparse constraint would increase the discrimination between sources. Moreover, since the complex pattern could usually be generating by simple patterns(for example, the white noise could be generated by the combination of all pitches), the sparsity activation would increase the robustness of the decomposition. The objective of sparse NMF is shown in 2.9, where the sparsity is introduced with an additional L1 penalty on activation H.

$$\begin{aligned} \min_{W,H} D(Y||WH) + \alpha \left\| H \right\|_1 \\ \text{s.t.} \quad W \geq 0, H \geq 0 \end{aligned} \tag{2.9}$$

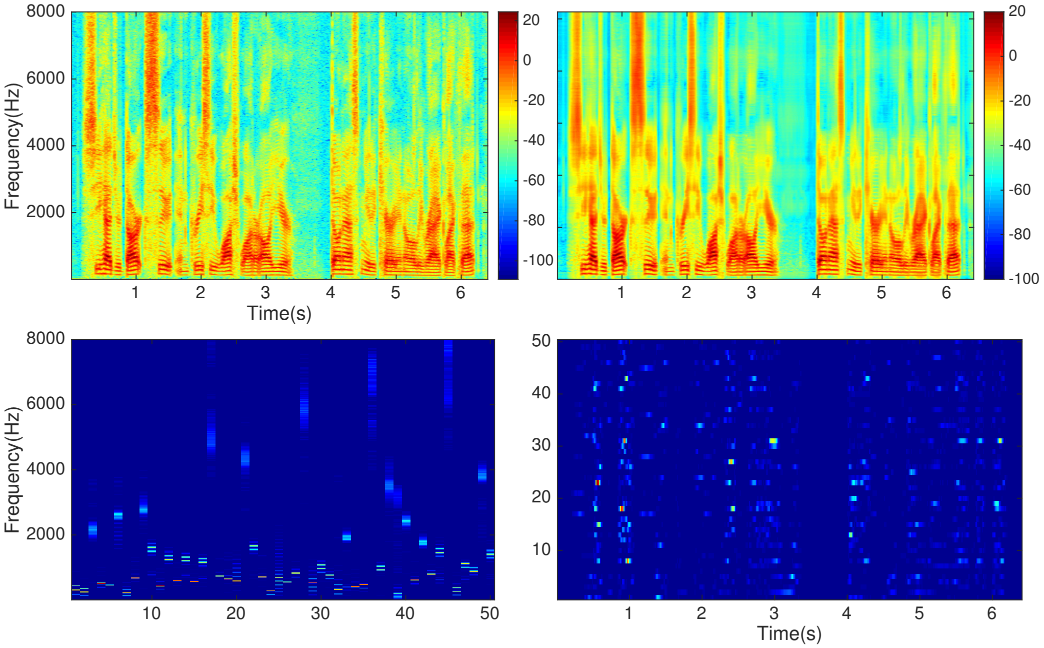


Figure 2.4: Upper left: The spectrogram of clean speech. Upper right: the reconstructed speech. Bottom left: the learnt bases through the decomposition. Bottom right: the learnt activation.

2.5.4 Convolutional NMF

In Convolutional NMF(Behnke, 2003; Bello, 2010; Chen et al., 2014), the spectrogram is decomposed into the convolution between the basis and activation, rather than the matrix multiplication, as shown in eqn. 2.10,Here, $\{W(\tau)\} \subset \mathbb{R}_+^{F\times K}, \tau = 1, ..., P$ is a set of time-varying basis elements, where each $W(\tau)$ encodes the spectra pattern of each patch at its τth frame. $H \in \mathbb{R}_+^{K\times T}$ is the corresponding set of non-negative convolutive activations, and $\overset{\tau\rightarrow}{H}$ refers the "shift" operation, which pads τ zero-columns to the left of H and truncates its rightmost $P - \tau$ columns to maintain shape, with $\overset{\leftarrow\tau}{H}$ defined analogously for left-shift. Compared with NMF model, the proposed model decomposes speech as the sum of convolutions between the dictionary elements and their corresponding activations. Rather than individual speech spectra, the dictionary now consists of two-dimensional "patches" of speech, which capture the energy distribution in each frequency bin over subsequent points in time. Modeling temporal dependencies in this way prevents the speech model from erroneously capturing transient noise bursts.

$$\begin{aligned} &\min_{W,H} D(Y|| \sum_{\tau=0}^{P-1} W(\tau) \overset{\tau\rightarrow}{H}) + \alpha \|H\|_1 \\ &\qquad\qquad \text{s.t.} \quad W \geq 0, H \geq 0 \end{aligned} \tag{2.10}$$

2.5.5 Robust NMF

Based on the observation that the noise usually has low rank structure, but hard to predict before hand, the robust non-negative matrix factorization(RNMF) combine the the NMF with robust principle component analysis(RPCA)(Candès et al., 2011; De la Torre and Black, 2001), another commonly used decomposition technic(Zhang et al., 2011; Chen and Ellis, 2013). In RNMF, the spectrogram is decomposed into a dictionary reconstruction and a low rank residual, as shown in eqn 2.11. RNMF is specifically designed for the problem of speech enhancement, where the low rank residual models the noise and the dictionary models the speech. In eqn 2.11, an additional noise $L$is incorporated in the objective function. The noise is constraint to have low rank structure, which is enforced by minimizing its nuclear norm.

$$\begin{aligned} &\min_{H,L,E} \|E\|_{\mathrm{F}}^2 + \lambda_H \|H\|_1 + \lambda_L \|L\|_* + \mathcal{I}_+(H) \\ &\qquad\qquad \text{s.t.} \qquad Y = WH + L + E \end{aligned} \tag{2.11}$$

2.5.6 Limitation

The main limitations for decomposition based method lays in three aspects.

Firstly, the decomposition is linear, in other words, the spectrogram is the linear combination of basis. Such assumption is made to simplify the computation, however, it omits several very important aspect of audio, for example, the long time dependence, the modulation etc.

More importantly, the representation learnt through the decomposition is “shallow”. In other words, to required number of parameter increases linearly with the data variation. Therefore, it is extremely difficult to model the audio signal in detail with reasonable model size. This limitation fundamentally prevent the decomposition model to achieve high quality separation.

Additionally, the run time complexity for the decomposition model is expensive. Most of the decomposition based models are solved through iterative method, which usually require dozens of iteration to converge, even during the testing time. Therefore it is difficult to build application for the real time application. To increase the speed, the complexity of the model has to decrease.

(This page intentionally left blank)

Chapter 3

Neural Network

This chapter mainly introduced several fundamental aspects about artificial neural network.

Inspired from the biology observation that, though the basic nerve cells are very simple. The most common neuron is named as perceptron(Hagan et al., 1996), which has the form in 3.1. In 3.1, o is the output, i_k refers different input, with corresponding weight w_k and bias b, $f(\cdot)$ refers a non-linear function, which convert the weighted sum into a binary decision. By combining a large amount of neurons, neural network was designed to have the ability of modeling arbitrary functions. With different architecture, the network has different properties.

$$o = f(\sum_k i_k w_k + b) \tag{3.1}$$

3.1 Feed forward network

The feedforward neural network was the first and simplest type of artificial neural network devised(Zhang et al., 1998; Morgan and Bourlard, 1990). In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network. Each layer of the feedforward network consists the concatenation of perceptrons, followed by a non-linear function. as shown in fig 3.1. The common choices of non-linearity are sigmoid, softmax and rectifier linear function, as shown in eqn 3.2.

$$\begin{aligned}
&\text{softmax: } f(x_i) = \frac{e^{x_i}}{\sum_i e^{x_i}} \\
&\text{sigmoid: } f(x_i) = \frac{1}{1 + e^{-x_i}} \\
&\text{ReLU: } f(x_i) = \max(0, x)
\end{aligned} \tag{3.2}$$

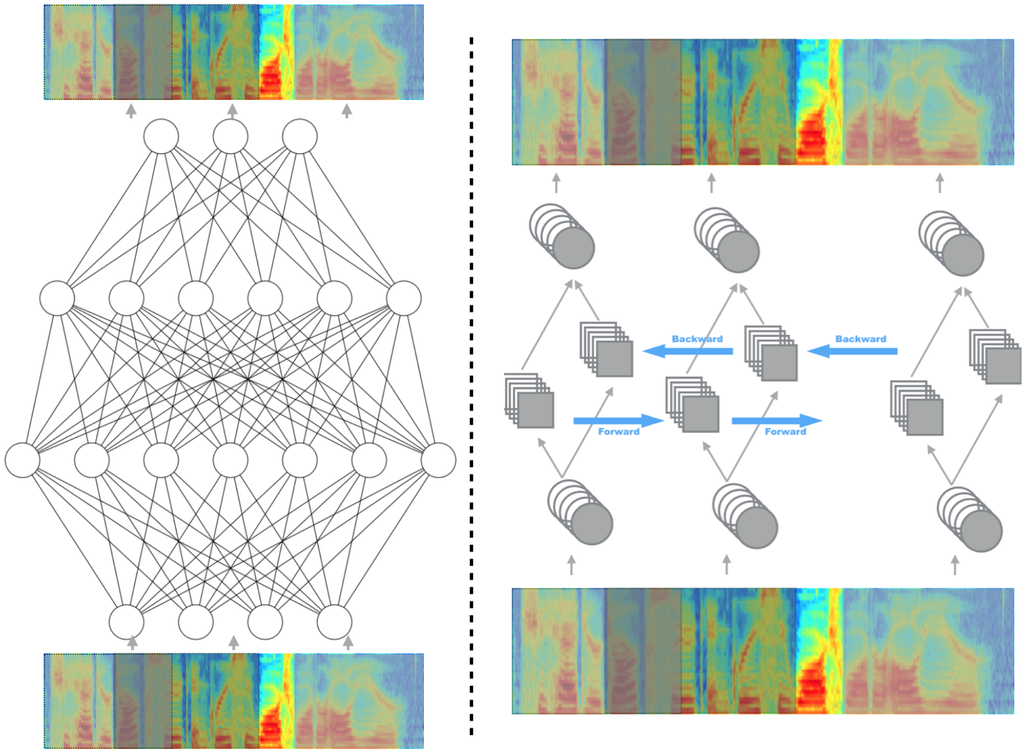


Figure 3.1: Left: An auto-encoder that consists of four layer fully connected network, with the input of the spectrogram with context window. Right: An auto-encoder of bi-directional recurrent neural network.

3.2 Recurrent network

A recurrent neural network (RNN) is a class of artificial neural network where connections between units form a directed cycle(Mikolov et al., 2010; Funahashi and Nakamura, 1993). This creates an internal state of the network which allows it to exhibit dynamic temporal behavior. Unlike feedforward neural networks, RNNs can use their internal memory to process arbitrary sequences of inputs. In each layer of RNN, the relation between input and the output is given in eqn 3.3. In eqn 3.3, t indexes the time step, x, h and y are input, hidden states and network output, W_h and the W_y are the weight with respect to the input for hidden state and output, with corresponding bias b_h and b_y, and U refers the weight between consecutive hidden states, finally, the non-linearity for the hidden state and output are denoted as $f(\cdot)$ and $g(\cdot)$.

As we can see from 3.3, compared with feed forward network, the recurrent network has an additional hidden state h, which is coupled by the additional weight U and passed through time steps. Therefore, the input from previous step could also affect the current output through this coupling, and the output at any time is the accumulation of all previous input, in other words, the network has the "memory" of the past input. This property makes the

RNN very appealing in audio processing since such dependence through time is one of the key feature in audio. A typical RNN is shown in fig 3.1.

$$\begin{aligned} h_t &= f(W_h x_t + U h_{t-1} + b_h) \\ y_t &= g(W_y h_t + b_y) \end{aligned} \tag{3.3}$$

3.2.1 Long short term memory

Although designed for the sequence processing, RNN usually suffers from the "vanishing gradient" problem, which prevent the network from capturing the long time dependency. The main reason for "vanishing gradient" comes from the coupling between time steps. In RNN, the time is captured by the multiplication of a weight matrix, i.e. W_h in 3.3. Then when the eigenvalue of $W_h W_h^T$ is not one, the gradient through time will explode or vanishing, a more detailed discussion can be found in (Bengio et al., 1994).

To fix this problem, in(Hochreiter and Schmidhuber, 1997), researchers proposed a novel structure to replace the multiplicative coupling, i.e. the long short term memory(LSTM) network. The standard LSTM memory block consists of one memory cell and three gates: input gate, output gate and forget gate. The memory cell stores the state of the memory block while the gates controls the flow of activation and error, which decide the behavior of memory cell. The input gate and output gate control the flow of the activation enter and leave the memory cell. To enable the memory block to model the dependency in subsequences, the forget gate controls the memory cell to "forget" the previous activations.

We consider an LSTM memory block at nth layer with an input vector $\boldsymbol{h}_t^{n-1}$ and output activation $\boldsymbol{h}_t^n$ at frame t (here, we omit the utterance index). Note that the input vector at the first layer corresponds to the observation vector, i.e., $\boldsymbol{h}_t^0 = \boldsymbol{y}_t$. We first define the concatenated vector of output activation $\boldsymbol{h}_{t-1}^n$ at previous time frame $t-1$ and the $n-1$th layer output activation $\boldsymbol{h}_t^{n-1}$ at current time frame t as $\boldsymbol{m}_t^n \triangleq [(\boldsymbol{h}_t^{n-1})^\top, (\boldsymbol{h}_{t-1}^n)^\top]^\top$. Then, the LSTM memory block has a memory cell (return: $\boldsymbol{c}_t$), which are obtained from the input gate (return: $\boldsymbol{i}_t$) and forget gate (return: $\boldsymbol{f}_t$):

$$\begin{aligned} \boldsymbol{i}_t^n &= \sigma(\boldsymbol{W}_{im}^n \boldsymbol{m}_t^n + \boldsymbol{W}_{ic}^n \boldsymbol{c}_{t-1}^n + \boldsymbol{b}_i^n), \\ \boldsymbol{f}_t^n &= \sigma(\boldsymbol{W}_{fm}^n \boldsymbol{m}_t^n + \boldsymbol{W}_{fc}^n \boldsymbol{c}_{t-1}^n + \boldsymbol{b}_f^n), \\ \boldsymbol{c}_t^n &= \boldsymbol{f}_t^n \odot \boldsymbol{c}_{t-1}^n + \boldsymbol{i}_t^n \odot \tanh(\boldsymbol{W}_{cm}^n \boldsymbol{m}_t^n + \boldsymbol{b}_c^n), \end{aligned} \tag{3.4}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are affine transformation parameters to be estimated at the training step. $\odot$, $\sigma(\cdot)$, and $\tanh(\cdot)$ denote the element-wise product operation, sigmoid function, and hyperbolic tangent function, respectively. The memory cell and input and forget gates are calculated from the concatenated activation vector $\boldsymbol{m}_t^n$ and the cell vector $\boldsymbol{c}_{t-1}^n$ at the previous frame. The relationship between $\boldsymbol{c}_t^n$ and $\boldsymbol{c}_{t-1}^n$ is controlled by the forget gate $\boldsymbol{f}_t^n$ dynamically, which enables to retain the long-range dependency of the cell, unlike the hidden state in standard RNNs.

Once we obtain cell vector $\boldsymbol{c}_t^n$, we can calculate output gate vector $\boldsymbol{o}_t^n$, and finally calculate output activation $\boldsymbol{h}_t^n$ as follows:

$$\begin{aligned} \boldsymbol{o}_t^n &= \sigma(\boldsymbol{W}_{om}^n \boldsymbol{m}_t^n + \boldsymbol{W}_{oc}^n \boldsymbol{c}_t^n + \boldsymbol{b}_o^n), \\ \boldsymbol{h}_t^n &= \boldsymbol{o}_t^n \odot \tanh(\boldsymbol{c}_t^n). \end{aligned} \tag{3.5}$$

A set of these equations is a basic feed-forward operation of the LSTM memory block at nth layer. At the top layer (N), output activation $\boldsymbol{h}_t^N$ is further calculated by the following affine transformation:

$$\hat{\boldsymbol{h}}_t = \boldsymbol{W}^N \boldsymbol{h}_t^N + \boldsymbol{b}^N. \tag{3.6}$$

This final activation $\hat{\boldsymbol{h}}_t$ would be used for the regression, classification through the softmax operation, or masking function through the sigmoid operation.

Similar to the LSTM, Bi-directional LSTM has the same memory block as the basic unit. Instead of propagating the information in one time direction, in BLSTM layer, there are two separated propagation sequences from the both time directions. Therefore, unlike equation (3.6), the BLSTM neural network obtains the final activation $\hat{\boldsymbol{h}}_t$ by using both the final activations from the past $\boldsymbol{h}_t^{N\rightarrow}$ and future $\boldsymbol{h}_t^{N\leftarrow}$, as follows:

$$\hat{\boldsymbol{h}}_t = \boldsymbol{W}^{N\rightarrow} \boldsymbol{h}_t^{\rightarrow} + \boldsymbol{W}^{N\leftarrow} \boldsymbol{h}_t^{\leftarrow} + \boldsymbol{b}^N. \tag{3.7}$$

This property enables the BLSTM network further explore the connection within contexts, and often lead to better performance than LSTM.

3.3 Objective function

For each neural network, a objective function is needed to measure the “correctness” of the model during the training stage. The difference between the network output and the objective can be used to update the network parameters, and lead to better accuracy. For speech enhancement, the most commonly used objective is the sum of square error(SSE), as defined in 3.8. In 3.8, y and x refer the label and network output, i indexes the dimension. The euclidian distance between y and x across each dimension is summed, and used to optimize the network.

$$\mathcal{L} = \sum_i (y_i - x_i)^2 \tag{3.8}$$

3.4 Back propagation

The back propagation(Chauvin and Rumelhart, 1995) is a common method of training artificial neural networks and used in conjunction with an optimization method such as gradient descent. The algorithm repeats a two phase cycle, propagation and weight update.

When an input vector is presented to the network, it is propagated forward through the network, layer by layer, until it reaches the output layer. Then the network parameters are updated with gradient based optimization methods. In the back propagation process, the chain rule is used to pass the gradient backward through layer, as shown in eqn 3.9

$$\begin{aligned} \mathcal{L} &= f(x) \\ \frac{\partial \mathcal{L}}{\partial x} &= \frac{\partial \mathcal{L}}{\partial f}\frac{\partial f}{\partial x} \end{aligned} \tag{3.9}$$

3.5 Regularization

Due to their tendency to overfit, some form of regularization is typically necessary to ensure that the optimized neural network's variance is not too high. A common regularizer which is used in many machine learning models is to include in the loss function a penalty on the norm of the model's parameters. In neural networks, adding the L^2 penalty $\lambda \sum_i \theta_i^2$ where λ is a hyperparameter and θ_i are the model's parameters (e.g. individual entries of the weight matrices and bias vectors) is referred to as "weight decay" (Hanson and Pratt, 1989). This can prevent the weight matrix of a given layer from focusing too heavily (via a very large weight value) on a single unit of its input. A related term is $\lambda \sum_i |\theta_i|$ which effectively encourages parameter values to be zero (Bengio, 2012).

A completely different regularization method which has recently proven popular in neural networks is "dropout" (Hinton et al., 2012b). In dropout, at each training iteration each unit of each layer is randomly set to zero with probability p. After training, the weights in each layer are scaled by $1/p$. Dropout intends to prevent the units of a given layer from being too heavily correlated with one another by randomly artificially removing connections. More simply, it provides a source of noise which prevents memorization of correspondences in the training set and has been shown empirically and theoretically (Wager et al., 2013) to be an effective regularizer.

In practice, the technique of "early stopping" is almost always used to avoid overfitting in neural network models (Prechelt, 2012). To utilize early stopping, during training the performance on a held-out "validation set" (over which the parameters of the network are not optimized by gradient descent) is computed. Overfitting is indicated by performance degrading on the validation set, which simulates real-world performance. As a result, early stopping effectively prevents overfitting by simply stopping training once the performance begins to degrade. The measure of performance and criteria for stopping my vary widely from task to task (and practitioner to practitioner) but the straightforwardness and effectiveness of this approach has led it to be nearly universally applied.

3.6 Neural network in audio processing

3.6.1 Context window

The context window is one the most commonly used trick in the audio processing(Hermansky et al., 2000; Hinton et al., 2012a). Context window refers the a moving window on the input, which extents the current input to a combination of feature within a time range. The intuition behind the context window is straightforward. In English, syllable is the the shortest meaningful acoustic unit, which usually has length around 100ms, while in most speech processing application, the spectrogram frame rate is 10ms. Therefore if viewed individually, each frame in spectrogram often contain limited information and large amount of randomness, which cause difficulties for the neural network to generalize. Adding the context window could form more robust feature, thus simplify the learning process. This trick has been shown to be very effective in applications such as automatic speech recognition, speech enhancement, machine translation etc.

3.6.2 Noisy auto-encoder

A typical auto-encoder network(Deng et al., 2010; Sainath et al., 2012; Lange and Riedmiller, 2010) is shown in fig 3.2. It consists of two parts, an encoder and a decoder. With the encoder, the network first embeds the input to a fixed dimension "code", which usually has fewer dimension than the input. Then another set of parameter, the decoder, is used to convert the embedded representation back to the input. The whole system maps the input to itself, thus called "auto-encoder".

Auto-encoder is an important model in neural network for two reasons. Firstly, the auto-encoder maps the input to a lower dimension representation, which is invertible. In other words, it can discover a more compact representation for the data, and remove the unnecessary variances. In other words, it can have a better generalization of the data, which is essential for data processing. More importantly, since the auto-encoder learns a mapping to itself, it doesn't require additional label to generate the gradient. Therefore, the data amount is always sufficient for auto-encoder network. Because of these two properties, the auto-encoder is usually used to initialize the network.

In practice, to improve the robustness and prevent overfit of the auto-encoder, a random noise is usually added into the input of the network, which converts the system to "noisy auto-encoder". Experiments showed that the noisy auto encoder can learn a significantly more robust embedding of the input(Bengio et al., 2013; Rifai et al., 2011; Vincent et al., 2010).

The noisy auto-encoder learns a transformation that can convert the noisy input to its clean version, which perfectly match the requirement for speech enhancement, where the task is to remove the noise from the noisy speech. Thus most of the neural network based speech enhancement system uses the auto-encoder based architecture(Xu et al., 2014; Narayanan and Wang, 2013b). The neural network based speech enhancement system is introduced in chapter 4.

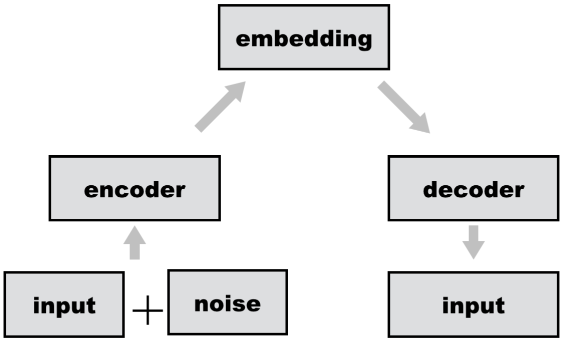


Figure 3.2: The input is firstly combined with additional noise and pass through the encoder to form the lower dimensional embedding. Then the embedding is used to generate the input through the decoder.

(This page intentionally left blank)

Chapter 4

Neural Network based speech enhancement

As an universal function approximater, neural network has been successfully applied in different applications. In the problem of speech enhancement, the neural network also achieved significant performance improvement, when compared with traditional models introduced in chapter 2.

As discussed in last chapter, most of the speech enhancement system use auto-encoder architecture. Among them, two branches are most commonly adopted, the feature mapping network and mask learning network.

4.1 Feature mapping

The feature mapping network is straightforward. The network takes the noisy spectrogram as input Y, and targets at the clean spectrogram X, with the objective function shown in 4.1. In 4.1, $\Phi(\cdot)$ refers the non-linear transformation of neural network, and $\mathcal{D}$ refers the distance between the output and the reference. Usually euclidian distance is chosen. The feature mapping network is simple but effective. The feature mapping network achieved significant performance improvement in comparison to the traditional models.

$$\mathcal{L} = \mathcal{D}(X||\Phi(Y)) \tag{4.1}$$

However, the feature mapping network also suffers from several drawbacks. The most obvious one is the unbounded dynamic range. Since the network directly output the clean spectrogram, which is unbounded, the output dynamic range has to be large enough to cover all possible volumes. Such procedure will largely increase the redundancy for the learning task. For example, for the same utterance with different amplitude, the feature mapping network need to generate completely different result, which will make the network more difficult to converge.

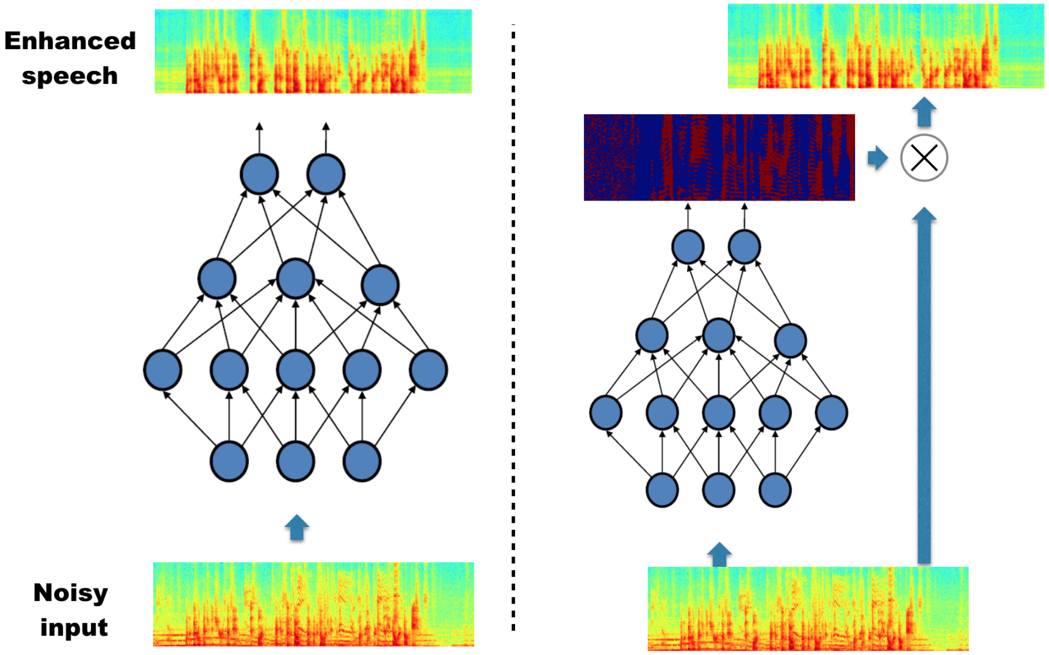


Figure 4.1: Left: the feature mapping architecture. Right: the mask learning architecture

4.2 Mask learning

Another commonly used architecture is the mask learning network. Different with the feature mapping network that directly output the clean spectrogram, the mask learning network output a mask, which can be applied back to the noisy signal and mask out the noise, as shown in figure 4.1. A typical objective function is shown in 6.2, where Y and X refer the input feature and clean reference accordingly. $\Phi(\cdot)$ refers the non-linear function learnt by the neural network and $\mathcal{D}(\cdot)$ refers the distance measure, which is usually euclidean distance.

$$\mathcal{L} = \mathcal{D}(X||\Phi(Y)) \tag{4.2}$$

There are two ways to learn the mask. The first one is called mask approximation (MA) which directly minimizes the distance between the learned mask and the target mask (Narayanan and Wang, 2013a, 2014b,a; Wang et al., 2014a). The second is called signal approximation (SA) which minimizes the distance between the target signal and the signal constructed by applying the estimated mask to the distorted signal (Weninger et al., 2014b; Erdogan et al., 2015b).

Since the mask is bounded(e.g. the mask usually has value in $[0, 1]$), the mask learning network has the fixed dynamic range. Therefore it is easier for the the mask learning network to generalize different noises, conditions. Moreover, as shown in fig 4.1, during the separation, since the mixture is re-introduced to the computation, the network only needs to filter out the noisy part. Compared with the feature mapping network, where the system need to both remove the noise and remember the clean reference, the learning task for mask based system is easier, and thus usually leads to better performance, as reported in (Narayanan and Wang, 2013b; Wang and Wang, 2013; Healy et al., 2013)

We choose signal approximation based mask learning in this study. It is shown in (Weninger et al., 2014b) that SA is better than MA as its final target is directly related with the source signal. The objective of SA based mask learning for speech enhancement is:

$$\mathcal{L} = \|X - \Phi(Y) \odot M\|_2^2 , \tag{4.3}$$

where X and M are clean speech and noisy speech in the mask learning *output* feature domain; Y is the noisy speech in the mask learning *input* feature domain; $\Phi(\cdot)$ is the mask estimation function learnt from neural networks. $\Phi(Y) \in [0, 1]$ is the learnt soft mask.

