COMPLEX NEURAL NETWORKS FOR AUDIO

A Thesis
Submitted to the Faculty
in partial fulfillment of the requirements for the
degree of
Doctor of Philosophy
in
Computer Science
by

Andy M. Sarroff
DARTMOUTH COLLEGE
Hanover, New Hampshire
May 2018

(This page intentionally left blank)

COMPLEX NEURAL NETWORKS FOR AUDIO

A Thesis
Submitted to the Faculty
in partial fulfillment of the requirements for the
degree of
Doctor of Philosophy
in
Computer Science
by
Andy M. Sarroff
DARTMOUTH COLLEGE
Hanover, New Hampshire
May 2018

Examining Committee:

Michael A. Casey, Ph.D. (Chair)

Daniel Rockmore, Ph.D.

Andrew T. Campbell, Ph.D.

Jonathan Le Roux, Ph.D.

F. Jon Kull, Ph.D.
Dean of Graduate and Advanced Studies

(This page intentionally left blank)

(This page intentionally left blank)

Abstract

Complex Neural Networks for Audio

Andy M. Sarroff

Audio is represented in two mathematically equivalent ways: the real-valued time domain (i.e., waveform) and the complex-valued frequency domain (i.e., spectrum). There are advantages to the frequency-domain representation, e.g., the human auditory system is known to process sound in the frequency-domain. Furthermore, linear time-invariant systems are convolved with sources in the time-domain, whereas they may be factorized in the frequency-domain.

Neural networks have become rather useful when applied to audio tasks such as machine listening and audio synthesis, which are related by their dependencies on high quality acoustic models. They ideally encapsulate fine-scale temporal structure, such as that encoded in the phase of frequency-domain audio, yet there are no authoritative deep learning methods for complex audio. This manuscript is dedicated to addressing the shortcoming.

Chapter 2 motivates complex networks by their affinity with complex-domain audio, while Chapter 3 contributes methods for building and optimizing complex networks. We show that the naive implementation of Adam optimization is incorrect for complex random variables and show that selection of input and output representation has a significant impact on the performance of a complex network.

Experimental results with novel complex neural architectures are provided in the second half of this manuscript. Chapter 4 introduces a complex model for binaural audio source localization. We show that, like humans, the complex model can generalize to different anatomical filters, which is important in the context of machine listening. The complex model's performance is better than that of the real-valued models, as well as real- and complex-valued baselines.

Chapter 5 proposes a two-stage method for speech enhancement. In the first stage, a complex-valued stochastic autoencoder projects complex vectors to a discrete space. In the second stage, long-term temporal dependencies are modeled in the discrete space. The autoencoder raises the performance ceiling for state of the art speech enhancement, but the dynamic enhancement model does not outperform other baselines. We discuss areas for improvement and note that the complex Adam optimizer improves training convergence over the naive implementation.

Preface

My path began as a music recording and mixing engineer. I had been spending my post-collegiate years working with musicians in New York when digital recording was first becoming available to the mainstream. In 2006 I joined the nascent Media and Audio Research Laboratory at NYU. Juan Bello, upon inviting me into his research group, introduced me to Music Information Retrieval, a corner of the computational humanities that I never knew existed. Juan was adamant that his students enroll in Machine Learning across the street at Courant and I followed his advice. We had discussions in 2008 about whether deep learning would be just another short-lived fad in the quick-moving world of machine learning. Somehow between then and now, I've found myself writing a dissertation on what I believe to be an incredibly exciting area, complex deep learning for audio.

I feel incredibly lucky to have been offered the opportunity to dive into such a fascinating field, but my research has only been possible with tremendous intellectual and emotional contributions from mentors, friends, and family who have supported me along the way. Foremost, my advisor Michael Casey has guided me through Dartmouth and the dissertation with intellectual generosity and professional wisdom. Drs. Jonathan Le Roux and John Hershey from MERL welcomed me into a vibrant laboratory, making me a better scientist. Dan Rockmore, through his direction of the Neukom Institute, has fostered an environment at Dartmouth where scientists and artists can thrive together. Andrew Campbell took a chance on me as I entered Dartmouth, welcoming me into his research group during the formative first year of study at Dartmouth.

Thanks also to the countless grad friends and colleagues that have seen me through the worst and best, and who have provided hours of conversation and insight. There are too many to list, but special mention goes to some of my ex-housemates: Carlos Dominguez, Jessica Thompsom, Beau Sievers, Phil Hermans, and Victor Shepardson. I'd also like to thank Colin Raffel who, through Dan Ellis, invited me to join him at LabROSA in its final summer and fall. The template for this thesis originated at LabROSA, as well as several ideas behind this thesis.

I am filled with love and gratitude for my family, who have been unwavering supports through the ups and downs. Thank you to my parents Eileen and Alan Sarroff, Amanda and Pierre Sarroff-Robion, and especially to Elsa Sarroff-Robion, who carries on the percussive tradition at joyful volumes. Finally, to Jen. From porcupines to top hats, I couldn't have done this without you.

(This page intentionally left blank)

Contents

(This page intentionally left blank)

List of Tables

List of Figures

CHAPTER 1

Introduction

1.1 Motivation

Neural networks and audio are tightly integrated into our lives. Because of deep neural networks, we're able to dictate messages to computers using automatic speech recognition (ASR) (Chan, Jaitly, Le, & Vinyals, 2016; Shan, Zhang, Wang, & Xie, 2017; Soltau, Liao, & Sak, 2016). We can query computers about prominent instruments in a music recording (Han, Kim, & Lee, 2017). With neural audio synthesis, computers are able to generate realistic-sounding speech (Shen et al., 2017) and musical tones (Engel et al., 2017), or to separate speech mixtures into single-speaker audio streams (P.-S. Huang, Kim, Hasegawa-Johnson, & Smaragdis, 2014).

The success of these technologies hinges on adequate machine listening and neural sound synthesis. These are related because they each depend on a high quality acoustic model, which represents the relationship between an audio signal and another entity of interest. In some cases the audio signal is the input to a model (e.g., ASR); in other cases audio is the output (e.g., neural audio synthesis); and some modeling tasks use audio as input and output (e.g., speech enhancement). Whichever the case, the performance of a system is bounded by the quality of an acoustic model.

Audio is represented in two mathematically equivalent ways: the real-valued time domain (i.e., waveform) and the complex-valued frequency domain (i.e., spectrum). There are several advantages to relying on the frequency-domain representation. First, the human auditory system processes sound in frequency bands (Zwicker & Fastl, 2010), with the cochlea providing a frequency decomposition similar to the Fourier transform, except on a logarithmic scale (McDermott, 2013). When we begin (or end) a computational pipeline with

a frequency-domain representation of audio, we are leveraging a schema that is positively correlated to human audition. From there, a network may learn the frequency-based features that are most informative to the task (Humphrey, Bello, & LeCun, 2013).

For many audio modeling tasks, we're particularly interested in linear time- or shift-invariant (LTI and LSI) systems. Room reverberation, vocal sounds, and binaural auditory cues are outputs of LTI systems, and are obtained by convolving an input with a system's impulse response. This brings us to the second motivation for using frequency-domain signals. Convolution in the time-domain is mathematically equivalent to multiplication in the frequency-domain. If we wish to characterize the system that produced a sound, the problem is reduced from deconvolution in the time-domain to factorization in the frequency-domain.

Here, we identify a shortcoming in the deep learning literature for audio. Most acoustic models favor a frequency-domain representation. Yet the application of neural networks in tandem with complex-valued audio is poorly understood, so it's common practice to truncate complex-valued spectra by discarding the phase components of frequency-domain audio.

We argue that the discarded phase components are informative in machine listening and audio synthesis tasks, but that they're often neglected because the deep learning community has not developed dependable methods for training neural networks using complex-valued data. We address this shortcoming, motivated by the following observations.

1. Without phase, audio is less intelligible, which impairs the quality of human and machine listening.
2. The quality of frequency-domain audio synthesis is limited when phase is reconstructed as an afterthought.

The following subsections provide further examination of the motivations for using complex neural networks on machine listening and audio synthesis tasks.

1.1.1 Machine Listening

The goal of machine listening is to endow computers with the ability to "hear." Computers should be able to extract meaningful information from, and make human-like decisions about, audio. In addition, they should be able to do this in a general way. For instance, humans are particularly adept at focusing on individual sounds within a mixture of sounds. We navigate the focus of our attention between multiple aspects of a live music performance with ease, picking out the lead vocalist or guitarist. Bregman (1994) refers to this phenomenon as "auditory streaming." We deftly identify the source of an auditory stream, often among interfering noises and among varying acoustic contexts. The audio features of a musical instrument recorded in a concert hall versus one that is recorded on a public street might be very different, but we'd expect a computer to identify them as the same, thereby generalizing across varying acoustic contexts.

External sounds are filtered by the human anatomy before arriving at the inner ear (Blauert, 1997). Due to distinguishing differences in the shapes of the torso, head, and ears, an individual's anatomical filters, characterized by their head-related transfer functions (HRTFs),

are unique. We can approximate an individual's HRTFs by taking measurements called head-related impulse responses (HRIRs), in which wide-band sounds arriving from various angles relative to a human subject are recorded using in-ear microphones. Time-domain HRIRs are subsequently converted into frequency-domain HRTFs using the Fourier transform.

The physical anatomy characterized by the HRTF is an LTI system that provides important localization cues, allowing one to estimate the direction of arrival of a sound source (Wenzel, Arruda, Kistler, & Wightman, 1993). According to duplex theory (Middlebrooks & Green, 1991), human source localization depends on two auditory cues: interaural level and time difference (ILD and ITD). The acoustic waves propagating from any sound source that is not located along the median plane must travel different distances to the ears and therefore arrives at different times. The ITD of binaural sound corresponds to interaural phase differences (IPD) in the frequency-domain. It's been established that the human auditory system "prefers" ITD and IPD over ILD in cases where both cues are available but provide conflicting information (Wightman & Kistler, 1992). In other words, phase differences are incredibly important to humans when localizing sound sources.

We can simulate localization cues through binaural processing, in which a sound source is convolved with the HRIR corresponding to a known elevation and azimuth (Wightman & Kistler, 1989a, 1989b). Although the HRTFs corresponding to an individual are unique, it's been shown that humans can adapt to the HRTFs of other "good" hearers (Wenzel et al., 1993). In other words, consider a subject who hears binaural audio associated with a second subject having high localization accuracy. Then the first subject is able to identify the virtual locations of sounds with similar accuracy as in free-field conditions. This is quite astonishing, showing that humans have an incredible ability to quickly learn new auditory localization cues.

We return to the requirement that computers make human-like decisions about sound and that they generalize to different acoustic contexts. We might ask whether a source localization algorithm can perform well at localizing sounds synthesized with HRTFs, that were not seen during the training phase. If so, then the algorithm is exhibiting an ability to generalize to different conditions, much as humans can. This question is answered in Chapter 4, in which we train a deep complex neural network to localize a sound source in synthesized binaural audio and test its accuracy with unseen binaural filters. Importantly, we show that complex neural networks outperform real networks.

1.1.2 Audio Synthesis

An ideal machine-generated sound is indistinguishable from an acoustically-produced exemplar. It is challenging to design a sufficient acoustic model, one that can produce natural-sounding audio, given a model's internal representation. This is a difficult problem because audio is extremely high-dimensional and it has short-term and long-term temporal dependencies.

The high dimensionality of digital audio is a consequence of the Nyquist-Shannon sampling theorem, which states that for a given digital signal, the maximum frequency of any sinusoidal component is one half of the sampling rate. Healthy human hearing extends to approximately 20 kHz, requiring a minimum sample rate of 40 kHz to cover the normal range of hearing. In

practice, there are several standard sample rates. For instance, audio CDs are recorded at a sample rate of 44.1 kHz.

Audio has structure existing on multiple temporal scales. For instance, the structural elements of speech exist at the phoneme, syllable, word, phrase, and sentence level. Towards one end of the scales, English syllables are a few hundred milliseconds in duration (Peterson & Lehiste, 1960), while at the other end sentences are typically several seconds in duration. Neural audio synthesis models must capture temporal dependencies at the sample-level, as well as at several higher orders of temporal magnitude.

In the literature, acoustic models are often based solely on the magnitude spectrogram. In such cases, the fine-scale temporal dependencies, which are encoded by the phase spectra, are not explicitly modeled. In the end, phase must somehow be added back to the spectrogram as a postprocessing step for audio waveform synthesis. For several decades, the canonical method for reconstructing waveforms from magnitude spectra has been the Griffin-Lim algorithm (Griffin & Lim, 1984). Griffin-Lim is an iterative reconstruction algorithm that suffers from several limitations, such as not considering cross-frame dependencies or local regularities within a frame (Gerkmann, Krawczyk-Becker, & Le Roux, 2015).

Recently, a spate of new algorithms have emerged that provide time-domain neural audio synthesis (Kalchbrenner et al., 2018; Mehri et al., 2016; van den Oord et al., 2016; van den Oord et al., 2017), bypassing the need for explicit phase reconstruction. At the heart of these models is sample-based autoregressive synthesis, where each time-domain audio sample is conditioned on previous samples, thus explicitly modeling fine-scale temporal dependencies. Several autoregressive methods have been proposed for modeling long-term, as well as fine-scale, temporal dependencies. SampleRNN (Mehri et al., 2016) uses several tiers of recurrent layers for modeling dependencies at hierarchically increasing temporal scales. WaveNet (van den Oord et al., 2016) is a convolutional network relying on dilated convolutions for multiscale temporal modeling.

The autoregressive nature of these models severely limits efficiency at inference time, since the sample-level input of the model at the current timestep depends on the sample-level output of the model at the previous timestep. These models, as first proposed, were not able to provide real-time inference on modern hardware, severely limiting their practical utility. Several solutions have since been suggested for increasing inference efficiency, but these too have drawbacks. For instance, Parallel WaveNet (van den Oord et al., 2017) provides real-time synthesis, but training it directly is infeasible. Instead, a "teacher" model trains a "student" model to "match the probability of its own samples under the distribution learned by the teacher" (van den Oord et al., 2017). There are yet other alternatives for efficient time-domain neural audio synthesis. WaveRNN (Kalchbrenner et al., 2018) weakens constraints on conditional dependency by using "subscaling." According to Kalchbrenner et al. (2018), WaveRNN's subjective performance approaches that of WaveNet, but does not surpass it.

Complex frequency-domain audio includes the fine-scale temporal structure needed for audio synthesis. Despite this, the complex-valued deep modeling of audio has not been extensively studied. Chapter 5 applies deep complex networks, for the first time, to the problem of speech enhancement, in which speech audio is enhanced by rejecting interfering sounds from a "noisy" audio signal and synthesizing a "clean" version. This task requires a strong acoustic model for synthesizing audio, one that is absent of artifacts. Our approach uses a deep model

to learn a stochastic dictionary of complex vectors. Speech is enhanced by inferring the best dictionary elements for synthesis.

1.1.3 Complex Neural Networks

Recent papers such as (Bittner, McFee, Salamon, Li, & Bello, 2017; Hawthorne et al., 2017; Jansson et al., 2017; Liang, Gotham, Johnson, & Shotton, 2017; Maezawa, 2017; Oramas, Nieto, Barbieri, & Serra, 2017) are employing increasingly deep networks for music information retrieval and music generation. Their successes may be attributed to the fact that, with at least one hidden layer of arbitrary width and nonlinear activation functions, neural networks are universal function approximators (Hornik, 1991; Sonoda & Murata, 2017). Arbitrarily deep architectures provide more expressive models with fewer computational units than shallow networks (Bengio, 2009). The combined power and expressivity of universal approximation and deep architectures has helped neural networks dominate recent advances in the machine learning literature and, consequently, music information, audio, and signal processing applications.

Nevertheless the literature on complex-valued networks is heavily dominated by the signal processing community, especially concerning topics on nonlinear adaptive filters. Applications include adaptive beamforming (Suksmono & Hirose, 2003), channel equalization (R.-C. Huang & Chen, 2000; Solazzi, Uncini, Claudio, & Parisi, 2001), source separation (Scarpiniti, Vigliano, Parisi, & Uncini, 2008), equalization of satellite communications (Benvenuto, Marchesi, Piazza, & Uncini, 1991b), and coherent-lightwave networks (Hirose, 1994; Hirose & Kiuchi, 2000). An adaptive noise reduction method for interferometric synthetic aperture radar is presented in (Suksmono & Hirose, 2002). Improvements to electrical power systems have been proposed with respect to load flow analysis (Ceylan, Çetinkaya, Ceylan, & Özbay, 2006) and power transformer modeling (Chistyakov, Kholodova, Minin, Zimmermann, & Knoll, 2011; Minin, Chistyakov, Kholodova, Zimmermann, & Knoll, 2012).

In the fields of computer vision and image processing, researchers have applied complex networks on optical flow (Miyauchi & Seki, 1992; Miyauchi, Seki, Watanabe, & Miyauchi, 1993). Multivalued neurons were used for deblurring (I. N. Aizenberg, Paliy, Zurada, & Astola, 2008), and character recognition was performed using Clifford networks (Rahman, Howells, & Fairhurst, 2001). Gray-scale image reconstruction was presented in (Tanaka & Aihara, 2009). Complex networks have been used in conjunction with holographic movies in (Hirose, Higo, & Tanizawa, 2006a, 2006b; Tay, Tanizawa, & Hirose, 2007). S. L. Goh, Popovic, and Mandic (2004) have used complex neural networks to predict wind direction and speed, and Minami and Hirose have constructed digital elevation maps (Minami & Hirose, 2003).

Unitary weight matrices have been used to alleviate the vanishing gradient problem for recurrent networks (Arjovsky, Shah, & Bengio, 2016; Wisdom, Powers, Hershey, Le Roux, & Atlas, 2016). These papers take advantage of properties associated with orthogonal matrices, and are able to improve space and time efficiency by utilizing complex-valued unitary matrices. Danihelka, Wayne, Uria, Kalchbrenner, and Graves (2016) have combined holographic memory, in which complex vectors are used to store and retrieve information from a memory bank, with Long Short-Term Memory (Hochreiter & Schmidhuber, 1997) models.

Recently, Trabelsi et al. (2017) discussed how to apply complex adaptations of real-valued activation functions, batch normalization, and weight initialization to deep complex networks.

Despite rich literature on complex neural networks, there are few examples of these being applied to audio or audio-like signals. Hirose and Yoshida (2012) demonstrated that complex networks are better than real networks for denoising incoherent, or noise-corrupted, waveforms. Al-Nuaimi, Amin, and Murase (2012) proposed an improvement to the MP3 codec using complex networks. Tsuzuki, Kugler, Kuroyanagi, and Iwata (2013) have used complex networks for audio source localization. Associative memories for musical temporal sequences has been suggested, but these models operated on a symbolic music representation, rather than on audio signal (Kataoka, Kinouchi, & Hagiwara, 1998; Kinouchi & Hagiwara, 1995, 1996). Finally, Trabelsi et al. (2017) evaluate deep complex networks with a music transcription and a speech spectrum prediction task.

Why use complex-valued models on audio, when a complex number $z = a + \mathrm{i}b$ is fully identified by an ordered pair of real-valued numbers (a, b)? Our motivations are twofold. First, as a consequence of key properties of the discrete Fourier transform, the real and imaginary parts of audio are statistically dependent on each other. For example, the Fourier transform is a linear map, and multiplying a time-domain signal by a scalar corresponds to multiplying its magnitude in the frequency-domain. Scaling the magnitude of a complex number is equivalent to multiplying its real and imaginary parts by equal values, implying statistical dependence. As a second example of statistical dependency, we may consider the shift theorem, i.e., a circular rotation of a time-domain signal corresponds to a linear phase shift in the frequency domain. The real and imaginary parts of a complex number are dependent on each other under any change in phase. We argue that using real-valued models on frequency-domain audio breaks assumptions, based on prior knowledge, about the dependence of the real and imaginary parts.

Second, a complex-valued model provides a more constrained system than one based on real numbers. If we know in advance that phase and magnitude are important to the learning objective, then it's sensible to employ a complex-valued model. To illustrate, the following example is borrowed from Hirose (2011). Suppose we would like to solve the linear equation $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$, with inputs $\boldsymbol{x} \in \mathbb{R}^2$, parameters $\boldsymbol{W} \in \mathbb{R}^{2\times 2}$, and outputs $\boldsymbol{y} \in \mathbb{R}^2$. There are more parameters than inputs, so the equation is underdetermined and has either no solutions or an infinite number of solutions. Now, if we interpret $\boldsymbol{W}$ as a univariate complex number, then its elements must have the form

$$\boldsymbol{W} = \begin{bmatrix} a & -b \\ b & a \end{bmatrix}, \tag{1.1}$$

where a and b are the real and imaginary coefficients of the complex number $w = a + \mathrm{i}b$. There are now only two degrees of freedom. Furthermore, $a = |w| \cos(\theta)$ and $b = |w| \sin(\theta)$, where $\theta = \arg(w)$. Therefore, by having $\boldsymbol{W}$ represent a univariate complex parameter rather than two bivariate real parameters, we have a solution space with less degrees of freedom and that is easily interpreted in terms of magnitude and phase.

Research into complex networks has developed in parallel with real networks since the late 1980s, yet there has been little development of complex neural networks for deep learning. Rather, such research, using shallow models, targets highly-specific signal-processing domains such as channel equalization and adaptive array processing. Several factors contribute to

the slow adoption of complex networks. First, they are difficult to train because complex-valued activation functions cannot be simultaneously bounded and complex-differentiable; second, there are few known methods for regularization and hyper-parameter optimization specifically developed for complex neural networks; third, there are no public libraries for training complex models. Indeed, there is a large gap in the recent literature between complex- and real-valued neural networks. Chapter 3 provides an overview of these challenges and various ways of training complex networks

1.2 Contributions

The origin of complex networks dates to the 1980s. Yet none of the existing implementations address the challenges and rewards of employing deep complex networks specifically for audio tasks. Modern deep learning libraries, which have undergone rapid development in the last five years, do not fully support complex learning. The application of complex networks on audio is therefore itself a novel contribution, one which will open paths to future research.

This manuscript provides three primary contributions.

1. We're the first to motivate deep complex networks by their affinity with complex-valued frequency-domain audio.

2. We provide working complex architectures, bridging a gap between theory and application.

3. We present experimental results using novel complex neural architectures on two highly relevant audio modeling tasks: audio source localization and speech enhancement.

The first two contributions are the content of Chapters 2 and 3, where we motivate complex networks for audio and examine architectural design choices and optimization schemes. In particular, we emphasize that holomorphicity is not a requirement for complex neural optimization. That is, there's a potentially large palette of activation functions from which to choose if we relax smoothness requirements. Instead, networks with complex components such as complex weights may be optimized using real-analytic activation functions. This may be accomplished using modern deep learning libraries after making minor modifications to optimization algorithms. In particular, we show that the naive implementation of Adam and other optimization algorithms that track the uncentered second moment of the gradient need to ensure that the statistic is correct for complex random variables. We also show that the choice of input and output representation may have a significant impact on the performance of a complex network.

The latter contribution is the content of Chapters 4 and 5, where we introduce novel architectures for binaural audio source localization and speech enhancement. The source localization model beats real-valued models, as well as real- and complex-valued baselines. We emphasize that the localization model is able to generalize across different anatomical filters, or head-related transfer functions. This is important because humans can do the same, and machine listening algorithms should perform comparably to humans.

Table 1.1: Phrases and their abbreviations.

Abbreviation	Phrase
ANN	Artificial neural network
BRIR	Binaural room impulse response
CE	Cross-entropy
CLMS	Complex least mean square
CPU	Central processing unit
DFT	Discrete Fourier transform
DSE	Dynamic speech enhancement
ETF	Elementary transcendental function
GPU	Graphics processing unit
GRU	Gated recurrent unit
HATS	Head and torso simulator
HRIR	Head-related impulse response
HRTF	Head-related transfer function
ILD	Interaural level difference
IPD	Interaural phase difference
ITD	Interaural time difference
LMS	Least mean square
LSTM	Long-short term memory
MA	Mask approximation
NMF	Nonnegative matrix approximation
POS	Part-of-speech
PSA	Phase-sensitive signal approximation
ReLU	Rectified linear unit
RNN	Recurrent neural network
SA	Signal approximation
SDR	Signal to distortion ratio
SNR	Signal to noise ratio
STFT	Short term Fourier transform

Speech is enhanced using a two-stage model. In the first stage, we train a complex-valued stochastic autoencoding dictionary to encode a frame of frequency-domain audio into a set of categorical probability distributions. Frames of audio are synthesized by sampling from the encoder and propagating the samples through a nonlinear decoder. We show that the autoencoder can generate high-quality audio from a discrete space, which is an important contribution because neural audio synthesis requires a strong acoustic model. We also show that the Complex Adam algorithm converges to a better optimum than a naive implementation of Adam for complex networks. The second stage of the model infers a temporal sequence of discrete categorical spaces associated with frames of enhanced speech. We find that our proposed complex speech enhancement model is not able to beat a similar real models. However, it is the first implementation of its kind, and it paves the way for other complex domain neural synthesis models.

Table 1.2: Summary of math typography, notation, and symbols.

Notation	Description
$\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$	Sets of integer, rational, real, and complex numbers
x, $\boldsymbol{x}$, $\boldsymbol{X}$	Scalar, vector, matrix
x_{ij}	The element in the i-th row and j-th column of $\boldsymbol{X}$
$\boldsymbol{x}^{(a)}$	A vector $\boldsymbol{x}$ that is identified by the label (a)
f, g, h	Functions
x, y, z	Variables
$\operatorname{Re} z$, $\operatorname{Im} z$	Real and imaginary parts of z
$\boldsymbol{Z}^{\mathsf{T}}$	Transpose of $\boldsymbol{Z}$
$\overline{\boldsymbol{Z}}$	Complex conjugate of $\boldsymbol{Z}$
$\boldsymbol{Z}^{*}$	Conjugate transpose of $\boldsymbol{Z}$
$\mathcal{F}$	Fourier transform
$\mathcal{H}$	Hilbert transform
$\mathrm{f_s}$	Sampling rate
i	Imaginary number
e	Euler's number

1.3 Preliminaries

Table 1.1 provides commonly abbreviated phrases found in the manuscript. We will adhere to the following notational conventions, a summary of which is provided in Table 1.2. Most mathematical notation is typeset in a Roman font. We use the following typesetting to indicate the dimensionality of data structures.

Scalar array elements. Lowercase slanted symbol, e.g. x;

Row or column vectors. Lowercase bold slanted symbol, e.g. $\boldsymbol{x}$; and

Multidimensional arrays, including matrices. Uppercase bold slanted symbol, e.g. $\boldsymbol{X}$.

Unless otherwise indicated, all vectors are to be treated as column vectors. In general the uppercase letters M and N will indicate the sizes of individual array dimensions, with i and j representing indices of rows and columns, respectively. The symbols f, g, and h will be used for functions. The typesetting of a function will be used, when convenient and informative, to indicate the dimensionality of its output, e.g. f, $\boldsymbol{f}$, and $\boldsymbol{F}$. Indices into arrays are indicated with subscripted letters. We use zero-based counting for all indices. Array labels are indicated using superscripted text surrounded by parentheses.

Variables will generally be introduced by the symbols (upper or lowercase) x, y, and z. We will describe them as (possibly multidimensional) arrays of values. At the same time, we will indicate the numerical set membership of array elements. For instance a matrix of complex numbers could be designated as $\boldsymbol{X} \in \mathbb{C}^{M \times N}$. The transpose of a matrix $\boldsymbol{X}$ is $\boldsymbol{X}^{\mathsf{T}}$.

An arbitrary complex matrix $\boldsymbol{Z} = \boldsymbol{U} + \mathrm{i}\boldsymbol{V} : \boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{M \times N}$ is composed of real coefficients $\boldsymbol{U}$ and imaginary coefficients $\mathrm{i}\boldsymbol{V}$. The complex conjugate is indicated with an overline, e.g.,

$\overline{\boldsymbol{Z}} = \boldsymbol{U} - \mathrm{i}\boldsymbol{V}$. Its conjugate transpose is indicated with an asterisk, e.g., $\boldsymbol{Z}^* = (\overline{\boldsymbol{Z}})^\mathsf{T}$. Real and imaginary components may be extracted respectively using the operators $\mathrm{Re}\{\boldsymbol{Z}\} = \frac{1}{2}(\boldsymbol{Z} + \overline{\boldsymbol{Z}})$ and $\mathrm{Im}\{\boldsymbol{Z}\} = \frac{1}{2\mathrm{i}}(\boldsymbol{Z} - \overline{\boldsymbol{Z}})$.

Any real number, denoted as $x \in \mathbb{R}$, is also a complex number. That is, $\mathbb{R} = \{x \in \mathbb{C} : \mathrm{Im}\, x = 0\}$. In this manuscript we'll sometimes use real numbers interchangeably with their "complex" versions. In other words, we may compose computational graphs that include variables that are defined as real as well as variables that are defined as complex.

When describing a particular neural network architecture, we will also make distinctions between the following elements, any of which may be real or complex:

- Domain or input;
- Internal processing units and weights; and
- Codomain or output.

When we refer to a model that has a complex domain, we may simply say that its inputs are complex-valued. A model that has complex processing units has complex-valued weights. A model with a complex codomain has complex outputs. Such a model is termed "complex-valued."

CHAPTER 2

Audio

Sound is the superposition of one or more oscillations propagated in a medium such as air (American National Standards Institute, 2013).[1] A sensor such as a microphone transduces mechanical energy to electrical current. A digital recorder stores the signal after band-limiting and sampling it at a fixed rate, f_s. The particular value of f_s is linked to the spectral, or frequency, content of the recorded audio signal. Due to the Nyquist-Shannon sampling theorem, f_s must be at least twice as large as the greatest frequency in the signal. The maximum frequency that humans can hear is approximately 20 kHz. Musical sound has frequency content that spans the frequency range of human hearing, which is one of the reasons that the sampling rate of 44.1 kHz has been a recording industry standard for decades. Speech signals, on the other hand, have most of their energy focused about a narrower frequency range; they are often recorded using smaller sampling rates, such as 16 or 8 kHz. Whatever the value of f_s for audio, it is considerably greater than other signals such as natural language (approximately 2.5 words per second); video (24 or 60 frames per second); or electroencephalography (EEG, 240 samples per second).

A "raw audio" recording is represented by a long string of samples, with typically many thousands of samples encoding a few seconds of sound. The high dimensionality of raw audio means that we rarely work with it directly. Rather, raw audio is subjected to a transformation motivated by, among other concepts,

- Reducing dimensionality;
- Representing the data in a domain that is easier to work with; or

[1] For the purposes of this manuscript only sounds that evoke an *auditory sensation* as opposed to, e.g., ultrasound, are considered.

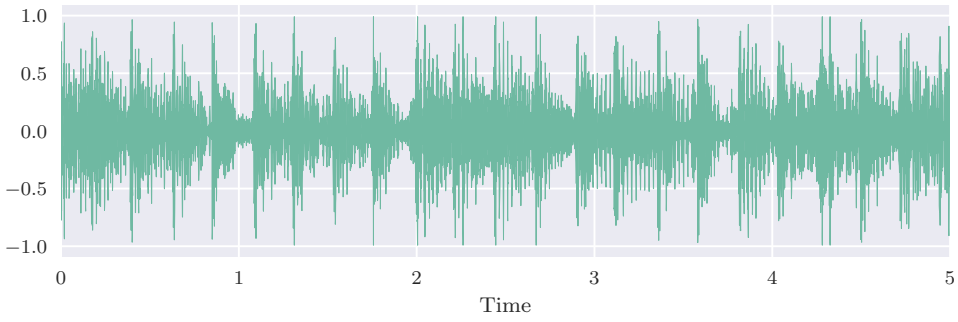

Figure 2.1: Waveform of an audio recording.

- Increasing the salience of signal characteristics that are correlated with an inference task.

The following sections provide an overview of real and complex representations, as well as motivations for using each.

2.1 Waveform

Consider an audio signal $x(t) \in \mathbb{R}$, evaluated at discrete time steps $t = 0, 1, \ldots, T-1$. Figure 2.1 depicts 5 seconds of a monophonic audio recording of pop music. We might decide (naively) to work with the unaltered digital audio waveform directly, expressed as $\begin{bmatrix} x(0) & x(1) & \cdots & x(T-1) \end{bmatrix}^\mathsf{T}$. However there are several reasons why the raw unwindowed waveform may not suffice.

The number of samples, T, is likely to be extraordinarily high. Even with methods designed to handle high-dimensional data, such as convolutional nets, the scale is likely to be too burdensome and problems may arise if we try to train an arbitrary model using the standard backpropagation algorithm. Meanwhile, we usually don't care to make inferences about entire audio files based upon signal characteristics occurring at the duration of microseconds. Therefore we may construct $N = \lfloor T/H \rfloor$ frames of length M from $x(t)$ using a hop size of H, and construct $\boldsymbol{X} \in \mathbb{R}^{M \times N}$ in the following way.

$$\boldsymbol{x}^{(n)} := \begin{bmatrix} x(nH) & x(nH+1) & \cdots & x(nH+M-1) \end{bmatrix}^\mathsf{T} \quad \text{and} \tag{2.1}$$

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}^{(0)} & \boldsymbol{x}^{(1)} & \cdots & \boldsymbol{x}^{(N-1)} \end{bmatrix}. \tag{2.2}$$

For instance, to accomplish full and non-redundant coverage of x we can extract adjacent non-overlapping frames by letting $H = M$. The best choices of H and M are not always immediately obvious and are generally dictated by the task at hand.

Assuming that H is small relative to M and that the value of $\mathrm{f_s}$ is high, the difference in sound between time-adjacent frames $\boldsymbol{x}^{(n)}$ and $\boldsymbol{x}^{(n+1)}$ would normally be imperceptible to a human. Yet any non-trivial waveform is non-stationary. Shifting the location of an arbitrary

frame $\boldsymbol{x}^{(n)}$ would result in a new frame that is numerically dissimilar, even if the two frames are not perceptually distinct. Audio has a correspondence problem, where it is difficult to determine whether vectors of audio are related to each other. For many tasks, it is therefore desirable that either the input to a model or the model itself be robust to such changes, exhibiting "time-shift invariance." One way of learning shift invariance is to use a model with high learning capacity and provide an enormous amount of data during training. For instance (Sainath, Weiss, Senior, Wilson, & Vinyals, 2015) showed that, using approximately 2000 hours of training data, a model given raw waveforms matches the performance of one using a reduced-dimensional frequency-domain time-shift invariant representation. Otherwise one may use a hierarchical approach, modeling dependencies at several temporal scales, as done in SampleRNN (Mehri et al., 2016). Another method is to preprocess the data so that it exhibits some properties of shift invariance, for instance by using the Discrete Fourier Transform.

2.2 Discrete Fourier Transform

The human auditory system acts as a filter bank, decomposing sound stimuli into critical frequency bands. The auditory apparatus helps us to distinguish components of spectral content. However it's not obvious from an unprocessed time-domain waveform which frequencies have the most energy. There's a mismatch between the processing performed by the human auditory system and the waveform representation of a sound. If our learning task corresponds to human perception, as it often does, then a corresponding signal representation, one that is similar to the way humans hear sound, may be considered useful.

We often aim to have an intermediate representation that shows an evolution of time and frequency, which more closely matches human cognition of sound. The usual approach is to segment the waveform into partitions of size M, each having $M \ll T$ (with T defined above) points and scaled by a tapered window function $w(m)$, such as the Hann function. By computing the M-point discrete Fourier transform (DFT, $\mathcal{F}$) on consecutive frames we produce short-term (intra-frame) evolution of amplitude over frequency along with long-term (inter-frame) evolution of frequency amplitude over time.

The DFT of the windowed frame $\boldsymbol{x}^{(n)}$ is an invertible transformation defined as

$$\hat{x}_k^{(n)} := (\mathcal{F}\boldsymbol{x}^{(n)})(k), \tag{2.3}$$

$$= \sum_{m=0}^{M-1} w(m) x_m^{(n)} \mathrm{e}^{-\mathrm{i}2\pi km/M}, \quad \text{and} \tag{2.4}$$

$$\hat{\boldsymbol{x}}^{(n)} = \begin{bmatrix} \hat{x}_0^{(n)} & \hat{x}_1^{(n)} & \dots & \hat{x}_{M-1}^{(n)} \end{bmatrix}^{\mathsf{T}}. \tag{2.5}$$

One can see that $\hat{x}_k^{(n)}$ is simply the cross correlation between the windowed frame and a complex sinusoid having frequency k/M radians per unit time, telling us how well the two signals "match." When the input to a DFT is real-valued (as is the case with audio), then $\hat{x}_k^{(n)} = \overline{\hat{x}}_{M-k}^{(n)}$ and we typically are only interested in the values associated with frequency bins $k = 0, 1, \dots, K-1$, with $K = M/2 + 1$.

The DFT is an invertible transform, and the samples of $\boldsymbol{x}^{(n)}$ can be recovered from the DFT in the following way.

$$x_m^{(n)} := (\mathcal{F}^{-1}\hat{\boldsymbol{x}}^{(n)})(m) \quad \text{and} \tag{2.6}$$

$$= \frac{1}{Mw(m)} \sum_{k=0}^{M-1} \hat{x}_k^{(n)} \mathrm{e}^{\mathrm{i}2\pi km/M}. \tag{2.7}$$

2.2.1 Short-Term Fourier Transform

We may construct a short-term Fourier transform (STFT) from time-adjacent frames, which provides temporal evolution of the spectrum of a waveform. Using $\boldsymbol{X}$ as defined above, we obtain a matrix of time and frequency coefficients,

$$\hat{\boldsymbol{X}} = \begin{bmatrix} \hat{x}_0^{(0)} & \hat{x}_0^{(1)} & \cdots & \hat{x}_0^{(N-1)} \\ \hat{x}_1^{(0)} & \hat{x}_1^{(1)} & \cdots & \hat{x}_1^{(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{x}_{K-1}^{(0)} & \hat{x}_{K-1}^{(1)} & \cdots & \hat{x}_{K-1}^{(N-1)} \end{bmatrix} \tag{2.8}$$

$$\in \mathbb{C}^{K \times N}. \tag{2.9}$$

Figure 2.2 shows the power spectrum in decibels (i.e. $20 \log_{10} \left|\hat{\boldsymbol{X}}\right|$) for the waveform depicted in Fig. 2.1, as well as its phase spectrum in radians.

Visual inspection of the power spectrogram in the upper panel of Fig. 2.2 reveals several interesting details about the recording. Rhythmic syncopation is expressed by wide-band bursts of energy occurring at regular time intervals. Harmonic components are apparent by the presence of energy that is regularly spaced along the frequency axis. Spectra based upon the magnitude coefficients, which do not take into account phase, provide course temporal detail about the signal under analysis. The lower image shows the phase spectra, which provide fine temporal details about the DFT of each audio frame. It's more difficult to visually parse this image, as the phases of the complex coefficients might appear random to the human eye.

2.2.2 Phase Spectra

In many learning and inference tasks the phases $\left(\arg \hat{\boldsymbol{X}}\right)$ of the DFT-transformed signal are discarded and the complex-valued coefficients are treated as real-valued and zero-phase. In discarding the phase, we obtain a representation that is invariant to small shifts in time, which is useful for some scenarios. But in other cases the fine-scale temporal indicators might be important.

In 1981 Alan Oppenheim published "The Importance of Phase in Signals," in which phase-only and magnitude-only reconstructions of images and audio are compared (Oppenheim & Lim, 1981). The conditions under which such signals may be reconstructed were examined and Oppenheim determined that phase plays a higher role than magnitude in intelligibility. One

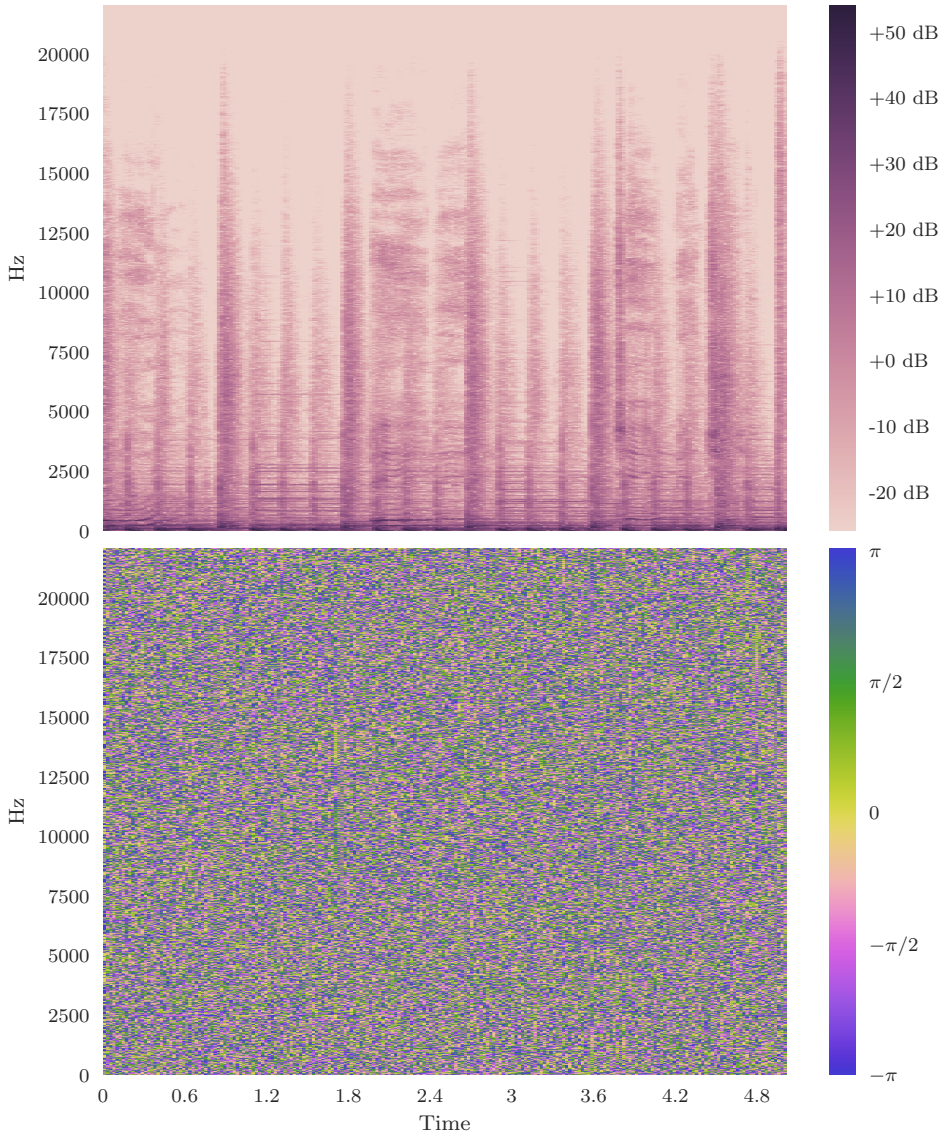

Figure 2.2: STFT corresponding to a few seconds the pop music recording depicted in Fig. 2.1. Top: power spectrogram, in decibels. Harmonics, seen as horizontal bands of energy are apparent in the lower frequencies. Transient sounds, such as drum hits, are represented by bursts of energy across all frequencies. Bottom: phase spectrogram, in radians. It is difficult to visually parse patterns from the phase spectra.

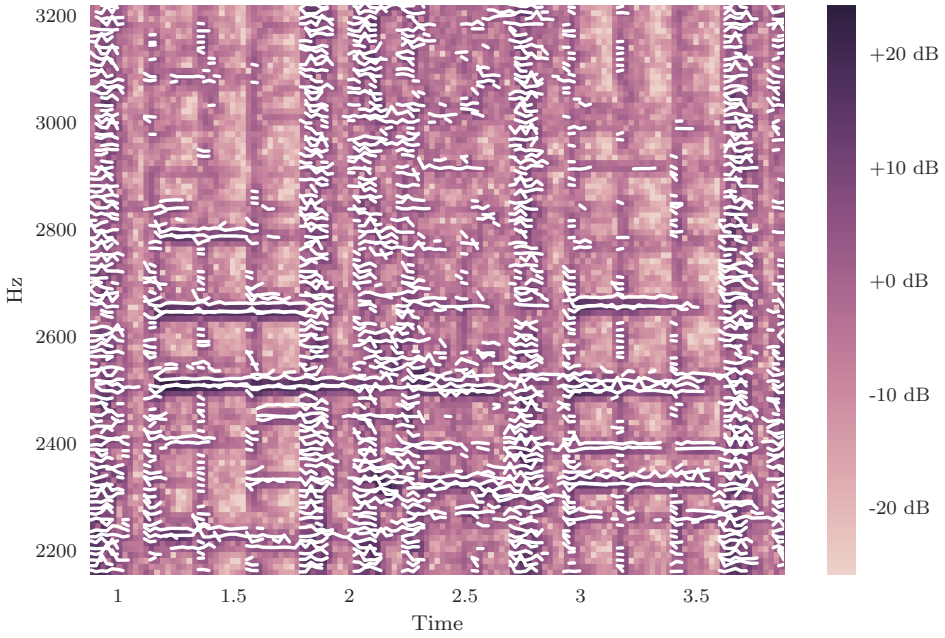

Figure 2.3: Zoomed-in power portion of spectrogram shown in Fig. 2.2. Bins having at least 0 dB energy are overlaid with phase-based frequency estimations (in white). With a small amount of processing, the phase spectra reveal an estimate of the true frequency that each bin's energy is associated with, information which is unavailable in the magnitude spectra.

may argue that for many analysis applications, the relative phases of components may not be informative; in some cases humans are not sensitive to such differences (von Helmholtz, 1863).

Yet when considering sinusoidal components that are near to each other in frequency, humans have a much higher sensitivity to differences in phase (Goldstein, 1967; Mathes & Miller, 1947). It follows that relative phase may play an especially high perceptual role for signals exhibiting wide-band spectral energy, e.g. musical audio, in which energy is distributed more evenly across the spectrum. In addition humans are sensitive to binaural phase differences, e.g. perceived pitches may be induced using binaural recordings of noise (Cramer & Huggins, 1958). In multichannel settings phase differences arising from different sources may yield information about direction of arrival. Hence discarding the phase of a transformed signal amounts to throwing away potentially important information.

For instance, the magnitude coefficients $\left|\hat{\boldsymbol{X}}\right|$ only give us a rough idea of how much energy is associated with each complex sinusoid in the set of Fourier basis functions. Due to spectral leakage and the coarse frequency-domain sampling of the DFT, we cannot make accurate predictions about the true frequencies associated with each bin by relying solely on the magnitude coefficients. However the phase spectra provide big hints about the frequency content. For each frequency bin k in $\hat{\boldsymbol{X}}$ we can estimate the true frequency bin $\hat{k}$ by analyzing

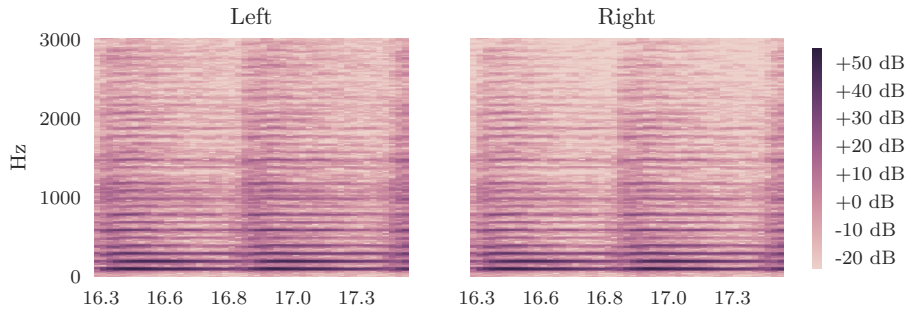

Figure 2.4: Log power spectrograms of a simulated binaural recording, left and right channels.

the differences of the bin phase with respect to time frame (Gunawan & Sen, 2005),

$$\Delta_k = \min_{\delta \in \left\{0, \pm \frac{M}{H}\right\}} \left\{ \left| \frac{M}{H} \left(\frac{\partial(\arg \boldsymbol{X})}{\partial n} \frac{1}{2\pi} \left\lfloor \frac{Hk}{M} \right\rfloor \right) - k + \delta \right| \right\}, \quad \text{given} \tag{2.10}$$

$$\hat{k} = k + \Delta_k. \tag{2.11}$$

Figure 2.3 provides a zoomed-in view of the power spectrogram shown in Fig. 2.2. The estimated frequencies of the spectrogram, computed using Eq. (2.11), are overlaid in white. (Values associated with low-energy bins have been masked for intelligibility.) Harmonic content and minute fluctuations in pitch are clearly visible.

The example shows that important monophonic cues are present in the phase derivative of the STFT of a monophonic recording. Informative phase-related cues are also present in stereophonic recordings. The human head, torso, and outer ears impose spectral filters on sound before it reaches our ears. We rely on these filters for localizing acoustic events. At frequencies above approximately 1.5 kHz, the size and density of the head impose a "shadowing" effect, causing frequency-dependent interaural level differences between the left and right ears. At frequencies below 1.5 kHz, sinusoidal wavelengths are greater than the diameter of the head, allowing them to pass through the head with minimal change in intensity. But there remain frequency-based phase differences between the ears due to differences in time of arrival of individual spectra.

Figure 2.4 shows the log power spectrograms for the left and right channels of a simulated binaural recording. The source audio is an excerpt of a polyphonic chamber music performance. The audio has been convolved with the head-related impulse response of a human subject. The impulse response encodes the subject's head filter corresponding to sounds arriving at an azimuthal angle of 15° and elevation of 0°. In other words, the recording has been manipulated to simulate the signal as if it were recorded in the subject's ears, and if the performance were being displayed at eye level and just to the right of the subject's head.

The details of the left and right log power spectrograms are very similar, with the right channel exhibiting a slightly reduced level of overall energy. Figure 2.5 shows the difference between the left and right log spectrograms in the top axes. The relative energy of the left channel is slightly higher, especially in some frequency bands above 1 kHz. However, not

much detail can be discerned. The inter-channel phase differences are plotted as arrows in the bottom axes, where it is more obvious that the two channels have nearly constant per-channel phase differences and group delay.

2.3 Other Representations

The analytic signal is one which has no negative frequency components. It allows us to observe the instantaneous phase and amplitude at any sample by computing its argument and modulus (Picinbono, 1997). It also yields, via first order derivative, the instantaneous frequency of the signal. The analytic signal of $x(t)$ is expressed with

$$s(t) = x(t) + \mathrm{i}\mathcal{H}x(t), \tag{2.12}$$

where $\mathcal{H}$ represents the Hilbert transform,

$$\mathcal{H}x(t) = -\frac{1}{\pi} \lim_{\varepsilon \to 0} \int_{\varepsilon}^{\infty} \frac{x(t+\tau) - x(t-\tau)}{\tau} \, \mathrm{d}\tau \, . \tag{2.13}$$

When x is a real discrete-time signal, the discrete-time analytic signal of a frame $\boldsymbol{x}^{(n)}$ (as defined above) is composed by using an approximation of the Hilbert transform, the DFT, and its inverse.

$$\hat{s}_k^{(n)} = \begin{cases} 2\big(\big(\mathcal{F}\boldsymbol{x}^{(n)}\big)(k)\big) & \text{if} \quad 1 \le k \le M/2 - 1 \\ \big(\mathcal{F}\boldsymbol{x}^{(n)}\big)(k) & \text{if} \quad k \in \{0, M/2\} \\ 0 & \text{otherwise} \end{cases} , \tag{2.14}$$

$$\hat{\boldsymbol{s}}^{(n)} = \begin{bmatrix} \hat{s}_0^{(n)} & \hat{s}_1^{(n)} & \cdots & \hat{s}_{M-1}^{(n)} \end{bmatrix}^{\mathsf{T}}, \quad \text{and} \tag{2.15}$$

$$\boldsymbol{s}^{(n)} = \Big(\mathcal{F}^{-1}\hat{\boldsymbol{s}}^{(n)}\Big). \tag{2.16}$$

Other complex transformations for audio include the complex wavelet (Kingsbury, 2001) and empirical mode decomposition (N. E. Huang et al., 1998). This manuscript focuses on waveforms and their frequency-domain transforms.

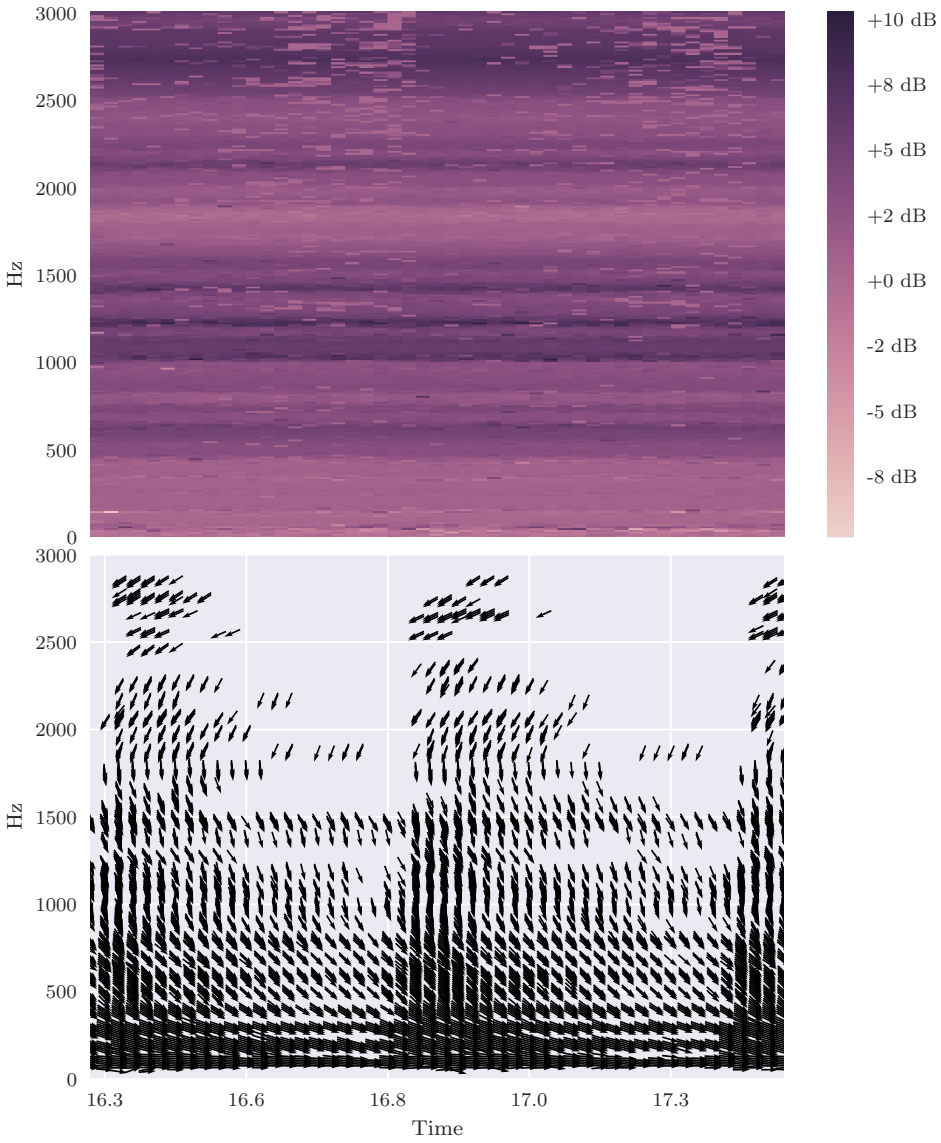

Figure 2.5: Top: difference of left and right log power spectrograms of a simulated binaural recording. Bottom: difference of phase spectrograms.

CHAPTER 3

Complex Neural Networks

Complex networks have been largely ignored by all except a few niche research communities, in spite of an accelerating flurry of research on deep real networks. Nevertheless the early history of complex networks, which is summarized in the following section, dates to the late 1980s. The following sections present activation functions and objective functions. We next review ways to optimize complex networks, along with how the Wirtinger Calculus may ease the implementation of certain types of networks. We also contribute a modification to the Adam algorithm (Kingma & Ba, 2014) for computing second-order statistics of the complex gradient. Finally, the relative merits of various input and output representations are discussed in the context of a toy waveform synthesis task.