The effectiveness of the auto-encoder based speech enhancement has been shown in different works(), one of the most representative one is on the 2nd CHiME speech separation and recognition challenge, where different speech enhancement methods were evaluated under different frame work.

The noisy data in ChiME challenge was constructed from the Wall Street Journal dataset. The clean 16kHz WSJ data was firstly convolved with the room impulse response to model the reverberation. Then the reverberated speech was mixed with the recorded background noise the at 6 different SNRs from $-6dB$ to $9dB$. The training set contains 7138 utterances from 83 speakers, totaling 14.5 hours. The development set contains 4.5 hours data, which consists of 2460 utterances from 10 speakers that are disjoint with the training set. The test set consists of 1980 utterance from 8 speakers, 4 hours in total.

Four models were evaluated(Le Roux et al., 2015a; Weninger et al., 2015; Chen et al., 2015), including two mask learning based models using feedforward network(DNN) and bi-directional long short term memory(BLSTM) network. The DNN consists of three layers, each layer had 1024 nodes, with hyperbolic tangent function for non-lineariry, followed by an output layer, which was a feedforward layer of 100 nodes with sigmoid function. The input feature for the DNN were the 100 dimension log mel-filterbank, with $T = 9$ context window. The spectrogram was calculated using 25 ms window size and 10 ms window shift. For the BLSTM network, each BLSTM layer had 300 forward LSTM cell and 300 backward cell. Similar to DNN network, an output layer was added after the BLSTM network to generate the mask. No context window is used for the BLSTM network. All the input feature was normalized with zero mean and unit variance.

The discriminate non-negative matrix factoration(DNMF) and plain NMF were included as the baseline. The spectrogram was calculated using 25 ms window size and 10 ms window shift. With 9 consecutive frames, the concatenated spectrogram was used as feature. The dictionary size was set to 1000. The performance of NMF based models on the same data was cited from (Le Roux et al., 2015a). And the full evaluation is shown in fig 4.2.

In fig 4.2, the neural network based methods outperformed the NMF based model by a large margin on every condition. As discussed in previous chapter, the LSTM network outperforms the DNN, because of the sequence modeling. The dash line showed in fig 4.2 refers the result

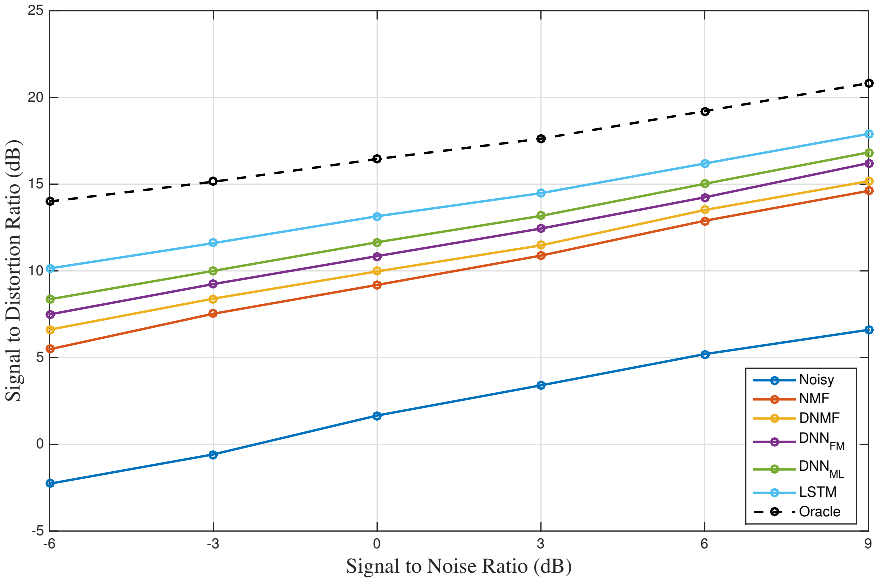


Figure 4.2: The CHiME 2 speech enhancement evaluation

using oracle ratio mask. Note that the LSTM is only 3.34Db on average lower than the oracle, showing its effectiveness.

4.3 Improving the mask learning network

The mask learning network showed significant performance in contrast to previous models. However, there are several limitations.

4.3.1 Scale Mismatch Problem

The main limitation for mask learning network lies in the assumption that the scale of the masked signal is the same as the clean target. Using masks as the training target is supposed to remove only the noise distortion. If the distorted signal is also impacted by the channel distortion, an additional feature mapping function has to be learned to remove the channel distortion in the enhanced speech feature estimated via mask learning. This problem is even more severe in the far-talking scenarios. With a few exceptions for some synthetic data, this assumption is usually not applicable for most real recorded parallel data due to the varied sound source location and the differences in microphones. We refer this as "scale mismatch problem".

When the mixture (M), clean speech (X), and the noise (N) are strictly additive, i.e. $M = X + N$, the clean speech can be perfectly recovered from the ideal mask (i.e. $\frac{X}{X+N}$) learned from the neural network. Nevertheless, the same scale assumption in mask learning is only true on some synthetic data.

For real recorded data, due to the sound source location and microphone differences between far-talk and far-talk channels, their recordings can vary significantly both in scale and spectrum. This is the "scale mismatch problem".

In real parallel recording, mixture from a far-talk microphone (M) and clean speech (X) can be represented as

$$M = f(g(X) + N), \tag{4.4}$$

where $f(\cdot)$ is the non-linear transformation introduced by the channel mismatch; $g(\cdot)$ represents the spectral difference between the two recordings.

The masking process $\Phi\left(Y\right) \odot M$ would fail to recover the clean speech due to the cascaded non-linearities, even with the ideal mask.

4.3.2 Mask Learning with Restoration Layer

We propose to extend the mask learning with two types of restoration layers before or after the mask to address the scale mismatch problem, namely pre- and post-restoration layers.

In the mask learning with pre-restoration layers, the noisy speech first passes through fully connected neural network layers with rectifier linear unit activation (ReLU); then combines with the learnt $[0, 1]$ mask $\phi(Y)$. Since ReLU is unbounded, the pre-restoration layer is designed to learn the scale mismatch between the mixture and the reference. Alternatively, in the mask learning with post-restoration layers, the scale restoration happens after the mask learning through fully-connected neural network layers.

Figure 4.3 depicts the architecture of the extended mask learning with pre- or post-restoration layers. Both pre- and post- restoration layers are designed to fix the scale mismatch problem in mask learning. The key differences lie in that post-restoration layers, as the last step in speech enhancement, have the potential to fix the additional scale-related spectral patterns re-introduced during masking.

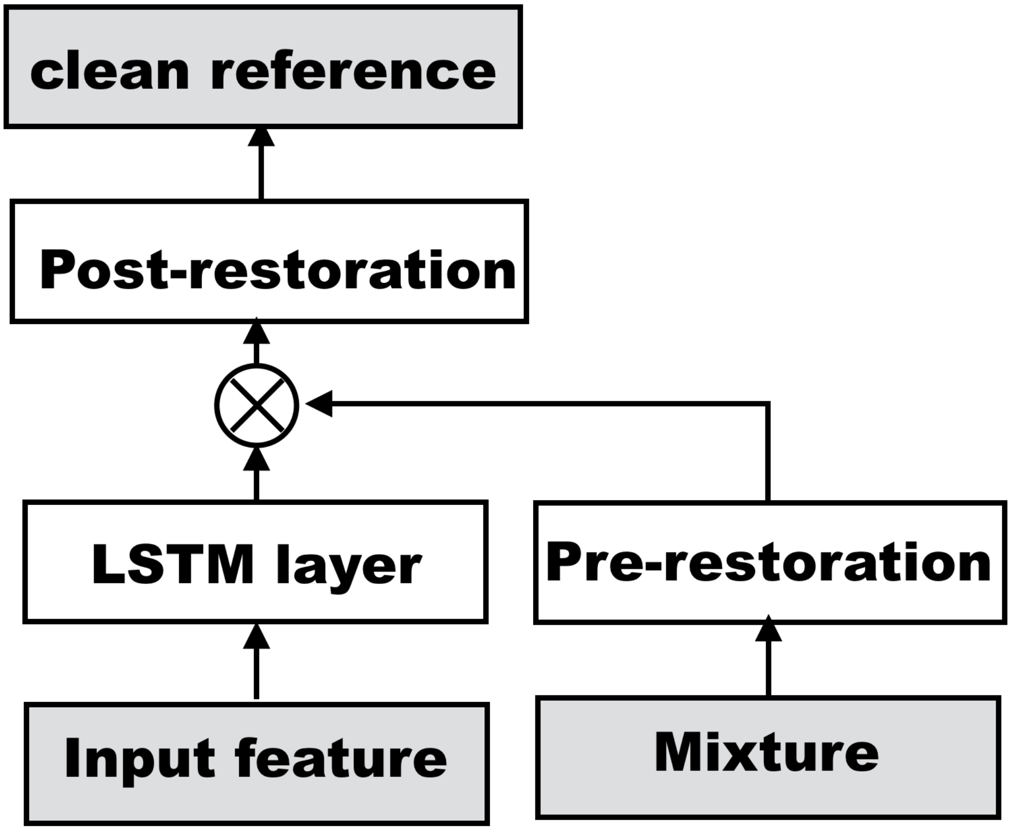


Figure 4.3: Input feature, mixture and clean reference blocks correspond Y, M and X in Equation 5.1, and the remaining white blocks are neural network parameters($\Phi(\cdot)$ in Equation 5.1)

With this extension, the mask learning based speech enhancement can be applied to a wide range of real world scenario. We will present results on the mask learning with restoration layers on CHiME3 task in Section 6.5.

4.4 Residual Learning Feature Mapping

In this section, we introduce applying the residue learning architecture to speech enhancement.

4.4.1 Background of Residual Network

Deep residual learning makes use of short cut connections between neural network layers for fast convergence. It was first proposed in (He et al., 2015) for image recognition. Lately its efficacy was also confirmed in large vocabulary speech recognition (Xiong et al., 2016a).

In residue network, neural network layers are explicitly reformulated to learn residual functions with respect to the layer inputs. The short-cut connection in deep residue learning effectively addresses gradient vanishing/exploding problem in very deep neural network. Thus it is generally believed that very deep neural networks with residue connections is easier to optimize. The residue learning helps to maintain consistently improved accuracy performance in increasingly deeper and more complicated neural network.

Most previous work in applying residue learning focuses on improving network optimization for very deep network.

4.4.2 Residual Learning for Speech Enhancement

Unlike solving gradient vanishing/exploding problems and ease of training of very deep network, the motivation behind our work in applying the residue learning in speech enhancement has straightforward physical meaning in signal reconstruction.

Multiplication in linear scale corresponds to summation when performed in logarithm scale. In a feature mapping network, when the input feature is in logarithmic scale, e.g. log-spectrogram, log-mel-filterbank, etc., adding the additive residual connection between layers is equivalent to perform the masking learning.

Based on this observation, we propose a residual learning based architecture for enhancement. Two types of residual connection are proposed: input residual connection and layer-wise residual connection. In the input residual connection, the shortcut connection between input and the output of each layer is incorporated in the network. In layer-wise residual connection, the shortcut is added between the output of each layer and its previous layer.

Figure 4.4 presents the architecture of residual learning based speech enhancement using *input* residue connection and/or *layer-wise* residue connection. Introducing residue connections in feature mapping network allows us to benefit from both feature mapping and mask learning. This architecture alternates between the feature mapping and mask learning cross different neural network layers. Therefore, it can potentially outperforms speech enhancement with the mask or the feature mapping only.

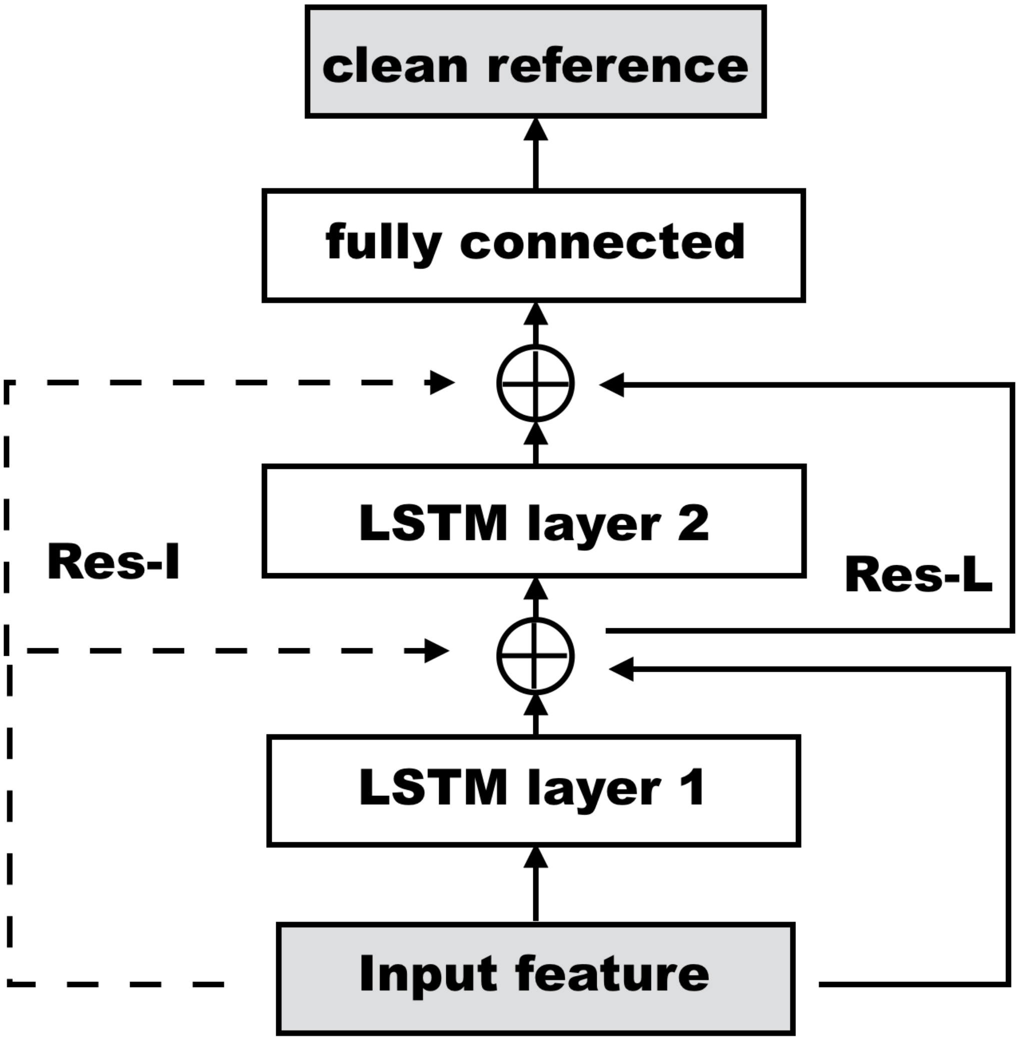


Figure 4.4: Architecture of the residue-learning based speech enhancement model

Architecture of the residue-learning based speech enhancement model

4.5 Experiment

In this section, we present our speech enhancement experimental results on the CHiME3 task.

4.5.1 CHiME 3 and ASR Back-End

CHiME 3 (Barker et al., 2015a) data is recorded using a 6-channel microphone array mounted on a tablet. The training data consists of 1600 real noisy utterances and 7138 simulated utterances. The real data is recorded in different live environments. The simulated data is obtained by mixing clean utterances into different background recordings. For both real and simulated data, four environments are selected: cafi (CAF), street (STR), public transport (BUS), and pedestrian area (PED).

We train a fully connected deep neural network (DNN) on close-talk clean speech. The DNN has 7-hidden layers, each with 2048 hidden units. The input consists of a 2640-dim feature vector formed by 80-dim LFB feature and its accelerating feature components with a context window of 11 frames (80*3*11=2640). The output layer has 3012 senone states. We adopt the RBM pre-training before the fine-tuning of the full network using the cross-entropy criteria. We use WSJ 5K word 3-gram LM for decoding throughout this paper.

The model is then evaluated on the multi-channel enhanced speech audio provided by the CHiME 3 speech challenge (Barker et al., 2015a) and the single-channel far-field noisy speech. The proposed single-channel speech enhancement models are applied as a plug-in module before decoding.

4.5.2 Extended Mask Learning with Restoration Layers

We use a two-layer 300-cell LSTM for masking learning and a feed-forward projection layer with sigmoid activation injected before or after the mask learning layers for scale restoration. The input is 100-dimension log mel-filterbank, calculated with 25ms window and 10ms hop. The setup is similar to a previous state-of-art mask learning system(Barker et al., 2015b).

We compare the plain mask learning with the extended mask learning with pre- or post-restoration layers. As shown in Table 4.1, without scale restoration, the mask learning completely fails on both real and simulated data. This confirms the scale mismatch problem described earlier. After introducing scale restoration layers, both pre-restoration and post-restoration layer improves the mask learning result. In particular, we found that the post-restoration outperforms the pre-restoration. This is likely due to the fact that the post-restoration layers have more information from bottom layers and can be better optimized globally.

In addition, we compare the mask learning based approach with the feature mapping approach. Here the feature mapping is conducted in the acoustic model feature domain. The input is the 240-dim log mel-filterbank formed by 80-dim log mel-filterbank with double delta, as described in Section 4.5.1; the output is the 80-dim log mel-filterbank feature. A similar two-layer 300-cell LSTM is used. It can be seen from Table 4.1 that feature mapping learning in the acoustic model feature domain significantly outperforms the mask-learning based approach, even with injected restoration layers. We believe that this is due to the extra noise introduced during the signal conversion in the mask learning based speech enhancement.

Lastly, we compare three feature mapping networks using DNN, LSTM, or bLSTM. All three models have comparable number of model parameters. The DNN has four hidden layers

Table 4.1: Speech recognition word error rate (WER) comparison mask learning with/without scale restoration.

method	**Real**	**Simu**
Baseline	31.12	15.78
Standard Mask	37.40	15.05
Mask + pre-Restoration	37.40	14.96
Mask + post-Restoration	30.87	13.41
Feature mapping (LSTM)	24.22	11.23

Table 4.2: Speech recognition WER performance comparison for feature mapping using DNN, LSTM, or bLSTM. The results in the brackets are relative WER reductions from the baseline setup.

method	**Real (WER.R)**	**Simu (WER.R)**
Noisy (Baseline)	31.12 (NA)	15.78 (NA)
Feature mapping(DNN)	30.02 (3.53)	14.26 (9.63)
Feature mapping(LSTM)	24.22 (22.17)	11.23 (28.83)
Feature mapping(bLSTM)	24.07 (22.65)	10.81 (31.50)

with 512 nodes; the LSTM is a two-layer 300-cell LSTM network; the bi-direction LSTM has two layers with 150 forward and 150 backward cells. Similarly, the feature mapping learning here is conducted in the acoustic model front-end domain. As shown in Table 4.2, The LSTM feature mapping learning significantly outperforms the DNN with 22.17 % WER reduction for real data and 28.83 % WER reduction for simulated data. This suggests that modeling long-span contextual information via re-current modeling and memory cell helps in learning the noise interference and thus improves the speech enhancement performance. Bi-directional LSTM speech enhancement yields additional accuracy gain due to the integration of bi-directional contextual information.

4.5.3 Residual Learning

Our residue learning based speech enhancement is developed based upon the best performed bLSTM feature mapping model as discussed in Section 4.5.2.

Specifically, we compare three residual learning networks with different types of reside connections: input residual connections only (Res-I); layer-wise residual connections only (Res-L); both input and layer-wise residual connections (Res-B).

Table 4.3 summarizes the experimental results on residual learning based speech enhancement. First, all three proposed residual networks yield small but consistent additional accuracy gain comparing to the state-of-art bLSTM feature mapping model. This suggests the efficacy of the residue connections in speech enhancement model. In particular, we found that the architecture with input residue connection only (Res-I) performs best with 2.91% additional WER reduction against the baseline bLSTM feature mapping model.

We further compare speech recognition performance of the residue learning based speech enhancement with input residue connections in different noisy environments. As shown in Figure 4.5:

Table 4.3: Speech recognition accuracy comparison for residue learning based speech enhancement: input residue connection (ResI), layer-wise residue connection (ResL), or both (ResB). The results in the brackets are relative WER reductions from the baseline setup.

method	Real (WER.R)	Simu (WER.R)
bLSTM (Baseline)	24.07 (NA)	10.81(NA)
bLSTM + Res-I	23.37 (2.91)	10.41 (3.70)
bLSTM + Res-L	23.74 (1.37)	10.88 (-0.65)
bLSTM + Res-B	23.52 (2.29)	10.56 (2.31)

- The residue-learning based speech enhancement improves the speech recognition accuracy performance across almost all noisy conditions. The average relative WER reduction is 24.90 % for real and 34.57 % for simulated data.

- For simulated data, we observe large performance gain for public transport (BUS) and pedestrian area (PED) with 40∼50 % relative WER reduction. Nevertheless, for cafi (CAF), the enhancement did not seem to yield noticeable performance improvement. We suspect this is likely due to certain data simulation specifics for cafe data.

- For real data, all noisy conditions receive consistent accuracy gain with relative WER reduction ranging from 18.07 % for cafi (CAF) to 28.58 % for street junction (STR). The gain is slightly smaller but yet significant comparing to in the simulated data. This suggests that the proposed approach is effective for a wide range of real life noisy environments with stationary, semi-stationary, or highly non-stationary noise.

It is worth noting that our architecture is still considered to be a shallow model. As more training data becomes available, we can increase the depth of the enhancement network layers, which is expected to benefit even more from residue connections.

4.5.4 Single-Channel Far-Field Speech Enhancement

In real world far-field applications, microphone array and multi-channel speech enhancement are not always available. We would like to find out how the proposed single-channel speech enhancement would perform on single-channel far-field noisy speech.

To this end, we further apply the proposed extended bLSTM mask learning with post-restoration layers and input residue connection to single far-talk noisy channel speech enhancement. As shown in table 4.4, the proposed best performed speech enhancement applied to single far-field channel yields 11.73 % WER reduction, comparing to 24.90 % WER reduction when applied to the multi-channel enhanced speech evaluated on the real testing part of CHiME 3.

The results suggest the proposed approach is beneficial even with single-channel far-field noisy speech. This makes the proposed single-channel speech enhancement approach practical in a wide rage of real world far-field ASR scenario with/without microphone array and enhanced far-field speech.

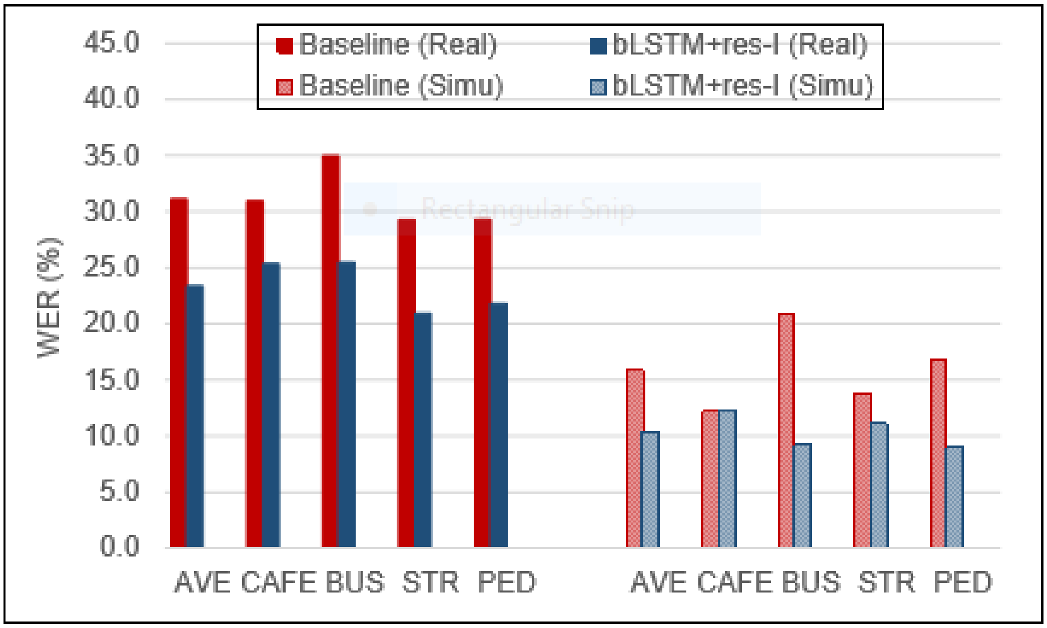


Figure 4.5: Speech recognition accuracy performance comparison of the residue-learning based speech enhancement in different types of noisy environments for real and simulated data

Table 4.4: Speech recognition accuracy performance comparison for single-channel far-field speech enhancement using bLSTM with input residue connection. The results in the brackets are relative WER reductions from the baseline setup.

method	**Real (WER.R)**	**Simu (WER.R)**
Noisy (Baseline)	31.12 (NA)	NA (NA)
Res-L(Enhanced)	23.37 (24.90)	24.90 (34.57)
Res-I(CH5)	27.47 (18.07)	11.73 (28.58)

4.6 Conclusion

In conclusion, we relaxed the matching scale constraint in mask learning based speech enhancement model by integrating two types of restoration layer. We proposed a novel residual learning for improved speech enhancement. We evaluated the proposed speech enhancement models on CHiME 3 task. Without retraining the acoustic model, the best performed bi-direction LSTM input residue connection yields 25.13% relative WER reduction on real data and 34.03% relative WER reduction on simulated data. Further analytic study suggests that the proposed approach is effective for a wide range of noisy environments.

Chapter 5

Deep clustering

In last chapter, we introduced the neural network based speech enhancement and several of its extensions, which showed very high separation quality. In this chapter, we further discuss the neural network based audio source separation system. We firstly show the limitations of the auto-encoder network, and introduce another family of neural network based system, which is designed for a more general scenario of auditory source separation.

5.1 Introduction

In real world perception, we often must selectively attend to objects whose features are intermingled in the incoming sensory signal. Nowhere is this more apparent than in hearing, where signals are densely mixed and can be challenging to separate. Nevertheless human listeners easily perceive separate sources in an acoustic mixture, and this ability has inspired a variety of computational approaches to the so-called *auditory scene analysis* or *cocktail party* problem (Bregman, 1990). We address the problem of "cocktail-party" speech separation in a deep learning framework we call *deep clustering*.

Single-channel speech separation is the task of estimating the individual speech signals that are mixed together and overlapping in a monaural signal. It is a challenging problem and many further assumptions have been used to make headway. Previous attempts have generally assumed that the number of sources is fixed. Some "speech separation" methods are about separating speech from challenging background noise (Weninger et al., 2014a; Wang and Wang, 2013; Wang et al., 2013) instead of separating multiple speakers. Many previous approaches have relied on speaker-dependent models (Hershey et al., 2010; Huang et al., 2015; Virtanen, 2006), although some also addressed the case of same-speaker mixtures (Hershey et al., 2010; Virtanen, 2006), or more than two speakers (Rennie et al., 2010). Furthermore, many of these addressed only tasks with limited vocabulary and grammar, as in (Cooke et al., 2010). Some were able to achieve impressive performance in these limited domains.

In this chapter, we consider a more open and difficult task of speaker-independent separation of two or more speakers, with no special constraint on vocabulary and grammar. Speaker-independent separation was addressed in (Weiss, 2009) by building speaker adaptation upon the model-based approach of (Hershey et al., 2010). In another direction, (Rennie et al., 2010) extended (Hershey et al., 2010) to handle more than two speakers. While both extensions are interesting, in general speed and learning are problematic.

Meanwhile, the state-of the art in enhancement and separation is currently done using deep networks (Weninger et al., 2015; Wang et al., 2014b; Xu et al., 2014), as introduced in last chapter. These *class-based* methods train on parallel sets of mixtures and their constituent target sources, so that the network predicts the source belonging to the target class, or classifies the type of source that dominates each time-frequency bin.

Although *class-based* methods can succeed in the speaker-dependent case, where each target is a speaker known at training time, they fail to learn in the speaker-independent case, as shown in our experiments. Two main problems prevents the class based system from working, which we refer as "permutation problem" and "output dimension mismatch problem", and will introduce in detail in later sections.

An important family of methods based on clustering may be more flexible in this regard. These include *computational auditory scene analysis* (CASA) approaches that use perceptual grouping cues (Cooke, 1991; Ellis, 1996a), and spectral clustering approaches (Bach and Jordan, 2006) that use affinity kernels. CASA approaches seek to explain perceptual grouping of regions in terms of their similarity(Wertheimer, 1938). Such methods are heuristic, and although carefully tuned systems perform surprisingly well on speech (Hu and Wang, 2013b), they still fall behind the class-based deep learning methods, as we show below. With no training, over-fitting is not a problem, but it is difficult to imagine accommodating different types of sources.