3.1 History

One of the earliest examples of a trainable neural network was Rosenblatt's perceptron (Rosenblatt, 1958). It subsequently influenced the ADALINE machine (Widrow & Hoff, 1960) for optimizing the coefficients of a linear adaptive filter. ADALINE was a single-layer single-neuron machine and it utilized least mean square (LMS) along with a stochastic gradient descent algorithm for learning an optimal configuration of weights. We describe it here. Denote inputs and weights as $\boldsymbol{x} \in \mathbb{R}^N$ and $\boldsymbol{w} \in \mathbb{R}^N$. The model infers

$$\hat{y} = \boldsymbol{x} \cdot \boldsymbol{w}. \tag{3.1}$$

Given a target output $y \in \mathbb{R}$. We denote error e and objective function $\mathcal{L}$. We wish to find the value of $\boldsymbol{w}$ that minimizes $\mathcal{L}$.

$$e = y - \hat{y}, \tag{3.2}$$
$$\mathcal{L} = e^2, \quad \text{and} \tag{3.3}$$
$$\boldsymbol{w} \leftarrow \arg\min_{\boldsymbol{w}}(\mathcal{L}). \tag{3.4}$$

The LMS algorithm optimizes $\mathcal{L}$ using gradient descent.

$$\boldsymbol{\nabla}\mathcal{L}_{\boldsymbol{w}} := -2e\boldsymbol{x} \quad \text{and} \tag{3.5}$$
$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha e\boldsymbol{x}, \tag{3.6}$$

where α is a learning rate that absorbs the scaling factor of 2.

Widrow, McCool, and Ball (1975) extended the LMS algorithm to the complex domain by providing a steepest descent derivation with respect to real and imaginary parts. Brandwood (1983) generalized the theory using descent with respect to complex variables, rather than their real components. His derivation, although uncited as such, depends on the Wirtinger Calculus (Wirtinger, 1927)[1], a complex-valued expression of the gradient that maintains an equivalence with the gradients of the real and imaginary components. The Wirtinger Calculus, which is discussed in more detail in Section 3.4, gives an equivalent but simplified method for deriving the gradient with respect to complex-valued functions. Denoting inputs, target, and weights in the complex domain, $\boldsymbol{x}, \boldsymbol{w} \in \mathbb{C}^N$ and $y \in \mathbb{C}$, the objective function, gradient, and update become

$$\mathcal{L} = e\overline{e}, \tag{3.7}$$
$$\boldsymbol{\nabla}\mathcal{L}_{\boldsymbol{w}} = -2e\overline{\boldsymbol{x}}, \quad \text{and} \tag{3.8}$$
$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha e\overline{\boldsymbol{x}}. \tag{3.9}$$

Despite Brandwood's early efforts, until recently much of the literature eschewed the Wirtinger Calculus in favor of more cumbersome derivations with respect to real and imaginary parts.

Noest (1987, 1988a, 1988b) is credited with introducing phasor networks in 1987, the first nonlinear neural networks defined over the field of complex numbers. Continuous and discrete phasors were presented in (Noest, 1988a) and (Noest, 1988b) in the context of associative retrieval. Kuroe, Hashimoto, and Mori (2001a, 2001b) analyzed the associative abilities of multivalued and continuous networks on synthetic data, as did Nemoto and Kubono (1996). Donq-Liang and Wen-June (1998) studied complex bidirectional associative memories, while Müezzinoğlu, Güzeliş, and Zurada (2003) investigated Hopfield-type complex networks. Jankowski, Lozowski, and Zurada (1996) used associative complex networks to recall gray-scale images. Kuroe and Taniguchi (2007) gave an analysis of performance of several types of complex associative memories.

One can trace the history of nonlinear complex-valued activation functions further back to Soviet Russia where, in the early 1970s, Naum Aizenberg and colleagues developed a theory of multithreshold logic defined for the field of complex numbers (N. N. Aizenberg & Ivas'kiv,

[1] Since the "Wirtinger" Calculus can be traced back to several authors, it is more generally described as the "$\mathbb{CR}$-Calculus" (or "Complex-Real Calculus") by Kreutz-Delgado (2009).

1977; N. N. Aizenberg, Ivas'kiv, & Pospelov, 1971; N. N. Aizenberg, Ivas'kiv, Pospelov, & Khudyakov, 1971, 1973). Due to restrictions placed by the USSR on global dissemination of scholarly work, complex domain multithreshold logic did not reach a wide audience and thus Noest's work was developed independently from Aizenberg's. From the early 1990s complex-parameterized networks and their variations were increasingly studied.

By showing that, with backpropagation, neural networks learn useful hidden features, Rumelhart, Hinton, and Williams (1986) paved the way for investigating other architectures. In the early 1990s several researchers independently proposed variations of the backpropagation algorithm for complex networks. Little, Gustafson, and Senn (1990) proposed a network that combined real parameters with real activations. M. S. Kim and Guest (1990) gave a derivation for complex nets having a single hidden layer. They showed that, just as in real nets, the XOR problem (Minsky & Papert, 1969) may be solved with a complex network having one hidden layer and sigmoidal activation functions.[2]

Leung and Haykin (1991) gave a derivation of the complex backpropagation algorithm using sigmoid functions. Benvenuto, Marchesi, Piazza, and Uncini (1991a), Benvenuto and Piazza (1992) gave their derivations for the real-imaginary split activation approach and apply their network to channel equalization of nonlinearly distorted satellite communications (Benvenuto et al., 1991b). Georgiou and Koutsougeras (1992) derive backpropagation using a novel non-holomorphic activation function. Nitta and Furuya (1991, in Japanese) and Nitta (1993a, 1993b, in English) also use the split real-imaginary approach and perform geometric transformations in the complex domain, which the authors claim cannot be satisfactorily reproduced in the real domain.

Donald Birx wrote a PhD thesis on the design of complex networks for eddy current characterization (Birx, 1990). John Sutherland introduced holographic neural networks (Sutherland, 1990). Clarke (1990) explored the relationship between real and complex systems. As an example of the utility of complex nets, Clarke suggests that a complex neural network should be able to readily determine whether an element is in a Mandelbrot set. The real time recurrent learning algorithm (Williams & Zipser, 1989) was broadened to the complex domain by Kechriotis and Manolakos (1994) using a split complex approach, where the real and imaginary parts are treated as independent.

Van den Bos (1994) extended Brandwood (1983)'s complex gradient with the complex Hessian, paving the way for second-order methods of optimization. Mandic and Goh (2009) have advocated "augmented," or widely linear, methods in order to accurately model second-order complex statistics, for instance in S. L. Goh and Mandic (2007), without which it's more difficult to capture complex impropriety (Schreier & Scharf, 2010). There is limited guidance on regularization of complex neural networks, but Hirose and Onishi (1999) advocate using noise.

More recently Zhang and Mandic (2016) suggest a complex learning rate. Trabelsi et al. (2017) are perhaps the first authors to advocate training deep partially complex networks. They provide complex extensions to batch normalization and weight initialization techniques that have been used in real networks.

[2] Igor Aizenberg demonstrates that, unlike its real counterpart, the XOR problem can be solved by a single complex neuron (I. N. Aizenberg, 2011).

3.2 Activation Functions

The basic building blocks of neural networks are affine transformations, followed by nonlinear "activation" functions which help lend a higher degree of expressivity to the model. Suppose we have input $\boldsymbol{x} \in \mathbb{C}^M$ and weights $\boldsymbol{W} \in \mathbb{C}^{N \times M}$, with M and N denoting the number of input and output dimensions, respectively. Then the output $\boldsymbol{y} \in \mathbb{C}^N$ of an arbitrary layer is determined by

$$\boldsymbol{z} = \boldsymbol{W}\boldsymbol{x} \quad \text{and} \tag{3.10}$$

$$\boldsymbol{y} = f(\boldsymbol{z}), \tag{3.11}$$

where f is often a nonlinear element-wise activation function.

Neural networks are known as universal function approximators, i.e., given enough hidden units, they can approximate any real continuous function to an arbitrary degree of precision. The first proofs of real neural networks as universal approximators were provided using "squashing" activation functions, typically sigmoids, which are bounded and monotonic (Barron, 1994; Cybenko, 1989; Funahashi, 1989; Hornik, Stinchcombe, & White, 1989). Entire functions are complex functions that are holomorphic everywhere. Liouville's theorem states that every bounded entire function must be a constant, which implies that we cannot have a complex network that is entire and that also utilizes the squashing activation functions.

Since the first complex neural networks were developed, authors have grappled with how to choose or design complex activation functions. If a complex-differentiable activation is desired, then one must forgo boundedness. Otherwise one may choose a real-differentiable activation. Clarke (1990) and other authors have proposed using holomorphic functions, along with strategies for avoiding singularities in the complex plane. Models using these activation functions are described here as "fully complex". There is no consensus on best activation functions for real or complex networks. In general, we seek a function that is nonlinear and that doesn't lead to vanishing or exploding gradients during training.

Figure 3.1 shows surface plots of the real and imaginary parts of the hyperbolic tangent function. The hyperbolic tangent function, which is fully complex, is a commonly used nonlinearity in real networks and has been promoted for complex networks in several manuscripts, e.g., (Mandic & Goh, 2009). It can be seen that values lying along the imaginary axis of the input cause regularly spaced singularities in the output. In order to use such an activation function, one must ensure that the input is scaled appropriately to avoid explosively large values.

Others have suggested that it is an unnecessarily strict constraint to require holomorphic activation functions. They suggest that it is sufficient to use activation functions that are differentiable with respect to their real and imaginary parts. The proposed methods include split real-imaginary and split amplitude-phase, referred respectively as "Type A" and "Type B" by Kuroe, Yoshid, and Mori (2003). Benvenuto and Piazza (1992), among many others, used a split real-imaginary approach with sigmoid functions on the real and imaginary parts of the signal. Figure 3.2 shows the surface plot of a split real-imaginary hyperbolic tangent function.

Georgiou and Koutsougeras (1992) proposed a split amplitude-phase nonlinear activation function (Fig. 3.3) that is phase-preserving and magnitude squashing. They make several

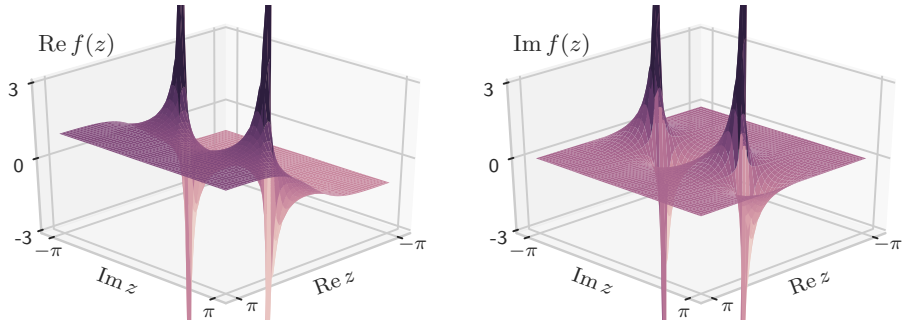

Figure 3.1: Surface plots for the real and imaginary parts of the hyperbolic tangent function.

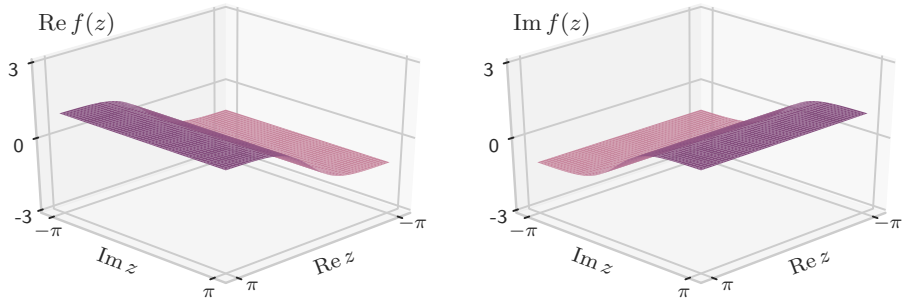

Figure 3.2: Surface plots for the real and imaginary parts of the split real-imaginary hyperbolic tangent function.

arguments against a fully complex approach, insisting that activation functions should be nonlinear, bounded, and have partial derivatives with respect to the real and imaginary parts. Hirose (1992a) proposed an alternate phase-preserving and magnitude squashing function.

The activation function of Georgiou and Koutsougeras is nonlinear with respect to magnitude and linear with respect to phase; there is also a sizable class of functions that are linear with respect to magnitude and nonlinear with respect to phase. These functions, introduced as phasor networks (Noest, 1988a), usually produce values on the origin and the unit circle (Hirose, 1992b; Jankowski et al., 1996; Miyajima, Baisho, Yamanaka, Nakamura, & Masahiro, 2000; Nemoto & Saito, 2002; Zemel, Williams, & Mozer, 1995). Takeda and Kishigami (1992, 1993) suggested similar networks motivated by the physical modeling of optical systems. Multivalued neurons, formulated from the multivalued threshold logic by Aizenberg and Aizenberg, also follow a similar paradigm (N. N. Aizenberg & Aizenberg, 1992).

Most early papers advocated split approaches using non-holomorphic functions. Yet fully complex activation functions lend themselves to complex gradient-preserving backpropagation. Kim and Adalı suggest that some elementary transcendental functions (ETFs) with isolated singularities are preferred for training neural networks if the domain of interest does not include the singularity (T. Kim & Adalı, 2001, 2002, 2003). Suggested ETFs and their derivatives are shown in Table 3.1.

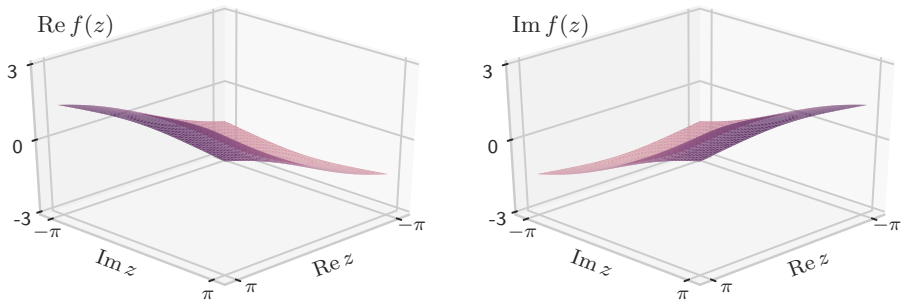

Figure 3.3: Surface plots for the real and imaginary parts of the split phase-preserving function proposed by (Georgiou & Koutsougeras, 1992).

Table 3.1: Elementary transcendental functions and their derivatives.

$f(z)$	$\frac{\mathrm{d}f}{\mathrm{d}z}$
Circular	
$\tan z = \frac{\mathrm{e}^{\mathrm{i}z}-\mathrm{e}^{-\mathrm{i}z}}{\mathrm{i}(\mathrm{e}^{\mathrm{i}z}+\mathrm{e}^{-\mathrm{i}z})}$	$\frac{\mathrm{d}}{\mathrm{d}z}\tan z = \sec^2(z)$
$\sin z = \frac{\mathrm{e}^{\mathrm{i}z}-\mathrm{e}^{-\mathrm{i}z}}{2\mathrm{i}}$	$\frac{\mathrm{d}}{\mathrm{d}z}\sin z = \cos z$
Inverse circular	
$\arctan z = \frac{1}{2}\mathrm{i}[\ln(1-\mathrm{i}z) - \ln(1+\mathrm{i}z)]$	$\frac{\mathrm{d}}{\mathrm{d}z}\arctan z = \frac{1}{1+z^2} : z \neq \pm\mathrm{i}$
$\arcsin z = -\mathrm{i}\ln\left(\mathrm{i}z + \sqrt{1-z^2}\right)$	$\frac{\mathrm{d}}{\mathrm{d}z}\arcsin z = \frac{1}{\sqrt{1-z^2}} : z \neq \pm 1$
$\arccos z = \frac{1}{2}\pi + \mathrm{i}\ln\left(\mathrm{i}z + \sqrt{1-z^2}\right)$	$\frac{\mathrm{d}}{\mathrm{d}z}\arccos z = -\frac{1}{\sqrt{1-z^2}} : z \neq \pm 1$
Hyperbolic	
$\operatorname{arctanh} z = \frac{1}{2}[\ln(1+z) - \ln(1-z)]$	$\frac{\mathrm{d}}{\mathrm{d}z}\operatorname{arctanh} z = \frac{1}{1-z^2} : z \neq \pm 1$
$\operatorname{arcsinh} z = \ln\left(z + \sqrt{1+z^2}\right)$	$\frac{\mathrm{d}}{\mathrm{d}z}\operatorname{arcsinh} z = \frac{1}{\sqrt{1+z^2}} : z \neq \pm 1$
Inverse hyperbolic	
$\operatorname{arctanh} z = \frac{1}{2}[\ln(1+z) - \ln(1-z)]$	$\frac{\mathrm{d}}{\mathrm{d}z}\operatorname{arctanh} z = \frac{1}{1-z^2} : z \neq \pm 1$
$\operatorname{arcsinh} z = \ln\left(z + \sqrt{1+z^2}\right)$	$\frac{\mathrm{d}}{\mathrm{d}z}\operatorname{arcsinh} z = \frac{1}{\sqrt{1+z^2}} : z \neq \pm 1$

Table 3.2: Common and recently proposed activation functions for complex networks. Elementary transcendental functions (ETFs) are holomorphic and all others are not. ReLU refers to the rectified linear unit.

activation	$f(z) =$
ETFs (T. Kim & Adalı, 2003)	see Table 3.1
Georgiou and Koutsougeras (1992)[a]	$\frac{z}{c+\frac{1}{r}\lvert z\rvert}$
Hirose (1992a)[b]	$\tanh\left(\frac{\lvert z\rvert}{m}\right)e^{\mathrm{i}\arg z}$
Type A (Kuroe et al., 2003)[c]	$f^{(r)}(\mathrm{Re}\, z) + \mathrm{i} f^{(i)}(\mathrm{Im}\, z)$
Type B (Kuroe et al., 2003)[d]	$\psi(\lvert z\rvert)e^{\mathrm{i}\varphi(\arg z)}$
modReLU (Arjovsky et al., 2016)[e]	$\mathrm{ReLU}(\lvert z\rvert + b)e^{\mathrm{i}\arg z}$
zReLU (Guberman, 2016)	$\begin{cases} z & \text{if} \quad \arg z \in [0, \pi/2] \\ 0 & \text{otherwise} \end{cases}$
$\mathbb{C}$ReLU (Trabelsi et al., 2017)	$\mathrm{ReLU}(\mathrm{Re}\, z) + \mathrm{i}\,\mathrm{ReLU}(\mathrm{Im}\, z)$

[a] c and r are constants.
[b] m is a constant.
[c] $f^{(r)}$ and $f^{(i)}$ are nonlinear real functions.
[d] ψ and φ are nonlinear non-negative real functions.
[e] b is a trainable bias parameter.

Radial basis functions are found in complex networks (Cha & Kassam, 1995; S. Chen, McLaughlin, & Mulgrew, 1994a, 1994b; Jianping, Sundararajan, & Saratchandran, 2002). Uncini, Vecci, Campolucci, and Piazza (1999) proposed spline-based activation functions, which was explored further by other authors (Vitagliano, Parisi, & Uncini, 2003). More recently Scardapane, Vaerenbergh, Hussain, and Uncini (2018) have proposed nonparametric kernel activation functions, on the real and imaginary components, or directly in the complex domain.

Clifford Networks generalize the K-algebras, such as real, complex, and quaternion (Buchholz, 2005; Pearson, 1995). Hyperbolic networks, in which the imaginary unit i is redefined as $\mathrm{i}^2 = +1$ are investigated in (Buchholz & Sommer, 2000; Nitta & Buchholz, 2008). There is an extensive literature on networks implementing hyperspheres, quaternions, qubits, multidimensional neurons, and others. The scope of this work is limited to networks implementing complex neurons. There is no consensus in the literature concerning which activation functions are suited for which applications.

The rectified linear unit (ReLU) (Hahnloser, Sarpeshkar, Mahowald, Douglas, & Seung, 2000) has become very popular for deep real networks,

$$\mathrm{ReLU}(x) := \max(x, 0), \tag{3.12}$$

as it avoids vanishing gradients associated with sigmoidal activations. At the same time, recent literature on complex networks has sought to utilize the ReLU. Arjovsky et al. (2016) proposed adding a trainable bias parameter to the modulus of a complex number, after which the ReLU is applied. In the recent paper by Trabelsi et al. (2017), the ReLU is applied independently on the real and imaginary components of a complex number. The

first approach has the effect of preserving phase while nonlinearly transforming magnitude. The latter approach provides a nonlinear mapping of phase when the input argument lies anywhere other than the upper right quadrant of the Argand plane. Neither type of activation is holomorphic. Table 3.2 provides an overview of these and other activation functions mentioned in this chapter.

3.3 Loss Functions

Much of the existing literature on complex networks focuses on adaptive filtering tasks, where the mean squared error is the preferred cost function. Given a target $\boldsymbol{y}$ and model output $\hat{\boldsymbol{y}}$, both in $\mathbb{C}^N$, and error

$$\boldsymbol{e} := \boldsymbol{y} - \hat{\boldsymbol{y}}, \tag{3.13}$$

the complex mean squared loss function is defined as:

$$\mathcal{L}(\boldsymbol{e}) = \sum_{i=0}^{N-1} |e_i|^2 \tag{3.14}$$

$$= \sum_{i=0}^{N-1} e_i \overline{e}_i. \tag{3.15}$$

Equation (3.15) is a non-negative scalar real-valued function that tends to zero as the magnitude of the complex error tends to zero. Alternatively Savitha, Suresh, and Sundararajan (2013) have proposed replacing Eq. (3.13) with the error between the log targets and log outputs,

$$\boldsymbol{e} := \log \hat{\boldsymbol{y}} - \log \boldsymbol{y}. \tag{3.16}$$

The loss function now reduces to

$$\mathcal{L}(\boldsymbol{e}) = (\log |\hat{y}_i| - \log |y_i|)^2 + (\arg \hat{y}_i - \arg y_i)^2. \tag{3.17}$$

Equation (3.17) has the nice property that the magnitude and phase errors are represented explicitly in the objective function.

Equations (3.13) and (3.16) may be appropriate error functions for a complex-valued regression model. What objective function should we use for a classification model that maps complex numbers to real numbers? There is no agreed-upon method, but one option is to project the outputs of the last layer to the real domain through a (necessarily non-holomorphic) transform and use any of the preexisting loss functions for real nets, such as cross entropy. We study this approach in Chapter 4.

3.4 Optimization

3.4.1 The Complex Gradient and Wirtinger Calculus

Brandwood and Van den Bos formulated early derivations of the complex gradient, Jacobian, and Hessian using the Wirtinger Calculus (Brandwood, 1983; van den Bos, 1994). Wirtinger (Wirtinger, 1927) provided an equivalent formalism that makes computing the derivative of complex-valued functions with respect to holomorphic and real-analytic non-holomorphic functions less burdensome.[3] The Wirtinger Calculus helps simplify cumbersome statements about the derivatives of complex functions by letting us describe them entirely within the complex field, rather than with respect to the real and imaginary components. Despite its apparent convenience, until recently few authors have opted to illustrate backpropagation of complex neural networks using the Wirtinger Calculus. The published non-Wirtinger derivations can be unwieldy to read. The Wirtinger Calculus appears to be gaining favor, with newer publications promoting its use, e.g. (Amin, Amin, Al-Nuaimi, & Murase, 2011), Minin's doctoral dissertation (Minin, 2012), and the newest edition of Haykin's *Adaptive Filter Theory* (Haykin, 2014). The following paragraphs summarize Wirtinger calculus and the associated computational graph, largely following Kreutz-Delgado (2009)'s exposition.

Denote $z \in \mathbb{C}$ and $x, y \in \mathbb{R}$ with $z = x + \mathrm{i}y$. We define several related functions of z, x, and y,

$$f(z) := f(z, \overline{z}) \tag{3.18}$$
$$= g(x, y) \tag{3.19}$$
$$= u(x, y) + \mathrm{i}v(x, y). \tag{3.20}$$

The second definition of f in Eq. (3.18) suggests that we will treat the variable z and its conjugate $\overline{z}$ as independent variables, even though they are clearly not independent of one another. Using this definition, the $\mathbb{R}$-derivative and $\overline{\mathbb{R}}$-derivative of f are defined as

$$\left.\frac{\partial f}{\partial z}\right|_{\overline{z} \text{ is constant}} \quad \text{and} \tag{3.21}$$
$$\left.\frac{\partial f}{\partial \overline{z}}\right|_{z \text{ is constant}} \tag{3.22}$$

Note that the $\mathbb{R}$-derivative and $\overline{\mathbb{R}}$-derivative are formalisms, as z cannot be independent of $\overline{z}$. However one is treated as constant when computing the derivative of other, applying the normal rules of calculus. Using these definitions, (Brandwood, 1983) shows that

$$\frac{\partial f}{\partial z} = \frac{1}{2}\left(\frac{\partial f}{\partial x} - \mathrm{i}\frac{\partial f}{\partial y}\right) \quad \text{and} \tag{3.23}$$
$$\frac{\partial f}{\partial \overline{z}} = \frac{1}{2}\left(\frac{\partial f}{\partial x} + \mathrm{i}\frac{\partial f}{\partial y}\right). \tag{3.24}$$

[3] Real analytic functions have partial derivatives with respect to their real and imaginary parts but are not necessarily complex-analytic.

Observe that the $\overline{\mathbb{R}}$-derivative is equal to zero for any holomorphic function. Recall the Cauchy-Riemann equations which state that for the complex derivative of $f(z) = g(x, y) = u(x, y) + \mathrm{i}v(x, y)$ to exist, the following identities must hold,

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \quad \text{and} \tag{3.25}$$

$$\frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y}. \tag{3.26}$$

If we expand Eq. (3.24) and substitute the Cauchy-Riemann equations, then the $\overline{\mathbb{R}}$-derivative vanishes. Thus an equivalent (and intuitive) statement about a holomorphic function is that it does not depend on the conjugate of its input. It is further shown by Brandwood that if $f : \mathbb{C} \to \mathbb{R}$ is a real-valued scalar function, either $\frac{\partial f}{\partial z} = 0$ or $\frac{\partial f}{\partial \overline{z}} = 0$ is a necessary and sufficient condition for f to have a stationary point. By extension if $f : \mathbb{C}^N \to \mathbb{R}$ is a real-valued function of a complex vector $\boldsymbol{z} = [z_0 \; z_1 \; \cdots \; z_{N-1}]^\mathsf{T}$ and we define the cogradient and conjugate cogradient as

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{z}} &:= \begin{bmatrix} \frac{\partial}{\partial z_0} & \frac{\partial}{\partial z_1} & \cdots & \frac{\partial}{\partial z_{N-1}} \end{bmatrix} \quad \text{and} \\ \frac{\partial}{\partial \overline{\boldsymbol{z}}} &:= \begin{bmatrix} \frac{\partial}{\partial \overline{z}_0} & \frac{\partial}{\partial \overline{z}_1} & \cdots & \frac{\partial}{\partial \overline{z}_{N-1}} \end{bmatrix}, \end{aligned} \tag{3.27}$$

then $\frac{\partial}{\partial \boldsymbol{z}} f = 0$ or $\frac{\partial}{\partial \overline{\boldsymbol{z}}} f = 0$ are necessary and sufficient to determine a stationary point.

If f is a function of a complex vector $\boldsymbol{z}$, its total differential is

$$\mathrm{d}f = \frac{\partial f}{\partial \boldsymbol{z}} \mathrm{d}\boldsymbol{z} + \frac{\partial f}{\partial \overline{\boldsymbol{z}}} \mathrm{d}\overline{\boldsymbol{z}}\,. \tag{3.28}$$

If f is real, as it is whenever it is a neural objective function that we wish to optimize, then we have

$$\mathrm{d}f = 2\,\mathrm{Re}\left\{ \frac{\partial f}{\partial z} \mathrm{d}z \right\}. \tag{3.29}$$

Defining the gradient operator as

$$\boldsymbol{\nabla}_{\boldsymbol{z}}(\cdot) := \left(\frac{\partial (\cdot)}{\partial \overline{\boldsymbol{z}}} \right)^\mathsf{T} \quad \text{and} \tag{3.30}$$

$$= \left(\frac{\partial (\cdot)}{\partial \boldsymbol{z}} \right)^*, \tag{3.31}$$

it can be shown using the Cauchy-Schwarz inequality that f has the greatest rate of change when it is in the direction of the gradient, Eq. (3.31). Kreutz-Delgado clarifies that the preceding only holds under the assumption of a Euclidean space[4]. With these definitions we can perform gradient descent using a real-valued cost function of complex arguments, even if some inner elements of the function are not holomorphic.

[4] The derivation in (Kreutz-Delgado, 2009) provides a more general framework, without posing assumptions about metric space.

Equation (3.28) should provide a hint that for a generic complex network, we need to keep track of two partial derivatives for each function in the computational graph. Doing so can be quite burdensome, especially for a composition of many functions. E.g., consider the composition of two vector-valued complex functions $f(\boldsymbol{g}(\boldsymbol{z}, \overline{\boldsymbol{z}}))$. Its differential can be characterized as,

$$\mathrm{d}f = \left(\frac{\partial f}{\partial \boldsymbol{g}}\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{z}} + \frac{\partial f}{\partial \overline{\boldsymbol{g}}}\frac{\partial \overline{\boldsymbol{g}}}{\partial \boldsymbol{z}}\right)\mathrm{d}\boldsymbol{z} + \left(\frac{\partial f}{\partial \boldsymbol{g}}\frac{\partial \boldsymbol{g}}{\partial \overline{\boldsymbol{z}}} + \frac{\partial f}{\partial \overline{\boldsymbol{g}}}\frac{\partial \overline{\boldsymbol{g}}}{\partial \overline{\boldsymbol{z}}}\right)\mathrm{d}\overline{\boldsymbol{z}}\,. \tag{3.32}$$

However there are several ways that we can simplify things. Lets consider what happens if f is a real-valued scalar objective function. Then, according to Eq. (3.31), and as Drude, Raj, and Haeb-Umbach (2016) point out, we only care about the left term on the right hand side of Eq. (3.32), which becomes

$$\mathrm{d}f \propto \left(\frac{\partial f}{\partial \boldsymbol{g}}\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{z}} + \frac{\partial f}{\partial \overline{\boldsymbol{g}}}\frac{\partial \overline{\boldsymbol{g}}}{\partial \boldsymbol{z}}\right)\mathrm{d}\boldsymbol{z}\,. \tag{3.33}$$

Now suppose rather that f is a holomorphic function. Considering that a holomorphic function cannot depend on the conjugate of its input, Eq. (3.32) simplifies to

$$\mathrm{d}f = \left(\frac{\partial f}{\partial \boldsymbol{g}}\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{z}}\right)\mathrm{d}\boldsymbol{z}\,. \tag{3.34}$$

For a more concrete example, consider a complex-valued function,

$$\boldsymbol{g}(\boldsymbol{z}, \overline{\boldsymbol{z}}) = \begin{bmatrix} g_0(\boldsymbol{z}, \overline{\boldsymbol{z}}) & g_1(\boldsymbol{z}, \overline{\boldsymbol{z}}) & \cdots & g_{M-1}(\boldsymbol{z}, \overline{\boldsymbol{z}}) \end{bmatrix}^{\mathsf{T}}, \quad \text{given} \tag{3.35}$$

$$\boldsymbol{z} = \begin{bmatrix} z_0 & z_1 & \cdots & z_{N-1} \end{bmatrix}^{\mathsf{T}} \quad \text{and} \tag{3.36}$$

$$\overline{\boldsymbol{z}} = \begin{bmatrix} \overline{z}_0 & \overline{z}_1 & \cdots & \overline{z}_{N-1} \end{bmatrix}^{\mathsf{T}}, \tag{3.37}$$

where we define the Jacobian and conjugate Jacobian matrices

$$\boldsymbol{J}_{\boldsymbol{g}} := \frac{\partial \boldsymbol{g}(\boldsymbol{z}, \overline{\boldsymbol{z}})}{\partial \boldsymbol{z}} \tag{3.38}$$

$$= \begin{bmatrix} \frac{\partial g_0(\boldsymbol{z}, \overline{\boldsymbol{z}})}{\partial \boldsymbol{z}} \\ \frac{\partial g_1(\boldsymbol{z}, \overline{\boldsymbol{z}})}{\partial \boldsymbol{z}} \\ \vdots \\ \frac{\partial g_{M-1}(\boldsymbol{z}, \overline{\boldsymbol{z}})}{\partial \boldsymbol{z}} \end{bmatrix} \quad \text{and} \tag{3.39}$$

$$\boldsymbol{J}^{(c)}_{\boldsymbol{g}} := \frac{\partial \boldsymbol{g}(\boldsymbol{z}, \overline{\boldsymbol{z}})}{\partial \overline{\boldsymbol{z}}} \tag{3.40}$$

$$= \begin{bmatrix} \frac{\partial g_0(\boldsymbol{z}, \overline{\boldsymbol{z}})}{\partial \overline{\boldsymbol{z}}} \\ \frac{\partial g_1(\boldsymbol{z}, \overline{\boldsymbol{z}})}{\partial \overline{\boldsymbol{z}}} \\ \vdots \\ \frac{\partial g_{M-1}(\boldsymbol{z}, \overline{\boldsymbol{z}})}{\partial \overline{\boldsymbol{z}}} \end{bmatrix}. \tag{3.41}$$

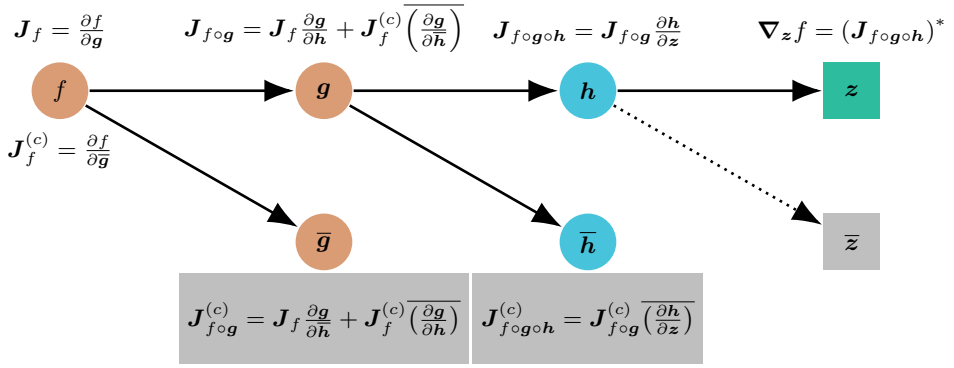

Figure 3.4: Computational dependency graph for a composition of holomorphic and non-holomorphic functions. Tan circular nodes: non-holomorphic functions. Blue circular nodes: holomorphic functions. Rectangular nodes: inputs. The non-holomorphic functions are dependent on their inputs and the conjugate of their inputs. The holomorphic functions are necessarily not dependent on the conjugate of their inputs.

In general, given arbitrary functions $\boldsymbol{f}$ and $\boldsymbol{g}$, we compose their Jacobians in the following way,

$$\boldsymbol{J}_{\boldsymbol{f}\circ\boldsymbol{g}} = \boldsymbol{J}_{\boldsymbol{f}}\boldsymbol{J}_{\boldsymbol{g}} + \boldsymbol{J}_{\boldsymbol{f}}^{(c)} \overline{\left(\boldsymbol{J}_{\boldsymbol{g}}^{(c)}\right)} \quad \text{and} \tag{3.42}$$

$$\boldsymbol{J}_{\boldsymbol{f}\circ\boldsymbol{g}}^{(c)} = \boldsymbol{J}_{\boldsymbol{f}}\boldsymbol{J}_{\boldsymbol{g}}^{(c)} + \boldsymbol{J}_{\boldsymbol{f}}^{(c)} \overline{(\boldsymbol{J}_{\boldsymbol{g}})}, \tag{3.43}$$

as illustrated by Amin et al. (2011), H. Li and Adalı (2008).

Now suppose we have a composition of functions $(f \circ \boldsymbol{g} \circ \boldsymbol{h})(\boldsymbol{z}, \overline{\boldsymbol{z}})$, with f being a real valued (i.e. non-holomorphic) cost function, $\boldsymbol{g}$ being a non-holomorphic complex-valued function, and $\boldsymbol{h}$ being a holomorphic function. We would like to compute the gradient of f with respect to $\boldsymbol{z}$. Figure 3.4 shows the dependency graph for backpropagating the gradient.

Keeping in mind that f is a real-valued function of complex variables and that $\boldsymbol{h}$ is holomorphic such that $\frac{\partial \boldsymbol{h}}{\partial \overline{\boldsymbol{z}}} = 0$, we apply the chain rule to the gradient and Jacobian matrices,

$$\boldsymbol{J}_f = \frac{\partial f}{\partial \boldsymbol{g}} \tag{3.44}$$

$$\boldsymbol{J}_f^{(c)} = \frac{\partial f}{\partial \overline{\boldsymbol{g}}} \tag{3.45}$$

$$\boldsymbol{J}_{f\circ g} = \boldsymbol{J}_f \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{h}} + \boldsymbol{J}_f^{(c)} \overline{\left(\frac{\partial \boldsymbol{g}}{\partial \overline{\boldsymbol{h}}}\right)} \tag{3.46}$$

$$\boldsymbol{J}_{f\circ g\circ h} = \boldsymbol{J}_{f\circ g} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{z}} \tag{3.47}$$

$$\nabla_{\boldsymbol{z}} f = (\boldsymbol{J}_{f\circ g\circ h})^*. \tag{3.48}$$

As depicted in Fig. 3.4, our objective function is real, which lets us get away with tracking a single gradient for each node in the graph. We've also taken advantage of the fact that

$\boldsymbol{h}$ is holomorphic, so its output cannot depend on the conjugate of its input and $\boldsymbol{J}_{\boldsymbol{h}}^{(c)} = 0$, allowing the simplification in Eq. (3.47).

The Wirtinger Calculus makes it a little easier to build a computational graph for complex networks having mixed compositions of holomorphic and non-holomorphic operations. The computation of gradients with respect to complex-valued inputs prevents the tedious exercise of moving computation between complex domains.

3.4.2 Deep Real-Analytic Optimization

We'd like to easily define deep neural networks that are internally composed of holomorphic functions (such as matrix multiplication) and non-holomorphic functions (such as absolute value). The preceding section makes clear that it's possible to optimize such a network by tracking only one gradient. The question is whether any of the existing deep learning libraries support such optimization. At this writing, there are no existing deep learning frameworks that fully support deep learning in the complex field. TensorFlow[5], however, gets us much of the way there.

For every forward operation defined by TensorFlow, there's an accompanying gradient operation. At the time of optimization, the gradient of a real-valued objective function is backpropagated through the network, obeying the chain rule at each operation (or node) in the computational graph. Using a real-valued objective function enables us (and TensorFlow) to compose a relatively simple backpropagation path that satisfies many use cases. Assuming that the loss function is $\mathcal{L}$, its gradient with respect to an arbitrary function $f(z)$ is (without loss of generality for vector-valued functions of vectors $\boldsymbol{f}(\boldsymbol{z})$),

$$\boldsymbol{\nabla}_z \mathcal{L} = (\boldsymbol{\nabla}_f \mathcal{L}) \overline{\left(\frac{\partial f}{\partial z} \right)} + \overline{(\boldsymbol{\nabla}_f \mathcal{L})} \overline{\left(\frac{\partial \overline{f}}{\partial z} \right)} \tag{3.49}$$

$$= (\boldsymbol{\nabla}_f \mathcal{L}) \overline{\left(\frac{\partial f}{\partial z} \right)} + \overline{(\boldsymbol{\nabla}_f \mathcal{L})} \frac{\partial f}{\partial \overline{z}}. \tag{3.50}$$

We'd like the following scenarios to be covered:

- Gradient with respect to a holomorphic function;
- Gradient with respect to an antiholomorphic function; and
- Gradient with respect to a real-valued, non-holomorphic, but real-analytic function.

The first case includes essential building blocks of neural networks such as matrix multiplication, convolution, and sigmoid functions. A function of $z \in \mathbb{C}$ defined on an open set is antiholomorphic if its derivative with respect to $\overline{z}$ exists in the neighborhood of every point in the set. The conjugate operator is an antiholomorphic function. Finally, the third class of functions includes the real, imaginary, absolute and argument functions, all of which are useful in defining activation functions. The gradient of $\mathcal{L}$ with respect to each of these function types is considered below.

[5] TensorFlow is available online at https://www.tensorflow.org/.

Table 3.3: Gradient of a real-valued loss function $\mathcal{L}$ with respect to the argument of several non-holomorphic real-analytic functions.

$f(z)$	$\boldsymbol{\nabla}_z \mathcal{L}$
$\mathrm{Re}(z) = \frac{1}{2}(z + \overline{z})$	$\mathrm{Re}\{\boldsymbol{\nabla}_f \mathcal{L}\}$
$\mathrm{Im}(z) = \frac{-\mathrm{i}}{2}(z - \overline{z})$	$\mathrm{i}\,\mathrm{Re}\{\boldsymbol{\nabla}_f \mathcal{L}\}$
$\lvert z\rvert = \sqrt{z\overline{z}}$	$\frac{z}{\lvert z\rvert}\mathrm{Re}\{\boldsymbol{\nabla}_f \mathcal{L}\}$
$\arg z = \frac{-\mathrm{i}}{2}(\log z - \log \overline{z})$	$\frac{\mathrm{i}}{z}\mathrm{Re}\{\boldsymbol{\nabla}_f \mathcal{L}\}$

When f is holomorphic the second term of Eq. (3.50) vanishes and we're left with

$$\boldsymbol{\nabla}_{\boldsymbol{z}} \mathcal{L} = (\boldsymbol{\nabla}_{\boldsymbol{f}} \mathcal{L}) \overline{\left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}\right)}. \tag{3.51}$$

Similarly, when f is antiholomorphic the first term of Eq. (3.50) vanishes and we have

$$\boldsymbol{\nabla}_{\boldsymbol{z}} \mathcal{L} = \overline{(\boldsymbol{\nabla}_{\boldsymbol{f}} \mathcal{L})} \frac{\partial \boldsymbol{f}}{\partial \overline{\boldsymbol{z}}}. \tag{3.52}$$

Finally when f is a real-valued, real-analytic function, then $\overline{\left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}\right)} \equiv \frac{\partial \boldsymbol{f}}{\partial \overline{\boldsymbol{z}}}$ and Eq. (3.50) reduces to

$$\boldsymbol{\nabla}_{\boldsymbol{z}} \mathcal{L} = \left(\boldsymbol{\nabla}_{\boldsymbol{f}} \mathcal{L} + \overline{(\boldsymbol{\nabla}_{\boldsymbol{f}} \mathcal{L})}\right) \frac{\partial \boldsymbol{f}}{\partial \overline{\boldsymbol{z}}} \tag{3.53}$$

$$= 2\,\mathrm{Re}\{\boldsymbol{\nabla}_{\boldsymbol{f}} \mathcal{L}\} \frac{\partial \boldsymbol{f}}{\partial \overline{\boldsymbol{z}}}. \tag{3.54}$$

Table 3.3 lists the gradients for several of these types of functions.

Luckily, TensorFlow handles the gradients for these functions correctly. Defining a complex network is as simple as declaring the appropriate data types. Yet we don't get full support from the library. In particular, at the time of this writing, none of TensorFlow's optimization algorithms (such as stochastic gradient descent) permit complex optimization. In order to accomplish optimization, we inherit a built-in optimization algorithm and add complex data types to the list of allowable data types. This change is necessary for stochastic gradient descent, and if we'd like to use a more advanced algorithm that relies on second-order moments, such as Adam (Kingma & Ba, 2014), we need to do a little more work.

The Adam optimizer provides adaptive learning rates for every parameter in the model. The algorithm maintains a moving average of the first and second uncentered moments of each parameter's gradient, using these to normalize the gradient and provide an effective adaptive learning rate. We follow Kingma and Ba (2014) in the following description. Let v be the biased second raw moment estimate of the gradient for an arbitrary parameter. Define g as the gradient and β_2 as a hyperparameter controlling the exponential decay of the moving average. Then Kingma and Ba (2014)'s paper provides the update rule

$$v \leftarrow \beta_2 v + (1 - \beta_2) g^2. \tag{3.55}$$

The problem with using this algorithm directly is that the square of the complex gradient g^2 is the pseudo-variance, a complex value that describes whether a random variable is proper, i.e., whether it is uncorrelated with its complex conjugate. In the typical real-valued scenario, g^2 provides the estimated uncentered variance of the gradient and helps to bound the learning rate of its parameter by scaling the gradient inversely proportionally to its norm. Therefore, for a complex network, we apply the following modification to Eq. (3.55),

$$v \leftarrow \beta_2 v + (1 - \beta_2) g\overline{g}. \tag{3.56}$$

We call this algorithm Complex Adam and it is used as a drop-in replacement for a complex network anywhere the Adam algorithm would be used for a similar real network. We show in Chapter 5 that Complex Adam performs better on a complex dictionary learning task than the naive implementation of Adam. Note that a similar modification may be applied to any other optimization algorithm that keeps an estimate of a second moment, e.g. Adagrad (Duchi, Hazan, & Singer, 2011); Adadelta (Zeiler, 2012); and RMSProp (Tieleman & Hinton, 2012).

3.5 Input, Internal, and Output Representations

The inputs to a network may be naturally complex, in which case the domain of the data is the set of complex numbers. For example, Fourier-transformed audio is complex-valued. In other cases, inputs are complex by design because they have phase and magnitude components that are not statistically independent. Wind data, as in (S. L. Goh et al., 2006; S. L. Goh et al., 2004), or acoustic direction of arrival, which is explored in Chapter 4, may be complex by design.

We're also faced with deciding how to represent the training targets for a network. If we seek to infer a probability distribution over complex variables, outputs characterizing the probabilities will be real-valued. However regression tasks may best be handled using complex output units. The network weights may be complex or real, independent of the chosen input/output representations. The determining factors for which of these to use may have more than anything to do with how constrained we want our system to be.

Finally, there are different activations appropriate to complex and real hidden units. It's not clear in which cases we'd want to use holomorphic functions, such as those from the elementary transcendental family of functions, versus nonholomorphic functions that more closely represent the nonlinearities favored in today's deep real-valued architectures.

This section characterizes the tradeoffs of some of these choices by framing them in a toy inference task. We denote a function $\boldsymbol{f} : \mathcal{X}^M \rightarrow \mathcal{Y}^N$, where $\mathcal{X}, \mathcal{Y} \in \{\mathbb{R}, \mathbb{C}\}$. The variables $\boldsymbol{x}$, $\boldsymbol{y}$, and $\hat{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{x})$ denote inputs, targets, and inferred outputs. Potential architectures are described below, along with their impact on a toy model that learns a function that sums sinusoids.

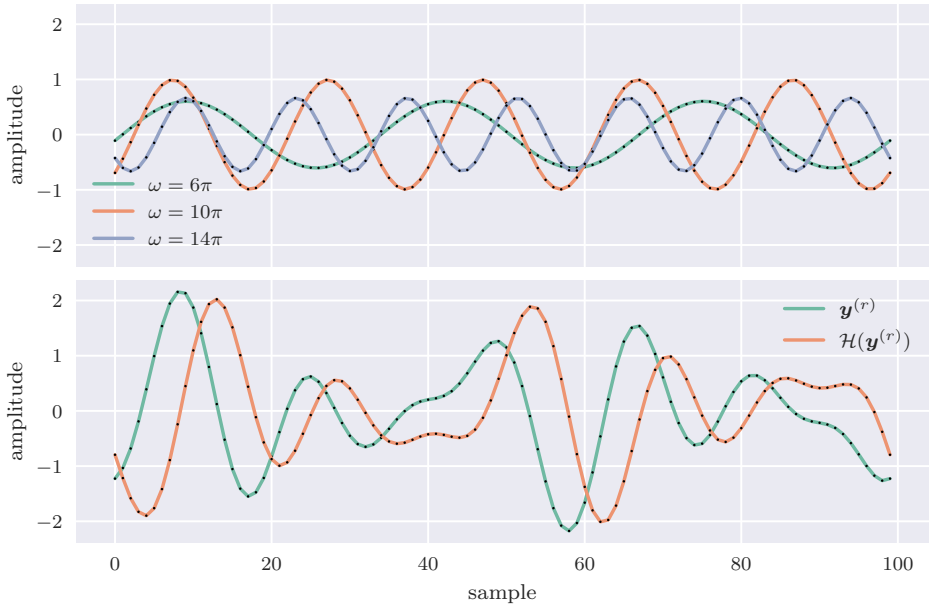

Figure 3.5: Top: three real sinusoids corresponding to arbitrary vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\varphi}$. Bottom: corresponding real-valued output $\boldsymbol{f}(\boldsymbol{\alpha}, \boldsymbol{\varphi})$ and the Hilbert transform of $\boldsymbol{f}$. Black markers indicate sample points corresponding to $\boldsymbol{t}$.

3.5.1 Sum of Sinusoids Task

For our toy example, we define the following function of the arguments $\boldsymbol{\alpha}, \boldsymbol{\varphi} \in \mathbb{R}^3$.

$$\boldsymbol{f}(\boldsymbol{\alpha}, \boldsymbol{\varphi}) = \sum_{i \in \{0,1,2\}} \alpha_i \sin(\omega_i \boldsymbol{t} + \varphi_i), \quad \text{given} \tag{3.57}$$

$$\boldsymbol{\omega} = \begin{bmatrix} 6\pi & 10\pi & 14\pi \end{bmatrix}^\mathsf{T} \quad \text{and} \tag{3.58}$$

$$\boldsymbol{t} = \begin{bmatrix} 0 & \frac{1}{100} & \cdots & \frac{99}{100} \end{bmatrix}, \tag{3.59}$$

where the sine function is applied element-wise. The top axis of Fig. 3.5 plots three sinusoids associated with arbitrary vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\varphi}$. The bottom axis plots their summation, $\boldsymbol{y}$, in green. The orange line plots the Hilbert transform of $\boldsymbol{f}$, which we discuss below.

We'd like to train a model that learns $\boldsymbol{f}$, and in the following sections we will explore various real and complex model configurations. The task is motivated by the fact that we often decompose audio into its sinusoidal components and that both the real and complex fields are equipped to handle periodicity (i.e., by trigonometric functions or complex argument). The following subsections describe design decisions for solving the task. Importantly, we show that particular combinations of input and output representations permit a closed-form solution. We argue that there are cases in which changing the input, internal, or output representation can make a task easier to solve, even when it doesn't have a closed-form solution.

3.5.2 Inputs

The function we'd like to learn takes input arguments $\boldsymbol{\alpha}$ and $\boldsymbol{\varphi}$, which are real-valued vectors. We can design the inputs of the model to be similar and real-valued, or we can make them complex-valued by design.

Amplitude-Phase. Arguably the simplest input representation is to simply concatenate the amplitude and phase offset parameters into a real-valued vector,

$$\boldsymbol{x}^{(ap)} := \begin{bmatrix} \cdots & \alpha_i & \varphi_i & \cdots \end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^{2K}. \tag{3.60}$$

Complex. Alternatively, the input parameters can be composed into complex numbers by letting the phase offset represent the phase of a complex phasor.

$$\boldsymbol{x}^{(c)} := \begin{bmatrix} \cdots & \alpha_i \mathrm{e}^{\mathrm{i}\varphi_i} & \cdots \end{bmatrix}^{\mathsf{T}} \in \mathbb{C}^{K}. \tag{3.61}$$

The vector $\boldsymbol{x}^{(c)}$ is complex by design, where we use the phase of a complex number to represent a parameter that represents rotation or angle.

Real-Imaginary. The complex vector in Eq. (3.61) could also be broken into its real and imaginary parts to form a real vector.

$$\boldsymbol{x}^{(ri)} := \begin{bmatrix} \mathrm{Re}\left\{\boldsymbol{x}^{(c)}\right\}^{\mathsf{T}} & \mathrm{Im}\left\{\boldsymbol{x}^{(c)}\right\}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} \tag{3.62}$$

$$= \begin{bmatrix} \cdots & \alpha_i \cos\varphi_i & \cdots & a_0 \sin\varphi_0 & \cdots \end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^{2K}. \tag{3.63}$$

Augmented Complex. Finally, we may use the augmented complex vector, which is the complex vector concatenated with its conjugate.

$$\boldsymbol{x}^{(ac)} := \begin{bmatrix} \left(\boldsymbol{x}^{(c)}\right)^{\mathsf{T}} & \left(\boldsymbol{x}^{(c)}\right)^{*} \end{bmatrix}^{\mathsf{T}} \in \mathbb{C}^{2K}, \tag{3.64}$$

$$= \begin{bmatrix} \cdots & \alpha_i \mathrm{e}^{\mathrm{i}\varphi_i} & \cdots & \alpha_i \mathrm{e}^{-\mathrm{i}\varphi_i} & \cdots \end{bmatrix}^{\mathsf{T}} \tag{3.65}$$

The mapping defined by complex conjugation is antilinear and, as such, it cannot be computed by a complex matrix multiplication. Providing the input along with its complex conjugate may provide a model additional necessary information to learn the task. Indeed, using the augmented complex vector provides an easier path to modeling the complete second-order statistical distribution of the data. Augmented inputs are used in widely linear estimation, e.g. (see Mandic & Goh, 2009, chaps. 12, 13, and 15).

3.5.3 Targets

We examine real- and complex-valued versions of two closely related targets. The first is a straightforward sum of sinusoids and the second uses the Hilbert transform to create an analytic signal.

Straightforward Target. We designate regular real and complex targets $\boldsymbol{y}^{(r)}$ and $\boldsymbol{y}^{(c)}$ as

$$y_t^{(r)} := \sum_i \alpha_i \sin(\omega_i t + \varphi_i) \tag{3.66}$$

$$= \sum_i \alpha_i (\sin \omega_i t \cos \varphi_i + \cos \omega_i t \sin \varphi_i) \quad \text{and} \tag{3.67}$$

$$\boldsymbol{y}^{(c)} := \boldsymbol{y}^{(r)} + \mathrm{i}\mathbf{0}, \tag{3.68}$$

where $\mathbf{0}$ indicates a vector of zeros.

Analytic Target. The analytic real and complex targets $\boldsymbol{y}^{(rh)}$ and $\boldsymbol{y}^{(ch)}$ are composed as follows.

$$y^{(rh)} := \left[\left(y^{(r)}\right)^\mathsf{T} \quad \mathcal{H}\left(y^{(r)}\right)^\mathsf{T} \right]^\mathsf{T} \quad \text{and} \tag{3.69}$$

$$y^{(ch)} := y^{(r)} + \mathrm{i}\mathcal{H}\left(y^{(r)}\right), \tag{3.70}$$

where $\mathcal{H}$ is the Hilbert transform (see Section 2.3). The green and orange lines in the bottom axis of Fig. 3.5 show an arbitrary output of $\boldsymbol{f}$ and its Hilbert transform.

3.5.4 Architectures

We examine feed-forward models having one hidden layer, defined as

$$\boldsymbol{h} = \boldsymbol{f}\left(\boldsymbol{W}^{(h)}\boldsymbol{x} + \boldsymbol{b}^{(h)}\right) \quad \text{and} \tag{3.71}$$

$$\hat{\boldsymbol{y}} = \boldsymbol{W}^{(o)}\boldsymbol{h} + \boldsymbol{b}^{(o)}, \tag{3.72}$$

where $\theta = \{\boldsymbol{W}^{(h)}, \boldsymbol{W}^{(o)}, \boldsymbol{b}^{(h)}, \boldsymbol{b}^{(o)}\}$ are the model parameters. Designating M as the number of input nodes (either $M = K$ or $M = 2K$ depending on the input type), and $N = 100$ as the number of output nodes,

$$\boldsymbol{W}^{(h)} \in \mathbb{R}^{M\times M} \quad \text{or} \quad \boldsymbol{W}^{(h)} \in \mathbb{C}^{M\times M} \tag{3.73}$$

$$\boldsymbol{b}^{(h)} \in \mathbb{R}^{M} \quad \text{or} \quad \boldsymbol{b}^{(h)} \in \mathbb{C}^{M} \tag{3.74}$$

$$\boldsymbol{W}^{(o)} \in \mathbb{R}^{N\times M} \quad \text{or} \quad \boldsymbol{W}^{(o)} \in \mathbb{C}^{N\times M} \tag{3.75}$$

$$\boldsymbol{b}^{(o)} \in \mathbb{R}^{N} \quad \text{or} \quad \boldsymbol{b}^{(o)} \in \mathbb{C}^{N}. \tag{3.76}$$

Figure 3.6 gives a diagrammatic overview of this simple model architecture. In all cases the inputs, weights, and outputs are in the same numerical domain. I.e., using the input $\boldsymbol{x}^{(r)}$ implies that the weights are real and that we are limited to the targets $\boldsymbol{y}^{(r)}$ and $\boldsymbol{y}^{(rh)}$.