In the area of spectral clustering, however, which is based on eigen-decomposition of the normalized affinity matrix (Shi and Malik, 2000), significant progress has been made in learning the relative weights of different affinity features (Bach and Jordan, 2006).

Unfortunately, the spectral clustering paradigm suffers from high computational cost, and shallow learning. These factors appear to be co-dependent: simple kernels tend to produce sparse affinity matrices, which require costly spectral methods to reduce to clusters. Conversely this complexity makes optimization of the front end processing a formidable challenge (Bach and Jordan, 2006).

In the speech separation problem, powerful front-end processing is indeed required, because of a pesky chicken and egg problem. To infer the segmentation requires features of neighboring regions of the same source, but the context regions for one source may contain intermingled parts of other sources. To extract uncorrupted features, then, would seem to require knowing the segmentation in advance.

Nevertheless, we know from from prior work that deep neural networks can learn their way out of this quandary, when the targets are distinct classes. So we propose to use more powerful front end processing to produce a lower-rank affinity matrix, which then may be amenable to clustering by simpler methods such as K-means. The simpler clustering methods in turn should provide for easier training, allowing a more complex front-end to be learned.

Learned feature transformations known as *embeddings* have recently been gaining significant interest in many fields. Unsupervised embeddings obtained by auto-associative deep networks, used with relatively simple clustering algorithms, have recently been shown to outperform spectral clustering methods (Tian et al., 2014; Huang et al., 2014a) in some cases.

In our framework a deep network assigns embedding vectors to each time-frequency region of the spectrogram, according to an objective function that minimizes the distances between embeddings of time-frequency bins dominated by the same source, while maximizing the distances between embeddings for those dominated by different sources. Thus the clusters in the embedding can represent the inferred spectral masking patterns of the sources, in a permutation-free way. Moreover, despite the fixed dimensionality of the network output, the embeddings can implicitly represent different numbers of sources.

This objective relates to spectral clustering in that the embeddings can be used to approximate an ideal affinity matrix given by the known segmentation. It is also closely related to the K-means objective function so that at test time we can infer the assignments given the embeddings using K-means algorithm.

The experiments show that the proposed method can separate speech using a speaker-independent model on an open set of speakers. We derive partition labels by mixing signals together and observing their spectral dominance patterns. After training on a database of mixtures of speakers trained in this way, we show that the model can generalize to three-speaker mixtures despite training only on two-speaker mixtures. Although results are preliminary, this suggests that we may hope to achieve class-independent segmentation of arbitrary sounds, with additional application to image segmentation and other domains.

5.2 Permutation problem and output dimension problem

Most neural network methods are trained to map the input signal to a unique target output which can be a label, a sequence, or regression reference. Permutation problem in speech separation arises due to the fact that the order of targets in the mixture is irrelevant. One example for the two problem is shown in fig 5.1. The upper figure in fig 5.1 shows a typical auto-encoder network for speech enhancement, where the input is the mixture between speech and noise, and the network targets at the clean speech. Since the speech and the noisy are from two audio class, which have very different statistical properties, the network shows the ability to distinguish the two. When separating two speakers, following the same methodology, an auto-encoder network should output the clean spectorgram of both speakers. In practice, it usually requires the last layer have the size that is twice of the spectrogram bins, and it first half is assigned to first speaker and the second half for the second speaker, as suggested in the middle figure in fig 5.1. However, when separating mixtures of speakers (A,B), (A,C), and (B,C), it will generate confusions by assigning speaker A to the first target position in (A,B) and (A,C) cause in mixture (B,C), both of the speaker need to be in the second position for consistency.

The second problem in using neural network framework for speech separation is the output dimension mismatch problem. Since the number of sources in the mixture can vary, a neural

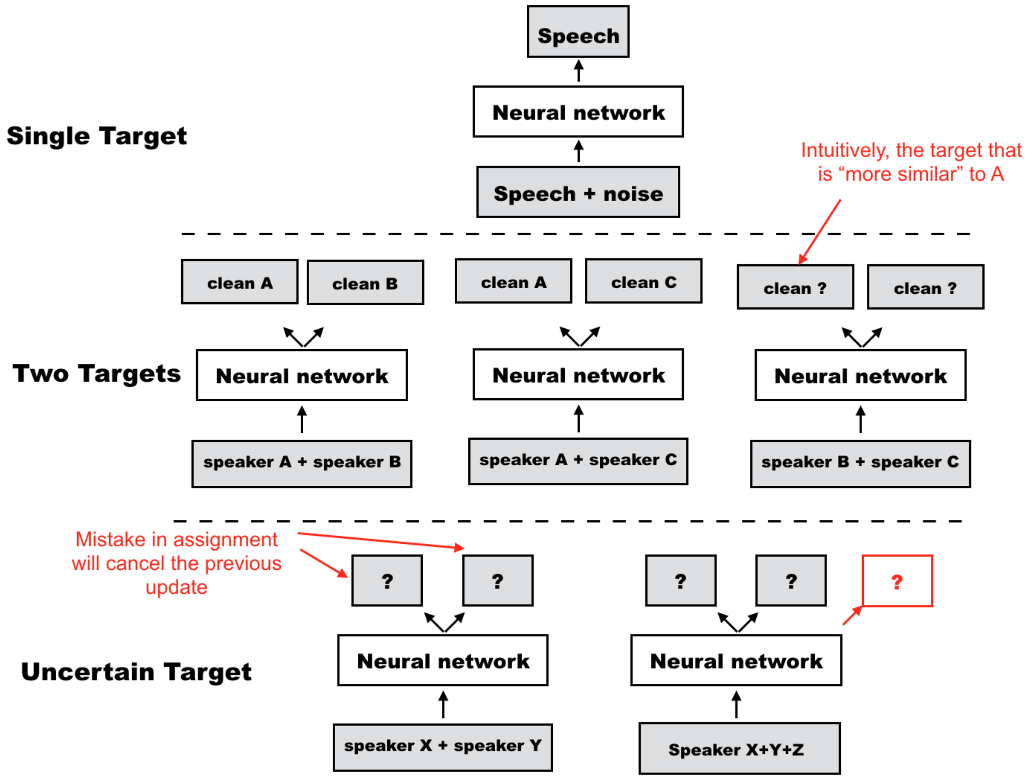


Figure 5.1: The permutation problem and the output dimension mismatch problem

network with fixed number of output nodes does not have the flexibility to separate arbitrary number of sources. As shown in the bottom figure of fig 5.1, an auto-encoder network trained for two speaker, despite the permutation problem can only handle the two speaker separation problem, since the output dimension for the network is fixed. When there are three speaker in the mixture, the two speaker network would not work at all because there is no dimension for the additional speaker. On the other hand, it would also be problematic if the network is trained with three speakers but facing two speaker mixtures. In that case, assuming the network is well trained, two out of three heads in the output layer should generate the separation of the two speaker and the remaining head should be completely removed. But it would be very difficult to distinguish which two heads are the valid separation during the testing time when the clean sources are not available, because it is very likely that all the three heads has speech liked spectrogram. Therefore, in short, for the auto-encoder based model, the number of mixing sources has be fixed in both training and testing data, which clearly is over-simplify the separation scenario.

5.3 Learning deep embeddings for clustering

We define x as a raw input signal and as $X_i = g_i(x), i \in \{1, \ldots, N\}$, a feature vector indexed by an element i. In the case of audio signals, i is typically a time-frequency index (t, f), where t indexes frame of the signal, f indexes frequency, and $X_i = X_{t,f}$ the value of the complex spectrogram at the corresponding time-frequency bin. We assume that there exists a reasonable partition of the elements i into regions, which we would like to find, for example to further process the features X_i separately for each region. In the case of audio source separation, these regions can be defined as the sets of time-frequency bins in which each source dominates, and estimating such a partition would enable us to build time-frequency masks to be applied to X_i, leading to time-frequency representations that can be inverted to obtain isolated sources.

To estimate the partition, we seek a D-dimensional embedding $V = f_\theta(x) \in \mathbb{R}^{N \times D}$, parameterized by θ, such that performing some simple clustering in the embedding space will likely lead to a partition of $\{1, \ldots, N\}$ that is close to the target. In this work, $V = f_\theta(X)$ is based on a deep neural network that is a global function of the entire input signal X. Thus our transformation can take into account global properties of the input, and the embedding can be considered a permutation- and cardinality-independent encoding of the network's estimate of the signal partition. Here we consider a unit-norm embedding, so that $|v_i|^2 = 1$ where $v_i = \{v_{i,d}\}$ and $v_{i,d}$ is the value of the d-th dimension of the embedding for element i. We consider the embeddings V to implicitly represent an $N \times N$ estimated affinity matrix VV^T.

The target partition is represented by the indicator $Y = \{y_{i,c}\}$, mapping each element i to each of C clusters, so that $y_{i,c} = 1$ if element i is in cluster c. In this case YY^T, is considered as a binary affinity matrix that represents the cluster assignments in a permutation-independent way: $(YY^T)_{i,j} = 1$ if elements i and j belong to the same cluster, and $(YY^T)_{i,j} = 0$ otherwise, and $(YP)(YP)^T = YY^T$ for any permutation matrix P.

We can learn affinity matrix VV^T, as a function of the inputs, X to match the affinities, YY^T, by minimizing, with respect to $V = f_\theta(X)$, the training cost function,

$$\mathcal{C}_Y(V) = \|VV^T - YY^T\|_\mathrm{F}^2 = \sum_{i,j} \left(\langle v_i, v_j \rangle - \langle y_i, y_j \rangle\right)^2 \tag{5.1}$$

$$= \sum_{i,j:y_i=y_j} \left(|v_i - v_j|^2 - 1\right) + \sum_{i,j} \langle v_i, v_j \rangle^2, \tag{5.2}$$

summed over training examples, where $\|A\|_\mathrm{F}^2$ is the squared Frobenius norm. For the true cluster labels $\mathring{Y}$, $\mathcal{C}_{\mathring{Y}}(V)$ minimizes the distance between the estimated affinity matrix VV^T and the ideal affinity matrix $\mathring{Y}\mathring{Y}^T$. The form (5.2) pulls the embeddings v_i and v_j closer together for elements within the same partition, whereas the second term pushes all elements apart, preventing collapse to a trivial solution.

Note that although this function ostensibly sums over all pairs of data points i, j, the low-rank nature of the objective leads to an efficient implementation:

$$\mathcal{C}_Y(V) = \|V^T V\|_\mathrm{F}^2 - 2\|V^T Y\|_\mathrm{F}^2 + \|Y^T Y\|_\mathrm{F}^2, \tag{5.3}$$

which avoids explicitly constructing the $N \times N$ affinity matrix. In practice, N is orders of magnitude greater than D, leading to a significant speedup. Derivatives with respect to V are also efficiently obtained due to the low-rank structure:

$$\frac{\partial \mathcal{C}_Y(V)}{\partial V^T} = 4V(V^TV) - 4Y(Y^TV) \tag{5.4}$$

This low-rank formulation also relates to spectral clustering in that the latter typically requires the Nyström low-rank approximation to the affinity matrix (Fowlkes et al., 2004) for efficiency. So, rather than making a low-rank approximation to a complicated full-rank model, deep clustering directly optimizes a low-rank model so that simple clustering can be used.

For inference, we compute the embeddings $V = f_\theta(X)$ on the test signal X, and cluster the rows $v_i \in \mathbb{R}^D$, by minimizing the K-means inference cost: $\bar{Y} = \arg\min_Y \mathcal{K}_V(Y) = \|V - YM\|_\mathrm{F}^2$, where $M = (Y^TY)^{-1}Y^TV$ are the $C \times D$ means of each cluster. The resulting cluster assignments $\bar{Y}$ are used as binary masks to separate the sources. The ideal mask used as our cluster reference $\mathring{Y}$, yields the optimal signal to noise ratio (SNR) among all binary masks. Although continuous masks can yield further improvement, here we first focus on solving the permutation problem, leaving refinement for future work.

The clustering error between the estimates $\bar{Y}$, and the labels $\mathring{Y}$, can be quantified by a variety measures, such as the χ^2 error,

$$d_{\chi^2}(\bar{Y}, \mathring{Y}) = \|\bar{Y}(\bar{Y}^T\bar{Y})^{-1}\bar{Y}^T - \mathring{Y}(\mathring{Y}^T\mathring{Y})^{-1}\mathring{Y}^T\|_\mathrm{F}^2. \tag{5.5}$$

(Meilă, 2012; Hubert and Arabie, 1985; Bach and Jordan, 2006). The minima of the training objective, $\mathcal{C}_{\mathring{Y}}(V)$, the K-means objective, $\mathcal{K}_V(\bar{Y})$, and the clustering error, $d_{\chi^2}(\bar{Y}, \mathring{Y})$, all coincide when $VV^T = \mathring{Y}\mathring{Y}^T$, leading to $\bar{Y} = \mathring{Y}$. More general bounds between the various objectives are derived in (Bach and Jordan, 2006; Meilă, 2012; Meila, 2014). See (Hershey et al., 2015) for further discussion in the present context. Note that one might consider directly optimizing the K-means objective as a function of V. Although $\mathcal{K}_V(\bar{Y})$ solely minimizes within-class variance, leading to a degenerate solution, this may be prevented by using the ratio of within-class to total variance, as in linear discriminant analysis. We leave this and other alternative objectives for future work.

5.4 Speech separation experiments

5.4.1 Experimental setup

We evaluate deep clustering (DC) on a speaker-independent speech separation task. Mixtures involving speech from same gender speakers can be extremely challenging since the pitch and vocal tract of the voices are in the same range. We here consider mixtures of two and three speakers, which include the same gender condition. Three types of experiments were performed, separating two unknown speakers, three unknown speakers, or three known speakers. In the latter case, the systems are trained on mixtures of the three known speakers at training time, whereas in the other cases training speakers and test speakers are different.

We created a new corpus of speech mixtures using utterances from the Wall Street Journal (WSJ0) corpus because existing speech separation challenge datasets are too limited for the evaluation of our model. For example, the speech separation challenge (Cooke et al., 2010) only contains two-speaker mixtures, with a limited vocabulary and insufficient training data.

A 30 h training set and a 10 h validation set consisting of two-speaker mixtures were generated by randomly selecting utterances by different speakers from the WSJ0 training set `si_tr_s`, and mixing them at various signal-to-noise ratios (SNR) between 0 dB and 10 dB. The validation set was used to optimize some tuning parameters and to evaluate the source separation performance in closed conditions (**CC**). Five hours of evaluation data were generated similarly using utterances from 16 speakers from the WSJ0 development set `si_dt_05` and evaluation set `si_et_05`. The speakers are different from those in our training and validation sets, and we thus use this set for open condition (**OC**) evaluation. Note that previous speech separation methods (e.g., (Smaragdis, 2007; Le Roux et al., 2015b)) cannot handle the open speaker problem, and require knowledge of the speakers in the evaluation.

We also created three sets of three-speaker mixtures. The first two sets are similar respectively to the two-speaker validation and evaluation sets, with 100 three-speaker mixtures obtained from a pool of many speakers in closed condition (**MS-CC**) and open condition (**MS-OC**). The third one consists in 5000 mixtures for training, 500 mixtures for validation, and 500 mixtures for test, using speech from a closed set of three known speakers in `si_et_05` (**3S-CC**) .

All data were downsampled to 8 kHz before processing to reduce computational and memory costs. The input features X were the log spectral magnitudes of the speech mixture, computed using a short-time Fourier transform (STFT) with 32 ms window length, 8 ms window shift, and the square root of the hann window. To ensure local coherency, a mixture is separately processed in half-overlapping segments of 100 frames, roughly the length of one word in speech, to output embeddings V based on the proposed model.

5.4.2 Training procedure

The binary masks were used to build the target Y to train our network. In each time-frequency bin, the mask values are set to 1 for the source with the maximum magnitude and 0 for the others. For the two-source case, this corresponds to the ideal binary mask (IBM)(Wang, 2005). To avoid training the network to assign embeddings to silence regions, a binary weight for each time-frequency bin was used during the training process, only retaining those bins such that magnitude of the mixture at that bin is greater than some ratio (arbitrarily set to -40 dB) of the maximum magnitude. The network structure used in our experiments has two bi-directional long short-term memory (BLSTM) layers, followed by one feedforward layer. Each BLSTM layer has 600 hidden cells and the feedforward layer corresponds with the embedding dimension D. Stochastic gradient descent with momentum 0.9 and fixed learning rate 10^{-5} was used for training. In each updating step, to avoid local optima, Gaussian noise with zero mean and 0.6 variance was added to the weight. We prepared several networks using different embedding dimensions from 5 to 60. In addition, two different activation functions (logistic and tanh) were explored to form the embedding V with different ranges for $v_{n,d}$. For each embedding dimension, the weights for the corresponding network were initialized randomly according to a normal distribution with zero mean and 0.1 variance with the tanh activation. In the experiments with the logistic activation, the network was initialized with the tanh network.

A state of the art class-based BLSTM speech enhancement network (Weninger et al., 2015) was included as baseline for both two-speaker and three-speaker experiments. Because of the inherent ambiguity in speaker-independent separation tasks, as to which output should be used for each speaker, we proposed two training schemes to help with learning using the class-based LSTM. In one case we used the stronger source as the training target for each 100 frame segment (**BLSTM stronger**). We also propose a permutation-free scheme (**BLSTM permute**), where we find the closest clean source to each output of the network, and use that source to measure the training error and compute the gradients.

To facilitate comparison, both deep clustering and the classifier system used the same architectures, except for the final output layers and objective function. Since deep clustering has a large embedding layer, we also formulated a class-based BLSTM with the same number of parameters by using an additional feedforward layer of the same size as the embedding layer used in deep clustering (**BLSTM permute***). In the three-known-speakers experiment, the speaker identities are known, so we used the stacked ideal soft mask for each speaker as target (**BLSTM stack**). For both experiments, squared Euclidean distance was used as error measurement for class-based network. All the BLSTM layers in the class-based model were initialized with the parameters of the trained deep clustering network (i.e. $D = 40$ tanh).

5.4.3 Speech separation procedure

At test time, speech separation was performed by re-filtering time-domain signals based on time-frequency masks for each speaker. The masks were obtained by clustering the row vectors of embedding V, where V was output from the proposed model for each segment (100 frames), similarly to the training stage. The number of clusters is set to the number of speakers in the mixture. We evaluated two types of clustering methods: global K-means on the embeddings of the whole utterance and local K-means, where clustering is done separately on each 100-frame segment. In both cases, we choose the best correspondence in the least-squares sense between the recovered sources and target signals.

Given that DC can represent an arbitrary number of clusters, an interesting question is whether it can generalize to the case of three-speaker mixtures without changing the model parameters. Speech separation experiments on three-speaker mixtures were thus conducted using the network trained with two-speaker mixtures, by simply changing the number of clusters from 2 to 3 in the clustering step.

Besides the class-based BLSTM, we used supervised sparse non-negative matrix factorization (SNMF) as another baseline (Smaragdis, 2007; Le Roux et al., 2015b). While SNMF is amenable to separating male-female mixtures when using a concatenation of bases trained on speakers of different genders, in preliminary experiments it failed for same-gender mixtures. We thus give SNMF an unfair advantage by using speaker dependent models with oracle information about the speakers present at test time. Wiener-filter like masks are built using the estimated models and applied to the mixture, and the separated signals are obtained by inverse STFT. We used 256 bases per speaker, and magnitude spectra with 8 consecutive

Table 5.1: SDR improvements (dB) for different separation methods

method	**CC**	**OC**
oracle NMF	5.1	-
CASA	2.9	3.1
DC local K-means	6.5	6.5
DC global K-means	5.9	5.8
BLSTM stronger	1.3	1.2
BLSTM permute	1.3	1.3
BLSTM permute*	1.4	1.2

Table 5.2: SDR improvements (dB) for different embedding dimensions D and activation functions

	CC		**OC**	
model	DC local	DC global	DC local	DC global
$D = 5$	-0.8	-1.0	-0.7	-1.1
$D = 10$	5.2	4.5	5.3	4.6
$D = 20$	6.3	5.6	6.4	5.7
$D = 40$	6.5	5.9	6.5	5.8
$D = 60$	6.0	5.2	6.1	5.3
$D = 40$ logistic	6.6	5.9	6.6	6.0

frames of left context as input features. We also included an unsupervised CASA-based system (Hu and Wang, 2013b) as another baseline for the two-speaker experiment.

For all experiments, performance was evaluated in terms of averaged signal-to-distortion ratio (SDR) using the `bss_eval` toolbox (Vincent et al., 2006). The initial SDR averaged over the mixtures was 0.2 dB for two-speaker mixtures and −3.0 dB for three-speaker mixtures.

Table 5.3: SDR improvement (dB) for mixtures of three speakers.

method	MS-CC	MS-OC	3S-OC
oracle NMF	4.4	-	4.5
DC local	3.5	2.8	7.0
DC global	2.7	2.2	6.9
BLSTM stack	-	-	6.8

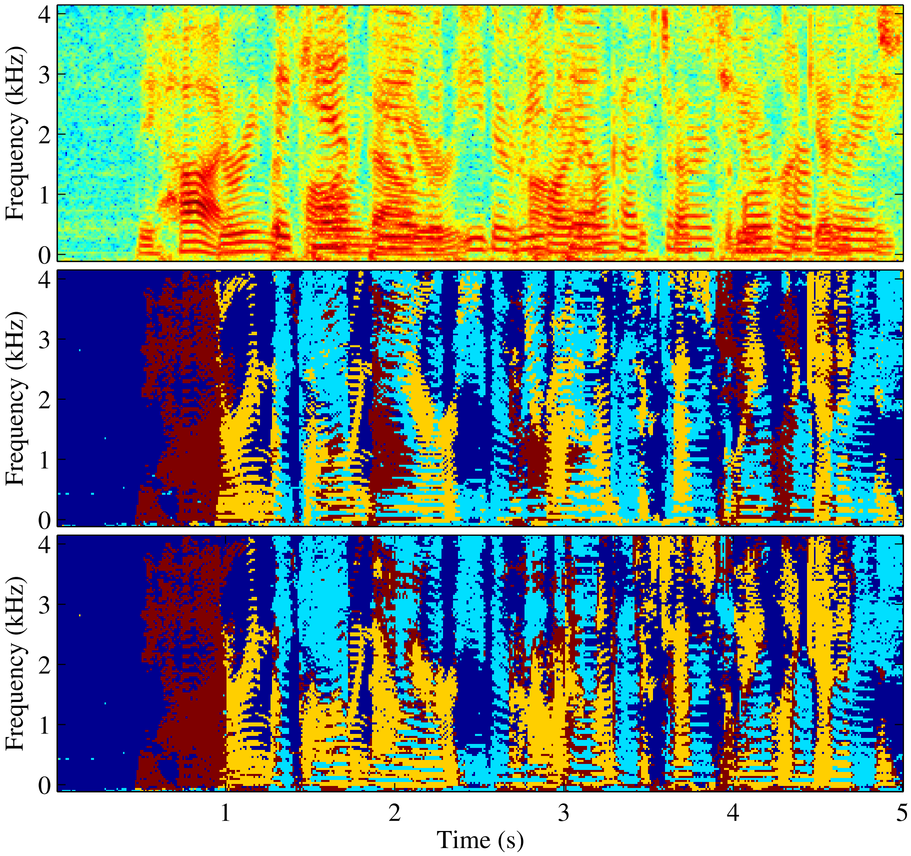


Figure 5.2: Top: mixture log spectrogram. Middle: IBM. Dark blue shows silence. Bottom: output mask from proposed system trained on two-speaker mixtures.

5.5 Results and discussion

As shown in Table 5.1, both local and global clustering methods significantly outperform all baselines. Note that due to stability issues with the CASA code provided by authors of (Hu and Wang, 2013b), evaluation could only be run on a subset of about 40 % of the data, but there was no significant difference for this subset in starting SNR or in the improvements of other algorithms. It should note that in later works(Yu et al., 2016), the author showed that after some fine tuning, the permutation free training proposed here could also generate high quality separation performance, despite its incapability to handle the output dimension mismatch problem.

The global K-means clustering of the whole utterance performs only slightly worse than local clustering. As the system was only trained with individual segments, this suggests that the network learns globally important features. The performance of DC is similar in open and closed conditions, indicating that it can generalize well to unknown speakers.

In Table 5.2, the $D = 5$ system completely fails, either because optimization of the current network architecture fails, or the embedding fundamentally requires more dimensions. The performance of $D = 20$, $D = 40$, $D = 60$ is similar, showing that the system can operate in a wide range of parameter values. We arbitrarily used tanh networks in most of the experiments because of their larger embedding space than logistic networks. However, in Table 5.2, we verify that the logistic network performs about the same.

All class-based BLSTMs performed poorly in non-speaker-dependent settings, even when carefully trained (Table 5.1, right). Only for the speaker-dependent 3S-CC set, the class-based model performed similarly to DC (Table 5.3). We can expect other speaker-dependent methods (Hu and Wang, 2013a; Huang et al., 2015) to follow the same trend. This confirms that class-based networks lack the ability to resolve the permutation problem introduced by same-class mixtures. In contrast, in DC the permutation is solved by the clustering step, which allows modeling power to focus on the distinction between sources.

We see in Table 5.3 (left) that DC remarkably can also separate three-speaker mixtures, even when only trained on two-speaker mixtures. Figure 5.2 shows an example of separation for three-speaker mixture in the open validation set. Of course, including mixtures involving more than two speakers at training time should improve performance further, but the method does surprisingly well even without retraining. Performance is now worse than oracle NMF, but is again much better once we allow DC to focus on a limited set of speakers, as shown in Table 5.3 (right): there, DC is trained on mixtures of the same three speakers used for test.

Alternative network architectures with different time and frequency dependencies, such as deep convolutional neural networks (Farabet et al., 2013) or hierarchical recursive embedding networks (Sharma et al., 2014), could be helpful in terms of learning and regularization. Finally, scaling up training on databases of more disparate audio types, as well as applications to other domains such as image segmentation, are prime candidates for future work.

5.6 Improving deep clustering

In this section we present improvements and extensions that enable a leap forward in separation quality, reaching levels of improvement that were previously out of reach (audio examples and scripts to generate the data used here are available at (Isik et al., 2016a)). In addition to improvements to the training procedure, we investigate the three speaker case, showing generalization between two- and three-speaker networks.

The original deep clustering system was intended to only recover a binary masks for each source, leaving recovery of the missing features to subsequent stages. In this section, we incorporate enhancement layers to refine the signal estimate. Using soft clustering, we can then train the entire system *end-to-end*, training jointly through the deep clustering embeddings, the clustering and enhancement stages. This allows us to directly use a signal approximation objective instead of the original mask-based deep clustering objective.

Below we present the deep clustering model and further investigate its capabilities. We then present extensions to allow end-to-end training for signal fidelity. The results are evaluated using an automatic speech recognition model trained on clean speech. The end-to-end signal approximation produces unprecedented performance, reducing the word error rate (WER) from close to 89.1% WER down to 30.8% by using the end-to-end training. This represents a major advancement towards solving the cocktail party problem.

5.7 Improvements to the Training Recipe

We investigated several approaches to improve performance over the baseline deep clustering method, including regularization such as drop-out, model size and shape, and training schedule. We used the same feature extraction procedure as in (Hershey et al., 2016a), with log-magnitude STFT features as input, and we performed global mean-variance normalization as a pre-processing step. For all experiments we used we used rmsprop optimization (Tieleman and Hinton, 2012a) with a fixed learning rate schedule, and early stopping based on cross-validation.

Regularizing recurrent network units: Recurrent neural network (RNN) units, in particular LSTM structures, have been widely adopted in many tasks such as object detection, natural language processing, machine translation, and speech recognition. Here we experiment with regularizing them using dropout.

LSTM nodes consist of a recurrent memory cell surrounded by gates controlling its input, output, and recurrent connections. The direct recurrent connections are element-wise and linear with weight 1, so that with the right setting of the gates, the memory is perpetuated, and otherwise more general recurrent processing is obtained.

Dropout is a training regularization in which nodes are randomly set to zero. In recurrent network there is a concern that dropout could interfere with LSTM's memorization ability; for example, (Zaremba et al., 2014) used it only on feed-forward connections, but not on the recurrent ones. *Recurrent dropout* samples the set of dropout nodes once for each sequence, and applies dropout to the same nodes at every time step for that sequence. Applying recurrent dropout to the LSTM memory cells recently yielded performance improvements on phoneme and speech recognition tasks with BLSTM acoustic models (Moon et al., 2015).

In this work, we sampled the dropout masks once at each time step for the forward connections, and only once for each sequence for the recurrent connections. We used the same recurrent dropout mask for each gate.

Architecture: We investigated using deeper and wider architectures. The neural network model used in (Hershey et al., 2016a) was a two layer bidirectional long short-term memory (BLSTM) network followed by a feed-forward layer to produce embeddings. We show that expanding the network size improves performance for our task.