3.5.5 Activation Functions

Identity. The identity function provides an entirely linear model.

$$f^{(-)}(x) := x. \tag{3.77}$$

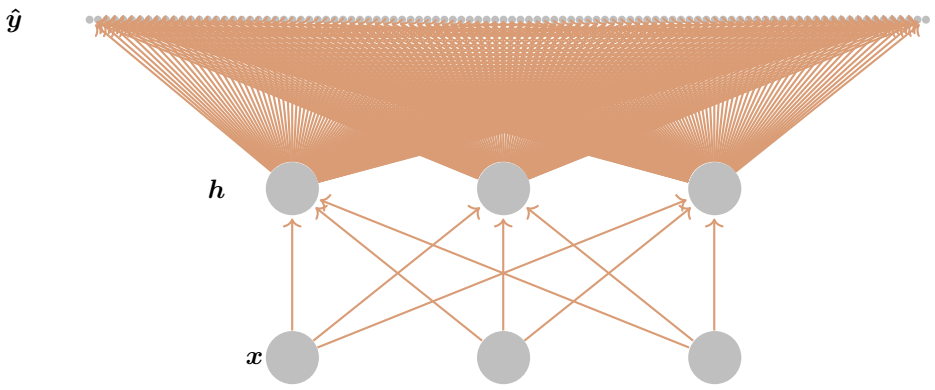

Figure 3.6: Basic model architecture showing $D = 3$ input and hidden nodes, with $N = 100$ output nodes.

Hyperbolic tangent. The hyperbolic tangent function is a sigmoid function that is real- and complex-differentiable. It is a widely used nonlinearity for machine learning.

$$f^{(\sigma)}(x) := \tanh x. \tag{3.78}$$

Split real-imaginary. We also test split real-imaginary activations on networks with complex weights,

$$f^{(ri)}(x) := \tanh(\operatorname{Re} x) + \mathrm{i} \tanh(\operatorname{Im} x), \tag{3.79}$$

where the nonlinearity is applied independently to the real and imaginary components. The split real-imaginary activation is not complex-differentiable, but it is real-differentiable with respect to its real and imaginary components.

Split amplitude-phase. The split amplitude-phase activation applies the nonlinearity to the modulus of a complex-valued number and doesn't modify the phase,

$$f^{(ap)}(x) := \tanh(|x|)\mathrm{e}^{\mathrm{i} \arg x}. \tag{3.80}$$

This function is also not complex-differentiable.

3.5.6 Results

We evaluated several model configurations using disjoint training and testing data, denoted $\mathcal{X}^{(tr)}$ and $\mathcal{X}^{(te)}$. That is, the $\boldsymbol{\alpha}$ and $\boldsymbol{\varphi}$ parameters did not overlap in the two datasets, as depicted in Fig. 3.7. Note that not only are the training and testing data disjoint, but the numerical ranges that they cover are disjoint. That is, the training data consists of a range of amplitudes that are non-overlapping in the test data. Similarly, the training data consists of a range of phase offsets that are non-overlapping with test data.

We judge the success of a trained model by how well it makes accurate inferences from the domain $\mathcal{X}^{(te)}$ when trained solely using the disjoint domain $\mathcal{X}^{(tr)}$. If a model can faithfully

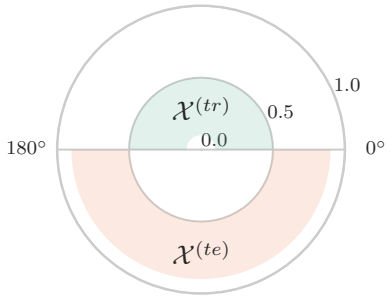

Figure 3.7: Non-overlapping training and testing data, shown in polar coordinates.

Table 3.4: closed-form linear solutions exist for some combinations of input and output representations.

	real	
	$\boldsymbol{y}^{(r)}$	$\boldsymbol{y}^{(rh)}$
$\boldsymbol{x}^{(ap)}$	no	no
$\boldsymbol{x}^{(ri)}$	yes	yes

	complex	
	$\boldsymbol{y}^{(c)}$	$\boldsymbol{y}^{(ch)}$
$\boldsymbol{x}^{(c)}$	no	yes
$\boldsymbol{x}^{(ac)}$	yes	yes

produce the correct outputs given the non-overlapping and unseen testing data, then it has successfully learned the underlying function.

All models were trained with data having angular frequencies $\boldsymbol{\omega} = \begin{bmatrix} 6\pi & 10\pi & 14\pi \end{bmatrix}$, which are not harmonically related. Training was performed for 100 epochs using mini-batch stochastic gradient descent with batch size $B = 10$ and learning rate <0.001. Denoting b as the index of a training batch, we used a mean squared error loss function for real and complex nets:

$$\boldsymbol{e}^{(b)} = \boldsymbol{y}^{(b)} - \hat{\boldsymbol{y}}^{(b)} \tag{3.81}$$

$$\mathcal{L} = \frac{1}{B} \sum_{b=0}^{B-1} \left(\boldsymbol{e}^{(b)} \right)^* \boldsymbol{e}^{(b)} \tag{3.82}$$

Weights were initialized as normally random with a standard deviation of <0.001.

For a simple task such as the sum of sinusoids, the input, inner, and output representations may matter quite a bit. First we note that there are five combinations of input and output representation for which a closed-form linear solution exists, as summarized in Table 3.4. We don't present numerical results for these combinations, as the models are trivially able to learn these configurations using SGD when the hidden layer is linear, i.e., the entire model is linear. However some insight is provided in the following paragraphs.

When either of the real-valued output targets is used, $\boldsymbol{y}^{(r)}$ or $\boldsymbol{y}^{(rh)}$, the real-imaginary input representation has a closed-form solution, while the amplitude-phase representation does not. The former expresses the input in terms of Cartesian coordinates and the latter expresses the input in terms polar coordinates. In the latter, phase is represented using a circular real

Table 3.5: Training error and testing error for combinations of input representation, output representation, and activation function for where a closed-form solution does not exist. All networks have six units in their hidden layer. The best error, across 100 epochs is indicated, along with which epoch the error was recorded. Results are sorted by ascending test error.

input	target	activation	train error	train epoch	test error	test epoch
$\boldsymbol{x}^{(c)}$	$f^{(c)}$	$f^{(ri)}$	<0.001	100	4.6	100
$\boldsymbol{x}^{(c)}$	$f^{(c)}$	$f^{(ap)}$	0.1	100	8.3	8
$\boldsymbol{x}^{(c)}$	$f^{(c)}$	$f^{(-)}$	0.3	72	8.3	7
$\boldsymbol{x}^{(c)}$	$f^{(c)}$	$f^{(\sigma)}$	0.3	99	8.5	6
$\boldsymbol{x}^{(ap)}$	$f^{(r)}$	$f^{(\sigma)}$	0.2	100	9.6	2
$\boldsymbol{x}^{(ap)}$	$f^{(rh)}$	$f^{(\sigma)}$	0.2	97	9.7	2
$\boldsymbol{x}^{(ap)}$	$f^{(r)}$	$f^{(-)}$	0.2	70	10.0	1
$\boldsymbol{x}^{(ap)}$	$f^{(rh)}$	$f^{(-)}$	0.2	58	10.1	1

value, wrapping around from $-\pi$, to π, and cannot be expressed through a linear operation. As Eqs. (3.67) and (3.69) show, the Cartesian representation closely corresponds to the target. Considering that, by Fourier theory, any signal is decomposable into a set of sinusoidal functions, we suggest that in cases where we have a choice between the two representations, one may consider the Cartesian coordinate representation first.

When the target is made complex by simply adding zero-valued imaginary components, as we do for the target $\boldsymbol{y}^{(c)}$, the model must essentially learn the real operator. As shown in the previous section, the real operator is a nonholomorphic function that relies on the conjugate of its input.

There's a closed-form solution to the task when the augmented complex input is used, and no closed-form solution for the regular complex input. That is, $\left(\boldsymbol{x}^{(ac)}\right)^{\mathsf{T}}\theta$ exactly equals $\boldsymbol{y}^{(c)}$ when we let $\theta = \begin{bmatrix}\cdots & \frac{-\mathrm{i}}{2}\mathrm{e}^{\mathrm{i}\omega_i} & \cdots & \frac{\mathrm{i}}{2}\mathrm{e}^{-\mathrm{i}\omega_i}\end{bmatrix}^{\mathsf{T}}$. We suggest that the augmented input be used any time the task may require a mapping from a complex representation through a nonholomorphic function.

We turn our attention to the networks' behavior when a closed-form solution does not exist with respect to the given input and output representations. Table 3.5 provides the best train and test errors across a combination of input types, output types, and activation functions. All models have six units in their hidden layer. The results are sorted by ascending test error. (Note that the absolute test error cannot be compared easily with the absolute train error because the average amplitude of the test targets is approximately twice as large as the training targets.) The table shows that, in all cases, the complex input representation does a better job at learning the sum-of-sinusoids function than the representations using real inputs. The best performing model has a split real-imaginary activation function. Notably the holomorphic hyperbolic tangent function gives the worst performance for the task. This might be due to the singularities along the imaginary axis of its output. By examining the values of the best epochs, it appears that only the first model has not suffered from overfitting. For all other models the epoch yielding the best training error is much greater than the epoch yielding the best test error.

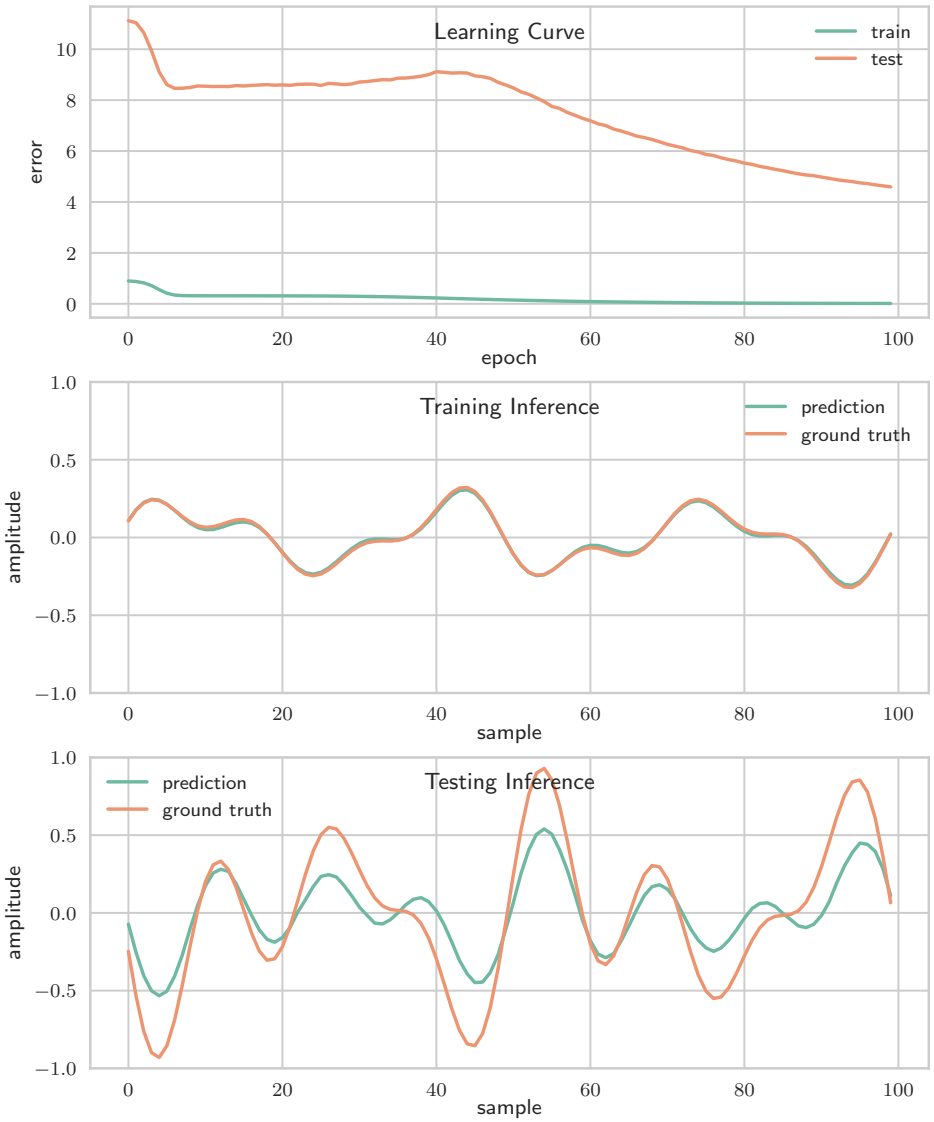

Figure 3.8: Model using regular complex inputs, regular complex outputs, and sigmoidal activation on the real and imaginary components. Top: learning curves on the training and testing sets. Middle: inference on a random example from the training set. Bottom: inference on a random example from the testing set.

The best model, which employs a sigmoidal activation function on the real and imaginary components, is depicted in Fig. 3.8. The top panel of the figure shows the learning curves for 100 epochs of training. It appears that the model has not converged and that, if trained longer, it would continue to improve. The middle panel shows the model's inference when given a random draw from the training set. The output of the model is quite similar to the ground truth. The bottom panel shows inference when given a random draw from the test set. The inferred waveform follows the general shape of the ground truth but there are major differences between the two signals.

Figure 3.9 shows a similar view of the best-performing model trained on real inputs. It's clear from the learning curve that the model is not able to learn the underlying sum-of-sinusoids function and that learning has failed completely. As the training error decreases across epochs, the testing error increases. Inference, when given a sample from the training set, produces a waveform that has the approximate shape of the ground truth target. Interestingly, the inferences from the training and test sets are very jagged compared to the best performing complex model. We hypothesize that the constraints imposed by complex weights provide regularization that helps to produce smoother inferences and to avoid overfitting to the individual coordinates of the targets.

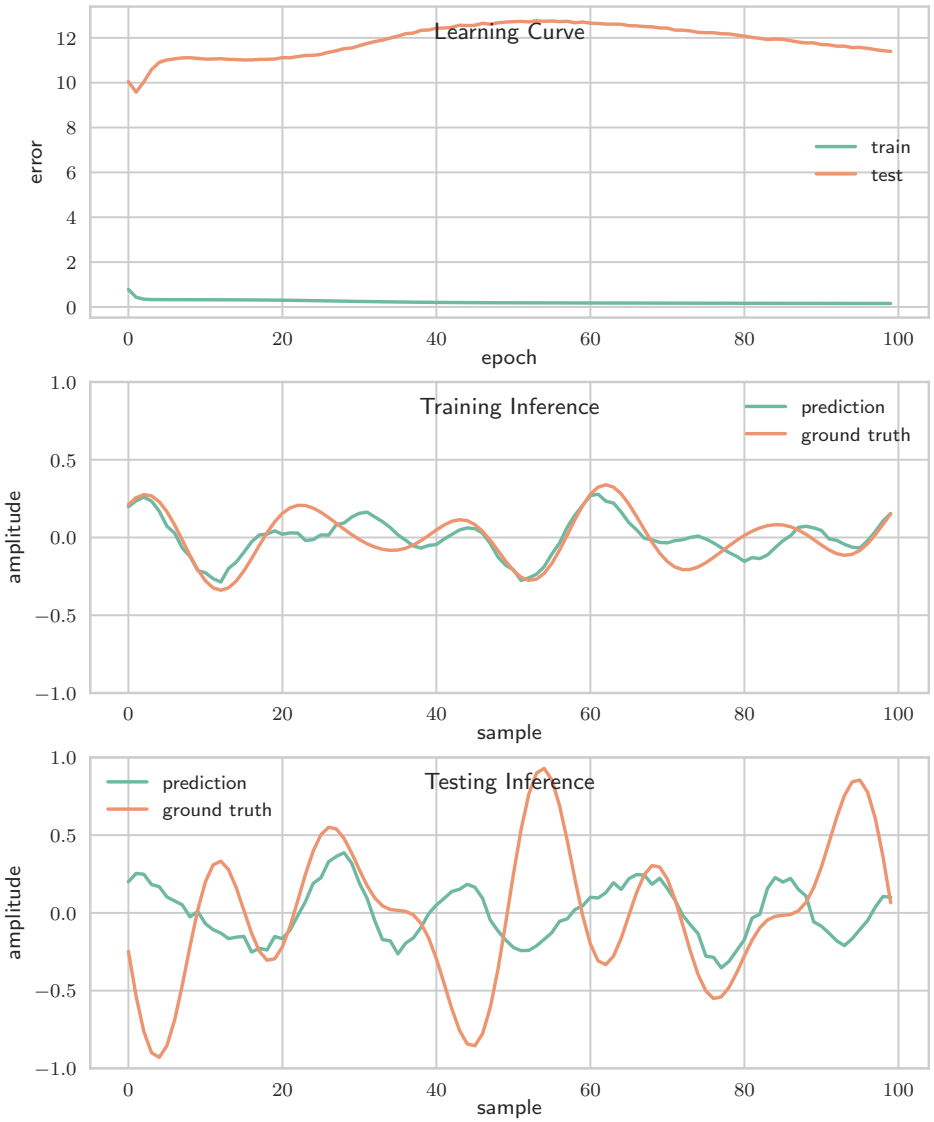

Figure 3.9: Model using real amplitude-phase inputs, regular real outputs, and sigmoidal activations. Top: learning curves on the training and testing sets. Middle: inference on a random example from the training set. Bottom: inference on a random example from the testing set.

CHAPTER 4

Binaural Source Localization

In this chapter we propose complex neural networks for binaural source localization of musical instruments, a task that has relevance to the fields of machine listening and music information retrieval (MIR). Machine listening aims to endow computers with the ability to make human-like decisions about sound. From the perspective of machine listening researchers, it's imperative for computers to succeed in listening skills that are innate for humans, like source localization.

Human azimuthal localization is dependent on binaural spatial cues related to fine-scale temporal differences of sounds arriving at the ears. These cues are encoded in audio, but many methods for source localization rely on hand-engineered techniques to extract cues and make inferences about source location. We suggest complex neural networks for this task, since the complex-valued frequency-domain binaural audio preserves inter- and intra-channel temporal differences. We show that feedforward networks with complex-valued weights and activations are more adept at localization than other real-valued models. We also show that the complex networks generalize better than real models across sounds filtered through unseen models of human heads. This is notable accomplishment for improved machine listening, as it's been shown that humans can adapt in a similar way (Wenzel et al., 1993). Other models in the literature do not show the same ability to generalize across varying anatomical contexts.

Main contributions: We propose a feedforward network with complex-valued weights and activations that takes as input the Discrete Fourier Transform (DFT) of a binaural audio frame. We provide experimental results showing that this network outperforms its real-valued counterpart. It also outperforms several baselines designed to extract binaural cues directly from the audio. We provide statistical evidence that the complex neural network generalizes

across unseen binaural filters, and that it does this better than the other models that we tested.

We take a data-driven approach in this work, where a multi-layered neural network with complex-valued activations is trained on a set of source signals convolved with a training set of binaural filters. The use of deep complex neural networks in this application is novel. Inference is fast since the model does not rely on temporal dependencies or learning at inference time. We show that it is unnecessary to apply specialized feature-engineering to extract spatial cues from the input signal. We let the model learn complex representations directly from complex inputs. We show that our neural network performs better than baseline neural models that explicitly extract complex cues (Tsuzuki et al., 2013) and that perform frontal-plane localization directly from binaural cues (Dietz, Ewert, & Hohmann, 2011).

Relationship to the literature: A complex network for stereo (non-binaural) microphone localization has been proposed by (Tsuzuki et al., 2013). The authors manufacture an input feature vector that summarizes inter-channel phase differences over a range of rotated signals, similar to the MUSIC (Schmidt, 1986) algorithm. The search space is determined as a function of distance between microphones, which should be known *a priori*. Their network is shallow (one hidden layer) while, in contrast, we use a deeper network, a different loss function, different nonlinearities, and train on a large set of binaural filters. We show that their features perform poorly in a complex network for binaural localization.

Direction-of-arrival inference from multichannel microphone arrays, in which the real-valued phases of an STFT are given as the input to a convolutional neural network, is presented in (Chakrabarty & Habets, 2017). The model is trained by exciting a microphone array with broad-spectrum noise aimed from various angles. Thus the model is not dependent on source material and is shown to be robust to different room conditions, noise conditions, and small perturbations of microphone placement. Their paper is not concerned with the specific challenge of generalizing across multiple heads or with microphone arrays of size less than four.

In contrast to the existing literature, we do not attempt to prove that the network generalizes to diverse source signals. Rather, we show that it generalizes to diverse and unseen binaural filters, or human heads. To this end, we do not examine the robustness of the network to corrupted input signals and reverberant environments. This thesis focuses on localization of a musical source in the full lateral plane and does not examine whether such models would also be suitable for inference of elevation.

In the following sections, we describe the binaural localization task, followed by a description of the complex model. Experimental results are provided in Section 4.3. We provide concluding remarks and discussion in Section 4.4.

4.1 Binaural Music Source Localization Task

Binaural source localization is a computational auditory scene analysis task concerned with determining the relative location(s) of sources by analyzing an audio signal that has been filtered by a human head. Musical source localization may provide useful information about

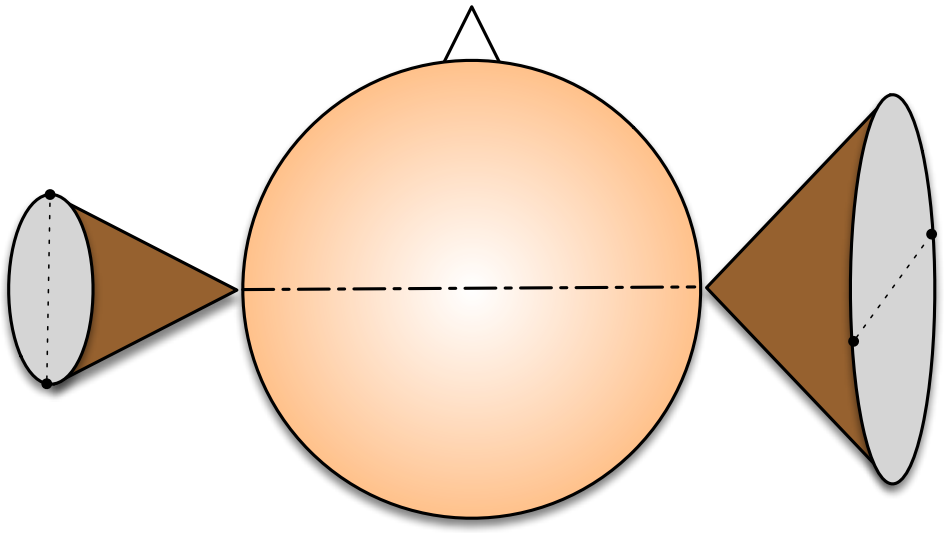

Figure 4.1: Cones of confusion. sounds arriving along the cone have similar ITD and ILD, making them difficult to distinguish.

the context of a live performance to a remote audience, or to performers who are not co-located. Concert hall design may benefit from analyses that take into account the apparent locations of musical sources relative to human listeners. Localization of musical objects in a multichannel recording can be precursor to multichannel source separation algorithms that rely on spatialization, as in (FitzGerald, Liutkus, & Badeau, 2016).

More generally, binaural localization finds applications in hearing aid design, humanoid robot perception, head tracking, and augmented reality. Computer-based localization may assist hearing-damaged individuals, provide augmented hearing acuity when ears are obstructed by devices such as headsets, or enhance ordinary hearing capabilities. Generalizable models are desirable so that a system need not be fine-tuned to the characteristics of each individual's head. We'd like source localization to work out-of-the-box, without taking measurements of each user's head or storing a lookup table of many head filters.

According to duplex theory (Middlebrooks & Green, 1991) human localization employs monaural and binaural psychoacoustic cues. The binaural cues combine frequency-dependent inter-aural time or phase differences (ITD; IPD) and inter-aural level differences (ILD). It's believed that we use IPD cues to locate frequencies approximately below 1.5 kHz, and ILD otherwise. Despite human adeptness, localization is susceptible to errors along "cones of confusion," depicted in Fig. 4.1, where it's common to mistake the locations of sources at symmetric angles about the ear. The irregular shapes of the outer ears provide an additional monaural filtering mechanism, helping to resolve such confusions. Binaural source localization systems typically model the way humans extract spatial cues and suffer from similar confusions.

Virtually all source localization systems include a feature-engineering stage at the front end that manipulates the time domain or frequency domain signals to extract ITD/IPD- and

ILD-like spatial cues. Many systems are non-parametric and require storage of a dataset of cues and associated locations, which are later matched during inference (D. Li & Levinson, 2003; Liu, Pang, & Zhang, 2015; Liu & Zhang, 2014; Liu, Zhang, & Fu, 2014; Nix & Hohmann, 2006; Raspaud, Viste, & Evangelista, 2010; Talagala, Wu, Zhang, & Abhayapala, 2014; Wu, Talagala, Zhang, & Abhayapala, 2015, 2016).

For supervised systems that do not store a table of spatial cues, it is generally assumed that the set of binaural filters will not differ between training and inference time. For example, (Deleforge, Forbes, & Horaud, 2015; Woodruff & Wang, 2012) show the efficacies of their systems by training and testing using the same sets of binaural filters. These systems cannot claim that their methods generalize to unseen heads. In contrast, a few unsupervised methods have been proposed, e.g. (Mandel, 2010; Talagala, Zhang, Abhayapala, & Kamineni, 2014). However they require sufficiently large data to accumulate statistics for decision making and therefore may not be well-suited to real-time inference.

4.2 Complex Feedforward Binaural Localization

4.2.1 Training signal

Given a frame of binaural audio having N samples, let l_n and r_n, with $n = 0, 1, \ldots, N-1$, denote the left and right channels. The channels are transformed to the frequency domain, truncated, and concatenated,

$$\hat{l}_k = \sum_{n=0}^{N-1} l_n \mathrm{e}^{-\mathrm{i}2\pi kn/N} \tag{4.1}$$

$$\hat{r}_k = \sum_{n=0}^{N-1} r_n \mathrm{e}^{-\mathrm{i}2\pi kn/N} \tag{4.2}$$

$$x_k = \begin{bmatrix} \hat{l}_k & \hat{r}_k \end{bmatrix}^{\mathsf{T}}, \tag{4.3}$$

with the index $k \in \{0, 1, \ldots, N/2\}$ representing a non-conjugate frequency bin.

Azimuth locations are discrete; each ground truth location in the dataset is represented as a binary-valued one-hot vector with D dimensions, as

$$\boldsymbol{y} \in \mathbb{Z}_2^D. \tag{4.4}$$

For example, if y_i has the value 1, then all other elements of $\boldsymbol{y}$ must have the value 0 and the target is a member of the i-th class. Thus the input to the model is the pairwise complex-valued DFTs of binaural audio, $\boldsymbol{x}$. The output of the model is the inferred azimuthal angle of the sound source represented as a one-hot vector. The model learns a mapping

$$\boldsymbol{f} : \boldsymbol{x} \in \mathbb{C}^{N+2} \mapsto \hat{\boldsymbol{y}} \in \mathbb{R}^D, \tag{4.5}$$

from complex spectra to a probability distribution over azimuth classes. We minimize the cross entropy between the training target $\boldsymbol{y}$ and the inference $\hat{\boldsymbol{y}}$. Aside from the initial

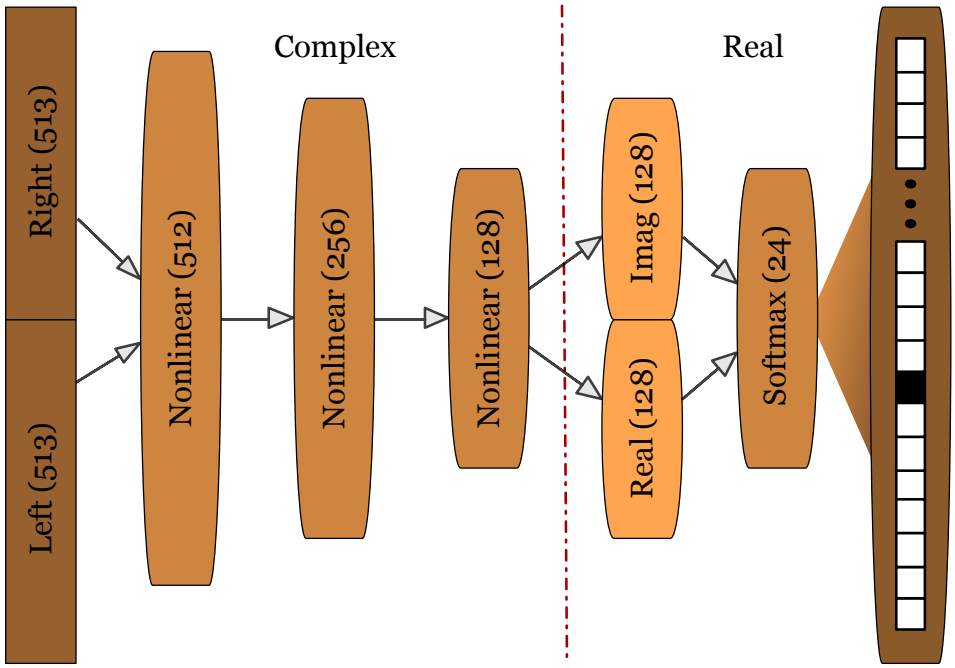

Figure 4.2: Architecture of the binaural source localization model. Numbers in parentheses indicate number of units. The left and right frequency-domain inputs are fed into three complex nonlinear layers. The real and imaginary parts of the third layer are concatenated into a real vector, which is followed by a final nonlinear layer with a softmax activation.

transformation to the Fourier domain, i.e., Eqs. (4.1) to (4.3), no hand-engineering is applied to produce the input features.

4.2.2 Model

We use a dense multilayer perceptron, depicted in Fig. 4.2, with activations computed on the complex field. The architecture of the neural network with L layers and indices $l = 0, \ldots, L$ is as follows. We designate parameters of the network as $\theta = (\{\boldsymbol{W}^{(l)}, \boldsymbol{b}^{(l)}\})_{l \neq 0}$, and let

$$\boldsymbol{z}^{(l)} = a\boldsymbol{W}^{(l)}\boldsymbol{h}^{(l-1)} + \boldsymbol{b}^{(l)} \tag{4.6}$$

$$\boldsymbol{h}^{(l)} = \sigma^{(l)}\left(\boldsymbol{z}^{(l)}\right), \tag{4.7}$$

where $\sigma^{(l)}$ is an element-wise activation function. We designate the cardinality of each layer as $M^{(l)}$, with $M^{(0)} = N + 2$. Each hidden layer has complex-valued weights and biases,

$$\boldsymbol{W}^{(l<L)} \in \mathbb{C}^{M \times M^{(l-1)}} \quad \text{and} \tag{4.8}$$

$$\boldsymbol{b}^{(l<L)} \in \mathbb{C}^{M^{(l)}}, \tag{4.9}$$

Table 4.1: Nonlinearities for complex neural networks.

Nonlinearity	$\sigma(z) =$
tanh	$\tanh z$
cartTanh	$\tanh(\operatorname{Re} z) + \mathrm{i}\tanh(\operatorname{Im} z)$
cartReLU	$\max(\operatorname{Re} z, 0) + \mathrm{i}\max(\operatorname{Im} z, 0)$
modTanh	$\tanh\lvert z\rvert \cdot \mathrm{e}^{\mathrm{i}\arg z}$
modReLU (Arjovsky et al., 2016)	$\begin{cases} (\lvert z\rvert + \beta)\cdot \mathrm{e}^{\mathrm{i}\arg z} & \text{if} \quad \lvert z\rvert > 0 \\ 0 & \text{otherwise} \end{cases}$

and the final output layer utilizes real weights and biases,

$$\boldsymbol{W}^{(L)} \in \mathbb{R}^{D\times 2M^{(L-1)}} \quad \text{and} \quad \boldsymbol{b}^{(L)} \in \mathbb{R}^{D}. \tag{4.10}$$

The complex weights are initialized with orthogonal vectors, as suggested in (Trabelsi et al., 2017). The model's inputs and outputs are denoted

$$\boldsymbol{x} \triangleq \boldsymbol{h}^{(0)} \quad \text{and} \tag{4.11}$$

$$\hat{\boldsymbol{y}} \triangleq \boldsymbol{h}^{(L)}. \tag{4.12}$$

Table 4.1 lists the activation functions examined in this work. The modReLU activation has been previously proposed in (Arjovsky et al., 2016) and includes an additional parameter β. There is no established consensus in the literature on the appropriate choice of activation function for complex networks. As a consequence of Liouville's theorem, which states that any holomorphic function is unbounded unless it is a constant, choosing a complex-analytic function implies choosing an unbounded function. Yet boundedness has been considered a desirable property for optimization in real networks. E.g., logistic sigmoids are bounded in the range $[0, 1]$ and rectified linear units (ReLUs) are bounded in the range $[0, \infty)$. This fact may make complex-analytic activations such as the hyperbolic tangent function, which has regularly spaced singularities along the imaginary axis, undesirable. We investigate both complex-analytic and real-analytic activations to try to gain some insight about whether one type is more desirable than the other.

4.2.3 Inference and loss

The final hidden layer is transformed to real by splitting the real and imaginary parts,

$$\hat{\boldsymbol{h}}^{(L-1)} = \left[\operatorname{Re}(\boldsymbol{h}^{(L-1)}) \quad \operatorname{Im}(\boldsymbol{h}^{(L-1)})\right]^{\mathsf{T}} \tag{4.13}$$

$$\boldsymbol{z}^{(L)} = \boldsymbol{W}^{(L)}\hat{\boldsymbol{h}}^{(L-1)} + \boldsymbol{b}^{(L)} \tag{4.14}$$

Finally we compute the softmax function to produce the output layer,

$$\sigma^{(L)}(\boldsymbol{z}) = \frac{\mathrm{e}^{z_d}}{\sum_{d=0}^{D-1} \mathrm{e}^{z_d}} \tag{4.15}$$

$$\hat{\boldsymbol{y}} = \sigma^{(L)}(\boldsymbol{z}^{(L)}). \tag{4.16}$$

The output layer is interpreted as a probability distribution and we compute the loss using cross-entropy,

$$\mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\sum_{d} y_d \log \hat{y}_d. \tag{4.17}$$

It can be shown using the Wirtinger Calculus (Wirtinger, 1927) that a scalar real-valued function of arbitrary complex variables $\boldsymbol{z} \in \mathbb{C}^N$ may be minimized by defining the cogradient and gradient operators respectively,

$$\frac{\partial}{\partial \boldsymbol{z}} := \begin{bmatrix} \frac{\partial}{\partial z_0} & \frac{\partial}{\partial z_1} & \cdots & \frac{\partial}{\partial z_{N-1}} \end{bmatrix} \quad \text{and} \tag{4.18}$$

$$\boldsymbol{\nabla}_{\boldsymbol{z}}(\cdot) = \left(\frac{\partial(\cdot)}{\partial \boldsymbol{z}} \right)^{*}. \tag{4.19}$$

The network is optimized using Eq. (4.19) and the Adam algorithm (Kingma & Ba, 2014).

4.3 Experiments

Two sets of source audio were used. The first set, "Music," was selected from a studio recording of Vivaldi's *Concerto No. 1 in E major, Op. 8, RV 269* found in the MedleyDB dataset (Bittner et al., 2014). It is comprised of five clips of raw audio recordings, 21 seconds in duration each, corresponding to the double bass, cello, viola, violin 1, and violin 2. The second set, "Noise" consists of white uniform noise. The "Noise" dataset was chosen as a control group to compare against the "Music" sources. Since it contains equal energy across the spectrum, Noise excites the full frequency range of the binaural filters. We expect that the relatively sparse Music data should be more difficult to learn because Music presents the model with material that excites fewer components of the binaural filter.

Binaural recordings were synthesized by convolving source audio with head-related impulse responses (HRIRs), which are recordings of wide-band sounds that have been directed at a human head and measured inside the ears using in-ear microphones. The sounds are propagated at regularly spaced angles with respect to the human head. The impulse responses therefore encode the sound filter created by the anatomical properties of the head. We used the binaural filters from IRCAM's Listen HRTF Database[1], which contains HRIRs for 51 human subjects recorded using in-ear microphones in an anechoic chamber. For each subject, 187 measurements have been recorded across a range of azimuths and elevations.

[1] Available online at http://recherche.ircam.fr/equipes/salles/listen/download.html

Table 4.2: Dataset partition.

Dataset	# Subjects	Hours	# Obs.
Train	25	18.6	2,823,000
Validation	13	9.3	1,467,960
Test	13	9.3	1,467,960

We convolved each of the five music audio recordings and an equal number of samples of noise with $D = 24$ compensated[2] azimuthal HRIRs at 0° elevation for each human subject in the IRCAM dataset. (HRIRs corresponding to other elevations were not used.) The azimuthal HRIRs were equally spaced along the lateral plane, with 15° spatial resolution. The sampling rates of the HRIRs and the source audio was 44,100 Hz. The synthesized binaural recordings were normalized such that the maximum amplitude in at least one channel was 1.0.

The binaural recordings were broken into rectangular non-overlapping frames of $N = 1024$ samples, and each frame was treated as an independent observation. In some experiments (see below), we applied a high-pass or low-pass filter to all of the observations in the dataset. The dataset was subsequently partitioned randomly by HRIR subject into disjoint training, validation, and test sets providing subsets of approximately 2.8 million training observations (25 subjects) and 1.5 million validation and testing observations (13 subjects each) for "Music" and "Noise" each. The partitioning is summarized in Table 4.2. Frames with low energy relative to the dataset are likely to lack meaningful content. E.g., they may correspond to silences near the endpoints of the source files or room noise between instrument onsets. To avoid training and evaluating on such data, we sorted the frames by increasing RMS energy and discarded all observations within the first 10th percentile.

We describe several experiments in the following subsections. In all cases we use accuracy as the evaluation metric. Denote $\boldsymbol{y}^{(i)}$ and $\hat{\boldsymbol{y}}^{(i)}$ the target and inference associated with the i-th test example in dataset $\mathcal{X}$. Accuracy is defined as follows.

$$a = \frac{100}{|\mathcal{X}|} \sum_{i=0}^{|\mathcal{X}|-1} \begin{cases} 1 & \text{if} \quad \boldsymbol{y}^{(i)} = \hat{\boldsymbol{y}}^{(i)} \\ 0 & \text{otherwise} \end{cases} . \tag{4.20}$$

In order to facilitate comparison, all models, except where noted, have a similar architecture, with $L = 4$ layers having sizes $M = (512, 256, 128, 24)$. Individual models were trained until no improvement was shown on the validation set for two consecutive training epochs.

Significant differences in accuracy are indicated by bold font in the tables of results. We use the Wilcoxon signed-rank test to determine significance. The test is applied on the thirteen pairs of average accuracies for two models, aggregated by the thirteen subjects in the test set. The null hypothesis is that the median difference in average accuracy between two models is zero. The null hypothesis is deemed incorrect, i.e., models were deemed significantly different, when the p-value of the Wilcoxon signed rank test was less than 0.1. We say with 95 % confidence that any model whose accuracy is shown in bold is better at generalizing to unseen heads than the comparison models.

[2]The "compensated" measurements have been equalized to remove the spectral characteristics of the recording system and reproduction system.

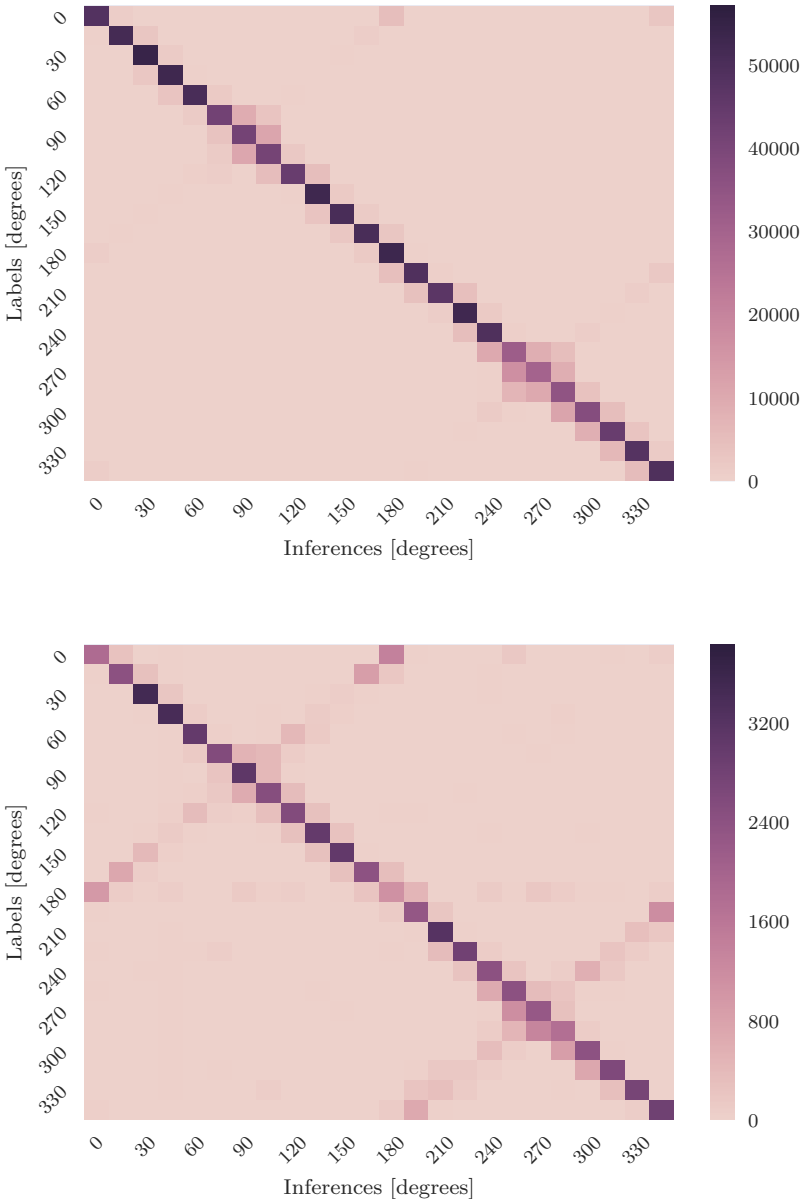

Figure 4.3: Confusion matrices. Top: best complex model. Bottom: baseline complex model using features proposed in (Tsuzuki, Kugler, Kuroyanagi, & Iwata, 2013).

Table 4.3: Comparison of signal domains and nonlinearities for models with complex inputs and weights. Table 4.1 gives the equations for the nonlinearities.

		accuracy[a]	
input transform	activation	Music	Noise
STFT	cartReLU	**0.817**	**0.829**
STFT	modTanh	0.793	—
waveform	cartReLU	0.776	—
waveform	modTanh	0.769	—
STFT	cartTanh	0.763	—
STFT	modReLU	0.753	—
waveform	modReLU	0.742	—
waveform	cartTanh	0.709	0.810
waveform	tanh	0.045	—
STFT	tanh	0.042	—

[a] **Highlighted** indicates Wilcoxon signed-rank test with p-value $< .05$ for per-subject mean accuracy.

4.3.1 Overall accuracy

The confusion matrix for our best performing complex model, which uses a cartReLU nonlinearity and has 81.7 % accuracy, is shown in the top axis of Fig. 4.3. Nearly all misclassifications are between the neighboring classes and along the cones of confusion, where not much information can be gleaned from inter-aural cues. If we consider predictions correct when they are no more than one azimuthal location away from the ground truth, the accuracy is 96.9 %. If we also consider errors along the cones of confusions to be correct, the accuracy is 99.7 %. The bottom axis of Fig. 4.3 shows the confusion matrix for a baseline complex model using features proposed in (Tsuzuki et al., 2013). The baseline model, which has 67.6 % accuracy, is discussed in more detail below.

4.3.2 Nonlinearity

Table 4.3 shows the effect of using various nonlinearities with our proposed model. (Table 4.1 provides the respective equations.) The cartReLU activation, in which a rectified linear unit is placed upon the real and imaginary parts of the pre-activation, provides significantly better performance than the other activations. The hyperbolic tangent function performs predictably bad because we made no attempt to clip the singularities that occur along the imaginary axis. All results presented below this point use the cartReLU nonlinearity for complex hidden layer activations and ReLU nonlinearity for real hidden layer activations.

4.3.3 Complex vs. real weights

We test whether using complex weights provides an advantage to the network over real weights. We also examine whether models that are trained on low-passed or high-passed

Table 4.4: Complex weights are better than real weights for music and high-passed noise inputs.

weight domain	input filter	accuracy[a] Music	Noise
complex	none	**0.817**	0.829
real	none	0.809	0.813
complex	lowpass	**0.734**	0.797
real	lowpass	0.724	0.795
complex	highpass	**0.831**	**0.817**
real	highpass	0.817	0.800

[a] **Highlighted** indicates Wilcoxon signed-rank test with p-value $< .05$ for per-subject mean accuracy.

Table 4.5: Real-imaginary features are better than magnitude-phase and univariate real features.

input transform	feature	accuracy[a] Music	Noise
STFT	real-imaginary	**0.809**	**0.813**
waveform	amplitude	0.775	0.799
STFT	magnitude	0.762	—
STFT	magnitude-phase	0.743	—
STFT	phase	0.586	—

[a] **Highlighted** indicates Wilcoxon signed-rank test with p-value $< .05$ for per-subject mean accuracy.

signal perform significantly different. The filters have a cutoff frequency of 1.5 kHz, which corresponds approximately to the frequency below which IPD cues dominate and above which ILD cues dominate. Table 4.4 shows that networks with complex weights perform significantly better than those with real weights when trained and evaluated on the Music data. Performance on low-passed data, where inter-aural phase cues should dominate, are lower regardless of the numerical domain. However the complex activations provide an advantage. We see no significant difference between models for fullband and lowpass noise. We expect that the additional excitation across spectral components gives the real model more information from which to learn.

4.3.4 Input feature for real models

We also examine whether models having real-valued inputs and weights gain an advantage when the features are similar to complex-valued DFT inputs, i.e., when we provide the model real and imaginary parts but treat them as independent. If the real-imaginary input representation is useful then we should be able to show a significant boost over other representations even when the weights and activations of the model are real. Table 4.5 shows

Table 4.6: Best complex-weight model outperforms baseline complex-weight model (Tsuzuki, Kugler, Kuroyanagi, & Iwata, 2013) and baseline auditory model (Dietz, Ewert, & Hohmann, 2011).

	accuracy[a]	
feature	Music	Noise
complex	**0.817**	**0.829**
Tsuzuki et al. (2013)	0.676	0.520
Dietz et al. (2011)	0.769	—

[a] **Highlighted** indicates Wilcoxon signed-rank test with p-value $< .05$ for per-subject mean accuracy.

that the best performing real model employs a "complex-like" input representation, where the real and imaginary parts of the complex DFT have been concatenated.

The models that have been trained with phase on the polar coordinate system perform the worst, but this isn't very surprising. The values 0 and $2\pi - \epsilon$, where ϵ is a small number, are close together on the polar coordinate system but may appear distant from each other when viewed on the real line. Optimizing a network to learn these relationships appears to be difficult using traditional techniques.

4.3.5 Baseline models

Finally, we compare our model against several others in the literature. Tsuzuki et al. (Tsuzuki et al., 2013) proposed a complex neural network for localization. There are significant differences between their work and ours. First, their system performs localization from a stereo microphone array; binaural audio is not considered. Secondly they focus on a regression, rather than classification, task. Finally, the foundation of their method relies on a costly complex-domain feature extraction step. We implemented their feature extraction method and used it in our network. Table 4.6 reports the results, which are significantly worse compared to our proposed model.

As a final baseline, we compare our model to one that doesn't rely on any neural network. Dietz et al. (Dietz et al., 2011) provide a binaural auditory model of ITD and ILD for estimation of *frontal plane* azimuthal sources. A series of spatial cue extraction steps are performed on the left and right channels of binaural audio. The ITD is estimated from the source signal and compared to a lookup table of responses computed using known HRIRs. We use the implementation provided with the Auditory Modeling Toolbox (Søndergaard & Majdak, 2013), which provides the code and model parameters associated with the paper. Table 4.6 shows that the auditory model achieves 76.9 % on the *frontal only* portion of our test set. Much worse performance occurs (not shown) when evaluated on the full test set. Note that a direct comparison between models is not yet possible because our models were trained on the full lateral plane. Considering that a large portion of our model's error is dominated by front-back confusions, we expect that the performance difference would be larger if we were to train our model only on the frontal plane.

4.4 Discussion and future work

We proposed a complex feedforward neural network that localizes azimuthal source location from binaural audio. We showed that the complex network has better generalization performance across unseen binaural head filters than a comparable real-valued model and several baselines. We believe this work is important because it shows that latent spatial cues of the complex-valued input signal can be directly learned by a neural network, and that these might be superior to hand-engineered features. Furthermore it shows that using complex weights and complex matrix multiplications may be advantageous over their real counterparts for a task that depends on phase-related cues.

We were genuinely surprised that the cartReLU nonlinearity provided best performance for our model. This function effectively limits the activation to the positive quadrant of the Cartesian coordinates. We will investigate this phenomenon further in future work. We're also eager to test our system on more diverse, noise-corrupted, and reverberant source signals. Doing so will help us understand whether the proposed model is robust to real-world environments.

Finally, we will investigate multi-source localization using neural networks with complex activations and multi-frame inputs. We believe that, by fashioning the model to perform location inference on individual time-frequency bins, a similar system might perform source separation and localization on an unknown number of sources.

CHAPTER 5

Speech Enhancement

Speech enhancement, sometimes also known as monaural speech separation, is the task of removing interfering noise from recorded speech. Humans are naturally able to focus their attention on speech in the presence of interfering sounds. We're incredibly good at this, even in quite challenging acoustic contexts, and our aim is to provide algorithms that can do the same. Given an audio recording of a speaker that the model has never seen, embedded in stationary and non-stationary noise that we'd find in the real world, an ideal speech enhancement algorithm should completely remove all non-speech audio, synthesizing "clean" speech that is free of artifacts. Good speech enhancement is related to good machine listening, where a computer is able to make human-like decisions about sound.

In this work, we report on a speaker-independent method for separating speech from highly non-stationary noise, such as blaring television sounds and laughing children. The solution to this task is highly relevant, as improved enhancement can bring better hearing aids and more accurate automatic speech recognition systems. The proposed method uses two deep neural stages. In the first, we train a generative vector quantization model. In the second, we train a dynamic speech enhancement model.

Recent speech enhancement systems have explored deep neural networks (Kolbæk, Tan, & Jensen, 2017; Pascual, Bonafonte, & Serrà, 2017; Vu, Bigot, & Chng, 2016; Xu, Du, Dai, & Lee, 2015; Xu, Du, Huang, Dai, & Lee, 2015). At the same time, many papers have focused on mask approximation (MA) or signal approximation (SA) (P.-S. Huang et al., 2014; Narayanan & Wang, 2013; Xu, Du, Huang, et al., 2015) techniques. In the first case, a model learns to infer a real or binary-valued mask that, when applied to a "noisy" mixture of interfering noise and "clean" speech, returns just the clean speech. In the second case, real-valued clean spectra are estimated directly from noisy speech. Its been recently shown that phase-sensitive signal approximation (PSA) improves enhancement performance

(Erdogan, Hershey, Watanabe, & Le Roux, 2015; Weninger et al., 2015), but otherwise speech enhancement tasks typically treat phase as an afterthought. One notable exception is Williamson, Wang, and Wang (2016), who propose complex domain mask estimation.

In this work, we depart from MA, MSA, and PSA approaches. Rather, we train a generative dictionary model using a stochastic autoencoder called a Conditional Vector Quantization (Conditional VQ) model. Conditional VQ is a stochastic autoencoder that is trained to learn a discrete latent space. It is conditioned on Short-Time Fourier Transform (STFT) frames extracted from clean training data. In the second stage, we train a dynamic speech enhancement (DSE) encoder to map noisy STFT spectra to Conditional VQ codes. The DSE encoder is also a generative model, where frames of clean speech are sampled, conditioned on a sequence of noisy spectra.

The following observations motivate us to project our audio to a discrete latent space. First, recent methods in audio synthesis methods, for example WaveNet (van den Oord et al., 2016), have shown the feasibility of transforming an audio regression task to an audio classification task by discretizing the signal. In that work, raw audio is preprocessed by applying a μ-law companding transformation and then quantizing it to 256 values. We observe that discretization allows some conveniences, such as allowing us to use a cross-entropy (CE) loss function, rather than mean squared error (MSE). MSE assumes that a model's residual will follow a Normal distribution with fixed variance. The cross entropy loss function doesn't carry the same assumptions, namely that "over-predicting" an output is equally as bad as "under-predicting" an output.

Secondly, a discrete representation allows the use of dictionary methods, where we can perform inference over full frames of audio, rather than samples. One criticism of the original WaveNet paper was that inference is too slow for practical use. WaveNet is a sample-level autoregressive model. Our model performs inference over vectors of audio, rather than samples. Using a stochastic vector quantization scheme opens doors to an autoregressive model that is more compact than WaveNet, although we reserve autoregressive models for future work.

Finally we investigate real and complex-field training and inference. In the former case, our real Conditional VQ model is trained on the magnitude spectra of clean training speech to infer a latent discrete representation. The real DSE encoder maps noisy magnitude spectra to the conditional latent space. We use the real Conditional VQ decoder to return our codes back to the audio magnitude spectral domain. Reconstruction is performed using the noisy signal phase. We also train the Conditional VQ and DSE models to map *complex* spectra. In this case, phase and magnitude are reconstructed jointly, without any need to use the noisy phase.

The proposed complex models are not able to beat the performance of the current state of the art speech enhancement model. However we note that complex-domain deep learning is a nascent field, one which has not been studied as closely or for as long as the broader domain of deep leaning. This chapter serves as a guide to integrating complex methods into a difficult task that might later benefit from complex representations.

In the following section we provide background to our methods and how they relate to the literature. We present the models in Section 5.2. Experiments are described in Section 5.3, followed by concluding remarks.

5.1 Related Work

Conditional VQ falls within the general category of dictionary and basis learning methods. There is a rich history of literature on dictionary methods for speech enhancement. Ellis and Weiss (2006) proposed training a speaker-dependent codebook for speech enhancement. Recent works focus on nonnegative matrix factorization (NMF), or variants thereof (Smaragdis, Raj, & Shashanka, 2007; Wang & Sha, 2014; Weninger, Le Roux, Hershey, & Watanabe, 2014), as well as proposals to combine NMF with deep learning methods (Kang, Kwon, Shin, & Kim, 2015; Le Roux, Hershey, & Weninger, 2015; Vu et al., 2016).

In this work, vector quantization is achieved using an autoencoder with a discrete binary-valued hidden layer. In principle, this brings the model under the umbrella of deep hashing binary autoencoders, such as found in (Carreira-Perpinan & Raziperchikolaei, 2015; Liong, Lu, Wang, Moulin, & Zhou, 2015), except that these models are motivated by creating distance-preserving binary encodings of their inputs, whereas there's no distance criteria in out model. Furthermore, our binary encodings are organized into groups representing K-class categorical distributions.