Temporal context: During training, the utterances are divided into fixed length non-overlapping segments, and gradients are computed using shuffled mini-batches of these segments, as in (Hershey et al., 2016a). Shorter segments increase the diversity within each batch, and may make an easier starting point for training, since the speech does not change as much over the segment. However, at test time, the network and clustering are given the entire utterance, so that the permutation problem can be solved globally. So we may also expect that training on longer segments would improve performance in the end.

In experiments below, we investigate training segment lengths of 100 versus 400, and show that although the longer segments work better, pretraining with shorter segments followed by training with longer segments leads to better performance on this task. This is an example of *curriculum learning* (Bengio et al., 2009), in which starting with an easier task improves learning and generalization.

Multi-speaker training: Previous experiments (Hershey et al., 2016a) showed preliminary results on generalization from two speaker training to a three-speaker separation task. Here we further investigate generalization from three-speaker training to two-speaker separation, as well as multi-style training on both two and three-speaker mixtures, and show that the multi-style training can achieve the best performance on both tasks.

5.8 Optimizing Signal Reconstruction

Deep clustering solves the difficult problem of segmenting the spectrogram into regions dominated by each source. It does not however solve the problem of recovering the sources in regions strongly dominated by other sources. Given the segmentation, this is arguably an easier problem. We propose to use a second-stage enhancement network to obtain better source estimates, in particular for the missing regions. For each source c, the enhancement network first processes the concatenation of the amplitude spectrogram x of the mixture and that $\hat{s}_c$ of the deep clustering estimate through a BLSTM layer and a feed-forward linear layer, to produce an output z_c. Sequence-level mean and variance normalization is applied to the input, and the network parameters are shared for all sources. A soft-max is then used to combine the outputs z_c across sources, forming a mask $m_{c,i} = \mathrm{e}^{z_{c,i}} / \sum_{c'} \mathrm{e}^{z_{c',i}}$ at each TF bin i. This mask is applied to the mixture, yielding the final estimate $\tilde{s}_{c,i} = m_{c,i} x_i$. During training, we optimize the enhancement cost function $\mathcal{C}_E = \min_{\pi \in \mathcal{P}} \sum_{c,i} (s_{c,i} - \tilde{s}_{\pi(c),i})^2$, where $\mathcal{P}$ is the set of permutations on $\{1, \dots, C\}$. Since the enhancement network is trained to directly improve the signal reconstruction, it may improve upon deep clustering, especially in regions where the signal is dominated by other sources.

5.9 End-to-End Training

In order to consider end-to-end training in the sense of jointly training the deep clustering with the enhancement stage, we need to compute gradients of the clustering step. In (Hershey et al., 2016a), hard K-means clustering was used to cluster the embeddings. The resulting binary masks cannot be directly optimized to improve signal fidelity, because the optimal masks are generally continuous, and because the hard clustering is not differentiable. Here we propose a soft K-means algorithm that enables us to directly optimize the estimated speech for signal fidelity.

In (Hershey et al., 2016a), clustering was performed with equal weights on the TF embeddings, although weights were used in the training objective in order to train only on TF elements with significant energy. Here we introduce similar weights w_i for each embedding v_i to focus the clustering on TF elements with significant energy. The goal is mainly to avoid clustering silence regions, which may have noisy embeddings, and for which mask estimation errors are inconsequential.

The soft weighted K-means algorithm can be interpreted as a weighted expectation maximization (EM) algorithm for a Gaussian mixture model with tied circular covariances. It alternates between computing the assignment of every embedding to each centroid, and updating the centroids:

$$\gamma_{i,c} = \frac{\mathrm{e}^{-\alpha|v_i-\mu_c|^2}}{\sum_{c'} \mathrm{e}^{-\alpha|v_i-\mu_{c'}|^2}}, \qquad \mu_c = \frac{\sum_i \gamma_{i,c} w_i v_i}{\sum_i \gamma_{i,c} w_i}, \tag{5.6}$$

where μ_c is the estimated mean of cluster c, and $\gamma_{i,j}$ is the estimated assignment of embedding i to the cluster c. The parameter α controls the hardness of the clustering. As the value of α increases, the algorithm approaches K-means.

The weights w_i may be set in a variety of ways. A reasonable choice could be to set w_i according to the power of the mixture in each TF bin. Here we set the weights to 1, except in silence TF bins where the weight is set to 0. Silence is defined using a threshold on the energy relative to the maximum of the mixture.

End-to-end training is performed by *unfolding* the steps of (5.6), and treating them as layers in a clustering network, according to the general framework known as *deep unfolding* (Hershey et al., 2014). The gradients of each step are thus passed to the previous layers using standard back-propagation.

5.10 Experiments

Experimental setup: We evaluate deep clustering on a single-channel speaker-independent speech separation task, considering mixtures of two and three speakers with all gender combinations. For two-speaker experiments, we use the corpus introduced in (Hershey et al., 2016a), derived from the Wall Street Journal (WSJ0) corpus. It consists in a 30 h training set and a 10 h validation set with two-speaker mixtures generated by randomly selecting utterances by different speakers from the WSJ0 training set `si_tr_s`, and mixing them at various signal-to-noise ratios (SNR) randomly chosen between 0 dB and 10 dB. The validation set was here used to optimize some tuning parameters. The 5 h test set consists in mixtures similarly generated using utterances from 16 speakers from the WSJ0 development set `si_dt_05` and evaluation set `si_et_05`. The speakers are different from those in our training and validation sets, leading to a speaker-independent separation task. For three-speaker experiments, we created a corpus similar to the two-speaker one, with the same amounts of data generated from the same datasets. All data were downsampled to 8 kHz before processing to reduce computational and memory costs. The input features X were the log spectral magnitudes of the speech mixture, computed using a short-time Fourier transform (STFT) with a 32 ms sine window and 8 ms shift.

The scores are reported in terms of signal-to-distortion ratio (SDR), which we define as scale-invariant SNR. As oracle upper bounds on performance for our datasets, we report in Table 5.4 the results obtained using two types of "ideal" masks: the ideal binary mask (ibm) defined as $a_i^{\text{ibm}} = \delta(|s_i| > \max_{j \neq i} |s_j|)$, which leads to highest SNR among all binary masks, and a "Wiener-like" filter (wf) defined as $a_i^{\text{wf}} = |s_i|^2 / \sum_j |s_j|^2$, which empirically leads to good SNR, with values in $[0, 1]$ (Wang, 2005; Erdogan et al., 2015a). Here s_i denotes the time-frequency representation of speaker i. CASA (Hu and Wang, 2013b) and previous deep clustering (Hershey et al., 2016a) results are also shown for the two-speaker set.

Table 5.4: SDR (dB) improvements using the ideal binary mask (ibm), oracle Wiener-like filter (wf), compared to prior methods dpcl (Hershey et al., 2016a) and CASA (Hu and Wang, 2013b) on the two- and three-speaker test sets.

# speakers	ibm	wf	dpcl v1 (Hershey et al., 2016a)	CASA (Hu and Wang, 2013b)
2	13.5	13.9	6.0	3.1
3	13.3	13.8	-	-

The initial system, based on (Hershey et al., 2016a), trains a deep clustering model on 100-frame segments from the two-speaker mixtures. The network, with 2 BLSTM layers, each having 300 forward and 300 backward LSTM cells, is denoted as 300×2. The learning rate for the rmsprop algorithm (Tieleman and Hinton, 2012a) was $\lambda = 0.001 \times (1/2)^{\lfloor \epsilon/50 \rfloor}$, where ϵ is the epoch number.

Regularization: We first considered improving performance of the baseline using common regularization practices. Table 5.5 shows the contribution of dropout ($p = 0.5$) on feed-forward connections, recurrent dropout ($p = 0.2$), and gradient normalization ($|\nabla| \leq 200$), where the parameters were tuned on development data. Together these result in a 3.3 dB improvement in SDR relative to the baseline.

Table 5.5: Decomposition of the SDR improvements (dB) on the two-speaker test set using 300×2 model.

rmsprop	+dropout	+recurrent dropout	+norm constraint
5.7	8.0	8.9	9.0

Architecture: Various network architectures were investigated by increasing the number of hidden units and number of BLSTM layers, as shown in Table 5.6. An improvement of 9.4 dB SDR was obtained with a deeper 300 × 4 architecture, with 4 BLSTM layers and 300 units in each LSTM.

Table 5.6: SDR (dB) improvements on the two-speaker test set for different architecture sizes.

model	same-gender	different-gender	overall
300×2	6.4	11.2	9.0
600×2	6.1	11.5	9.0
300×4	7.1	11.5	9.4

Pre-training of temporal context: Training the model with segments of 400 frames, after pre-training using 100-frame segments, boosts performance to 10.3 dB, as shown in Table 5.7, from 9.9 dB without pre-training. Results for the remaining experiments are based on the pre-trained 300×4 model.

Table 5.7: SDR (dB) improvements on the two-speaker test set after training with 400 frame length segments.

model	same-gender	different-gender	overall
600×2	7.8	11.7	9.9
300×4	8.6	11.7	10.3

Multi-speaker training: We train the model further with a blend of two- and three-speaker mixtures. For comparison, we also trained a model using only three-speaker mixtures, again training first over 100-frame segments, then over 400-frame segments. The performance of the models trained on two-speaker mixtures only, on three-speaker mixtures only, and using the multi-speaker training, are shown in Table 5.8. The three-speaker mixture model seems to generalize better to two speakers than vice versa, whereas the multi-speaker trained model performed the best on both tasks.

Table 5.8: Generalization across different numbers of speakers in terms of SDR improvements (dB).

Training data	**Test data**	
	2 speaker	3 speaker
2 speaker	10.3	2.1
3 speaker	8.5	7.1
Mixed curriculum	10.5	7.1

Soft clustering: The choice of the clustering hardness parameter α and the weights on TF bins is analyzed on the validation set, with results in Table 5.9. The use of weights to ignore silence improves performance with diminishing returns for larg α. The best result is for $\alpha = 5$.

Table 5.9: Performance as a function of soft weighted K-means parameters on the two-speaker validation set.

weights	$\alpha = 2$	$\alpha = 5$	$\alpha = 10$	hard K-means
all equal	5.0	10.1	10.1	10.3
mask silent	9.1	10.3	10.2	10.3

End-to-end training: Finally, we investigate end-to-end training, using a second-stage enhancement network on top of the deep clustering ('dpcl') model. Our enhancement network features two BLSTM layers with 300 units in each LSTM layer, with one instance per source followed by a soft-max layer to form a masking function. We first trained the enhancement network separately ('dpcl + enh'), followed by end-to-end fine-tuning in combination with the dpcl model ('end-to-end'). Table 5.10 shows the improvement in SDR as well as *magnitude SNR* (SNR computed on the magnitude spectrograms).

The magnitude SNR is insensitive to phase estimation errors introduced by using the noisy phases for reconstruction, whereas the SDR might get worse as a result of phase errors, even

Table 5.10: SDR / Magnitude SNR improvements (dB) and WER with enhancement network.

model	same-gender	different-gender	overall	WER
dpcl	8.6 / 8.9	11.7 / 11.4	10.3 / 10.2	87.9 %
dpcl + enh	9.1 / 10.7	11.9 / 13.6	10.6 / 12.3	32.8 %
end-to-end	9.4 / 11.1	12.0 / 13.7	10.8 / 12.5	30.8%

if the amplitudes are accurate. Speech recognition uses features based on the amplitudes, and hence the improvements in magnitude SNR seem to predict the improvements in WER due to the enhancement and end-to-end training. Fig. 5.3 shows that the SDR improvements of the end-to-end model are consistently good on nearly all of the two-speaker test mixtures.

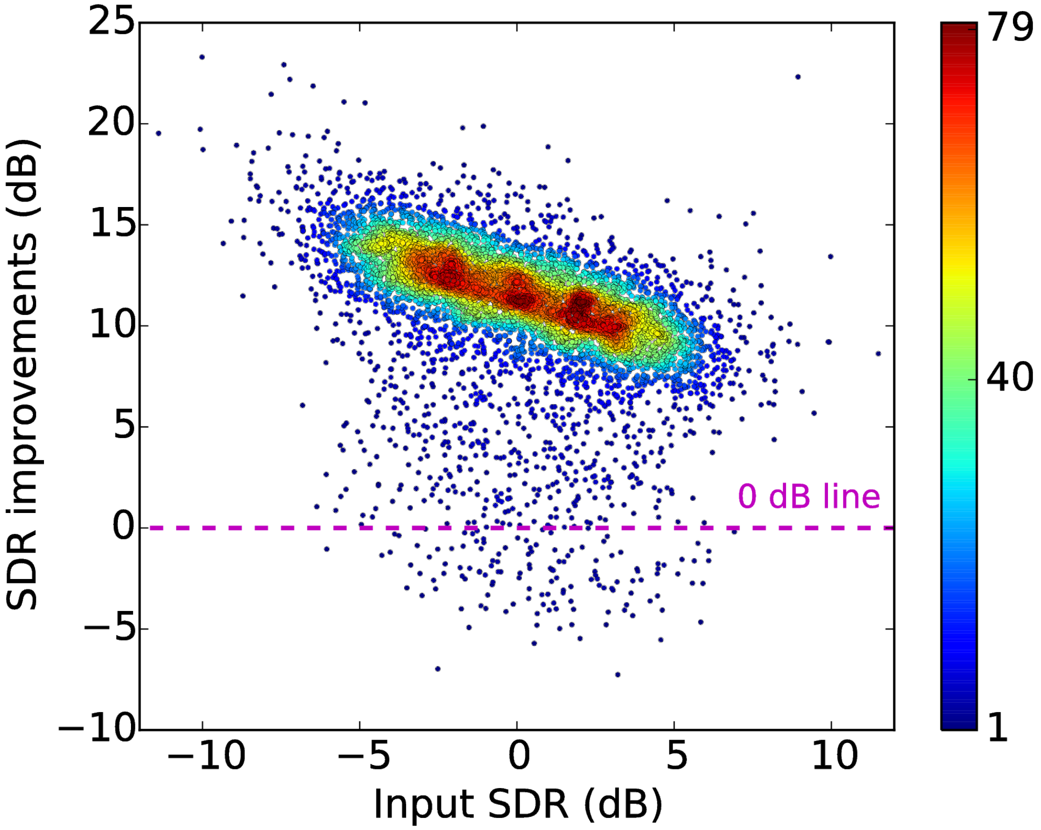
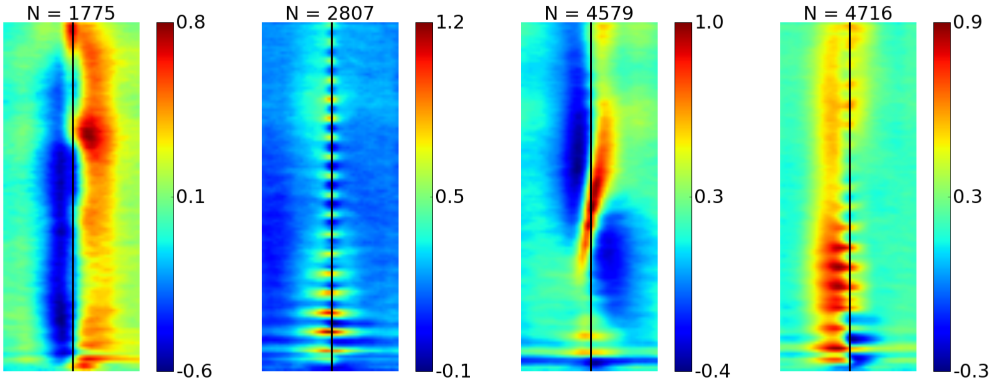


Figure 5.3: Scatter plot for the input SDRs and the corresponding improvements. Color indicates density.

ASR performance: We evaluated ASR performance (WER) with GMM-based clean-speech WSJ models obtained by a standard Kaldi recipe (Povey et al., 2011). The noisy baseline result on the mixtures is 89.1 %, while the result on the clean speech is 19.9 %. The raw output from dpcl did not work well, despite good perceptual quality, possibly due to the effect of near-zero values in the masked spectrum, which is known to degrade ASR performance. However, the enhancement networks significantly mitigated the degradation, and finally obtained 30.8 % with the end-to-end network.

Visualization: To gain insight into network functioning, we performed reverse correlation experiments. For each node, we average the 50-frame patches of input centered at the time when the node is active (e.g., the node is at 80% of its maximum value). Fig. 5.4 shows a variety of interesting patterns, which seem to reflect such properties as onsets, pitch, frequency chirps, and vowel-fricative transitions.



(a) onset **(b)** pitch **(c)** chirp **(d)** transition

Figure 5.4: Example spike-triggered spectrogram averages with 50-frame context, for active LSTM nodes in the second layer. N is the number of active frames for the corresponding node.

Chapter 6

Deep attractor network

In previous chapter, two deep learning based methods have been introduced to resolve these problems, which are known as ”deep clustering (DC)(Hershey et al., 2016b)” and ”permutation invariant training (PIT)(Yu et al., 2016)”. In deep clustering, a network is trained to generate discriminative embedding for each time-frequency (T-F) bin with points belonging to the same source forced to be closer to each other. DC is able to solve both permutation and output dimension problem to produce the state of the art separation performance. The main drawback of DC is its inefficiency to perform end-to-end mapping, because the objective function is the affinity between the sources in the embedded space and not the separated signals themselves. Minimizing the separation error is done with an unfolding clustering system and a second network, which is trained iteratively and stage by stage to ensure convergence (Isik et al., 2016b). The PIT algorithm solves the permutation problem by pooling over all possible permutations for N mixing sources ($N!$ permutations), and use the permutation with lowest error to update the network. PIT was first proposed in (Hershey et al., 2016b), and was later shown to have comparable performance as DC (Yu et al., 2016). However, PIT approach suffers the output dimension mismatch problem because it assumes a fixed number of sources. PIT also suffers from its computation efficiency, where the prediction window has to be much shorter than context window due to the inconsistency of the permutation both across and within sample segments.

In this chapter, we propose a novel deep learning framework which we refer to as the attractor network to solve the source separation problem. The term attractor refers to the well-studied perceptual effects in human speech perception which suggest that the brain circuits create perceptual attractors (magnets) that warp the stimulus space such that to draws the sound that is closest to it, a phenomenon that is called Perceptual Magnet Effect (Kuhl, 1991). Our proposed model works on the same principle by forming a reference point (attractor) for each source in the embedding space which draws all the T-F bins toward itself. Using the similarity between the embedded points and each attractor, a mask is estimated for each sources in the mixture. Since the mask is directly related to the attractor point, the proposed framework can potentially be extended to arbitrary number of sources without the permutation problem. Moreover, the mask learning enables a very efficient end-to-end training scheme and highly reduces the computation complexity compared with DC and PIT.

In Section 6.1, the proposed model is explained and discussed in more detail. In Section 6.5, we evaluate the performance of proposed system.

6.1 Attractor Neural Network

6.2 Model

The neural network is trained to map the mixture sound X to a k dimensional embedding space, such that it minimizes the following objective function:

$$\mathcal{L} = \sum_{f,t,c} \|S_{f,t,c} - X_{f,t} \times M_{f,t,c}\|_2^2 \tag{6.1}$$

where S is the clean spectrogram (frequency F × time T) of C sources, X is the mixture spectrogram (frequency F × time T), and M is the mask formed to extract each source. The mask is estimated in the K dimensional embedding space of each T-F bin, represented by $V \in \mathbb{R}^{F\times T\times K}$:

$$M_{f,t,c} = Sigmoid(\sum_k A_{c,k} \times V_{f,t,k}) \tag{6.2}$$

where $A \in \mathbb{R}^{C\times K}$ are the attractors for the C sources in the embedding space, learnt during training, which are defined as

$$A_{c,k} = \frac{\sum_{f,t} V_{k,f,t} \times Y_{c,f,t}}{\sum_{f,t} Y_{c,f,t}} \tag{6.3}$$

which $Y \in \mathbb{R}^{F\times T\times K}$ is the source membership function for each T-F bin, i.e., $Y_{t,f,c} = 1$ if source c has the highest energy at time t and frequency f compare to the other sources.

The objective function in Equation 6.1 consists of three parts. During training, we first compute an embedding V through a forward pass of the neural network for each given mixture. Then an attractor vector is estimated for each source using Equation 6.3. This can be done in several ways which we will elaborate in Section 6.3. The most straightforward method for attractor generation is to find the source centroid, as defined in Equation 6.3.

Next, we estimate a reconstruction mask for each source by finding the similarity of each T-F bin in the embedding space to each of the attractor vectors A, where the similarity metric is defined in Equation 6.2. This particular metric uses the inner product followed by a sigmoid function which monotonically scales the masks between $[0, 1]$. Intuitively, if an embedding of a T-F bin is closer to one attractor means that it belongs to that source, and the resulting mask for that source will produce larger values for that T-F bin. Since it is usually useful for source separation system to have constraint that the summation of each mask equal one for each TF bin, especially for difficult mixtures, the sigmoid function in mask forming step in 6.2 could be replaced with softmax function, leads to eqn. 6.4.

$$M_{f,t,c} = Softmax(\sum_k A_{c,k} \times V_{f,t,k}) \tag{6.4}$$

Finally, a standard $L2$ reconstruction error is used to generate the gradient, as shown in Equation 6.1. Therefore, the error for each source reflects the difference between the masked signal and the clean reference, forcing the network to optimize the global reconstruction error for better separation. We refer the proposed net as deep attractor network (DANet). Figure 6.1 shows the structure of the proposed system.

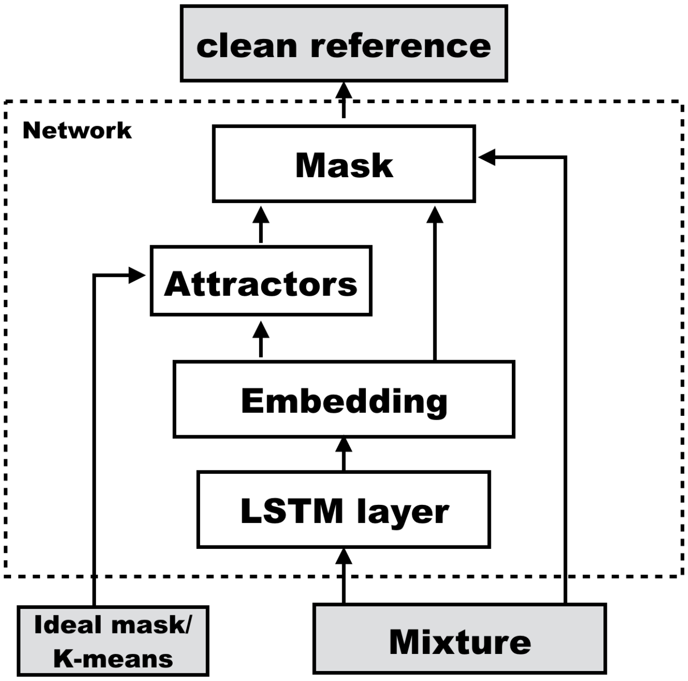


Figure 6.1: The system architecture. In the training time, a ideal mask is applied to form the attractor, while during the testing time, Kmeans is used to form the attractor. Alternatives for Kmeans is further discussed in Section 6.3

In comparison with previous methods, DANet network has two distinct advantages. Firstly, DANet removes the stepwise pre-training required in DC method to enable end-to-end training. Another big advantage of DANet arises from the flexibility in source dependent training, where the source-dependent knowledge could be easily incorporated by the attractor (e.g. speaker identity).

6.3 Estimation of attractor points

Attractor points can be estimated using various methods other than the average used in Equation 6.3. One possibility is to use weighted average. Since the attractors represents the source center of gravity, we can include only the embeddings of the most salient T-F bins, which leads to more robust estimation. We investigate this strategy by using an amplitude threshold in the estimation of the attractor. Alternatively, a neural network model may also be used to pick the representative embedding for each source, an idea which shares similarities with encoder-decoder attention networks (Bahdanau et al., 2014; Cho et al., 2015).

During test time, because the true assignment Y is unknown, we incorporate two strategies to form the attractor points. The first is similar to the strategy used in DC, where the centers are found using post K-means algorithm. The second method is based on the observation that the location of the attractors in the embedding space is relatively stable. This observation is shown in Figure 6.2, where each pair of dots corresponds to the attractor found for the two speakers in a given mixture. Figure 6.2 shows two principle stable attractor pairs for all the mixtures used, however, this observation needs to be tested in more depth and different tasks and datasets.

6.4 Relation with DC and PIT

The objective function of DC is shown in Equation 6.5, where Y is the indicator function which is equivalent to a binary mask, and V is the learnt embedding:

$$\mathcal{L} = \left\|YY^T - VV^T\right\|_2^2 \tag{6.5}$$

Since Y is orthogonal and constant for each mixture, by multiplying Y^T and a normalizer $U = (Y^TY)^{-1}$ to both term, we can get an objective function that is a special case of the attractor network, as in Equation 6.6:

$$\mathcal{L} = \left\|Y^T - UY^TVV^T\right\|_2^2 \tag{6.6}$$

In Equation 6.6, UY^TV can be viewed as an averaging step, where the embeddings are summed according to the label, and the resulted center is multiplied with the embedding matrix V to measure the similarity between each embedding and the center, and compared with the ground truth binary mask. When the learnt V is optimum, i.e, $VV^T = YY^T$, equation 6.5 and 6.6 are equivalent.

On the other hand, when the attractor vectors are considered as free parameters in the network, DANet reduces to a classification network (Hori et al., 2015; Chen et al., 2015), and Equation 6.1 becomes a fully-connected layer. In this case, PIT becomes necessary since the mask has no information about the source and the problem of fixed output dimension arises. In contrast, the freedom of the network to form attractor points during the training allows the system to use the affinity between samples where no constraint is on the number

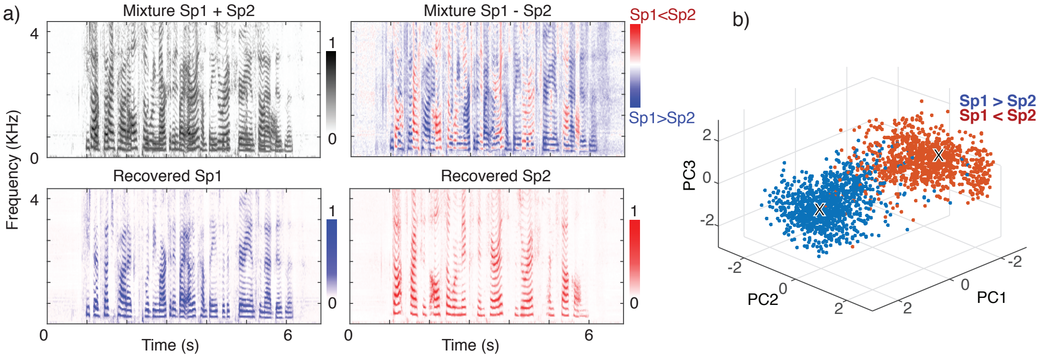


Figure 6.2: Location of T-F bins in the embedded space. Each dot visualizes the first three principle components of one T-F bin, where colors distinguish the relative power of speakers, and the location of attractors is marked with X.

of patterns found, therefore allowing the network to be independent of the number of sources. The flexibility of the network in choosing the attractor points is helpful even in two-source separation problem, because the two sources may have very different structures. As can be seen in Figure 6.3, our proposed method trained in speaker separation tasks has ended up finding 2 attractor pairs (4 points in the embedding space), which can be expected to increase in harder problems.

6.5 Evaluation

6.5.1 Experimental setup

We evaluate our proposed model on the task of single-channel overlapped two-speaker separation. We use the corpus introduced in (Hershey et al., 2016b), which contains a 30 h training set and a 10 h validation set generated by randomly selecting utterances from different speakers in the Wall Street Journal (WSJ0) training set si_tr_s, and mixing them at various signal-to-noise ratios (SNR) randomly chosen between 0 dB and 10 dB. 5 h evaluation set is generated similarly as above, using utterances from 16 unseen speakers from si_dt_05 and si_et_05 in WSJ0 dataset. Additionally, we construct a three speaker mixture dataset for three speaker separation evaluation from same WSJ set, which has 30h training, 10 hours validation and 5 hours testing data, with mixing SNR at -5 5 dB. We ensure that in each mixture, there exist both female and male speakers. All data are resampled to 8 kHz to reduce computational and memory costs. The log spectral magnitude is served as input feature, computed using short-time Fourier transform (STFT) with 32 ms window length, 8 ms hop size, and the square root of hanning window.

The network contains 4 Bi-directional LSTM (Hochreiter and Schmidhuber, 1997) layers with 600 hidden units in each layer. The embedding dimension is set to 20, resulting in a fully-connected feed-forward layer of 2580 hidden units (20 × 129) after the BLSTM layers.