Our vector-based approach may be related to Efficient Neural Audio Synthesis, which brought improvements to WaveNet efficiency by using tensor sub-scaling (Kalchbrenner et al., 2018). Both models infer vectors rather than samples of audio. However our model infers the Conditional VQ discrete latent space, whereas Efficient Neural Synthesis infers audio more directly.

Our Conditional VQ model is a sparse, stochastic autoencoder. The encoder maps audio to the probabilities of a set of categorical distributions. We sample from these in parallel and the decoder provides a nonlinear deterministic mapping back to the audio domain. Training a stochastic model brings many additional complications. In particular, the gradient of a stochastic layer is intractable and we therefore must employ a gradient approximation technique. One popular approach is the score function estimator (Kleijnen & Rubinstein, 1996), also known as likelihood ratio method (Glynn, 1990) and REINFORCE (Williams, 1992). Score function estimators are known to suffer from high variance.

Another approach, when available, is to generate samples from a surrogate distribution, one that does not depend on the parameters of the model. The samples are subsequently transformed, using model parameters, to a different distribution that exhibits desirable properties. For instance the variational autoencoder (Kingma & Welling, 2013) uses the reparameterization trick by approximating the posterior with a standard centered Gaussian distribution that is subsequently transformed to a non-standard uncentered distribution. Our model uses the reparameterization trick by replacing the Categorical distribution with the Concrete distribution (Maddison, Mnih, & Teh, 2016), equivalently known as the Gumbel-Softmax distribution (Jang, Gu, & Poole, 2016). This method provides a rather straightforward way for training a stochastic network.

The DSE encoder is composed of a stack of bidirectional gated recurrent units (GRUs) (Cho et al., 2014). Our model is not autoregressive, like WaveNet. Rather, it infers the independent probabilities of a set of relaxed categorical distributions, conditioned on a sequence of frames of noisy speech. Recent temporally dynamic speech enhancement models have shown to be very successful (Erdogan et al., 2015; Weninger et al., 2015).

Deep complex models are just beginning to gain popularity (Trabelsi et al., 2017). Drude et al. (2016) investigated the feasibility of using partially complex networks for speech enhancement. The architectures of their models are quite different, including different choices for input representation, activation function, and model architecture. Williamson et al. (2016) propose complex mask estimation, where a deep network infers the real and imaginary coefficients of a complex mask. The inputs to the model are a set of real-valued hand-engineered features. The model itself is also real-field, but it is optimized using a phase-sensitive cost function.

5.2 Model

We perform speech enhancement by training two models. The first, Conditional VQ, is a stochastic autoencoder. Its innermost layer provides the log probabilities to N K-class distributions. The Conditional VQ model infers a latent discrete representation for frames of clean speech. The DSE encoder maps a sequence of dirty frames to the latent discrete space of the clean speech.

5.2.1 Inputs and Outputs

We consider an STFT-transformed utterance of noisy speech and an accompanying STFT-transformed utterance of clean speech $(\boldsymbol{Y}, \boldsymbol{S}) \in \mathbb{C}^{F,T}$, where F is the number of frequency bins and T is the number of time frames. Considering for a moment the i-th frame, the inputs to our models are one of the following:

Real. The magnitude spectrum.

$$\boldsymbol{s}^{(r)} := \begin{bmatrix} |S_{0,i}| & |S_{1,i}| & \cdots & |S_{F/2,i}| \end{bmatrix}^{\mathsf{T}} \tag{5.1}$$

Real-imaginary. The real and imaginary components of the spectrum.

$$\boldsymbol{s}^{(ri)} := \begin{bmatrix} \operatorname{Re} S_{0,i} & \operatorname{Re} S_{1,i} & \cdots & \operatorname{Re} S_{F/2,i} & \operatorname{Im} S_{0,i} & \operatorname{Im} S_{1,i} & \cdots & \operatorname{Im} S_{F/2,i} \end{bmatrix}^{\mathsf{T}} \tag{5.2}$$

Complex. The complex spectrum, with the conjugate frequencies truncated.

$$\boldsymbol{s}^{(c)} := \begin{bmatrix} S_{0,i} & S_{1,i} & \cdots & S_{F/1,i} \end{bmatrix} \mathsf{T} \tag{5.3}$$

Augmented complex. The full complex spectrum. This is referred called "augmented" by convention, as the conjugate frequencies are typically discarded.

$$\boldsymbol{s}^{(ac)} := \boldsymbol{S}_{\cdot,i} \tag{5.4}$$

We designate the *estimated* clean speech with a hat symbol, e.g. the estimated augmented complex speech is $\hat{\boldsymbol{s}}^{(as)}$.

The DSE encoder takes a sequence of noisy frames as input and makes an inference in the discrete latent space of the Conditional VQ encoder. The same labels as above are used to designate the outputs. E.g. a frame of real-valued dirty speech is $\boldsymbol{y}^{(r)}$.

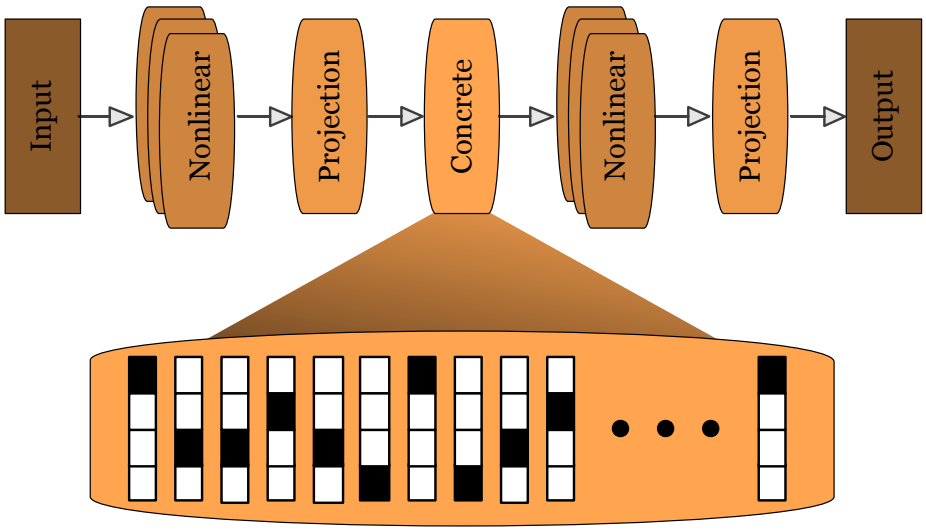

Figure 5.1: Conditional VQ model.

5.2.2 Conditional VQ

The encoder is an M-layer feedforward neural network that acts as a preconditioner to N relaxed categorical distributions having K classes. The distributions are jointly sampled to provide excitation signals to an M'-layer nonlinear decoder. The following sections explain the encoder and decoder, as well as the optimization procedure. Figure 5.1 provides a high-level diagram of the model.

5.2.2.1 Encoder

The encoder is a fully-connected nonlinear preconditioning network followed by a layer of stochastic cells. The i-th preconditioning layer, with $i = 0, 1, \ldots, M$, is composed as follows.

$$\boldsymbol{f}^{(0)} := \boldsymbol{s}, \quad \text{and} \tag{5.5}$$

$$\boldsymbol{f}^{(i)} = \sigma^{(i)}\Big(\boldsymbol{W}^{(i)}\boldsymbol{f}^{(i-1)} + \boldsymbol{b}^{(i)}\Big), \tag{5.6}$$

where $\sigma^{(i)}$ is an activation function, with the condition that $\sigma^{(M)}$ is a linear activation function. If this is a complex Conditional VQ model, then the real and imaginary parts of $\boldsymbol{f}^{(M-1)}$ are concatenated into a real vector before the final linear projection.

$$\boldsymbol{f}^{(M-1)} := \begin{bmatrix} \operatorname{Re}\boldsymbol{f}^{(M-1)} \\ \operatorname{Im}\boldsymbol{f}^{(M-1)} \end{bmatrix} \quad \text{if the model is complex.} \tag{5.7}$$

The final layer $\boldsymbol{f}^{(M)}$ is split into N non-overlapping groups of size K. For $j = 0, 1, \ldots, N-1$,

$$\boldsymbol{f}^{(M_j)} = \begin{bmatrix} f_{jK}^{(M)} \\ f_{jK+1}^{(M)} \\ \vdots \\ f_{jK+K-1}^{(M)} \end{bmatrix}. \tag{5.8}$$

Let $\boldsymbol{\alpha}^{(j)} = \exp\left\{\boldsymbol{f}^{(M_j)}\right\}$ be the exponentiation of one such group. We treat $\boldsymbol{\alpha}^{(j)}$ as the unnormalized locations to a Concrete distribution, Concrete($\boldsymbol{\alpha}^{(j)}, \lambda$), which we briefly describe below. For more detail see (Maddison et al., 2016).

The Concrete distribution uses a surrogate distribution, the Gumbel distribution, to approximate a Categorical distribution. We cannot use the Categorical distribution directly because it is not differentiable. The Concrete distribution employs the "reparameterization trick," in which a non-differentiable function is approximated by reparameterizing it with a differentiable function. Consider a sample $u_k^{(j)}$ drawn from the Uniform distribution that corresponds to the k-th element of $\boldsymbol{\alpha}^{(j)}$,

$$u_k^{(j)} \sim \mathcal{U}(0, 1). \tag{5.9}$$

The sample $u_k^{(j)}$ is transformed into a sample $g_k^{(j)}$ from the Gumbel distribution in the following way,

$$g_k^{(j)} = -\log\left(-\log\left(u_k^{(j)}\right)\right). \tag{5.10}$$

The corresponding sample from the Concrete Distribution is provided by

$$z_k^{(j)} \sim \text{Concrete}(\boldsymbol{\alpha}^{(j)}, \lambda) \tag{5.11}$$

$$= \frac{\exp\left(\left(\log \alpha_k^{(j)} + g_k^{(j)}\right)\Big/\lambda\right)}{\sum_{i=0}^{K-1} \exp\left(\left(\log \alpha_i^{(j)} + g_i^{(j)}\right)\Big/\lambda\right)}, \tag{5.12}$$

where λ is a temperature hyperparameter. Lowering λ makes the distribution more discrete, and raising the temperature softens the distribution. Hence the encoder provides a mapping from $\boldsymbol{s}$ to a set of N K-class categorical samples, which are encoded as one-hot vectors,

$$\boldsymbol{s} \mapsto \left\{\boldsymbol{z}^{(j)} : \boldsymbol{z}^{(j)} \in \mathbb{Z}_2^K \quad \text{and} \quad j = 0, 1, \ldots, N-1\right\}. \tag{5.13}$$

5.2.2.2 Decoder

The decoder is a deterministic feedforward network. It maps the categorical encodings of Eq. (5.13) through a densely connected network to produce inferences about the encodings.

First we concatenate the samples, which are represented by one-hot vectors,

$$\boldsymbol{f}^{(M+1)} := \text{concat}\left(\left\{\boldsymbol{z}^{(j)}\right\}\right) \tag{5.14}$$

$$:= \begin{bmatrix} \boldsymbol{z}^{(0)} \\ \boldsymbol{z}^{(1)} \\ \vdots \\ \boldsymbol{z}^{(N-1)} \end{bmatrix}. \tag{5.15}$$

If this is a complex Conditional VQ model, then the first layer is made complex by adding zero-valued imaginary coefficients,

$$\boldsymbol{f}^{(M+1)} := \boldsymbol{h}^{(M+1)} + \mathrm{i} \cdot \boldsymbol{0} \quad \text{if model is complex.} \tag{5.16}$$

The decoder is composed of M' layers, i.e. for $i = M+2, \ldots, M+M'+2$,

$$\boldsymbol{f}^{(i)} = \sigma^{(i)}\Big(\boldsymbol{W}^{(i)}\boldsymbol{h}^{(i-1)} + \boldsymbol{b}^{(i)}\Big), \tag{5.17}$$

where $\sigma^{(i)}$ is an activation function, with the condition that $\sigma^{(M+M'+2)}$ is a linear activation function. We define the output of the Conditional VQ model as

$$\hat{\boldsymbol{s}} := \boldsymbol{f}^{(M+M'+2)}. \tag{5.18}$$

5.2.2.3 Optimization

The Conditional VQ objective function is Mean Squared Error,

$$\boldsymbol{e} = \boldsymbol{s} - \hat{\boldsymbol{s}}, \tag{5.19}$$

$$\mathcal{L}^{(\text{CVQ})} = \frac{1}{D}\boldsymbol{e}^{*}\boldsymbol{e} \tag{5.20}$$

Because the model employs the reparameterization trick, the Jacobian of the Concrete distribution with respect to the model parameters is well-defined and tractable. We therefore backpropagate the error e in the same way that we would for a non-stochastic network.

5.2.3 DSE Encoder

The DSE model, which is depicted in Fig. 5.2, encodes a sequence of noisy audio frames. Let

$$\boldsymbol{g}^{(0)} := \boldsymbol{y}. \tag{5.21}$$

If the model is complex, then we precondition the encoder using a fully connected L'-layer complex nonlinear network and concatenate the real and imaginary components at the

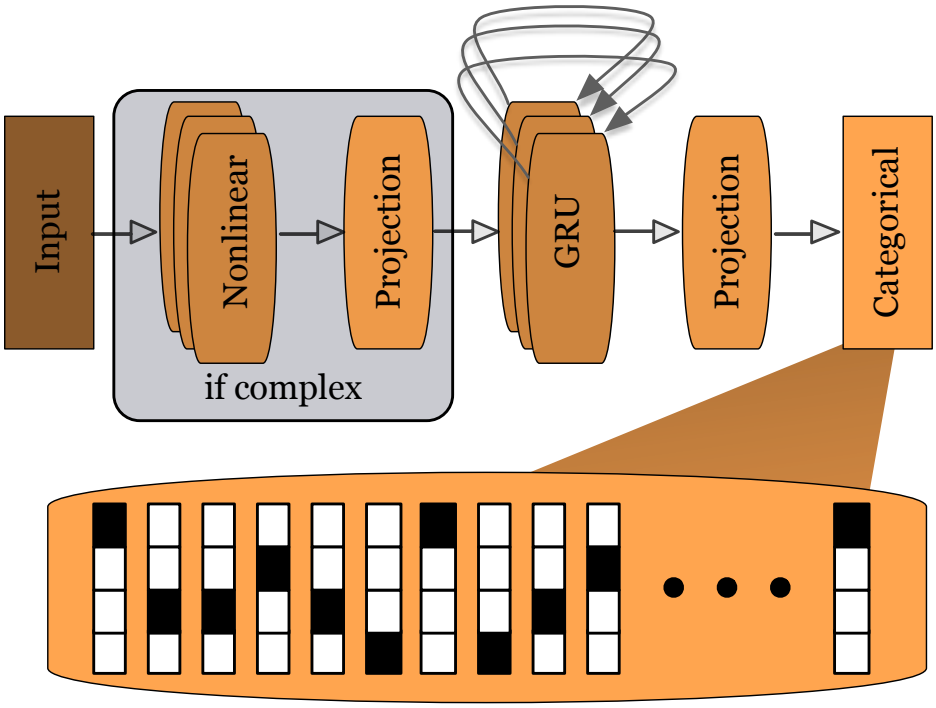

Figure 5.2: DSE Encoder model.

output. For $i = 1, 2, \ldots, L'$, with $L' > 0$ if the model is complex,

$$\boldsymbol{g}^{(i)} = \sigma^{(i)}\left(\boldsymbol{W}^{(i)}\boldsymbol{g}^{(i-1)} + \boldsymbol{b}^{(i)}\right), \quad \text{and} \tag{5.22}$$

$$\boldsymbol{g}^{(L')} := \begin{bmatrix} \operatorname{Re}\boldsymbol{g}^{(L')} \\ \operatorname{Im}\boldsymbol{g}^{(L')} \end{bmatrix}. \tag{5.23}$$

We follow with L layers, where the first $L-1$ are a stack of bidirectional Gated Recurrent Units (GRUs) and the L-th layer is a linear projection. That is, for $i = L'+1, L'+2, \ldots, L'+L-1$,

$$\boldsymbol{g}^{(i)} = \mathrm{GRU}\left(\boldsymbol{g}^{(i-1)}\right) \quad \text{and} \tag{5.24}$$

$$\boldsymbol{g}^{(L'+L-1)} = \boldsymbol{W}^{(L'+L)}\boldsymbol{g}^{(L'+L-1)} + \boldsymbol{b}^{(L'+L)}. \tag{5.25}$$

The output is split into N groups of k-class categorical distributions

$$\boldsymbol{g}^{((L'+L)_j)} = \begin{bmatrix} g_{jK}^{((L'+L)_j)} \\ g_{jK+1}^{((L'+L)_j)} \\ \vdots \\ g_{jK+K-1}^{((L'+L)_j)} \end{bmatrix} \tag{5.26}$$

where we denote $\tilde{\alpha}_k^{(j)} = \exp\left\{g_k^{((L'+L)_j)}\right\}$ the unnormalized probability for the k-th class of the j-th Categorical distribution. Inference is performed by sampling N distributions in parallel.

$$\hat{\boldsymbol{z}}^{(j)} \sim \text{Categorical}(\tilde{\boldsymbol{\alpha}}^{(j)}), \tag{5.27}$$

We optimize the DSE decoder by minimizing the average of the cross entropies from the probability distributions $\boldsymbol{\alpha}^{(j)}$ to the distributions $\tilde{\boldsymbol{\alpha}}^{(j)}$. Designating $|\cdot|$ as the softmax operator,

$$\mathcal{L}^{(\text{DSE})} = -\frac{1}{N}\sum_{j=0}^{N-1}\left|\boldsymbol{\alpha}^{(j)}\right| \cdot \log\left|\tilde{\boldsymbol{\alpha}}^{(j)}\right|. \tag{5.28}$$

5.3 Experiments

5.3.1 Data

Our experiments rely on the left channel of the CHiME-2 medium vocabulary dataset (Vincent et al., 2013), consisting of spoken utterances that have been mixed with interfering background sounds. The background sounds are collected from a portion of the CHiME corpus (Christensen, Barker, Ma, & Green, 2010) and consist of over five hours of domestic sounds recorded in a living room environment using a B&K head and torso simulator (HATS). Typical interfering noises include children laughing; electronic sounds; and street noises. The HATS was also used to catalog binaural room impulse responses (BRIRs).

The CHiME-2 dataset builds an extensive catalog by creating noisy mixtures, in which interfering noises are mixed with clean spoken utterances from the Wall Street Journal (WSJ0) 5k vocabulary read speech corpus (Garofalo, Graff, Paul, & Pallett, 2007). The clean sources are first convolved with the BRIRs. Following the procedure established in (Barker, Vincent, Ma, Christensen, & Green, 2013), the noisy mixtures are created at signal to noise (SNR) levels of −6 dB, −3 dB, 0 dB, 3 dB, 6 dB and 9 dB, where SNR is computed in the following way.

$$\text{SNR} = 10\log_{10}\left(\frac{\boldsymbol{s}^{(l)}\cdot\boldsymbol{s}^{(l)} + \boldsymbol{s}^{(r)}\cdot\boldsymbol{s}^{(r)}}{\boldsymbol{y}^{(l)}\cdot\boldsymbol{y}^{(l)} + \boldsymbol{y}^{(r)}\cdot\boldsymbol{y}^{(r)}}\right), \tag{5.29}$$

with the l and r labels indicating the left and right channels.

The desired SNR levels were achieved by scanning the data and matching the utterances to appropriate segments of interference. This is in contrast to a less ecologically valid but more common technique of scaling interfering noises relative to the utterances. As a consequence, the nature of the interfering noises are somewhat dependent on SNR level, with nonstationary sounds dominating utterances with lower SNRs and almost-stationary noises dominating utterances with higher SNRs.

Train, development, and test sets are created using disjoint sets of utterances, where each utterance is a unique speaker paired with a unique text. The interfering noises are shared

Table 5.1: Description of CHiME-2 dataset and its partitions. Utterances are disjoint between datasets. Each utterance in the training set is associated with one interfering noise. Each utterance in the test and development sets is associated with six interfering noises.

dataset	speakers	unique utterances	total utterances	average SNR
train	83	7138	7138	3.2 dB
development	10	409	2454	1.6 dB
test	12	330	1980	2.3 dB

across datasets. Every utterance in the training set is paired with a background noise related to a randomly chosen SNR level, with no overlap between background noises. The utterances in the test and development sets are associated with background noises for each of the six SNR levels, with minor overlap between background noises. The data is recorded at 16,000 Hz and transformed to STFTs using a frame size of 512 and frame shift of 128 samples. Table 5.1 summarizes the dataset partitioning.

5.3.2 Training

The weights of real feedforward layers are initialized using the Glorot method (Glorot, Bordes, & Bengio, 2011). Complex layers are initialized using complex orthogonal weights (Trabelsi et al., 2017). Real networks are optimized using the Adam optimizer, and complex networks use the Complex Adam optimizer discussed in Chapter 3. Models are trained until no decrease in the objective function is observed for ten consecutive epochs on the validation set.

5.3.3 Metrics

Our models are evaluated using the signal to distortion ratio (SDR) between estimated and reference waveforms, after reconstruction by the overlap-add method. They're also evaluated using the magnitude signal to distortion ratio (MSDR) between the magnitude estimated spectra and magnitude reference spectra. Let $\boldsymbol{X}_{f,t}$ and $\hat{\boldsymbol{X}}_{f,t}$ be reference and estimated spectra. Let $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ be centered (i.e. mean-subtracted) corresponding reference and centered estimated waveforms. Then SDR and MSDR are computed in the following way.

$$\alpha = \frac{\hat{\boldsymbol{y}} \cdot \boldsymbol{y}}{\boldsymbol{y} \cdot \boldsymbol{y}} \tag{5.30}$$

$$\text{SDR}(\boldsymbol{y}, \hat{\boldsymbol{y}}), = 10 \log_{10} \left(\frac{(\alpha \boldsymbol{y}) \cdot (\alpha \boldsymbol{y})}{(\hat{\boldsymbol{y}} - \alpha \boldsymbol{y}) \cdot (\hat{\boldsymbol{y}} - \alpha \boldsymbol{y})} \right) \quad \text{and} \tag{5.31}$$

$$\text{MSDR}(\boldsymbol{X}, \hat{\boldsymbol{X}}) = 10 \log_{10} \left(\frac{\sum_{t=0}^{T-1} |\alpha \boldsymbol{x}_t| \cdot |\alpha \boldsymbol{x}_t|}{\sum_{t=0}^{T-1} (|\hat{\boldsymbol{x}}_t| - |\alpha \boldsymbol{x}_t|) \cdot (|\hat{\boldsymbol{x}}_t| - |\alpha \boldsymbol{x}_t|)} \right). \tag{5.32}$$

Table 5.2: Conditional VQ models and their settings.

Parameter	Model Name Cplx-CVQ-1	Cplx-CVQ-2	Cplx-CVQ-3	Real-CVQ
Model Weights	complex	complex	real	real
Inputs	$\boldsymbol{s}^{(ac)}$	$\boldsymbol{s}^{(ac)}$	$\boldsymbol{s}^{(ri)}$	$\boldsymbol{s}^{(r)}$
Outputs	$\hat{\boldsymbol{s}}^{(ac)}$	$\hat{\boldsymbol{s}}^{(c)}$	$\hat{\boldsymbol{s}}^{(ri)}$	$\hat{\boldsymbol{s}}^{(r)}$
Inputs Dim	512	512	514	257
Outputs Dim	512	257	514	257
Enc Units	512	512	512	512
Dec Units	1024	512	1024	1024
Enc Act[a]	modReLU[c]	modReLU[c]	modReLU[c]	tanh
Dec Act[b]	modReLU[c]	modReLU[c]	modReLU[c]	ReLU
Temp	0.7	0.7	1.3	1.0
Dropout	0.0	0.0	0.0	0.1

[a] Encoder activation.
[b] Decoder activation.
[c] Modulus rectified linear unit, see Table 3.2.

5.3.4 Conditional VQ

Complex and real Conditional VQ models were trained with one hidden layer (512 units) in the encoder and two hidden layers (1024 units each) in the decoder. We investigate two complex models (Cplx-CVQ-1 and Cplx-CVQ-2). They're both given augmented complex inputs. The first model is trained to reproduce augmented complex signal at its output, and the second model produces non-augmented complex signal. Despite the fact that we measure performance based upon the non-augmented output, we hypothesize that using augmented output in the objective function might constrain the model to find a better solution, as it is forced to learn the relationship between non-conjugate and conjugate units. We additionally investigate two real models. The first encodes and decodes real and imaginary components, therefore encoding a complex-valued signal using real weights. The second model is real, learning to encode and decode the magnitude signal.

Table 5.2 provides a listing of the Conditional VQ models, along with some of their settings. In particular, we found that the choice of temperature during training can have a big influence on the success of training a stochastic network. Our models were trained at temperatures 1/3, 2/3, 3/3 and 4/3, and only the best models are shown. (During testing the temperature is set to a very small value, which gives approximately discrete samples.)

Models employed a stochastic code layer consisting of $N = 512$ categorical codes of size $K = 4$. These code sizes were chosen after evaluating the performance on a range of code sizes. Figure 5.3 shows that, for a fixed code size, the reconstruction SDR of a complex Conditional VQ network improves approximately logarithmically with the number codes.

Table 5.3 shows the final SDR results for complex and real conditional VQ models. We first examine the complex models, which are depicted in the upper portion of the table. SDR measures the signal to distortion ratio of the reconstructed signal (including phase) with respect to the original signal. There are three models—Cplx-CVQ-1, Cplx-CVQ-2 and

Figure 5.3: Complex Conditional VQ. SDR for different values of N (number of codes) and a fixed code size of $K = 4$.

Table 5.3: Conditional VQ results. The models listed in the upper portion are complex. The model in the lower portion is real. The performances of the complex and real models are directly comparable only with respect to MSDR and SDR-OP. All results are in dB.

Model	SDR	MSDR	SDR-OP[a]	SDR-NP[b]
Cplx-CVQ-1[c]	19.6	18.0	22.1	—
Cplx-CVQ-2[c]	18.1	16.5	20.4	—
Cplx-CVQ-2[d]	16.9	15.6	19.1	—
Cplx-CVQ-3	15.0	14.0	16.8	—
Real-CVQ[c]	—	19.6	23.6	16.8

[a] Signal to distortion ratio using oracle phase. The ground truth phase replaces the reconstructed phase (complex nets) or is added to the reconstructed signal (real nets).
[b] Signal to distortion ratio using noisy phase. The phase of the corresponding noisy signal is added to the reconstructed signal.
[c] Using Complex Adam optimization.
[d] Using naive Adam optimization.

Cplx-CVQ-3—that reconstruct phase *and* magnitude. The model that employs real weights (Cplx-CVQ-3) underperforms the Cplx-CVQ-1 by more than 4.5 dB. The CplX-CVQ-1 model, which uses an augmented complex output, outperforms one with a truncated complex output by about 1.5 dB.

The default method of optimizing complex models was using the Complex Adam algorithm discussed in Chapter 3. We also trained Cplx-CVQ-2 using the naive implementation of Adam, in which a moving average of the uncentered pseudo-variance of the gradients is stored, rather than the uncentered variance. Cplx-CVQ-2 converges to a worse solution using the naive Adam algorithm.