We split the input features into non-overlapping chunks of 100-frame length as the input to the network. RMSprop algorithm (Tieleman and Hinton, 2012b) is used for training with an exponential learning rate decaying strategy, where the learning rate starts at 10^{-4} and ends at 3×10^{-6}. The total number of epochs was set to be 150, and we used the cost function in Equation 6.1 on the validation set for early stopping. The criteria for early stopping is no decrease in the loss function on validation set for 10 epochs. We constructed a Deep Clustering (DC) network with the same configuration which is used as the baseline.

We report the results in terms of signal-to-distortion ratio (SDR, which we define as scale-invariant SNR here), signal-to-artifacts ratio (SAR), and signal-to-interference ratio (SIR). The results are shown in Table 6.1

6.5.2 Separation examples

Figure 6.2 shows an example of mixture, the difference between the two speakers in the mixture, and the separated spectrograms of the two speakers using DANet. Also visualized in Figure 6.2 is the embeddings of the mixture projected onto its first Principal Components. In Figure 6.2, each point represents one T-F bin in the embedding space. Red and blue dots correspond to the T-F bins where speaker one or two have greater energy accordingly. The location of attractors are marked by x. It shows that two symmetric attractor centers are formed, each corresponding to one of the speakers in the mixture. A clear boundary can be observed in the figure, showing that the network successfully pulled the two speakers apart toward their corresponding attractor points.

Figure 6.3 shows the location of attractors for 10,000 mixture examples, mapped onto the 3-dimensional space for visualization purpose using Principal Component Analysis. It suggests that the network may have learned two attractor pairs (4 symmetric centers), marked by A1 and A2. This observation confirms our intuition of the DANet mentioned in Section , that DANet has the ability to discover different number of attractors in an unsupervised way, and therefore, form complex separation strategies. Although the task considered in this study is already challenging, one can imagine much more difficult separation scenarios, where the number of speakers in the mixture is large and can change over time. The ability of DANet to form new attractors may prove to be crucial in such cases, because any effort in pre-setting the number of mixing patterns, as done in methods such as PIT, will hinder the generalization ability of the network. Figure 6.3 also suggests that hierarchical clustering methods can be more suitable, where attractors can drive a hierarchical grouping of sources, allowing a better representation of audio signals. We will explore these issues in future work.

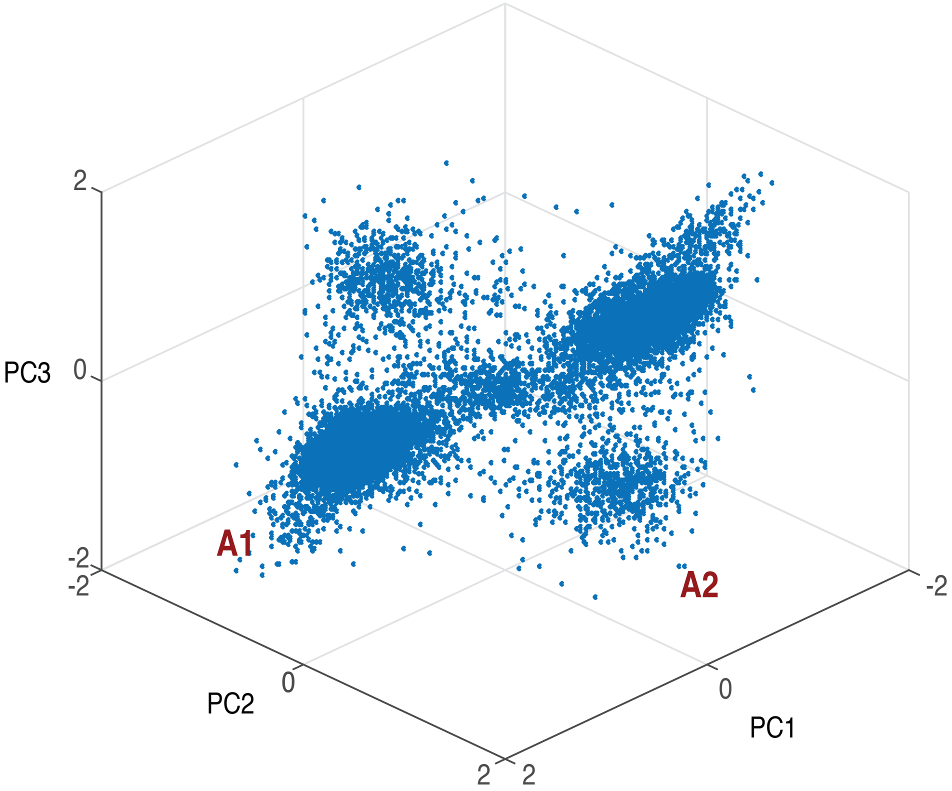


Figure 6.3: Location of attractor points in the embedding space. Each dot corresponds to one of the 10000 mixtures sounds, visualized using the first three principal components. Two distinct attractor pairs are visible (denoted by A1 and A2).

6.6 Results

Table 6.1 shows the evaluation results for different networks (example sounds can be found here (web)). Although the plain DANet already outperforms the DC baseline, adding a simple threshold on T-F samples included in the formation of the attractor yields further improved performance, presumably due to the increased influence of salient segments. On the other hand, the performance boost suggests that better attractor formation procedures can be utilized to estimate the attractors, such as joint optimizing of the network parameters. Moreover, by applying curriculum training strategy (Isik et al., 2016b), which we continue training the network with 400-frame length input, DANet achieves the best overall performance.

In the last experiment in Table 6.1, a fixed pair of attention vector collected from the training data is used, corresponding to the A1 pair in Figure 6.3. This pre-set attractor is able to generalize well to the unseen mixtures and produced high quality separation, however it was slightly worse than the best model. Compared with K-means, this has the advantage that it can be easily implemented in real-time using a frame-by-frame pipeline. Based on this observation, when more attractors are required (e.g. in more complex tasks), a collection of attractor codebook and a simple classifier could be implemented for real-time processing.

In three speaker separation experiment, shown in Table 6.2, the proposed system significantly outperform the deep clustering baseline. This result is natural since deep clustering was trained to estimate binary mask, while the deep attractor network focuses on the signal reconstruction. When the mixture is relative simple, the binary mask could generate high quality separation. However, for more complex mixtures, or when one source is significantly louder than the other, the binary mask usually lead to large bias to the loudest source, and thus result in unsatisfiable separation for weaker source. Note that in the three speaker experiment, the network was trained using softmax objective as shown in (6.4).

	GNSDR	GSAR	GSIR
DC	9.1	9.5	**22.2**
DANet	9.4	10.1	18.8
DANet-50%	9.4	10.4	17.3
DANet-70%	9.6	10.3	18.7
DANet-90%	9.6	10.4	18.1
DANet-90%‡	**10.5**	**11.1**	20.3
fix-DANet-90%	9.5	10.4	17.8

Table 6.1: Evaluation metrics for networks with different configurations. The percentage suffixes stand for the salient weight threshold used during training. ‡: curriculum training with 400-frame length input. fix-DANet: use fixed attractor points calculated by training set.

	GNSDR	GSAR	GSIR
DC	6.3	2.3	12.6
DANet	7.7	3.9	13.2
DANet‡	**8.8**	**5.0**	**15.0**

Table 6.2: Evaluation results for three speaker separation. ‡: curriculum training with 400-frame length input.

Chapter 7

Other application

In previous chapters, we introduced different neural network based auditory source separation systems, with the applications in speech enhancement and speaker separation. In this chapter, we introduce the applications using proposed models from chapter 4 ∼6, in music separation, which attract raising interests in recent years. The task for music separationis to separate the sining voice and the background music. Such system shows important value in applications like Karaoke application, and 3D audio creation, where single sources of audio is required.

7.1 Monaural music separation

Monaural music source separation has been the focus of many research efforts for over a decade. This task aims at separating a music recording into several tracks where each track corresponds to a single instrument. A related goal is to design algorithms that can separate vocals and accompaniment, where all the instruments are considered as one source. Music source separation algorithms have been successfully used for predominant pitch tracking (Fan et al., 2016), accompaniment generation for Karaoke systems (Tachibana et al., 2016), or singer identification (Berenzweig et al., 2002).

Despite these advances, a system that can successfully generalize to different music datasets has thus far remained unachievable, due to the tremendous variability of music recordings, for example in terms of genre or types of instruments used. Unsupervised methods, such as those based on computational auditory scene analysis (CASA) (Li and Wang, 2007), source/filter modeling (Durrieu et al., 2010), or low-rank and sparse modeling (Huang et al., 2012), have difficulty in capturing the dynamics of the vocals and instruments, while supervised methods, such as those based on non-negative matrix factorization (NMF) (Sprechmann et al., 2012), F0-based estimation (Hsu and Jang, 2010), or Bayesian modeling (Yang et al., 2014), suffer from generalization and processing speed issues.

Recently, deep learning has found many successful applications in audio source separation. Conventional regression-based networks try to infer the source signals directly, often by inferring time-frequency (T-F) masks to be applied to the T-F representation of the mixture so as to recover the original sources. These mask-inference networks have been shown to produce superior results compared to the traditional approaches in singing voice separation (Huang et al., 2014b). These networks are a natural choice when the sources can be characterized as belonging to distinct classes.

Another promising approach designed for more general situations is the so-called deep clustering framework (Hershey et al., 2016b). Deep clustering has been applied very successfully to the task of single-channel speaker-independent speech separation (Hershey et al., 2016b). Because of its use of pair-wise affinities as a separation criterion, deep clustering can handle conditions with multiple sources of the same type, and an arbitrary number of sources. Such difficult conditions are endemic to music separation.

In this study, we explore the use of both deep clustering and conventional mask-inference networks to separate the singing voice from the accompaniment, grouping all the instruments as one source and the vocals as another. The singing voice separation task that we consider here is amenable to separation based on classes, and would not seem to require the extra flexibility in terms of source types and number of sources that deep clustering would provide. However, in addition to opening up the potential to apply to more general settings, the additional flexibility of deep clustering may have some benefits in terms of regularization. To investigate this possibility, we develop a two-headed "Chimera" network with both a deep clustering head and a mask-inference head attached to the same network body. Each head has its own objective, but the whole hybrid network is trained in a joint fashion akin to multi-task training. Our findings show that the addition of the deep clustering criterion greatly improves upon the performance of the mask-inference network.

7.2 Model Description

7.2.1 Multi-task learning and Chimera networks

Whereas the deep clustering objective function has been shown to enable the training of neural networks for challenging source separation problems, a disadvantage of deep clustering is that the post-clustering process needed to generate the mask and recover the sources is not part of the original objective function. On the other hand, for mask-inference networks, the objective function minimized during training is directly related to the signal recovery quality. We seek to combine the benefits of both approaches in a strategy reminiscent of multi-task learning, except that here both approaches address the same separation task.

In (Hershey et al., 2016b) and (Isik et al., 2016b), the typical structure of a deep clustering network is to have multiple stacked recurrent layers (e.g., BLSTMs) yielding an N-dimensional vector at the top layer, followed by a fully-connected linear layer. For each frame t, this layer outputs a D-dimensional vector for each of the F frequencies, resulting in a $F \times D$ representation $\mathbf{Z}_t$. To form the embeddings, $\mathbf{Z}$ then passes through a tanh non-linearity, and unit-length normalization independently for each T-F bin. Concatenating across time results in the $TF \times D$ embedding matrix $\mathbf{V}$ as used in deep clustering.

We extend this architecture in order to create a two-headed network, which we refer to as "Chimera" network, with one head outputting embeddings as in a deep clustering network,

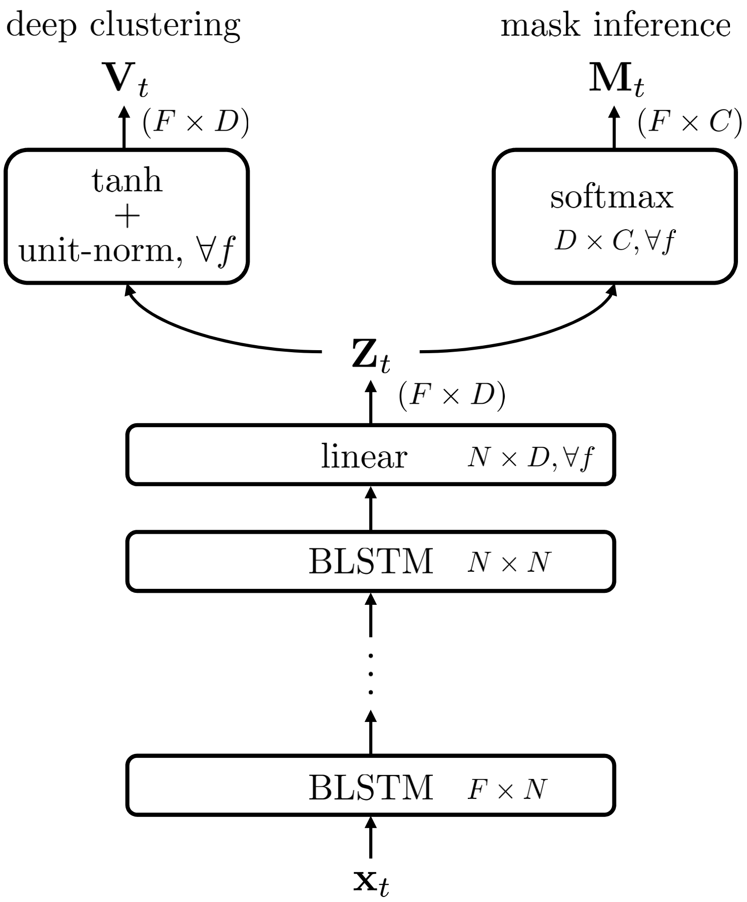


Figure 7.1: Structure of the Chimera network.

and the other head outputting a soft mask, as in a mask-inference network. The new mask-inference head is obtained starting with $\mathbf{Z}$, and passing it through F fully-connected $D \times C$ mask estimation layers (e.g., softmax), one for each frequency, resulting in C masks $\mathbf{M}^{(c)}$, one for each source. The structure of the Chimera network is illustrated in Figure 7.1.

The body of the network, up to the layer outputting $\mathbf{Z}$, can be trained with each head separately. For the deep clustering head, we use the objective $\mathcal{L}_{\text{dc}}$. For the mask-inference head, we can use the magnitude spectrum approximation objective function (Huang et al., 2012; Weninger et al., 2014c):

$$\mathcal{L}_{\text{mi}} = \sum_c ||\mathbf{R}^{(c)} - \mathbf{M}^{(c)} \odot \mathbf{S}||_2^2 \tag{7.1}$$

where $\mathbf{R}^{(c)}$ denotes the T-F representation of the c-th clean source and $\mathbf{S}$ that of the mixture. We can also define a global objective for the whole network as

$$\mathcal{L}_{\text{chi}} = \alpha \frac{\mathcal{L}_{\text{dc}}}{TF} + (1 - \alpha)\mathcal{L}_{\text{mi}} \tag{7.2}$$

where $\alpha \in [0, 1]$ controls the importance between the two objectives. Note that here we divide $\mathcal{L}_{\text{dc}}$ by TF because the objective for deep clustering calculates the pair-wise loss for each pair of T-F bins, while the spectrum approximation objective calculates end-to-end loss. For $\alpha = 1$, only the deep clustering head gets trained together with the body, resulting in a deep clustering network. For $\alpha = 0$, only the mask-inference head gets trained together with the body, resulting in a mask-inference network.

At test time, if both heads have been trained, either can be used. The mask-inference head directly outputs the T-F masks, while the deep clustering head outputs embeddings on which we perform clustering using, e.g., K-means.

7.3 Evaluation and discussion

7.3.1 Datasets

For training and evaluation purposes, we built a remixed version of the DSD100 dataset for SiSEC (DSD), which we refer to as DSD100-remix. For evaluation only, we also report results on two other datasets: the hidden iKala dataset for the MIREX submission, and the public iKala dataset for our newly proposed models.

The DSD100 dataset includes synthesized mixtures and the corresponding original sources for 100 professionally produced and mixed songs. To build the training and validation sets of DSD100-remix, we use the DSD100 development set (50 songs). We design a simple energy-level-based detector (Ramirez et al., 2007) to remove silent parts in both the vocal and accompaniment tracks, so that the vocals and accompaniment fully overlap in the generated mixtures. After that, we downsample the tracks from 44.1 kHz to 16 kHz to reduce computational cost, and then randomly mix the vocals and accompaniment together at 0 dB SNR, creating a 15 h training set and a 0.5 h validation set. We build the evaluation set of DSD100-remix from the DSD100 test set using a similar procedure, generating 50 pieces (one for each song) of fully-overlapped recordings with 30 seconds length each.

The input feature we use is calculated by the short-time Fourier transform (STFT) with 512-point window size and 128-point hop size. We use a 150-dimension mel-filterbank to reduce the input feature dimension. First-order delta of the mel-filterbank spectrogram is concatenated into the input feature. We used the ideal binary mask calculated on the mel-filterbank spectrogram as the target Y matrix.

7.3.2 System architecture

The Chimera network's body is comprised of 4 bidirectional long-short term memory (BLSTM) layers with 500 hidden units in each layer, followed by a linear fully-connected layer with a $D = 20$-dimension vector output for each of the frame's $F = 150$ T-F bins. In the mask-inference head, we set $C = 2$ for the singing voice separation task, and use softmax as the non-linearity. We use the rmsprop algorithm (Tieleman and Hinton, 2012c) as optimizer and select the network with the lowest loss on the validation set.

At test time, we split the signal into fixed-length segments, on which we run the network independently. We also tried running the network on the full input feature sequence, as in (Hershey et al., 2016b), but this lead to worse performance, probably due to the mismatch in context size between the training and test time. The mask-inference head of the network directly generates T-F masks. For deep clustering, the masks are obtained by applying K-means on the embeddings for the whole signal. We apply the mask for each source to the mel-filterbank spectrogram of the input, and recover the source using an inverse mel-filterbank transform and inverse-STFT with the mixture phase, followed by upsampling.

7.3.3 Results for the MIREX submission

We first report on the system submitted to the Singing Voice Separation task of the Music Information Retrieval Evaluation eXchange (MIREX 2016) (sin, b). That system only contains the deep clustering part, which corresponds to $\alpha = 1$ in the hybrid system. In the MIREX system, dropout layers with probability 0.2 were added between each feed-forward connection, and sequence-wise batch normalization (Laurent et al., 2015) was applied in the input-to-hidden transformation in each BLSTM layer. Similarly to (Isik et al., 2016b), we also applied a curriculum learning strategy (Bengio et al., 2009), where we first train the network on segments of 100 frames, then train on segments of 500 frames. As distinguishing between vocals and accompaniment was part of the task, we used a crude rule-based approach: the mask whose total number of non-zero entries in the low frequency range ($<$ 200 Hz) is more than a half is used as the accompaniment mask, and the other as the vocals mask.

The hidden iKala dataset has been used as the evaluation dataset throughout MIREX 2014-2016, so we can report, as shown in Table 7.1, the results from the past three years, comparing the best two systems in each year's competition to our submitted system for 2016. The official MIREX results are reported in terms of global normalized SDR (GNSDR), global SIR (GSIR), global SAR (GSAR) (sin, a).

Due to time limitations at the time of the MIREX submission, we submitted a system that we had trained using the DSD100-remix dataset described in Section 7.3.1. However, as mentioned in the MIREX description, the DSD100 dataset is different from both the hidden and public parts of the iKala dataset (sin, a). Nonetheless, our system not only won the 1st place in MIREX 2016 but also outperformed the best systems from past years, even without training on the better-matched public iKala dataset, showing the efficacy of deep clustering for robust music separation. Note that the hidden iKala dataset is unavailable to the public, and it is thus unfortunately impossible to evaluate here what the performance of our system would be when trained on the public iKala data.

7.3.4 Results for the proposed hybrid system

We now turn to the results using the Chimera networks. During the training phase, we use 100 frames of input features to form fixed duration segments. We train the Chimera network in three different regimes: a pure deep clustering regime (DC, $\alpha = 1$), a pure mask-inference regime (MI, $\alpha = 0$), and a hybrid regime (HD$_\alpha$, $0 < \alpha < 1$). All networks are trained from random initialization, and no training tricks mentioned above for the MIREX system are

Table 7.1: Evaluation metrics for different systems in MIREX 2014-2016 on the hidden iKala dataset. V denotes vocals and M music.

	GNSDR		GSIR		GSAR	
	V	M	V	M	V	M
DC	6.3	11.2	14.5	25.2	10.1	7.3
MC2 (sin, b)	5.3	9.7	10.5	19.8	11.2	6.1
MC3 (sin, b)	5.5	9.8	10.8	19.6	11.2	6.3
FJ1 (Fan et al., 2016)	6.8	10.1	13.3	11.2	11.5	10.0
FJ2 (Fan et al., 2016)	6.3	9.9	13.7	11.7	10.6	9.1
IIY1 (Ikemiya et al., 2016)	4.2	7.8	15.5	12.4	7.7	5.4
IIY2 (Ikemiya et al., 2016)	4.5	7.9	13.3	14.3	8.6	5.0

Table 7.2: SDRi (dB) on the DSD100-remix and the public iKala datasets. The suffix after HD$_\alpha$ denotes which head of the Chimera network is used for generating the masks.

	DSD100-remix		iKala	
	V	M	V	M
DC	4.9	7.2	6.1	10.0
MI	4.8	6.7	5.2	8.9
HD$_{0.05}$-DC	4.8	7.2	6.0	9.8
HD$_{0.05}$-MI	5.5	7.8	6.4	10.6
HD$_{0.1}$-DC	4.9	7.3	6.1	9.9
HD$_{0.1}$-MI	**5.6**	**7.9**	**6.5**	**10.7**

added. We report results on the DSD100-remix test set, which is matched to the training data, and the public iKala dataset, which is not.

By design, deep clustering provides one output for each source, without having to identify which is which. Therefore, the scores are computed by using the best permutation between references and estimates at the file level. Table 7.2 shows the results. We compute the source-to-distortion ratio (SDR), defined as scale-invariant SNR (Isik et al., 2016b), for each test example, and report the length-weighted average over each test set of the improvement of SDR in the estimate with respect to that in the mixture (SDRi).

As can be seen in the results, MI performs competitively with DC on DSD100-remix, however DC performs significantly better on the public iKala data. This shows the better generalization and robustness of the deep clustering method in cases where the test and training set are not matched. The best performance is achieved by HD$_\alpha$-MI, the MI head of the Chimera network. Interestingly, the performance of the DC head does not change significantly for the values of α tested. This suggests that joint training with the deep clustering objective allows the body of the network to learn a more powerful representation than using the mask-inference objective alone; this representation is then best exploited by the mask-inference head thanks to its signal approximation objective. Audio examples are available at (Luo).

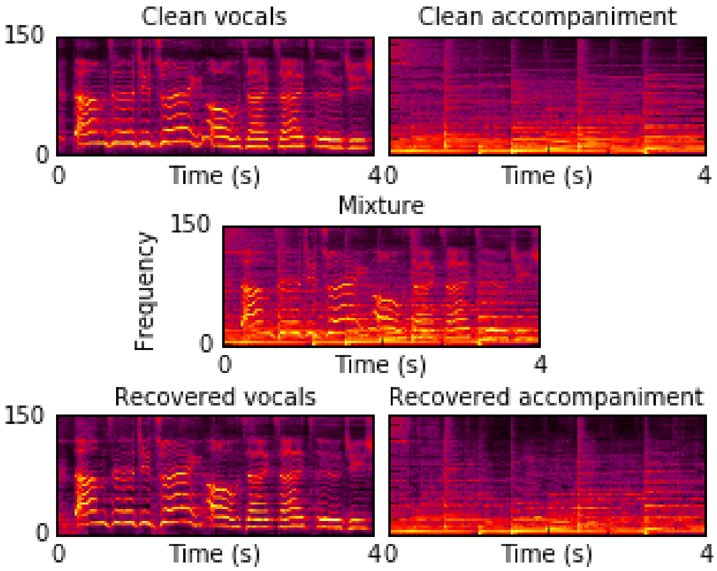


Figure 7.2: Example of separation results for a 4-second excerpt from file 45378_chorus in the public iKala dataset.

7.4 Neural decoding separation

In a natural auditory environment, most people can easily attend to a particular speaker out of many, and even switch their attention with ease(Bregman, 1994). However, this task can be extremely challenging for those suffering from hearing impairments, which in turn can lead to a substantial deterioration of ones mental health and wellbeing (Mackersie, 2003; Alain et al., 2006). According to the World Health Organization, approximately 5.3% of the worldwide population has a debilitating hearing impairment, and 33% of people over the age of 65 (Organization et al., 2013). While damage or age-related changes to the peripheral auditory system (e.g. the cochlea) can account for some of these deficits (Humes and Roberts, 1990; Abel et al., 2000), there is extensive evidence that a large proportion of hearing impairments are caused by a degradation of the neural mechanisms responsible for attending to speech, rather than simply an inability to process a weak and noisy signal from the ear (Mackersie, 2003; Alain et al., 2006). Therefore, simply amplifying all of the sounds in an environment is not enough to help such a user understand a conversation. Modern hearing aids can suppress certain types of background noise, and can even adapt automatically to the environment(Clark and Swanepoel, 2014). However, they do little to help a user attend to a single conversation among many, because they do not know which speaker a user is trying to attend to. Moreover, even if a device could detect the target speaker, they cannot enhance that voice because they cannot separate that speaker in a multi-speaker environment. One suggested solution has been to simply amplify the sound emanating from the direction in which the user is looking. However, this scheme breaks down in natural auditory environments: a user's gaze may shift in order to interact with various objects (e.g., while eating), or to observe something that may be the topic of conversation (e.g., at an art gallery)(Morla, 2011). Alternatively, a hand held controller could be used, with which a user could select a particular direction to amplify. However, such a device would be cumbersome and difficult to use effectively, which is contrary to user's wishes for a "wear and forget" device [8]. Instead, the optimal solution would be to automatically determine to whom a user is trying to attend by inferring it from their neural activity. Once established, it is necessary to selectively amplify that speaker while suppressing all others. Many studies over the past number of years have revealed a dynamic and selective representation of an attended speaker in auditory cortex (Ding and Simon, 2012b,a; Horton et al., 2013; Power et al., 2012; Golumbic et al., 2013; Mesgarani and Chang, 2012). More recently, it has been shown that is possible to decode attention over relatively short time-scales using non-invasive magnetoencephalography (MEG)(Ding and Simon, 2012a) and electroencephalography (EEG) data (O'Sullivan et al., 2015a; Horton et al., 2014). This has led to an upsurge in attention-decoding research, with many groups around the world attempting to refine and improve these findings [(Biesmans et al., 2016; O'Sullivan et al., 2015b; Aroudi et al., 2016; Ekin et al., 2016; Van Eyndhoven et al., 2016; Mirkovic et al., 2015; Dijkstra et al., 2015; Das et al., 2016). However, all of these attention-decoding studies have had access to the separated sound sources with which to infer whom the subject was attending to. This situation is obviously not realistic in real-world scenarios. The challenge therefore is to develop methods that can automatically separate each sound source in an environment, and use them to subsequently identify and amplify an attended speaker. Beamforming has been proposed as a possible solution to this problem (Van Eyndhoven et al., 2016; cocoha.org, 2016). This method can selectively enhance sounds in arbitrary directions by utilizing temporal delays in the recordings of multiple microphones in spatially separate locations. While successful in many regards, this approach is sub-optimal in three crucial aspects: (1) It requires multiple microphones, which are ideally placed as far apart as possible: an obvious limitation for a user wishing to wear a single hearing aid device. (2) With no prior knowledge of the environment, the location that should be amplified is unknown. (3) As it can only amplify a single direction in space, it fails when the target and interfering speakers are in the same location. This issue is confounded as both the user and target can move arbitrarily through space. What is required is a method for automatically segregating one speaker from another, regardless of their spatial location, and with a single microphone. Using a single acoustic channel to segregate a mixture of sound sources into their component parts is an incredibly difficult task; one which has yet to be fully resolved (Hershey et al., 2016b). In contrast, the human brain appears to perform this function admirably (Bregman, 1994). As such, there has been much research into emulating this ability over the past few decades. For example, computational auditory scene analysis (CASA) approaches attempt to emulate the assumed functions of the brain by grouping spectrotemporal features of sound based on similarity of onset/offset, pitch, and spectral envelope(Bach and Jordan, 2006; Krishnan et al., 2014). Other methods use model based approaches such as non-negative matrix factorization (Schmidt and Olsson, 2006). While these areas of research have proven successful in some situations, they often fail to generalize, and cannot operate in real-time. As a result, these methods have not been applied to the problem of decoding attention. Here, we address these issues by applying the state-of-the-art in single-channel speech separation algorithms to the attention-decoding platform. In a natural acoustic environment, there can be any number of interfering speakers and noise sources in any spatial location. However, here we limited the scope of this research to tackling one specific case that other methods such as beamforming fail to address: when a target speaker and an interfering speaker are overlapping in space. In order to separate a target speaker, we implemented a class of deep neural network (DNN) known as a long

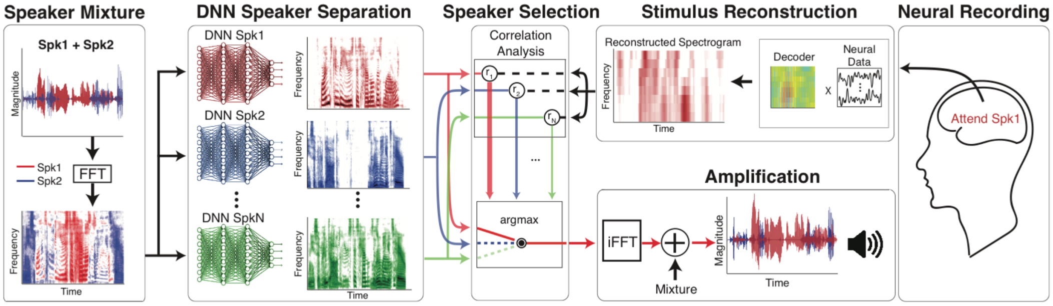


Figure 7.3: Two speakers, Spk1 (red) and Spk2 (blue), are mixed together into a single acoustic channel. In order to separate the speakers, a spectrogram of the mixture is first obtained (the two speakers have been marked red and blue for visualization purposes only). The spectrogram is then input to each of several DNNs, each trained to separate a specific speaker from a mixture. Simultaneously, a user is attending to one of the speakers (in this case, Spk1; red). A spectrogram of this speaker is reconstructed from the neural recordings of the user. This reconstruction is then compared with the outputs of each of the DNNs using a normalized correlation analysis in order to select the appropriate spectrogram, which is then converted into an acoustic waveform and added to the mixture so as to amplify the attended speaker

short-term memory recurrent neural network (LSTM-RNN). Each network is trained to separate one specific speaker from arbitrary mixtures of unknown interfering speakers.

Figure 7.3 illustrates a schematic of our proposed system. Two speakers, Spk1 (blue) and Spk2 (red), are mixed together. In order to separate the speakers, a spectrogram of the mixture is first obtained (see methods). This spectrogram is then fed to each of several DNNs (DNN Spk1 to DNN SpkN), each trained to separate a specific speaker from a mixture. Simultaneously, a user is attending to one of the speakers (in this case, Spk2; red). A spectrogram of this speaker is reconstructed from the neural recordings of the user (Mesgarani and Chang, 2012). This reconstruction is then compared with the outputs of each of the DNNs using a normalized correlation analysis (see methods) in order to select the appropriate spectrogram. This is then converted into an acoustic waveform and added to the mixture so as to amplify the attended speaker. For this study, we used invasive neural recordings, a methodology that allows for the rapid determination of a user's direction of attention (Mesgarani and Chang, 2012). However, previous studies have shown that this work could also be extended to non-invasive recordings (O'Sullivan et al., 2015a). This study provides the foundation for hearing aid devices that can automatically and dynamically track a user's direction of attention, and amplify an attended speaker.

7.5 Methods

7.5.1 Participants

6 patients who were undergoing clinical treatment for epilepsy took part in this study. All patients gave their written informed consent to partake in research. 5 patients were situated at North Shore Long Island Jewish (LIJ) hospital, and 1 patient was situated at the Columbia University Medical Center (CUMC). Two patients were implanted with high-density subdural electrode arrays over the left (language dominant) temporal lobe, with coverage over superior temporal gyrus (STG). These two patients will later be referred to as patients 3 and 5. The remaining 4 patients partook in stereotactic EEG (sEEG) in which they were implanted bilaterally with depth electrodes. This resulted in varying amounts of coverage over the left and right auditory cortices (STG and Heschls gyrus) for each patient.

7.5.2 Stimuli and Experiments

Each patient partook in two experiments for this study: a single-speaker (SS) and a multi-speaker (MS) experiment. The SS experiment was used as a control, in which each patient listened to 4 stories read by a female and male speaker (hereafter referred to as $Spk1_F$ and $Spk2_M$, respectively). This resulted in each patient listening to a total of 8 stories (4 stories twice). Both $Spk1_F$ and $Spk2_M$ were native American-English speakers, and were recorded in-house. In order to ensure the attentional engagement of each patient, the stories were randomly stopped, and the patient was instructed the repeat the last sentence. For the MS experiment, subjects were presented with a mixture of the same female and male speakers ($Spk1_F$ and $Spk2_M$), with no spatial separation between them. All stimuli were presented using a single Bose SoundLink Mini 2 speaker situated directly in front of the patient.

The MS experiment was divided into 4 blocks. Before each block, the patient was instructed to focus their attention on one speaker, and to ignore the other. All patients began the experiment by attending to the male speaker, and switched their attention to the alternate speaker on each subsequent block. In order to ensure that the patients were engaged in the task, the story was randomly paused and the patients were asked to repeat the last sentence of the attended speaker. The locations of the pauses were predetermined, and were the same for all patients. The patients were informed that the story would be paused, but were unaware of when the pauses would occur. The order in which the stories were presented was the same for each patient: The first block consisted of story 1 read by the male speaker mixed with story 2 read by the female speaker ($s1_m + s2_f$), the second block consisted of $s3_m + s4_f$, the third block consisted of $s4_m + s3_f$, and finally the fourth block consisted of $s2_m + s1_f$. In total, there were 11 minutes and 37 seconds of audio presented to each patient for the MS experiment. The SS experiment lasted twice as long, as each patient was required to listen to each story read by each speaker independently.

7.5.3 Data Preprocessing and Hardware

The patients at LIJ were recorded using Tucker Davis Technologies (TDT) hardware, and sampled at 2441Hz. The patient at CUMC was recorded using Xltek hardware, and sampled at 500Hz. All further processing steps were performed offline. All TDT data were resampled to 500Hz. A 2nd order Butterworth high-pass filter with a cut-off frequency at 1Hz was used to eliminate DC drift. Data were subsequently referenced using a local scheme, whereby the voltage at each electrode was referenced to the average of its immediately neighboring electrodes. Line noise at 60Hz and its harmonics (up to 240Hz) were removed using notch filters. A period of silence was recorded before the first (SS) experiment. All data were subsequently normalized by subtracting the mean and dividing by the standard deviation of this pre-stimulus period.

Data were then filtered into two frequency bands known to be responsive to speech (Golumbic et al., 2013): the high gamma band (70-150Hz), and the low frequency delta and theta bands (1-8Hz). The power of the high gamma band is known to be modulated by speech, in contrast to the phase of low frequency activity (Golumbic et al., 2013). In order to obtain the power of the high gamma activity, the data were first filtered into 8 frequency bands between 70 and 150Hz, each with a bandwidth of 10Hz. The power (analytic amplitude) in each band was then obtained using a Hilbert transform. We took the average of all 8 frequency bands as the total high gamma power. To obtain the low-frequency phase, we simply applied a low-pass filter to the data with a corner frequency at 8Hz (since we had already applied a high-pass filter at 1Hz in the preprocessing stage). We will hereafter refer to these two frequency bands as HGP (high gamma power), and LFP (low-frequency phase). It is important to clarify that we did not obtain the instantaneous phase of the data using a Hilbert transform; rather we are simply emphasizing the distinction between using the raw waveform of the low-frequency data, versus obtaining the power of the data in the high-gamma band.

7.5.4 Single-Channel Speaker Separation

In order to automatically separate each speaker from the mixture, we employed a method of single-channel speech separation that utilizes deep neural networks (DNNs)(Weninger et al., 2015). Each DNN was trained to separate one specific speaker from arbitrary mixtures. In our experiment, there were only two speakers ($Spk1_F$ and $Spk2_M$) presented to each patient. However, we are proposing a system that could work in a real-world situation, where a device would contain multiple DNNs, each trained to separate specific speakers, any of whom may or may not be present in the environment. Because of this, we trained 4 DNNs to separate 4 specific speakers: two female and two male, hereafter referred to as $Spk1_F$, $Spk2_M$, $Spk3_F$, and $Spk4_M$. This allowed us to test what would happen if a user had a device with 4 DNNs, only two of which were trained to separate the speakers that were actually present in the mixture. All speakers were native American-English speakers. As stated before, two of the speakers ($Spk1_F$ and $Spk2_M$) were recorded in-house. However, $Spk3_F$ and $Spk4_M$ were taken from the Wall Street Journal (WSJ) corpus (Paul and Baker, 1992).

In order to be utilized by the DNNs, the speech waveforms (sampled at 16kHz) were converted into 100-dimensional Mel-frequency spectrograms. The goal is then to obtain an estimation of a target spectrogram S from a mixture M. To do so, a soft mask $\hat{Y}$ is learnt that is applied to the mixture, so as to mask the interfering speech. To learn the mask, we used a type of DNN referred to as a long short-term memory recurrent neural network (LSTM-RNN). The difference between the masked spectrogram and the clean target spectrogram was treated as the error in order to generate the gradient that was back propagated into the network to update the parameters. The objective function is shown in 7.3:

$$E(\hat{Y}) = ||\hat{Y}M - S||^2 \tag{7.3}$$

The input to the DNNs was the logarithm (base 10) of the spectrograms, normalized so that each frequency band had zero mean and unit variance. Each network contained 4 layers with 300 nodes each, followed by a single layer containing 100 nodes with logistic activation in order to output a spectrogram. An acoustic waveform was generated from the output of the DNN by obtaining the magnitude of the complex spectrogram in combination with the phase of the original mixture. See (Weninger et al., 2014b) for further information.

For training, twenty minutes of speech from the target speakers was used, and ∼5 hours of speech from 103 interfering speakers was taken from the WSJ corpus. The target speaker was always mixed with one interfering speaker, and both were mixed into the same channel and with the same root mean squared (RMS) intensity. Unseen utterances were used for testing (for both the target and interfering speakers). It is important to clarify that the networks never saw any of the other target speakers during training. E.g., the network trained to separate $Spk1_F$ never saw $Spk2_M$, $Spk3_F$, or $Spk4_M$.

7.5.5 Stimulus-Reconstruction

In order to determine the attended speaker, we employed a method known as stimulus-reconstruction (Ding and Simon, 2012a; O'Sullivan et al., 2015a; Mesgarani and Chang, 2012; Mesgarani et al., 2009). This method applies a spatiotemporal filter (decoder) to neural recordings in order to reconstruct an estimate of the spectrogram of the attended speaker for each patient. The 100-dimensional log Mel-frequency spectrograms were downsampled by a factor of 10, resulting in 10 frequency bands. Each decoder was trained using the data from the single-speaker (SS) experiment only. Electrodes were chosen if they were significantly more responsive to speech than to silence. For the high frequency (HGP) data, this meant that the average power during speech was greater than that during silence. However, for the low frequency (LFP) data, electrodes were chosen if the average power was either significantly greater or lower than that during silence. In order to perform statistical analyses, the data were segmented into 500ms chunks, and divided into two categories; speech and silence. Significance was determined using an unpaired t-test (false discovery rate (FDR) corrected, $q<0.05$). This resulted in varying amounts of retained electrodes for each frequency band, and for each patient (see Table 7.3). The decoders were trained using time-lags from -400 to 0 ms, which is causal due to the fact that stimulus-reconstruction is a reverse mapping from the neural data back to the stimulus. See (Mesgarani et al., 2009) for further information on the stimulus-reconstruction algorithm.

7.5.6 Neural Correlation Analysis

As stated before, we trained decoders using the data from the SS experiment. These same decoders could then be used to reconstruct spectrograms from the MS experiment (Mesgarani and Chang, 2012). Determining to whom the patient is attending requires a correlation

Table 7.3: The number of electrodes retained for each frequency band and each patient. Also shown is the number of electrodes common between the HG and LF bands

Frequency Band	Patient					
	1	2	3	4	5	6
HG	32	11	84	51	72	30
LF	31	19	69	47	56	37
HG	29	7	59	40	41	28

analysis, commonly using Pearson's r-value (Mesgarani and Chang, 2012). Because there were 10 frequency bands in the downsampled spectrograms, all correlation values reported are the average of the r-values obtained across frequency. However, we excluded the lowest two frequency bands (50 $\sim$ 200Hz) in order to exclude any bias towards the male speaker, as the pitch of the male occupied this region.

Typically, whichever spectrogram has the largest correlation with the reconstructed spectrogram is taken to be the attended speaker (O'Sullivan et al., 2015a). However, because we are using 4 networks, each trained to separate a speaker that may or may not be present in the mixture, the analysis becomes slightly more complex. Crucially, it was necessary to normalize the correlation values with respect to the mixture, because the correlation between the reconstructed spectrograms and the mixture was very large. For clarity, we will first define some terminology: a spectrogram outputted from the kth DNN will be referred to as S_{DNN_k}, the spectrogram of the mixture as S_{MIX} and the reconstructed spectrogram (from neural responses) as S_{RECON} . In order to emphasize large correlations, we applied a Fisher transformation (inverse hyperbolic tangent) to each r-value.

The normalization procedure involved five steps. First, we obtained the correlation between S_{RECON} and each S_{DNN_k}, which we will refer to as ρ_{1_k} 7.4

$$\rho_{1_k} = \operatorname{arctanh}\{r(S_{RECON}, S_{DNN_k})\} \tag{7.4}$$

where $r(x, y)$ is Pearson's correlation between the variables and , and is the inverse hyperbolic tangent function.

Next, we obtained the correlation between S_{RECON} and the difference between S_{DNN_k} and S_{MIX}, which we will refer to as ρ_{2_k} 7.5.

$$\rho_{2_k} = \operatorname{arctanh}\{r(S_{RECON}, S_{MIX} - S_{DNN_k})\} \tag{7.5}$$

Intuitively, this value should be close to zero if a network is outputting the mixture, negative if a network is correctly separating the attended speaker, and positive if it separates the unattended speaker. Therefore, taking the difference of ρ_{1_k} and ρ_{2_k}, and dividing by their sum, should produce a score α_k that can differentiate between each of these cases7.6.

$$\alpha_k = \frac{\rho_{1_k} - \rho_{2_k}}{\rho_{1_k} + \rho_{1_k}} \tag{7.6}$$

This is followed by a test-normalization(t-norm)(Lotia and Khan, 2012), in which the α score for each network is normalized relative to the distribution of *alpha* scores from all networks7.7, where μ_α and σ_α are the mean and standard deviation of the distribution of α scores from all networks. Finally, we subtract the correlation between S_{DNN_k} and S_{MIX}, and add the constant 1, resulting in the final normalized correlation value P_k for each network7.8.

$$\beta_k = \frac{\alpha_k - \mu_\alpha}{\sigma_\alpha} \tag{7.7}$$

$$P_k = \beta_k - \operatorname{arctanh}\{r(S_{MIX}, S_{DNN_k})\} + 1 \tag{7.8}$$

This last normalization step will further penalize a network that is simply outputting the mixture, rather than separating the speakers, and is independent of the neural data. This could occur if a network's trained speaker was not in the mixture. The addition of the constant 1 is simply used in order to make the final result more intuitive, as the values would typically be less than zero otherwise.

7.5.7 DNN Correlation Analysis

In our experiment, there were only two speakers present in the mixture ($Spk1_F$ and $Spk2_M$). To ensure generalization, we trained 2 additional networks to separate 2 different speakers, one male and one female ($Spk3_F$ and $Spk4_M$). We wanted to test how each neural network behaved when given various mixtures. To do this, we created a data set consisting of 103 random speakers (taken from the WSJ corpus) mixed with target speakers $Spk2_M$, $Spk4_M$, $Spk1_F$, and $Spk3_F$, as well as another two male and female speakers. We created 200 mixtures for each target speaker, resulting in 1200 mixtures in total. We fed every mixture through each of the 4 networks, and tested how well each network separated the target speaker in each case, by obtaining a correlation coefficient (Pearson's r-value) between the output of the network and the spectrogram of the clean target speaker. Reported r-values are averaged across frequency.

We also obtained normalized correlation values, calculated in the same way as in the previous section. The only difference is that in equations 7.4 and 7.5, the reconstructed spectrogram S_{RECON} is replaced with the spectrogram of the clean target speaker.

In order to simulate a dynamic scenario in which a patient was switching attention between the two speakers, we artificially divided and concatenated the neural data into several consecutive segments in which the patients were attending to either speaker. Specifically, we divided the data into 10 segments, each lasting 60 seconds. The patients attended to the male speaker for the first segment. To assess our ability to track the attentional focus of each patient, we used a sliding window approach, whereby we obtained the normalized correlation values every second. We used various window sizes ranging from 5 to 30 seconds (5 second increments; 6 window sizes in total). Larger window sizes should lead to more consistent (less noisy) correlation values, and therefore provide a better estimate of the attended speaker. However, they should also be slower at detecting a switch in attention.

7.5.8 Objective Measurement

In order to determine the efficacy of the speech-separation algorithm we employed in separating each speaker, we used a commonly used objective measure of speech quality known as the Perceptual Evaluation of Speech Quality (PESQ) score (Rix et al., 2001). The PESQ algorithm produces a score between 1.0 and 4.5, with high values indicating better quality. This score is known to correlate well with subjective listening tests, and has proven to be the most reliable objective measure for the assessment of speech quality (Loizou, 2011), and to some degree, speech intelligibility (Ma et al., 2009).

In order to obtain a measure of our ability to determine the attended speaker from neural recordings, we first segmented the reconstructed spectrogram from the MS experiment into 20-second bins, resulting in 34 alternating segments (17 where the patients attended to male speaker, and 17 to the female speaker). As mentioned before, we had trained 4 DNNs to separate two female ($Spk1_F$ and $Spk3_F$) and two male ($Spk2_M$ and $Spk4_M$) speakers from random mixtures. Therefore, we obtained 4 normalized correlation values for each segment: P_f1, P_m1, P_f2 and P_m2 . Because $Spk1_F$ and $Spk2_M$ were the only speakers that were actually presented to the patient, we would expect that and would be the largest, depending on whom the patient was attending to. Therefore, we considered a segment to be correctly decoded if was largest when the patient was attending to $Spk1_F$, and if was largest when the patient was attending to $Spk2_M$. We define decoding-accuracy as the percentage of segments that were correctly decoded. Because there were 4 networks, chance performance is 25%. We determined that a decoding-accuracy of 41% is significantly above chance (14 out of 34 segments correctly decoded), based on a binomial test at the 5% significance level ($P(X \geq 14) = 0.028$). For comparison, we also determined the decoding-accuracy that would be achieved if we had access to the ideal spectrograms of $Spk1_F$ and $Spk2_M$.

7.5.9 Dynamic Switching of Attention

In order to simulate a dynamic scenario in which a patient was switching attention, we artificially divided and concatenated the neural data into consecutive segments in which the patients were attending to either speaker. Specifically, we divided the data into 10 segments, each lasting 60 seconds. The patients attended to the male speaker for the first segment. To assess our ability to track the attentional focus of each patient, we used a sliding window approach whereby we obtained normalized correlation values every second. We used window sizes ranging from 5 to 30 seconds (in 5 second increments for 6 window sizes in total). Larger windows should lead to more consistent (less noisy) correlation values and provide a better estimate of the attended speaker. However, they should also be slower at detecting a switch in attention.

7.5.10 Psychoacoustic Experiment

Although the PESQ score is known to be a reliable measure of speech quality (Loizou, 2011), we still wanted to test if users would actually prefer to use our proposed system in a multi-speaker scenario, when those users were presented with the same speakers as the patients in this study. To do so, we performed a psychoacoustic experiment on healthy controls. X subjects took part (Y female), aged between 20 and 28 years (mean ± SD, 22 ± 2.5). All subjects reported normal hearing, and provided written informed consent. The stimuli used for this experiment were the same as those used for the neural experiment, in that subjects were always presented with a mixture of $Spk1_F$ and $Spk2_M$. However, the manner in which the stimuli were presented was altered so as to obtain as much information as possible about the subjects' perception. The experiment was divided into four blocks, each containing 15 trials. Each trial consisted of a single sentence. Before each block, the subjects were instructed to pay attention to one of the speakers, starting with the male, and switching attention on each successive block. After each trial (sentence) the subjects were presented with a transcription of the sentence of the attended speaker, but with one word missing. Subjects were instructed to type the missing word. They also had to indicate the difficulty with which it was to understand the attended speaker on a scale from 1 to 5: very difficult, difficult, not difficult, easy, and very easy. This allowed us to obtain both an objective measure of intelligibility, and a subjective measure of listening effort, with the latter being equivalent to the Mean Opinion Score (MOS)(Rothauser et al., 1969). For half of the experiment, both speakers were presented at the same RMS power. For the other half, we attempted to amplify the attended speaker. The order in which this occurred was counterbalanced across subjects. In total, subjects were presented with 4 minutes and 11 seconds of audio, and the experiment lasted approximately 10 minutes.

In order to amplify the attended speaker, we decided not to perform a simulation; i.e., choosing when to perform the amplification. Instead, we wanted to use real neural data, in order to demonstrate how the overall system could be implemented. We elected to use the neural data from patient 1. In order to dynamically track the attentional focus of the patient, we implemented a strategy similar to the artificial switching of attention section discussed earlier. I.e., we used a sliding window approach, in which we attempted to decode the attentional focus of the patient every second. We decided to use the LFP data, as opposed to the HGP data, in order to be as comparable as possible with non-invasive technologies, which typically only have access to low frequency neural activity (O'Sullivan et al., 2015a). We also chose to use a window-size of 20 seconds, in order to be consistent with our decoding strategy discussed earlier. Whenever we could correctly classify the attended speaker from the neural data for that patient, we presented the output from the correct DNN added to the mixture. However, if a mistake was made, and we misclassified the patient's attentional focus, we would present the output from whichever DNN produced the largest normalized correlation. The DNN output was added at a level of +12dB relative to the mixture.

Subjects were informed before the experiment that they would have to report which half of the experiment required less effort to understand the attended speaker, and they were reminded half way through. At the end of the experiment, after they had reported their preference, they were then asked one final question: "For the (1st/2nd) half of this experiment, a system was turned on, that tried to amplify the attended speaker. It is not perfect, and may have sometimes amplified the incorrect speaker. Which would you prefer: to have this system turned on or off"

7.6 Results

7.6.1 DNN Correlation Analysis

Figure 7.4 displays the results of the DNN correlation analysis. All statistical analyses were performed using Wilcoxon rank-sum tests. The left figure in 7.4 displays the results when each DNN was presented with a mixture of a random speaker and the designated target speaker that the DNN was trained to separate. On the left, each boxplot shows the correlation between the spectrograms that were outputted from a DNN and the clean target spectrograms (averaged across frequency). On the right, each boxplot shows the correlation of the DNN outputs with the raw mixture. The top of the figure displays the results when a DNN is trained to separate a female speaker from a mixture, and the bottom when a DNN is trained to separate a male speaker (labeled DNN Gender: male/female). Because of the characteristic differences between male and female speakers (e.g., pitch, spectral envelope), it is typically more difficult to separate two speakers of the same gender. This is illustrated by dividing the results into two categories: when the interfering (masker) speaker was male or female. As can be seen, the DNNs perform slightly better when separating the target speaker from an interfering speaker of the opposite gender. However, this difference was only significant for the female DNNs (female DNNs, $p \gg 0.001$; male DNNs, $p = 0.11$). We also tested the behavior of the DNNs when the designated target speaker was not present in the mixture (The right figure in 7.4, Undesignated Target Speaker). One would expect that the DNNs would fail to separate the target speaker in this situation, and this was indeed the case. However, each DNN was slightly better at separating an undesignated target speaker when that speaker was the same gender as the speaker that the DNN was trained on. I.e., for a female DNN, the correlations for an undesignated female speaker were significantly larger than for an undesignated male speaker ($p \gg 0.001$). Similarly, for a male DNN, the correlations for an undesignated male speaker were significantly larger than for an undesignated female speaker ($p \gg 0.001$).

7.6.2 Neural Correlation Analysis

The neural correlation analysis was used to determine to whom the patients were attending, by comparing the reconstructed spectrograms (from the neural data) with the output of each DNN. A in figure 7.5 illustrates the results of the neural correlation analysis. The top of the figure (raw r-values) shows the average correlation between the reconstructed spectrograms and the outputs of the networks for each subject (where each subject is represented by a colored dot). Because the patients alternated their attention between two speakers, the r-values labeled as attended and unattended come from the DNNs that were trained on $Spk1_F$ and $Spk2_M$. The r-values that are labelled as untrained come from the DNNs that were trained on $Spk3_F$ and $Spk4_M$. Therefore, untrained in this sense simply means that the networks were not trained to separate either of the speakers that the patients actually listened to. Although the attended r-values are typically larger than the unattended r-values, there is also a very large correlation with the mixture, and therefore with the networks that were not trained on $Spk1_F$ and $Spk2_M$ (labeled untrained). This is because these networks typically outputted spectrograms that were very close to the mixture. In order to take this into account, we normalized the r-values with respect to the mixture. The bottom of the

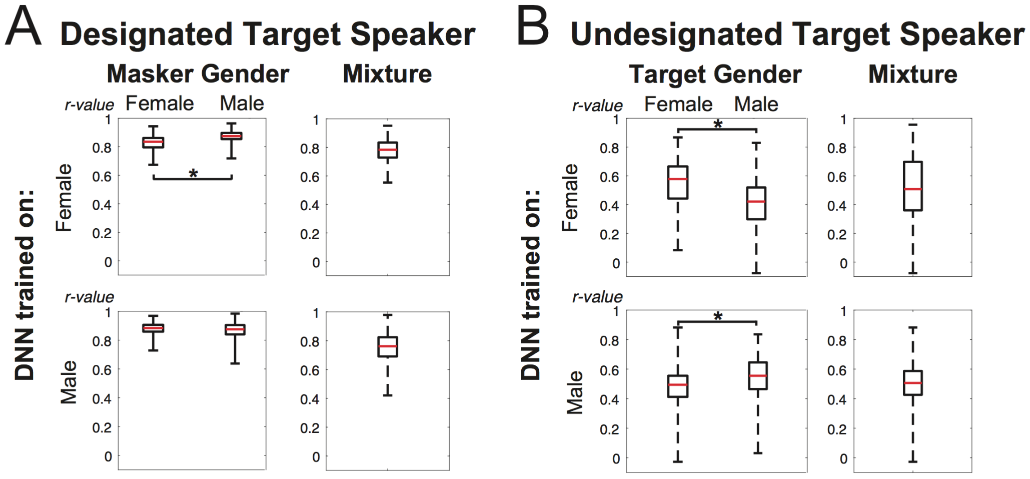


Figure 7.4: To test the performance of the DNNs at single-channel speaker-separation, we created multiple mixtures of random speakers with 4 target speakers (2 male and 2 female) and passed every mixture through each of the 4 DNNs (each trained to separate one of the target speakers). Performance was measured by obtaining the correlation (r-value) between the output of each DNN and the spectrogram of the clean target speaker. The resulting r-values are displayed as a series of boxplots. (A) The results when a DNN is presented with a mixture containing the speaker that it was trained to separate (designated target speaker). Results are separated into instances when the DNN was trained to separate a female (top panel) or a male (bottom panel) speaker. Results are further separated into when the interfering (masker) speaker was female (left boxplots) or male (right boxplots). In addition, we display the correlations of the DNN outputs with the raw mixture (right panels). (B) The results when a DNN is presented with a mixture containing a speaker that it was not trained to separate (undesignated target speaker). Results are displayed similarly to before, with DNNs trained to separate female and male speakers displayed on the top and bottom panels, respectively. However, rather than showing the effect of the masker speaker's gender, we show the effect of the target speaker's gender. These results show that a DNN is better at separating an undesignated target speaker whose gender matches the speaker that it was trained on.

figure (normalized r-values) shows the results of this analysis. Applying this normalization method caused the attended values to be far larger than either the unattended or untrained values.

7.6.3 Decoding-Accuracy

Given that the normalized correlation values could differentiate between attended, unattended, and untrained networks, it was then possible to decode the attentional focus of the patients. After segmenting the data into 20 second chunks, we would consider a segment to be correctly decoded if the normalized correlation between the reconstructed spectrograms and the output of the DNN that was trained to separate the attended speaker was larger than all other DNN outputs. Figure 7.5 shows the results of this analysis. We define decoding-accuracy as the percentage of segments that were correctly decoded. We considered the attentional focus of a patient to be successfully decoded if the decoding-accuracy for both the male and female speakers was greater than the 41% significance threshold (although chance performance is 25%, significant performance was calculated to be 41% based on a binomial test; see methods). Out of the 6 patients who participated in this study, the attentional focus of 4 patients (1,3,4 and 5) could be decoded using HGP data, and 3 patients (1,3 and 5) using LFP data. Notably, the decoding-accuracy achieved when using the ideal spectrograms was identical.

7.6.4 Artificial Switching of Attention

In order to simulate a dynamic situation where the patient was alternating their attention between the two speakers, we artificially segmented and concatenated the data into 60-second chunks, with the patient's attention switching at the beginning of each chunk. Figure 7.6 shows the results of the artificial switching of attention for an example subject (patient 1) using LF data and a 20 second window size. Each solid black line marks a switch in attention. The shaded blue and red bars at the top indicate segments where the patient was attending to $Spk2_M$ and $Spk1_F$, respectively. Plotted beneath are the normalized correlation values for each of the 4 DNNs. Ideally, the blue ($P2_m$) and red ($P1_f$) lines would alternate in having the largest values, and the cyan ($P4_m$) and magenta ($P3_f$) lines would be close to zero.

A in Figure 7.7 displays the same results, but averaged over all sections when the patient was attending to $Spk2_m$ (-60s $\sim$ 0s) and $Spk1f$ (0s $\sim$ 60s). On the left of the figure (Ideal), the blue and red lines show results when using ideal spectrograms for $Spk2_M$ and $Spk1_F$, respectively. On the right (DNN Output), the blue and red lines illustrate the results using the outputs of the DNNs. Shaded regions denote standard error. As can be seen, the DNN outputs normalized correlation values that are close to the ideal scenario. B figure in figure 7.7 shows the effect of the window size on decoding-accuracy and transition time (how long it takes to detect a switch in attention) for each subject whose attentional focus we could decode. These results indicate that there is an optimal window-size for decoding-accuracy, which is about 20 seconds in this case. However, as expected, the transition time monotonically increases as the window size increases. The transition times were calculated as the time at which the blue ($Spk2_M$) and red (Spk1F) lines intersect in the averaged data

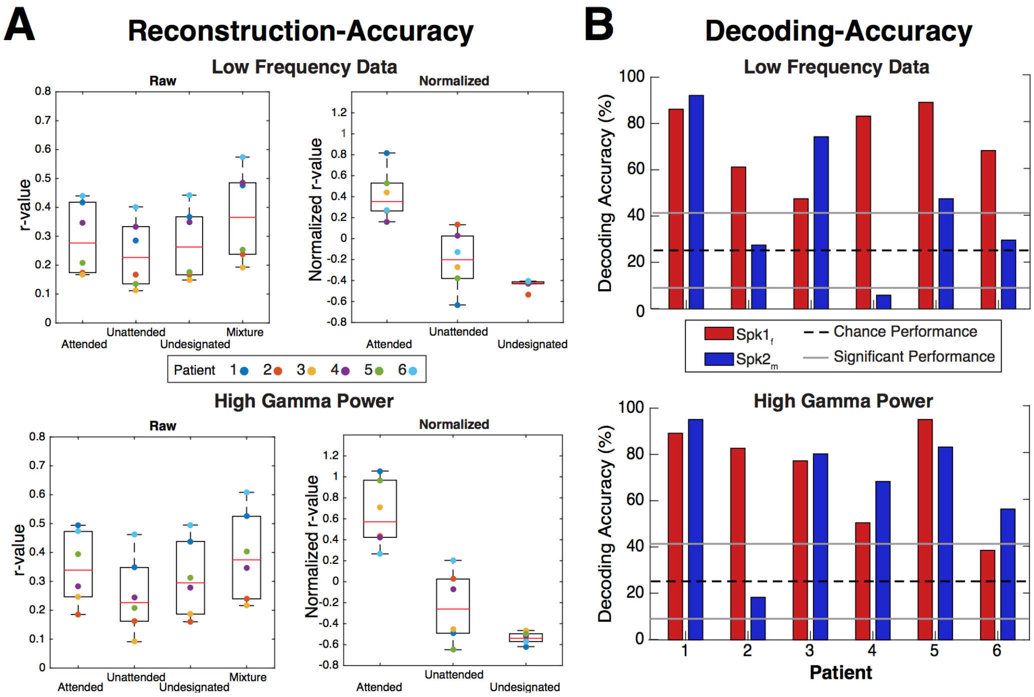


Figure 7.5: (A) The correlations between the reconstructed spectrograms (from neural data) and the outputs of the DNNs. Top panels show results when using LF data, and bottom panels when using HG data. Left panels show the raw r- values, and right panels show normalized r-values (see methods). Each subject is represented by a colored dot. Because the patients alternated their attention between two speakers, the r-values labeled as attended and unattended come from the DNNs trained on $Spk1_F$ and $Spk2_M$, whereas the r-values labeled as undesignated come from the DNNs that were trained on $Spk3_F$ and $Spk4_M$. Therefore, undesignated in this sense means that the DNNs were not trained to separate either of the speakers that the patients actually listened to. (B) Decoding-Accuracy: the percentage of segments (20s) in which the attentional focus of each patient could be correctly determined using LF data (top panel) and HG data (bottom panel). Results are separated according to which speaker (Spk1f or Spk2m) was being attended to. Because there were outputs from 4 DNNs chance performance is 25% and significant performance was determined to be 41% based on a binomial test at the 5% significance level.

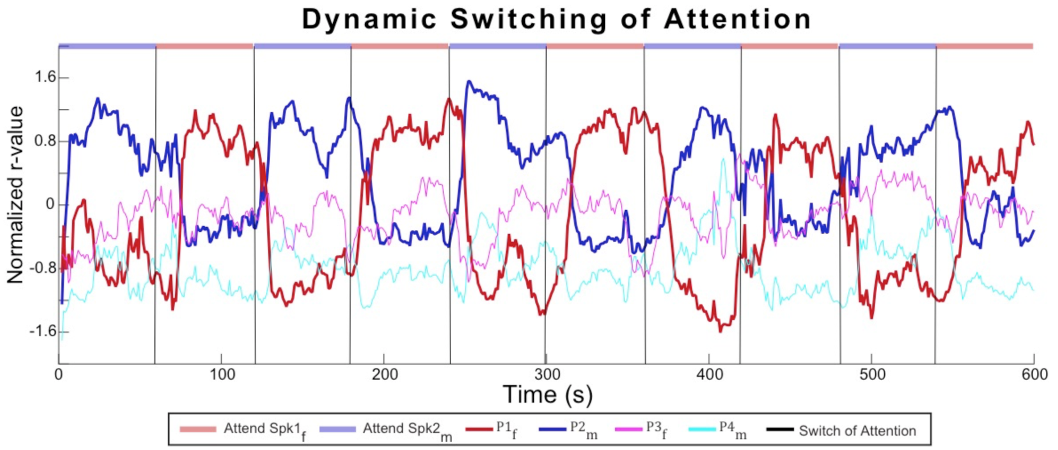


Figure 7.6: Data were segmented into 60-second chunks and concatenated into consecutive blocks that alternated with regards to the speaker being attending to. The results shown here are for an example patient (patient 1) using LF data. Black lines indicate a switch in attention and the colored bar on top indicates the speaker being attended to (Red: $Spk1_f$. Blue: $Spk2_m$). We used a sliding window (20s width) to obtain normalized r-values every second for each of the 4 DNNs. Ideally, $P2_m$ (blue) and $P1_f$ (red) would alternate in being the largest (corresponding to the speaker being attended), and $P3_f$ (magenta) and $P4_m$ (cyan) would be smallest.

(e.g., Figure 7.7). C and D in figure 7.7 show the same results using the HG data. For patient 1, the results using HG data are very similar to when using LF data. However, for the other patients, HG data provides a better decoding-accuracy than LF data.

7.6.5 Objective Measure of Speech Separation Quality

Using the Perceptual Evaluation of Speech Quality (PESQ) score, the speaker separation algorithm used in this study produced a speech signal that was objectively cleaner: the raw mixture produced scores of (mean ± SD) 1.91±0.13 and 1.80±0.11 for the female and male networks, respectively, whereas the separation algorithm produced scores of 2.60 ± 0.11 and 2.59 ± 0.08 for the female and male networks, respectively, which is a 41% increase on average.

7.6.6 Psychoacoustic Experiment

To test whether users would actually prefer to use our proposed system in a multi-speaker scenario, we performed a psychoacoustic experiment (see methods). Figure 7.8 displays these results. In the missing word task, there was no significant difference in the numbers of words that were correctly reported when the system was on versus off (A in Figure 7.8; Wilcoxon

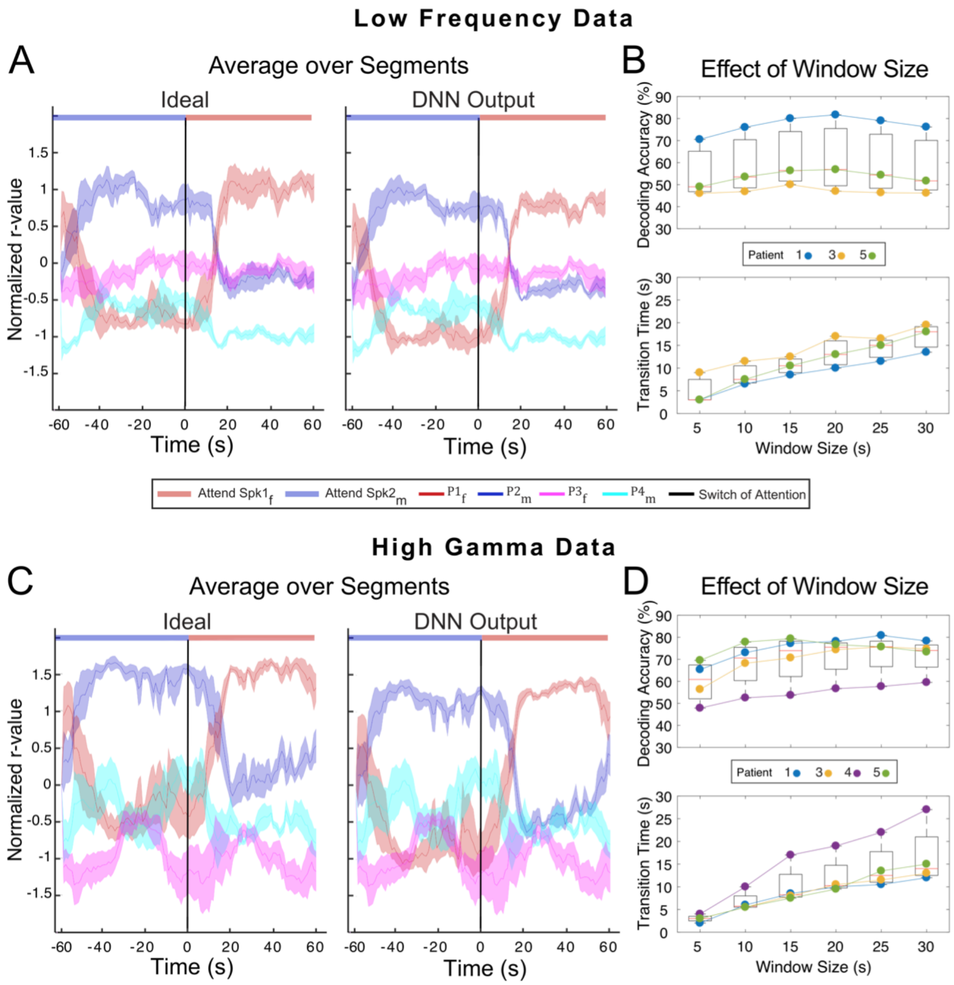


Figure 7.7: (A) Average over segments. Here we show the same data as in Figure 5, but with the average of all segments in which the male speaker was attended on the left of the black line, and the segments in which the female speaker was attended on the right. Data displayed is from the same example patient as in Figure 5 (patient 1). The left panel displays the results that would be obtained using the ideal (clean) spectrograms of each target speaker, and the right panel displays the results obtained using the outputs of the DNNs. We plot the values for P3f (magenta) and P4m (cyan) along with the ideal values for visualization purposes. (B) Effect of Window Size. The data in Figure 5 and in Figure 6A were obtained using a window-size of 20s. Here we display the decoding-accuracies and transition-times obtained using a range of window-sizes, and for each patient whose attention we could decode (patients 1, 3 and 5). (C-D) The same results as in A and B but when using HG data. The attention of one additional patient (patient 4) could be decoded using HG data, as is shown in D.

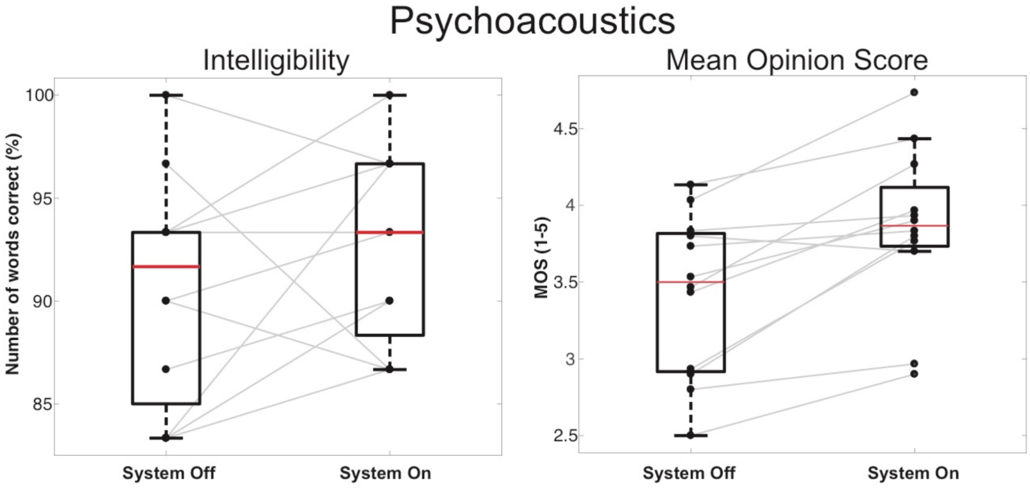


Figure 7.8: The left panel displays the intelligibility results (the number of missing words correctly reported) when the system was off (left) and on (right). There was no significant difference between the two cases. The right panel displays the mean opinion score (MOS) when the system was off (left) and on (right). There was a significant increase in the MOS when the system was on (Right tailed Wilcoxon signed-rank test, p = 9.7×10^{-4}).

signed-rank test, p = 0.09). This is because performance was close to ceiling for both conditions (93% for system on and 91% for system off). Therefore, it can be concluded that the intelligibility did not change when the system was on, which is a known phenomenon in speech enhancement research. However, all but one subject reported reduced listening effort when the system was on, with a median MOS of 3.87 (25th percentile, 3.73; 75th percentile, 4.12) versus a median MOS of 3.5 (25th percentile, 2.9; 75th percentile, 3.8) when the system was off (B in Figure 7.8). This was a significant increase in MOS (Wilcoxon signed-rank test, p = 9.7×10^{-4}). The one subject who reported no reduction in listening effort had almost identical MOS scores for both system on (3.7) and system off (3.8). Furthermore, the majority of subjects reported a preference for the segments in which the system was turned on (9 of 12), and the majority reported a preference for using the system once they were informed of its nature (10 of 12).

7.7 Discussion

We have presented a system that incorporates the latest automatic speech-separation algorithms into the neural attention-decoding platform, which is an important step towards developing cognitively controlled hearing aids. We show that we can automatically separate specific speakers from arbitrary mixtures, and use the separated signal to decode the attentional focus of a user. Moreover, we have shown that using the separated sound to amplify the attended speaker results in less effort on the behalf of listeners. This work provides the foundation for realistic hearing aid devices that can automatically and dynamically track a user's direction of attention, and amplify the attended speaker. Although the patients were only presented with mixtures of $Spk1_F$ and $Spk2_M$, we have demonstrated that this work can be extended to a more general case, because none of the networks ever saw any of the other target speakers during training. I.e., the network trained to separate $Spk1_F$ had never seen $Spk2_M$. However, a limitation with our proposed system is the fact that it requires pre-training on designated speakers, and therefore cannot generalize to unseen speakers. In addition, hardware constraints could limit the number of networks that could be housed inside a portable hearing aid. (However, modern hearing aids are capable of performing off-board computing by interfacing with a cell phone. (Clark and Swanepoel, 2014)) Furthermore, DNNs rely heavily on the data used to train them. Therefore, additional training would be required to separate speakers under different environmental conditions (Li et al., 2014). In addition, people tend to involuntarily speak louder in noisy situations, which not only effects loudness but also other acoustic features such as pitch, rate and the duration of syllables (the Lombard effect (Brumm and Zollinger, 2011)). This would also need to be taken into account during training of the networks. One potential solution is to use a new class of speech-separation algorithms known as Deep Clustering (Hershey et al., 2015). This approach does not require pre-training on specific speakers, and even shows promise in separating multiple speakers. However, it is currently not applicable in real-time. Future work will aim to see if it is possible to incorporate this class of algorithms into the attention-decoding platform.

7.7.1 Decoding-accuracy

Because of the large correlation between the reconstructed spectrograms and the mixture, it was crucial to normalize the r-values. Omitting this procedure resulted in almost no ability to decode the attentional focus of each patient. This is an important finding given that most research in this area typically only considers the attended and unattended spectrograms. The normalization procedure involved two key steps: (i) incorporating the mixture into the correlation analysis (equations 7.5 and 7.6) and (ii) performing a test-normalization (t-norm) to equalize the outputs of each DNN with respect to each other. The final normalization step (subtracting the correlation between the DNN output and the mixture; equation 7.8 was not as crucial, but we found that it did provide an improvement in decoding-accuracy. We empirically chose to use a 20-second window-size in order to obtain a measure of decoding-accuracy. Different window-sizes would of course affect decoding-accuracy, as evidenced by the changing decoding-accuracy in Figure 4B for the switching of attention simulation. However, for this analysis, we wanted to simply obtain a baseline decoding-accuracy for each subject in order to determine if his or her attentional state could be ascertained at all. Interestingly, the decoding-accuracy achieved using the ideal spectrograms was identical to what was achievable using the DNN outputs. What is important for decoding-accuracy is that the values for $Spk2_M$ and $Spk1_F$ far exceed those of $Spk3_F$ and $Spk4_M$. Therefore, although the values obtained using the ideal spectrograms are slightly higher than those obtained using the DNNs, this increase has no effect on decoding-accuracy because the DNN correlations for $Spk1_F$ and $Spk2_M$ are sufficiently large relative to $Spk3_F$ and $Spk4_M$. Some patients showed a bias towards one speaker (e.g., patient 2). This is likely due to electrode coverage issues; not all electrodes that are responsive to speech are modulated by attention (Golumbic et al., 2013), and could simply be showing a preference for features that are characteristic to a particular speaker. Therefore, although it appears that we can decode the attentional state of patient 2 when that patient is attending to the female speaker, it is probable that the reconstructed spectrograms simply represented the female speaker alone, regardless of the attentional focus of the patient. There is also a dichotomy between the HGP and LFP data for some patients (e.g., patient 4), where the attentional focus can be determined using HGP data, but not LFP, and indeed a large bias toward the female speaker is apparent using LFP data in this case. This is because low frequency and high frequency data are not necessarily representative of the same neural generators (Golumbic et al., 2013; Buzsáki et al., 2012). Indeed, there was a large disparity between the two frequency bands with regards to the number of electrodes that were responsive to speech (see Table 7.3). Therefore, it's possible that the electrodes that were responsive to speech in the high gamma band were modulated by attention, but that those same electrodes either weren't modulated by attention in the low frequency band, or were not responsive to speech at all.

7.7.2 Artificial Switching of Attention

In a real-world situation, it is likely that a user would want to dynamically switch their attention between multiple speakers as a conversation progresses. Although the patients in our study did alternate their attention between two speakers, they did not do so in a dynamic fashion; indeed, there was a substantial break in between each block of the experiment. When simulating switching of attention (Figures 7.5 and 7.6), it is clear that the window-size has an effect on both the decoding-accuracy and the transition-time (the time it takes to detect a switch in attention). As can be seen, there is an optimal window-size for decoding-accuracy, which in this case is approximately 20 seconds. Shorter window-sizes produce r-values that are too noisy, and longer window-sizes prohibit the rapid detection of switches in attention. However, there is a clear linear trend in transition time: the longer the window-size, the longer it takes to detect switches in attention. It is possible that more elaborate models that use shorter window-sizes but that smooth the resulting correlation values could provide a better trade off between decoding-accuracy and transition-time detection (Akram et al., 2014).

7.7.3 Psychoacoustics

One important characteristic of speech enhancement techniques is to ensure that the resulting speech is not distorted or corrupted, as users often prefer no enhancement over an amplified but distorted signal (reference). We observed a significant increase in the mean opinion score (MOS) when using our system, and almost every subject reported that they would prefer to have the system turned on. This supports our proposal for this being a useful and effective system for amplifying an attended speaker. The DNN output was added to the mixture at a level of +12dB relative to the mixture. This level has been shown to significantly increase the intelligibility of an attended speaker in a two-talker scenario (from ≈88% to 98% (Brungart, 2001)). Importantly, an unattended speaker should still be audible so that a user could switch their attention should they choose to do so. It has also been shown that it is still possible to understand a speaker when they are attenuated by 12dB, although intelligibility does drop to ≈78% (Brungart, 2001).

(This page intentionally left blank)

Chapter 8

Conclusion and feature work

The aim of this these is to advance the state of the art for the problem of single channel auditory source separation. In particular, the source separation with neural network. In this thesis, we firstly reviewed the previous proposed algorithms and showed their limitation and deficiency for the problem. We showed that the separation performance with the auto encoder based mask learning network significantly outperformed the traditional model. Then we discussed the limitation for the mask learning system for real recorded data, and proposed two extension of the mask learning, the additional restoration layer and the residual learning network. We showed that the proposed extension helps the mask learning network significantly on the separation of real recorded data.

We introduced the permutation problem and the output dimension mismatch problem, which are the main limitation for auto-encoder network for application in a more generalized separation scenario. We then presented a novel framework, deep clustering to help those two problem and showed that deep clustering can successfully generate high-quality separation result even under very challenging mixtures, which was mixed from three unknown speakers. And we found that the deep clustering can largely increase the intelligibility of the speech. Then we introduced an end to end extension to the deep clustering, i.e. the deep attractor network. We found that after deep attractor network outperformed the deep clustering on the same task. We also introduced several variations in forming the attractors. Lastly, we presented two applications for the neural network based source separation system. We firstly introduce the application of deep clustering and auto encoder network in music separation and showed that it can generate the state of the art performance in singing and background music separation. Then we introduced a novel application for hearing aid, which combined the high performance source separator and an brain signal decoder. We showed such system could decode the attention of the listener, which was used for selection of targeting sources.

There are several very interesting perspective remaining, which we hope to further explore in the further. The main problem for deep clustering and deep attractor network, although they can generate high quality performance, is the center mismatch between the training and the testing. Since for both system, during the testing, the K-means is required to form the center and assignment, while during the training such information is provided. Therefore a slight mismatch between the clustered center and the oracle center. Although no observation of performance degradation was found for this mismatch, we believe, in more challenging task, such phenomenon could cause more severe performance drop. This problem could be solved by using the fixed attractor, as discussed in chapter 6. Another solution could be the semi-fixed attractor, where a set of free attractor is pre-defined, and during both training and testing, the system is only allow to pick attractor from the attractor pool. This strategy could ensure the flexibility of the system, while solving the center mismatch problem.

Another very important and interesting direction is the extension on multi-channel processing. It is well known that the spatial information is very helpful for the source separation. Combining the beamforming and the single channel system such as deep clustering and deep attractor network is clearly very attempting. Interestingly, the beamforming is also a affinity based method, where a looking direction is firstly calculated based on the comparison between microphones. Therefore, the beamforming and the deep clustering/attractor network shares much in their nature, which might simplify the combination. One difficulty for the multi-channel processing is that the network need to process phase, which is usually complex number, and the complex neural network is still in early stage. However, the recent work on time domain processing (van den Oord et al., 2016) and deep beamforming(Xiao et al., 2016) provided new thoughts on this problem.

An obvious extension is on the application perspective. It would be very interesting to see how the system performs under more challenging mixtures. And it would be also important to test the generalization of the proposed system to a broader kinds of audio. Eventually, an universal audio separator should be the target for this system. A combination with other system is also important application. For example, the combination with speaker id system and speech recognition could be a very competitive candidate for the meeting transcription system.

Lastly, we would like to explore the unsupervised or semi-supervised training of the proposed model. Since for human, very limited clean source is available in daily life, while human can still develop almost perfect separator. One hypothesis for this phenomena is that human can combine the information from the same source from different mixtures. For example, the mixture between source A, B and Source A, C shares the information on A. Therefore, by comparing the two mixtures, there is a possibility that no clean source is required.

Bibliography

SiSEC 2016 Professionally produced music recordings task. https://sisec.inria.fr/home/2016-professionally-produced-music-recordings/. accessed: 2016-09-11. (Cited on page 64.)

(Cited on page 66.)

MIREX 2016: Singing Voice Separation Task. http://www.music-ir.org/mirex/wiki/2016:Singing_Voice_Separation, a. accessed: 2016-09-11. (Cited on page 65.)

MIREX 2016: Singing Voice Separation Results. http://www.music-ir.org/mirex/wiki/2016:Singing_Voice_Separation_Results, b. accessed: 2016-09-11. (Cited on pages 65 and 66.)

naplab.ee.columbia.edu/anet. (Cited on page 59.)

Sharon M Abel, Andrea Sass-Kortsak, and Jennifer J Naugler. The role of high-frequency hearing in age-related speech understanding deficits. *Scandinavian audiology*, 29(3): 131–138, 2000. (Cited on page 67.)

IAC acoustics. http://www.industrialnoisecontrol.com/comparative-noise-examples.htm. , 2016. (Cited on page 1.)

Sahar Akram, Jonathan Z Simon, Shihab A Shamma, and Behtash Babadi. A state-space model for decoding auditory attentional modulation from meg in a competing-speaker environment. In *Advances in Neural Information Processing Systems*, pages 460–468, 2014. (Cited on page 85.)

Claude Alain, Benjamin J Dyson, and Joel S Snyder. Aging and the perceptual organization of sounds: A change of scene. *Handbook of models for human aging*, pages 759–769, 2006. (Cited on page 67.)

Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. *arXiv preprint arXiv:1512.02595*, 2015. (Cited on page 1.)

Xavier Anguera, Chuck Wooters, and Javier Hernando. Acoustic beamforming for speaker diarization of meetings. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):2011–2022, 2007. (Cited on page 2.)

Ali Aroudi, Bojana Mirkovic, Maarten De Vos, and Simon Doclo. Auditory attention decoding with eeg recordings using noisy acoustic reference signals. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 694–698. IEEE, 2016. (Cited on page 68.)

Francis R Bach and Michael I Jordan. Learning spectral clustering, with application to speech separation. *Journal of Machine Learning Research*, 7(Oct):1963–2001, 2006. (Cited on pages 36, 40, and 68.)

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. (Cited on page 56.)

J. Barker, R. Marxer, E. Vincent, and S. Watanabe. The third chime speech separation and recognition challenge: Dataset, task and baselines. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 504–511. IEEE, 2015a. (Cited on page 31.)

J. Barker, R. Marxer, E. Vincent, and S. Watanabe. Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks. In *Interspeech 2015*. ISCA, 2015b. (Cited on page 31.)

Sven Behnke. Discovering hierarchical speech features using convolutional non-negative matrix factorization. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 4, pages 2758–2763. IEEE, 2003. (Cited on page 12.)

Ron Weiss Juan Bello. Identifying repeated patterns in music using sparse convolutive non-negative matrix factorization. 2010. (Cited on page 12.)

Jacob Benesty, Jingdong Chen, Yiteng Huang, and Jacek Dmochowski. On microphone-array beamforming from a mimo acoustic signal processing perspective. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3):1053–1065, 2007. (Cited on page 2.)

Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade*, pages 437–478. Springer, 2012. (Cited on page 19.)

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994. (Cited on page 17.)

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proc. ICML*, pages 41–48, 2009. (Cited on pages 47 and 65.)

Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising autoencoders as generative models. In *Advances in Neural Information Processing Systems*, pages 899–907, 2013. (Cited on page 20.)

Adam L. Berenzweig, Daniel P.W. Ellis, and Steve Lawrence. Using voice segments to improve artist classification of music. In *Proc. AES 22nd International Conference: Virtual, Synthetic, and Entertainment Audio*. Audio Engineering Society, 2002. (Cited on page 61.)

Wouter Biesmans, Neetha Das, Tom Francart, and Alexander Bertrand. Auditory-inspired speech envelope extraction methods for improved eeg-based auditory attention detection in a cocktail party scenario. 2016. (Cited on page 68.)

Albert S Bregman. *Auditory scene analysis: The perceptual organization of sound*. MIT press, 1990. (Cited on page 35.)

Albert S Bregman. *Auditory scene analysis: The perceptual organization of sound*. MIT press, 1994. (Cited on pages 8, 67, and 68.)

Guy J Brown and Martin Cooke. Computational auditory scene analysis. *Computer Speech & Language*, 8(4):297–336, 1994. (Cited on pages 2 and 8.)

Henrik Brumm and Sue Anne Zollinger. The evolution of the lombard effect: 100 years of psychoacoustic research. *Behaviour*, 148(11-13):1173–1198, 2011. (Cited on page 84.)

Douglas S Brungart. Informational and energetic masking effects in the perception of two simultaneous talkers. *The Journal of the Acoustical Society of America*, 109(3):1101–1109, 2001. (Cited on page 85.)

György Buzsáki, Costas A Anastassiou, and Christof Koch. The origin of extracellular fields and currents?eeg, ecog, lfp and spikes. *Nature reviews neuroscience*, 13(6):407–420, 2012. (Cited on page 85.)

Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011. (Cited on page 13.)

Yves Chauvin and David E Rumelhart. *Backpropagation: theory, architectures, and applications*. Psychology Press, 1995. (Cited on page 18.)

Zhuo Chen and Daniel PW Ellis. Speech enhancement by sparse, low-rank, and dictionary spectrogram decomposition. In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 1–4. IEEE, 2013. (Cited on pages 2 and 13.)

Zhuo Chen, Brian McFee, and Daniel PW Ellis. Speech enhancement by low-rank and convolutive dictionary spectrogram decomposition. In *INTERSPEECH*, pages 2833–2837, 2014. (Cited on page 12.)

Zhuo Chen, Shinji Watanabe, Hakan Erdoğan, and John R Hershey. Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks. ISCA, 2015. (Cited on pages 25 and 56.)

Zhuo Chen, Yi Luo, and Nima Mesgarani. Deep attractor network for single-microphone speaker separation. *arXiv preprint arXiv:1611.08930*, 2016. (Cited on page 2.)

E Colin Cherry. Some experiments on the recognition of speech, with one and with two ears. *The Journal of the acoustical society of America*, 25(5):975–979, 1953. (Cited on page 2.)

Kyunghyun Cho, Aaron Courville, and Yoshua Bengio. Describing multimedia content using attention-based encoder-decoder networks. *IEEE Transactions on Multimedia*, 17(11): 1875–1886, 2015. (Cited on page 56.)

Jackie L Clark and De Wet Swanepoel. Technology for hearing loss–as we know it, and as we dream it. *Disability and Rehabilitation: Assistive Technology*, 9(5):408–413, 2014. (Cited on pages 67 and 84.)

cocoha.org. *Cognitive Control of a Hearing Aid.* https://cocoha.org/the-need/, 2016. (Cited on page 68.)

M. P. Cooke. *Modelling auditory processing and organisation.* PhD thesis, Univ. of Sheffield, 1991. (Cited on page 36.)

Martin Cooke, John R Hershey, and Steven J Rennie. Monaural speech separation and recognition challenge. *Computer Speech & Language*, 24(1):1–15, 2010. (Cited on pages 35 and 41.)

Neetha Das, Wouter Biesmans, Alexander Bertrand, and Tom Francart. The effect of head-related filtering and ear-specific decoding bias on auditory attention detection. 2016. (Cited on page 68.)

Fernando De la Torre and Michael J Black. Robust principal component analysis for computer vision. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 362–369. IEEE, 2001. (Cited on page 13.)

Li Deng, Michael L Seltzer, Dong Yu, Alex Acero, Abdel-rahman Mohamed, and Geoffrey E Hinton. Binary coding of speech spectrograms using a deep auto-encoder. In *Interspeech*, pages 1692–1695. Citeseer, 2010. (Cited on page 20.)

KV Dijkstra, P Brunner, A Gunduz, W Coon, AL Ritaccio, J Farquhar, and G Schalk. Identifying the attended speaker using electrocorticographic (ecog) signals. *Brain-Computer Interfaces*, 2(4):161–173, 2015. (Cited on page 68.)

Nai Ding and Jonathan Z Simon. Emergence of neural encoding of auditory objects while listening to competing speakers. *Proceedings of the National Academy of Sciences*, 109 (29):11854–11859, 2012a. (Cited on pages 68 and 72.)

Nai Ding and Jonathan Z Simon. Neural coding of continuous speech in auditory cortex during monaural and dichotic listening. *Journal of neurophysiology*, 107(1):78–89, 2012b. (Cited on page 68.)

Jean-Louis Durrieu, Gaël Richard, Bertrand David, and Cédric Févotte. Source/filter model for unsupervised main melody extraction from polyphonic audio signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):564–575, 2010. (Cited on page 61.)

Bradley Ekin, Les Atlas, Majid Mirbagheri, and Adrian KC Lee. An alternative approach for auditory attention tracking using single-trial eeg. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 729–733. IEEE, 2016. (Cited on page 68.)

D. P. W. Ellis. *Prediction-Driven Computational Auditory Scene Analysis.* PhD thesis, MIT, 1996a. (Cited on page 36.)

Daniel PW Ellis. Prediction-driven computational auditory scene analysis. 1996b. (Cited on pages 2 and 8.)

Yariv Ephraim and David Malah. Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(2):443–445, 1985. (Cited on pages 2 and 7.)

Hakan Erdogan, John R. Hershey, Shinji Watanabe, and Jonathan Le Roux. Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. In *Proc. ICASSP*, April 2015a. (Cited on page 49.)

Hakan Erdogan, John R Hershey, Shinji Watanabe, and Jonathan Le Roux. Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 708–712. IEEE, 2015b. (Cited on page 24.)

Zhe-Cheng Fan, Jyh-Shing Roger Jang, and Chung-Li Lu. Singing voice separation and pitch extraction from monaural polyphonic audio music via DNN and adaptive pitch tracking. In *Proc. IEEE International Conference on Multimedia Big Data (BigMM)*, pages 178–185. IEEE, 2016. (Cited on pages 61 and 66.)

Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE Trans. PAMI*, 35(8):1915–1929, 2013. (Cited on page 45.)

Sven Fischer and Klaus Uwe Simmer. Beamforming microphone arrays for speech acquisition in noisy environments. *Speech communication*, 20(3):215–227, 1996. (Cited on page 2.)

Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the nyström method. *IEEE Trans. PAMI*, 26(2):214–225, 2004. (Cited on page 40.)

Ken-ichi Funahashi and Yuichi Nakamura. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural networks*, 6(6):801–806, 1993. (Cited on page 16.)

Elana M Zion Golumbic, Nai Ding, Stephan Bickel, Peter Lakatos, Catherine A Schevon, Guy M McKhann, Robert R Goodman, Ronald Emerson, Ashesh D Mehta, Jonathan Z Simon, et al. Mechanisms underlying selective neuronal tracking of attended speech at a ?cocktail party? *Neuron*, 77(5):980–991, 2013. (Cited on pages 68, 71, 84, and 85.)

Martin T Hagan, Howard B Demuth, Mark H Beale, and Orlando De Jesús. *Neural network design*, volume 20. PWS publishing company Boston, 1996. (Cited on page 15.)

Stephen José Hanson and Lorien Y. Pratt. Comparing biases for minimal network construction with back-propagation. In *Advances in Neural Information Processing Systems*, pages 177–185, 1989. (Cited on page 19.)

K. He, Zhang X., S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. (Cited on page 29.)

Eric W Healy, Sarah E Yoho, Yuxuan Wang, and DeLiang Wang. An algorithm to improve speech recognition in noise for hearing-impaired listeners. *The Journal of the Acoustical Society of America*, 134(4):3029–3038, 2013. (Cited on page 25.)

Hynek Hermansky, Daniel PW Ellis, and Sangita Sharma. Tandem connectionist feature extraction for conventional hmm systems. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, volume 3, pages 1635–1638. IEEE, 2000. (Cited on page 20.)

John R Hershey, Steven J Rennie, Peder A Olsen, and Trausti T Kristjansson. Super-human multi-talker speech recognition: A graphical modeling approach. *Comput. Speech Lang.*, 24(1):45–66, 2010. (Cited on pages 35 and 36.)

John R. Hershey, Jonathan Le Roux, and Felix Weninger. Deep unfolding: Model-based inspiration of novel deep architectures. September 2014. arXiv:1409.2574. (Cited on page 48.)

John R. Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. September 2015. arXiv:1508.04306. (Cited on pages 40 and 84.)

John R. Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *Proc. ICASSP*, March 2016a. (Cited on pages 46, 47, 48, and 49.)

John R. Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *Proc. ICASSP*, pages 31–35, 2016b. (Cited on pages 2, 53, 57, 62, 65, and 68.)

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012a. (Cited on page 20.)

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012b. (Cited on page 19.)

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9 (8):1735–1780, 1997. (Cited on pages 17 and 57.)

Takaaki Hori, Zhuo Chen, Hakan Erdogan, John R Hershey, Jonathan Le Roux, Vikramjit Mitra, and Shinji Watanabe. The merl/sri system for the 3rd chime challenge using beamforming, robust feature extraction, and advanced speech recognition. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 475–481. IEEE, 2015. (Cited on page 56.)

Cort Horton, Michael D'Zmura, and Ramesh Srinivasan. Suppression of competing speech through entrainment of cortical oscillations. *Journal of neurophysiology*, 109(12):3082–3093, 2013. (Cited on page 68.)

Cort Horton, Ramesh Srinivasan, and Michael D?Zmura. Envelope responses in single-trial eeg indicate attended speaker in a ?cocktail party? *Journal of neural engineering*, 11(4): 046015, 2014. (Cited on page 68.)

Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research*, 5(Nov):1457–1469, 2004. (Cited on page 11.)

Chao-Ling Hsu and Jyh-Shing Roger Jang. On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(2):310–319, 2010. (Cited on page 61.)

Ke Hu and DeLiang Wang. An iterative model-based approach to cochannel speech separation. *EURASIP Journal on Audio, Speech, and Music Processing*, 2013(1):1–11, 2013a. (Cited on page 45.)

Ke Hu and DeLiang Wang. An unsupervised approach to cochannel speech separation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 21(1):122–131, 2013b. (Cited on pages 36, 43, 44, and 49.)

Yi Hu and Philipos C Loizou. Subjective comparison and evaluation of speech enhancement algorithms. *Speech communication*, 49(7):588–601, 2007. (Cited on pages 2 and 7.)

Yi Hu and Philipos C Loizou. Evaluation of objective quality measures for speech enhancement. *IEEE Transactions on audio, speech, and language processing*, 16(1):229–238, 2008. (Cited on pages 2 and 7.)

Hank Chang-Han Huang and Frank Seide. Pitch tracking and tone features for mandarin speech recognition. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, volume 3, pages 1523–1526. IEEE, 2000. (Cited on page 9.)

Peihao Huang, Yan Huang, Wei Wang, and Liang Wang. Deep embedding network for clustering. In *Proc. ICPR*, pages 1532–1537, 2014a. (Cited on page 37.)

Po-Sen Huang, Scott Deeann Chen, Paris Smaragdis, and Mark Hasegawa-Johnson. Singing-voice separation from monaural recordings using robust principal component analysis. In *Proc. ICASSP*, pages 57–60, 2012. (Cited on pages 61 and 63.)

Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis. Singing-voice separation from monaural recordings using deep recurrent neural networks. In *Proc. ISMIR*, pages 477–482, 2014b. (Cited on page 62.)

Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis. Joint optimization of masks and deep recurrent neural networks for monaural source separation. *arXiv preprint arXiv:1502.04149*, 2015. (Cited on pages 35 and 45.)

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1): 193–218, 1985. (Cited on page 40.)

Larry E Humes and Lisa Roberts. Speech-recognition difficulties of the hearing-impaired elderlythe contributions of audibility. *Journal of Speech, Language, and Hearing Research*, 33(4):726–735, 1990. (Cited on page 67.)

Yukara Ikemiya, Katsutoshi Itoyama, and Kazuyoshi Yoshii. Singing voice separation and vocal f0 estimation based on mutual combination of robust principal component analysis and subharmonic summation. *arXiv preprint arXiv:1604.00192*, 2016. (Cited on page 66.)

Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R. Hershey. Deep clustering: Audio examples. `http://www.merl.com/demos/deep-clustering`, 2016a. [Online]. (Cited on page 45.)

Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R. Hershey. Single-channel multi-speaker separation using deep clustering. In *Proc. Interspeech*, 2016b. (Cited on pages 2, 53, 59, 62, 65, and 66.)

Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002. (Cited on page 10.)

Walter Kellermann. Strategies for combining acoustic echo cancellation and adaptive beamforming microphone arrays. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, volume 1, pages 219–222. IEEE, 1997. (Cited on page 2.)

Ulrik Kjems, Jesper B Boldt, Michael S Pedersen, Thomas Lunner, and DeLiang Wang. Role of mask pattern in intelligibility of ideal binary-masked noisy speech. *The Journal of the Acoustical Society of America*, 126(3):1415–1426, 2009. (Cited on page 9.)

Lakshmi Krishnan, Mounya Elhilali, and Shihab Shamma. Segregating complex sound sources through temporal coherence. *PLoS Comput Biol*, 10(12):e1003985, 2014. (Cited on page 68.)

Patricia K Kuhl. Human adults and human infants show a ?perceptual magnet effect? for the prototypes of speech categories, monkeys do not. *Perception & psychophysics*, 50(2): 93–107, 1991. (Cited on page 53.)

Sascha Lange and Martin Riedmiller. Deep auto-encoder neural networks in reinforcement learning. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2010. (Cited on page 20.)

César Laurent, Gabriel Pereyra, Philémon Brakel, Ying Zhang, and Yoshua Bengio. Batch normalized recurrent neural networks. *arXiv preprint arXiv:1510.01378*, 2015. (Cited on page 65.)

Jonathan Le Roux, John R Hershey, and Felix Weninger. Deep nmf for speech separation. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 66–70. IEEE, 2015a. (Cited on page 25.)

Jonathan Le Roux, Felix J. Weninger, and John R. Hershey. Sparse NMF – half-baked or well done? Technical Report TR2015-023, MERL, Cambridge, MA, USA, March 2015b. (Cited on pages 41 and 42.)

Byung Suk Lee and Daniel PW Ellis. Noise robust pitch tracking by subband autocorrelation classification. In *Interspeech*, pages 707–710, 2012. (Cited on page 9.)

Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001. (Cited on page 10.)

Bo Li, Tara N Sainath, Ron J Weiss, Kevin W Wilson, and Michiel Bacchiani. Neural network adaptive beamforming for robust multichannel speech recognition. In *Proc. Interspeech*, 2016. (Cited on page 5.)

Jinyu Li, Li Deng, Yifan Gong, and Reinhold Haeb-Umbach. An overview of noise-robust automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4):745–777, 2014. (Cited on page 84.)

Yipeng Li and DeLiang Wang. Separation of singing voice from music accompaniment for monaural recordings. *IEEE Transactions on Audio, Speech, and Language Processing*, 15 (4):1475–1487, 2007. (Cited on page 61.)

Philipos C Loizou. Speech quality assessment. In *Multimedia analysis, processing and communications*, pages 623–654. Springer, 2011. (Cited on page 75.)

Piyush Lotia and MR Khan. A review of various score normalization techniques for speaker identification system. *International Journal of Advances in Engineering & Technology*, 3 (2):650, 2012. (Cited on page 74.)

Jianfen Ma, Yi Hu, and Philipos C Loizou. Objective measures for predicting speech intelligibility in noisy conditions based on new band-importance functions. *The Journal of the Acoustical Society of America*, 125(5):3387–3405, 2009. (Cited on page 75.)

Carol L Mackersie. Talker separation and sequential stream segregation in listeners with hearing losspatterns associated with talker gender. *Journal of speech, language, and hearing research*, 46(4):912–918, 2003. (Cited on page 67.)

Marina Meila. The stability of a good clustering. *University of Washington Department of Statistics*, Technical Report 624, 2014. (Cited on page 40.)

Marina Meilă. Local equivalences of distances between clusterings?a geometric perspective. *Machine Learning*, 86(3):369–389, 2012. (Cited on page 40.)

Nima Mesgarani and Edward F Chang. Selective cortical representation of attended speaker in multi-talker speech perception. *Nature*, 485(7397):233–236, 2012. (Cited on pages 1, 68, 69, 72, and 73.)

Nima Mesgarani, Stephen V David, Jonathan B Fritz, and Shihab A Shamma. Influence of context and behavior on stimulus reconstruction from neural activity in primary auditory cortex. *Journal of neurophysiology*, 102(6):3329–3339, 2009. (Cited on page 72.)

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3, 2010. (Cited on page 16.)

Bojana Mirkovic, Stefan Debener, Manuela Jaeger, and Maarten De Vos. Decoding the attended speech stream with multi-channel eeg: implications for online, daily-life applications. *Journal of neural engineering*, 12(4):046007, 2015. (Cited on page 68.)

Taesup Moon, Heeyoul Choi, Hoshik Lee, and Inchul Song. RnnDrop: A novel dropout for RNNs in ASR. *Proc. ASRU*, 2015. (Cited on page 46.)

Nelson Morgan and Herve Bourlard. Continuous speech recognition using multilayer perceptrons with hidden markov models. In *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, pages 413–416. IEEE, 1990. (Cited on page 15.)

Alex Morla. Four transformative patient demands: convenience, size, simplicity, and flexibility. *Hearing Review*, 18(4):36–42, 2011. (Cited on page 68.)

A. Narayanan and D. Wang. Investigation of speech separation as a front-end for noise robust speech recognition. *IEEE/ACM Trans. on Audio, Speech and Language Processing*, 22(4):826–835, 2014a. (Cited on page 24.)

A. Narayanan and D.L. Wang. Ideal ratio mask estimation using deep neural networks. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, pages 7092–7096, 2013a. (Cited on page 24.)

A. Narayanan and D.L. Wang. Joint noise adaptive training for robust automatic speech recognition. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, 2014b. (Cited on page 24.)

Arun Narayanan and DeLiang Wang. Ideal ratio mask estimation using deep neural networks for robust speech recognition. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7092–7096. IEEE, 2013b. (Cited on pages 20 and 25.)

World Health Organization et al. Millions of people in the world have hearing loss that can be treated or prevented. *Geneva: WHO*, pages 1–17, 2013. (Cited on page 67.)

James A O'Sullivan, Alan J Power, Nima Mesgarani, Siddharth Rajaram, John J Foxe, Barbara G Shinn-Cunningham, Malcolm Slaney, Shihab A Shamma, and Edmund C Lalor. Attentional selection in a cocktail party environment can be decoded from single-trial eeg. *Cerebral Cortex*, 25(7):1697–1706, 2015a. (Cited on pages 1, 68, 69, 72, 73, and 76.)

James A O'Sullivan, Richard B Reilly, and Edmund C Lalor. Improved decoding of attentional selection in a cocktail party environment with eeg via automatic selection of relevant independent components. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 5740–5743. IEEE, 2015b. (Cited on page 68.)

Douglas B Paul and Janet M Baker. The design for the wall street journal-based csr corpus. In *Proceedings of the workshop on Speech and Natural Language*, pages 357–362. Association for Computational Linguistics, 1992. (Cited on page 71.)

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The Kaldi speech recognition toolkit. In *Proc. ASRU*, 2011. (Cited on page 51.)

Alan J Power, John J Foxe, Emma-Jane Forde, Richard B Reilly, and Edmund C Lalor. At what time is the cocktail party? a late locus of selective attention to natural speech. *European Journal of Neuroscience*, 35(9):1497–1503, 2012. (Cited on page 68.)

Lutz Prechelt. Early stopping - but when? In *Neural Networks: Tricks of the Trade*, pages 53–67. Springer, 2012. (Cited on page 19.)

Bhiksha Raj, Tuomas Virtanen, Sourish Chaudhuri, and Rita Singh. Non-negative matrix factorization based compensation of music for automatic speech recognition. In *Interspeech*, pages 717–720, 2010. (Cited on page 2.)

Javier Ramirez, Juan Manuel Górriz, and José Carlos Segura. *Voice activity detection. fundamentals and speech recognition system robustness*. INTECH Open Access Publisher, 2007. (Cited on page 64.)

S. J. Rennie, J. R. Hershey, and P. A. Olsen. Single-channel multitalker speech recognition. *IEEE Signal Process. Mag.*, 27(6):66–80, 2010. (Cited on pages 35 and 36.)

Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 833–840, 2011. (Cited on page 20.)

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 *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 2, pages 749–752. IEEE, 2001. (Cited on page 75.)

EH Rothauser, WD Chapman, N Guttman, KS Nordby, HR Silbiger, GE Urbanek, and M Weinstock. Ieee recommended practice for speech quality measurements. *IEEE Trans. Audio Electroacoust*, 17(3):225–246, 1969. (Cited on page 76.)

Tara N Sainath, Brian Kingsbury, and Bhuvana Ramabhadran. Auto-encoder bottleneck features using deep belief networks. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4153–4156. IEEE, 2012. (Cited on page 20.)

Tara N Sainath, Ron J Weiss, Andrew Senior, Kevin W Wilson, and Oriol Vinyals. Learning the speech front-end with raw waveform cldnns. In *Proc. Interspeech*, 2015a. (Cited on page 5.)

Tara N Sainath, Ron J Weiss, Kevin W Wilson, Arun Narayanan, Michiel Bacchiani, et al. Speaker location and microphone spacing invariant acoustic modeling from raw multichannel waveforms. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 30–36. IEEE, 2015b. (Cited on page 5.)

Ralf Schluter and Hermann Ney. Using phase spectrum information for improved speech recognition performance. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 1, pages 133–136. IEEE, 2001. (Cited on page 6.)

Mikkel N Schmidt and Rasmus Kongsgaard Olsson. Single-channel speech separation using sparse non-negative matrix factorization. In *Spoken Language Proceesing, ISCA International Conference on (INTERSPEECH)*, 2006. (Cited on pages 11 and 68.)

Björn Schuller, Felix Weninger, Martin Wöllmer, Yang Sun, and Gerhard Rigoll. Non-negative matrix factorization as noise-robust feature extractor for speech recognition. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4562–4565. IEEE, 2010. (Cited on page 2.)

Yang Shao, Soundararajan Srinivasan, Zhaozhang Jin, and DeLiang Wang. A computational auditory scene analysis system for speech segregation and robust speech recognition. *Computer Speech & Language*, 24(1):77–93, 2010. (Cited on page 9.)

Abhishek Sharma, Oncel Tuzel, and Ming-Yu Liu. Recursive context propagation network for semantic scene labeling. In *Proc. NIPS*, pages 2447–2455, 2014. (Cited on page 45.)

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. PAMI*, 22(8):888–905, 2000. (Cited on page 36.)

Paris Smaragdis. Convolutive speech bases and their application to supervised speech separation. *IEEE Trans. Audio, Speech, Language Process.*, 15(1):1–12, 2007. (Cited on pages 41 and 42.)

Pablo Sprechmann, Alexander M Bronstein, and Guillermo Sapiro. Real-time online singing voice separation from monaural recordings using robust low-rank modeling. In *ISMIR*, pages 67–72, 2012. (Cited on page 61.)

Hideyuki Tachibana, Yu Mizuno, Nobutaka Ono, and Shigeki Sagayama. A real-time audio-to-audio karaoke generation system for monaural recordings based on singing voice suppression and key conversion techniques. *Journal of Information Processing*, 24(3): 470–482, 2016. (Cited on page 61.)

Fei Tian, Bin Gao, Qing Cui, Enhong Chen, and Tie-Yan Liu. Learning deep representations for graph clustering. In *Proc. AAAI*, 2014. (Cited on page 37.)

T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012a. (Cited on pages 46 and 49.)

T. Tieleman and G. Hinton. Lecture 6.5-rmsprop. *COURSERA: Neural networks for machine learning*, 2012b. (Cited on page 58.)

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2), 2012c. (Cited on page 64.)

Aäron 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. *arXiv preprint arXiv:1609.03499*, 2016. (Cited on page 88.)

Simon Van Eyndhoven, Tom Francart, and Alexander Bertrand. Eeg-informed attended speaker extraction from recorded speech mixtures with application in neuro-steered hearing prostheses. *arXiv preprint arXiv:1602.05702*, 2016. (Cited on page 68.)

Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *IEEE Trans. Audio, Speech, Language Process.*, 14(4):1462–1469, 2006. (Cited on page 43.)

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010. (Cited on page 20.)

T. Virtanen. Speech recognition using factorial hidden markov models for separation in the feature space. In *Proc. Interspeech 2006*, Pittsburgh, 2006. (Cited on page 35.)

Tuomas Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3):1066–1074, 2007. (Cited on page 11.)

Stefan Wager, Sida Wang, and Percy S. Liang. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems*, pages 351–359, 2013. (Cited on page 19.)

Chao Wang and Stephanie Seneff. Robust pitch tracking for prosodic modeling in telephone speech. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, volume 3, pages 1343–1346. IEEE, 2000. (Cited on page 9.)

DeLiang Wang. On ideal binary mask as the computational goal of auditory scene analysis. In *Speech separation by humans and machines*, pages 181–197. Springer, 2005. (Cited on pages 9, 41, and 49.)

DeLiang Wang and Guy J Brown. *Computational auditory scene analysis: Principles, algorithms, and applications*. Wiley-IEEE Press, 2006. (Cited on pages 2 and 8.)

Y. Wang, A. Narayanan, and D. Wang. On training targets for supervised speech separation. *IEEE/ACM Trans. on Audio, Speech and Language Processing*, 22(12):1849–1858, 2014a. (Cited on page 24.)

Yuxuan Wang and DeLiang Wang. Towards scaling up classification-based speech separation. *IEEE Trans. Audio, Speech, Language Process.*, 21(7):1381–1390, 2013. (Cited on pages 25 and 35.)

Yuxuan Wang, Kun Han, and DeLiang Wang. Exploring monaural features for classification-based speech segregation. *IEEE Trans. Audio, Speech, Language Process.*, 21(2):270–279, 2013. (Cited on page 35.)

Yuxuan Wang, Arun Narayanan, and DeLiang Wang. On training targets for supervised speech separation. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, 22(12):1849–1858, 2014b. (Cited on page 36.)

Ron J Weiss. *Underdetermined source separation using speaker subspace models*. PhD thesis, Columbia University, 2009. (Cited on page 36.)

Felix Weninger, Florian Eyben, and Bjorn Schuller. Single-channel speech separation with memory-enhanced recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 3709–3713. IEEE, 2014a. (Cited on page 35.)

Felix Weninger, John R Hershey, Jonathan Le Roux, and Björn Schuller. Discriminatively trained recurrent neural networks for single-channel speech separation. In *Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on*, pages 577–581. IEEE, 2014b. (Cited on pages 24, 25, and 72.)

Felix Weninger, Jonathan Le Roux, John R. Hershey, and Björn Schuller. Discriminatively trained recurrent neural networks for single-channel speech separation. In *Proc. IEEE GlobalSIP 2014 Symposium on Machine Learning Applications in Speech Processing*, December 2014c. (Cited on page 63.)

Felix Weninger, Hakan Erdogan, Shinji Watanabe, Emmanuel Vincent, Jonathan Le Roux, John R Hershey, and Björn Schuller. Speech enhancement with lstm recurrent neural networks and its application to noise-robust asr. In *Latent Variable Analysis and Signal Separation*, pages 91–99. Springer, 2015. (Cited on pages 25, 36, 42, and 71.)

Max Wertheimer. Laws of organization in perceptual forms. In Willis A Ellis, editor, *A Source book of Gestalt psychology*, pages 71–88. Routledge and Kegan Paul, 1938. (Cited on page 36.)

Xiong Xiao, Shinji Watanabe, Hakan Erdogan, Liang Lu, John Hershey, Michael L Seltzer, Guoguo Chen, Yu Zhang, Michael Mandel, and Dong Yu. Deep beamforming networks for multi-channel speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5745–5749. IEEE, 2016. (Cited on page 88.)

W Xiong, J Droppo, X Huang, F Seide, M Seltzer, A Stolcke, D Yu, and G Zweig. The microsoft 2016 conversational speech recognition system. *arXiv preprint arXiv:1609.03528*, 2016a. (Cited on page 29.)

Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig. Achieving human parity in conversational speech recognition. *arXiv preprint arXiv:1610.05256*, 2016b. (Cited on page 1.)

Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee. An experimental study on speech enhancement based on deep neural networks. *Signal Processing Letters, IEEE*, 21(1):65–68, 2014. (Cited on pages 20 and 36.)

Po-Kai Yang, Chung-Chien Hsu, and Jen-Tzung Chien. Bayesian singing-voice separation. In *Proc. ISMIR*, pages 507–512, 2014. (Cited on page 61.)

Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. *arXiv preprint arXiv:1607.00325*, 2016. (Cited on pages 2, 44, and 53.)

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014. (Cited on page 46.)

Lijun Zhang, Zhengguang Chen, Miao Zheng, and Xiaofei He. Robust non-negative matrix factorization. *Frontiers of Electrical and Electronic Engineering in China*, 6(2):192–200, 2011. (Cited on page 13.)

Zhengyou Zhang, Michael Lyons, Michael Schuster, and Shigeru Akamatsu. Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pages 454–459. IEEE, 1998. (Cited on page 15.)