The magnitude SDR (MSDR) shows us the relative performance of the models if we strip away the phase, providing an idea of how well the models are able to learn magnitude only. The three complex models show the same trend in performance, i.e. the model using augmented complex outputs performs the best. Finally, SDR-OP shows the performance

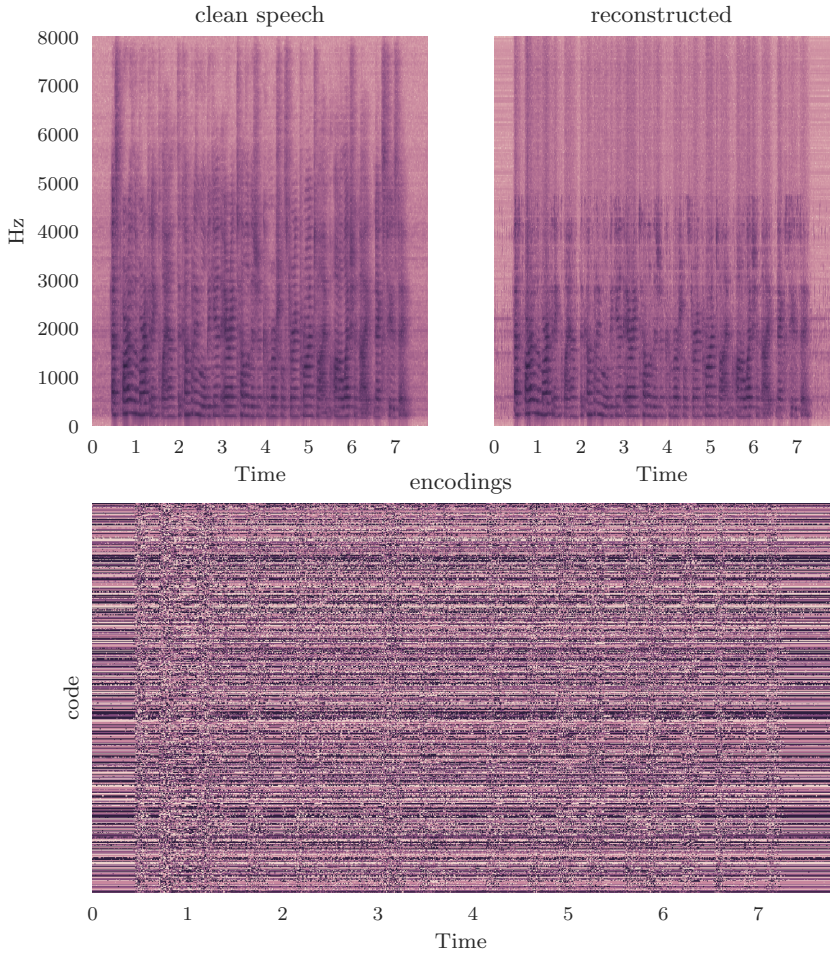

Figure 5.4: Complex Conditional VQ predictions. Top left: power spectrogram of a clean speech utterance from the test set. Top right: a power spectrogram reconstructed using Cplx-CVQ-1. Bottom: stochastic codes inferred by the model.

of the models if we replace the reconstructed phase with an oracle, or ground truth, phase. Again, the four models follow the same trends in performance.

Figure 5.4 shows three images. The upper left image is the power spectrogram of a clean speech utterance from the test set. The upper right panel shows a spectrogram reconstructed using Cplx-CVQ-1. The model seems to be able to reproduce much of the detail residing in the lower frequencies, but accuracy drops off steeply above approximately 2.5 kHz. The bottom panel shows the stochastic codes inferred by the model. The stationary areas in the clean speech are also mostly stationary in the encodings but could perhaps be more stationary.

The bottom portion of Table 5.3 shows the performance of the real Conditional VQ network. It is difficult to provide a direct comparison between this and the complex networks since the real network does not reconstruct phase. However we can make a comparison between the quality of magnitude reconstruction for the networks. By observing the MSDR, we can see that the real network performs better than any of the complex networks. The SDR-OP also shows that if we had access to the oracle phase at reconstruction time the real network would perform better than the best complex network by approximately 1.5 dB. The complex models, which learn phase and magnitude jointly do a poorer job at learning magnitude than the real model, which only learns magnitude.

Finally, we observe SDR-NP, the signal-to-distortion ratio if we reconstruct the signal using the noisy phase, or the phase of the speech-noise mixture. The real network achieves 16.8 dB. Since most speech enhancement models have no mechanism for modeling phase, it is typical to use the noisy phase at reconstruction time. This metric provides a performance ceiling for our speech enhancement task. That is, our real DSE Encoder will not be able to achieve better than 16.8 dB using the Real-CVQ dictionary. Notably, the complex models Cplx-CVQ-1 and Cplx-CVQ-2 each provide a higher performance ceiling, 19.6 dB and 18.1 dB.

5.3.5 DSE Encoder

Table 5.4 shows the settings for two DSE Encoder models (Cplx-DSE and Real-DSE), as well as a baseline speech enhancement model. The complex DSE model is relatively deep, with 1 nonlinear complex feedforward layer, followed by a stack of four GRUs. The real DSE encoder is a little shallower, with no complex nonlinear subnetwork and only 2 stacked gated recurrent unit layers. Both models were trained by truncating sequences to 512 frames and employing the backpropagation-through-time method. The output of both models are the 2048 logits associated with 512 groups of 4-class categorical distributions.

We compare our results with the recurrent network proposed by Erdogan et al. (2015). The baseline model, which uses a phase-sensitive objective function, takes a 100-bin Mel filter bank as its input and produces enhanced speech at its output. It has two stacked LSTM layers and trains through sequences of at least 50 frames and no more than 100 frames. At the time of this writing, the baseline method is the best performing speech enhancement model.

Table 5.5 shows the performance results for our two DSE Encoders and the baseline. It's difficult to make direct comparisons between Real-DSE and Cplx-DSE because the former jointly models phase and magnitude and the latter infers magnitude only. However, it's

Table 5.4: DSE models and their settings.

Parameter	Model Name Cplx-DSE	Real-DSE	Baseline[a]
Model Weights	complex	real	real
Inputs	$\boldsymbol{y}^{(ac)}$	$\boldsymbol{y}^{(r)}$	mfb[b]
Outputs	$\hat{\boldsymbol{z}}$	$\hat{\boldsymbol{z}}$	$\hat{\boldsymbol{s}}^{(r)}$
Inputs Dim	512	257	100
Outputs Dim	2048	2048	unknown
Cplx Layers	1	—	—
Cplx Units	512	—	—
Cplx Act	ModReLU[c]	—	—
RNN Layers	4	2	2
RNN Units	512	1024	384
RNN Act	GRU	GRU	LSTM
Unroll[d]	512	512	50 to 100

[a] Erdogan, Hershey, Watanabe, and Le Roux (2015).
[b] Mel filter bank.
[c] Modulus rectified linear unit, see Table 3.2.
[d] Number of frames used for a training step.

Table 5.5: DSE encoder results. All results are in dB.

Model	SDR	MSDR	SDR-OP[a]	SDR-NP[b]
Cplx-DSE	10.4	10.9	13.6	—
Real-DSE	—	13.2	15.9	13.0
Baseline	—	—	—	14.1

[a] Signal to distortion ratio using oracle phase. The ground truth phase replaces the reconstructed phase (complex nets) or is added to the reconstructed signal (real nets).
[b] Signal to distortion ratio using noisy phase. The phase of the corresponding noisy signal is added to the reconstructed signal.

clear that Real-DSE is a better speech enhancement model. The Real-DSE encoder is able to achieve 13.0 dB SDR when using noisy phase, whereas the complex model achieves only 10.4 dB when inferring phase and magnitude jointly. The real model is also clearly better when examining the magnitude-only SDR (MSDR) and SDR-OP, where the oracle phase is substituted. In addition, the baseline model performs better than both of the DSE encoders, with approximately 1 dB better performance than the real DSE.

5.4 Discussion and Future Work

We presented a two-stage speech enhancement system for real and complex-domain inference. The first stage of enhancement involves training a discrete stochastic autoencoder for projecting real or complex-valued spectra to a categorical latent space. There are a few motivations

for this architecture. First, we'd like to be able to perform enhancement in a discrete space, where a cross-entropy loss function does not enforce the same Gaussian assumptions about a model's residual as a mean-squared loss function. Second, an autoencoder provides the opportunity to learn a compact representation of the signal, where we perform inference over vectors, rather than samples. Third, the proposed architecture allows for jointly modeling phase and magnitude as complex spectra. Finally, by providing a generative model of multiple distributions, we open the door in future work to taking advantage of an autoregressive model that learns a conditional dependence between latent categorical distributions.

We showed that, when modeling complex spectra, the Conditional VQ model benefits from a complex architecture with complex weights. It also benefits from using an augmented complex output, as opposed to an non-augmented complex output. The best complex model (Cplx-CVQ-1) was not able to infer magnitude spectra as well as the real model. This might be related to the fact that nearly the same number of parameters was used for both models, whereas the complex model must learn a much larger space of signal.

We show in Fig. 5.3 that it was necessary to use 512 4-class distributions to achieve approximately 20 dB reconstruction SDR. This number of distributions provides an astronomically large space over which the autoencoder can make inferences. There are 512^4, or more than 1×10^{308}, unique combinations available to the model, which is many orders of magnitude greater than the size of the training data (approximately 6.8×10^6 frames). It seems that we should accomplish the same task with far fewer codes. We suspect that part of the difficulty lies in a noisy gradient. Gradient approximation for stochastic networks is an active area of research, and while we tested several alternative methods such as the score gradient trick, variational methods, and the straight-through gradient, reparameterization via the Concrete distribution gave the best results. We note that while the number of potential code combinations is huge, the best complex model did not simply learn the training data. The model's inferences on the training, development, and test datasets were disjoint, i.e., no two inputs caused the model to generate the same encoding. Nevertheless, more work is needed to make the representation more compact.

The baseline speech enhancement is very strong. Regardless, we were disappointed that neither the complex or real DSE models were able to beat its performance. The real model approaches the performance of the baseline, with about 1 dB difference in performance, although the DSE encoder uses many more parameters. We think that leveraging the chain rule via conditional modeling of latent codes will give us a boost in performance, but this is reserved for future work. In general, we were disappointed that the complex DSE performed so much worse than the real DSE, even though the complex Conditional VQ codes provide a higher performance ceiling. We identify two areas of potential improvement. First, it's clear from Fig. 5.4 that the Conditional VQ model does not do a good job at inferring the detail of the higher frequency components. In future work we will apply pre-emphasis to the input signal to encourage the model to do a better job at learning the upper frequencies. Second, the complex Conditional VQ model learns to encode frames of audio with absolute phase. The phase of an individual frame out of context is arbitrary. Future models will attempt to capture inter-frame phase dependencies by either explicitly modeling relative phase differences between neighboring frames or using a sequence-based autoencoder.

It's clear that there are many further directions for exploration. Nevertheless our models provide real-domain speech enhancement that is competitive with the current state-of-the-art.

This chapter also provides a concept case for performing speech enhancement directly on complex-valued spectra. Considering that the best speech enhancement models do not have mechanisms for modeling phase, and that they usually fall back to mixing dirty phase back in, complex-domain speech enhancement represents and exciting and novel direction worthy of additional investigation.

CHAPTER 6

Conclusion

Complex neural networks have existed for some time, but until recently they have gone largely unnoticed by the deep learning community. There are recent stirrings, e.g., (Arjovsky et al., 2016; Lee, Wang, Wang, Wang, & Wu, 2017; Trabelsi et al., 2017), that suggest that deep complex networks will continue to gain increased attention. But at the moment the literature is relatively sparse. This is the first work to examine deep complex neural networks in explicit relationship to audio.

In Chapter 2 we find that such work is well-motivated. There are many advantages to modeling audio in the frequency-domain, as opposed to the time-domain. Unfortunately, doing so usually involves truncating the phase components of complex-valued audio, as deep complex-valued networks have not been well-studied. Phase encodes fine-scale temporal structure, which is important to tasks that depend on a strong acoustic model, for instance machine listening and audio synthesis. These are areas where the success of a model may be determined by whether the temporal structure within and between audio channels and frequency bins are represented well.

The outstanding literature doesn't provide much guidance on suitable architectures for deep complex networks. Notably, one must make an informed decision about the relative importance of holomorphicity in a network. On the one hand, holomorphic activation functions, like those from the class of elementary transcendental functions, ensure that a complex gradient exists. But we show in Chapter 3 that one can compose holomorphic and non-holomorphic functions in a single neural network. Doing so opens us to a much richer palette of possibilities for network design, allowing us to designate portions of the network that process magnitude independently from the phase.

Until recently, it hasn't been clear how to optimize deep complex networks without the burdensome task of writing code from scratch. We show using the Wirtinger derivatives that,

as long as the objective function is real, backpropagation is relatively simple. We also show that optimization of such networks requires little more than modifying a few lines of code in TensorFlow, a widely-used deep learning library. For instance, the Adam optimization algorithm makes an estimate of the second order statistics of the gradient. For complex optimization, the current implementation needs to be tweaked to ensure that a running average of each gradient's variance, and not pseudo-variance, is monitored. We show in our speech enhancement task in Chapter 5 that the naive method converges to a worse solution.

The guidelines set forth in Chapters 2 and 3 are borne out in Chapters 4 and 5, where we present the results of two real-world and relevant tasks using real and complex networks. In the first task we infer the lateral locations of musical audio objects embedded in binaural audio. The complex model outperforms a comparable real model, as well as several baselines. We show that the task is not successful when using a fully complex network with hyperbolic tangent activation functions, but is successful when using a partially complex network. This result underscores that, whereas some practitioners have espoused fully complex models, deeper networks may be too constrained when fully complex.

In the second task, we tackle speech enhancement by proposing a novel autoencoding architecture that combines a complex network with a stochastic hidden layer. The autoencoder is a generative dictionary that jointly provides discrete encodings of phase and magnitude. The autoencoder's latent space is used to train a dynamic speech enhancement model, which is a generative temporal model of complex variables. The complex models do not outperform comparable real models, nor do they outperform a state of the art baseline model. Yet we show that the complex model raises the theoretical ceiling of performance over the real model. We also show, in further support of our analysis in Chapter 3, that the augmented complex signal provides improvement over a non-augmented complex signal, as well as a real model that has a real-imaginary input signal representation.

In summary, this manuscript makes important contributions to a nascent field. It provides guidance on how to design and train deep complex networks for audio, enabling compositions of arbitrarily deep networks. It also provides use case evidence for relevant real-world tasks, opening a path to future research. While our complex models cannot claim to beat state of the art methods using real networks, we note that the mere application of complex networks is novel, and that a great deal of future work will be necessary in order for complex networks to become competitive with their real counterparts. It took decades for real-valued deep networks to reach their current state of maturity and we expect that complex networks will become more viable as the research community contributes to their advancement.

We believe that complex neural training and inference of deep models is an exciting field of machine learning. There remain a number of research directions to follow. The activation functions presented in this thesis are modeled off of functions that have proven to work well for real networks, but there's no concrete evidence that these functions are the best types of nonlinearities for complex training. This manuscript hasn't studied regularization in complex networks. Complex networks are more regularized than real networks since a complex-valued function has less degrees of freedom than independent functions of real and imaginary components. It's clear that complex networks open a number of potential research questions and directions.

Bibliography

Aizenberg, I. N. (2011). *Complex-valued neural networks with multi-valued neurons*. doi:10.1007/978-3-642-20353-4. (Cited on page 23)

Aizenberg, I. N., Paliy, D. V., Zurada, J. M., & Astola, J. T. (2008). Blur identification by multilayer neural network based on multivalued neurons. *IEEE Transactions on Neural Networks*, *19*(5), 883–898. doi:10.1109/tnn.2007.914158. (Cited on page 5)

Aizenberg, N. N., & Aizenberg, I. N. (1992). CNN based on multi-valued neuron as a model of associative memory for grey scale images. In *CNNA '92 proceedings second international workshop on cellular neural networks and their applications*. doi:10.1109/cnna.1992.274330. (Cited on page 25)

Aizenberg, N. N., & Ivas'kiv, Y. L. (1977). *Multiple-valued threshold logic*. In Russian. Kiev: Naukova Dumka Publisher House. (Cited on page 22).

Aizenberg, N. N., Ivas'kiv, Y. L., & Pospelov, D. A. (1971). About one generalization of the threshold function. In *Doklady Akademii Nauk SSSR (Proceedings of the USSR Academy of Sciences)* (Volume 196, *6*, Pages 1287–1290). In Russian. (Cited on page 23).

Aizenberg, N. N., Ivas'kiv, Y. L., Pospelov, D. A., & Khudyakov, G. F. (1971). Multi-valued threshold functions. *Cybernetics*, *7*(4), 626–635. doi:10.1007/bf01071034. (Cited on page 23)

Aizenberg, N. N., Ivas'kiv, Y. L., Pospelov, D. A., & Khudyakov, G. F. (1973). Multivalued threshold functions. *Cybernetics*, *9*(1), 61–77. doi:10.1007/bf01068667. (Cited on page 23)

American National Standards Institute. (2013). ANSI/ASA S1.1-2013: American national standard on acoustical terminology. American Naional Standards Institute. (Cited on page 11).

Amin, M. F., Amin, M. I., Al-Nuaimi, A. Y. H., & Murase, K. (2011). Wirtinger calculus based gradient descent and Levenberg-Marquardt learning algorithms in complex-valued neural networks. In *Neural information processing* (Pages 550–559). doi:10.1007/978-3-642-24955-6_66. (Cited on pages 29, 32)

Arjovsky, M., Shah, A., & Bengio, Y. (2016). Unitary evolution recurrent neural networks. In M. F. Balcan & K. Q. Weinberger (Editors), *Proceedings of the 33rd international conference on machine learning* (Volume 48, Pages 1120–1128). Proceedings of Machine Learning Research. New York, New York, USA: PMLR. Retrieved from http://proceedings.mlr.press/v48/arjovsky16.html. (Cited on pages 5, 27, 50, 77)

Barker, J., Vincent, E., Ma, N., Christensen, H., & Green, P. D. (2013). The PASCAL CHiME speech separation and recognition challenge. *Computer Speech & Language*, *27*(3), 621–633. doi:10.1016/j.csl.2012.10.004. (Cited on page 67)

Barron, A. R. (1994). Approximation and estimation bounds for artificial neural networks. *Machine Learning*, *14*(1), 115–133. doi:10.1007/bf00993164. (Cited on page 24)

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, *2*(1), 1–127. doi:10.1561/2200000006. (Cited on page 5)

Benvenuto, N., Marchesi, M., Piazza, F., & Uncini, A. (1991a). A comparison between real and complex neural networks in communication applications. In *Artificial neural networks* (Pages 1177–1180). doi:10.1016/b978-0-444-89178-5.50046-4. (Cited on page 23)

Benvenuto, N., Marchesi, M., Piazza, F., & Uncini, A. (1991b). Non linear satellite radio links equalized using blind neural networks. In *ICASSP 91: 1991 international conference on acoustics, speech, and signal processing*. doi:10.1109/icassp.1991.150526. (Cited on pages 5, 23)

Benvenuto, N., & Piazza, F. (1992). On the complex backpropagation algorithm. *IEEE Trans. Signal Processing*, *40*(4), 967–969. doi:10.1109/78.127967. (Cited on pages 23, 24)

Birx, D. L. (1990). *The design of a neural network that performs a complex mapping for phase sensitive detection and characterization of eddy current impedance plane data* (Doctoral dissertation, University of Dayton). Retrieved from http://search.proquest.com/docview/303876508?accountid=10422. (Cited on page 23)

Bittner, R. M., McFee, B., Salamon, J., Li, P., & Bello, J. P. (2017). Deep salience representations for F0 estimation in polyphonic music. In S. J. Cunningham, Z. Duan, X. Hu, & D. Turnbull (Editors), *Proceedings of the 18th international society for music information*

retrieval conference (Pages 63–70). Retrieved from https://ismir2017.smcnus.org/wp-content/uploads/2017/10/85_Paper.pdf. (Cited on page 5)

Bittner, R. M., Salamon, J., Tierney, M., Mauch, M., Cannam, C., & Bello, J. P. (2014). MedleyDB: A multitrack dataset for annotation-intensive MIR research. In *Proceedings of the 15th international society for music information retrieval conference* (Pages 155–160). Retrieved from http://www.terasoft.com.tw/conf/ismir2014/proceedings/T028_322_Paper.pdf. (Cited on page 51)

Blauert, J. (1997). *Spatial hearing: The psychophysics of human sound localization*. MIT press. (Cited on page 2).

Brandwood, D. H. (1983). A complex gradient operator and its application in adaptive array theory. *IEE Proceedings F Communications, Radar and Signal Processing*, *130*(1), 11. doi:10.1049/ip-f-1.1983.0003. (Cited on pages 22, 23, 29)

Bregman, A. S. (1994). *Auditory scene analysis: The perceptual organization of sound*. A Bradford Book. (Cited on page 2).

Buchholz, S. (2005). *A theory of neural computation with Clifford algebras* (Doctoral dissertation, University of Kiel). Retrieved from http://e-diss.uni-kiel.de/diss_1402/index.htm. (Cited on page 27)

Buchholz, S., & Sommer, G. (2000). A hyperbolic multilayer perceptron. In *Proceedings of the IEEE-INNS-ENNS international joint conference on neural networks. IJCNN 2000. Neural computing: New challenges and perspectives for the new millennium.* doi:10.1109/ijcnn.2000.857886. (Cited on page 27)

Carreira-Perpinan, M. A., & Raziperchikolaei, R. (2015). Hashing with binary autoencoders. In *2015 IEEE conference on computer vision and pattern recognition (CVPR)*. doi:10.1109/cvpr.2015.7298654. (Cited on page 61)

Ceylan, M., Çetinkaya, N., Ceylan, R., & Özbay, Y. (2006). Comparison of complex-valued neural network and fuzzy clustering complex-valued neural network for load-flow analysis. In *Artificial intelligence and neural networks* (Pages 92–99). doi:10.1007/11803089_11. (Cited on page 5)

Cha, I., & Kassam, S. A. (1995). Channel equalization using adaptive complex radial basis function networks. *IEEE Journal on Selected Areas in Communications*, *13*(1), 122–131. doi:10.1109/49.363139. (Cited on page 27)

Chakrabarty, S., & Habets, E. A. P. (2017). Broadband DOA estimation using convolutional neural networks trained with noise signals. In *2017 IEEE workshop on applications of signal processing to audio and acoustics (WASPAA)*. doi:10.1109/waspaa.2017.8170010. (Cited on page 46)

Chan, W., Jaitly, N., Le, Q., & Vinyals, O. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE international*

conference on acoustics, speech and signal processing (ICASSP). doi:10.1109/icassp.2016.7472621. (Cited on page 1)

Chen, S., McLaughlin, S., & Mulgrew, B. (1994a). Complex-valued radial basic function network, part I: Network architecture and learning algorithms. *Signal Processing*, *35*(1), 19–31. doi:10.1016/0165-1684(94)90187-2. (Cited on page 27)

Chen, S., McLaughlin, S., & Mulgrew, B. (1994b). Complex-valued radial basis function network, part II: Application to digital communications channel equalisation. *Signal Processing*, *36*(2), 175–188. doi:10.1016/0165-1684(94)90206-2. (Cited on page 27)

Chistyakov, Y. S., Kholodova, E. V., Minin, A., Zimmermann, H.-G., & Knoll, A. (2011). Modeling of electric power transformer using complex-valued neural networks. *Energy Procedia*, *12*, 638–647. doi:10.1016/j.egypro.2011.10.087. (Cited on page 5)

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014, June 3). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv: 1406.1078v3 [cs.CL]. (Cited on page 61)

Christensen, H., Barker, J., Ma, N., & Green, P. D. (2010). The CHiME corpus: A resource and a challenge for computational hearing in multisource environments. In T. Kobayashi, K. Hirose, & S. Nakamura (Editors), *INTERSPEECH 2010, 11th annual conference of the international speech communication association* (Pages 1918–1921). ISCA. Retrieved from http://www.isca-speech.org/archive/interspeech_2010/i10_1918.html. (Cited on page 67)

Clarke, T. L. (1990). Generalization of neural networks to the complex plane. In *1990 IJCNN international joint conference on neural networks*. doi:10.1109/ijcnn.1990.137751. (Cited on pages 23, 24)

Cramer, E. M., & Huggins, W. H. (1958). Creation of pitch through binaural interaction. *The Journal of the Acoustical Society of America*, *30*(5), 413–417. doi:10.1121/1.1909628. (Cited on page 16)

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, *2*(4), 303–314. doi:10.1007/bf02551274. (Cited on page 24)

Danihelka, I., Wayne, G., Uria, B., Kalchbrenner, N., & Graves, A. (2016). Associative long short-term memory. In M. F. Balcan & K. Q. Weinberger (Editors), *Proceedings of the 33rd international conference on machine learning* (Volume 48, Pages 1986–1994). Proceedings of Machine Learning Research. New York, New York, USA: PMLR. Retrieved from http://proceedings.mlr.press/v48/danihelka16.html. (Cited on page 5)

Deleforge, A., Forbes, F., & Horaud, R. (2015). Acoustic space learning for sound-source separation and localization on binaural manifolds. *International Journal of Neural Systems*, *25*(01), 1440003. doi:10.1142/s0129065714400036. (Cited on page 48)

Dietz, M., Ewert, S. D., & Hohmann, V. (2011). Auditory model based direction estimation of concurrent speakers from binaural signals. *Speech Communication*, *53*(5), 592–605. doi:10.1016/j.specom.2010.05.006. (Cited on pages 46, 56)

Donq-Liang, L., & Wen-June, W. (1998). A multivalued bidirectional associative memory operating on a complex domain. *Neural Networks*, *11*(9), 1623–1635. doi:10.1016/s0893-6080(98)00078-1. (Cited on page 22)

Drude, L., Raj, B., & Haeb-Umbach, R. (2016). On the appropriateness of complex-valued neural networks for speech enhancement. In *Interspeech 2016*. doi:10.21437/interspeech.2016-300. (Cited on pages 31, 62)

Duchi, J. C., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, *12*, 2121–2159. Retrieved from http://dl.acm.org/citation.cfm?id=2021068. (Cited on page 35)

Ellis, D. P. W., & Weiss, R. J. (2006). Model-based monaural source separation using a vector-quantized phase-vocoder representation. In *2006 IEEE international conference on acoustics speed and signal processing proceedings*. doi:10.1109/icassp.2006.1661436. (Cited on page 61)

Engel, J., Resnick, C., Roberts, A., Dieleman, S., Eck, D., Simonyan, K., & Norouzi, M. (2017, April 5). Neural audio synthesis of musical notes with WaveNet autoencoders. arXiv: 1704.01279v1 [cs.LG]. (Cited on page 1)

Erdogan, H., Hershey, J. R., Watanabe, S., & Le Roux, J. (2015). Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. doi:10.1109/icassp.2015.7178061. (Cited on pages 60, 61, 72, 73)

FitzGerald, D., Liutkus, A., & Badeau, R. (2016). PROJET — Spatial audio separation using projections. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. doi:10.1109/icassp.2016.7471632. (Cited on page 47)

Funahashi, K.-I. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks*, *2*(3), 183–192. doi:10.1016/0893-6080(89)90003-8. (Cited on page 24)

Garofalo, J., Graff, D., Paul, D., & Pallett, D. (2007). Continous speech recognition (CSR-I) Wall Street Journal (WSJ0) news, complete. (Cited on page 67).

Georgiou, G. M., & Koutsougeras, C. (1992). Complex domain backpropagation. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, *39*(5), 330–334. doi:10.1109/82.142037. (Cited on pages 23–27)

Gerkmann, T., Krawczyk-Becker, M., & Le Roux, J. (2015). Phase processing for single-channel speech enhancement: History and recent advances. *IEEE Signal Processing Magazine*, *32*(2), 55–66. doi:10.1109/msp.2014.2369251. (Cited on page 4)

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. In G. Gordon, D. Dunson, & M. Dudík (Editors), *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (Volume 15, Pages 315–323). Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR. Retrieved from http://proceedings.mlr.press/v15/glorot11a.html. (Cited on page 68)

Glynn, P. W. (1990). Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, *33*(10), 75–84. doi:10.1145/84537.84552. (Cited on page 61)

Goh, S. L., Chen, M., Popović, D. H., Aihara, K., Obradovic, D., & Mandic, D. P. (2006). Complex-valued forecasting of wind profile. *Renewable Energy*, *31*(11), 1733–1750. doi:10.1016/j.renene.2005.07.006. (Cited on page 35)

Goh, S. L., & Mandic, D. P. (2007). An augmented CRTRL for complex-valued recurrent neural networks. *Neural Networks*, *20*(10), 1061–1066. doi:10.1016/j.neunet.2007.09.015. (Cited on page 23)

Goh, S. L., Popovic, D. H., & Mandic, D. P. (2004). Complex-valued estimation of wind profile and wind power. In *Proceedings of the 12th IEEE mediterranean electrotechnical conference*. doi:10.1109/melcon.2004.1348231. (Cited on pages 5, 35)

Goldstein, J. L. (1967). Auditory spectral filtering and monaural phase perception. *The Journal of the Acoustical Society of America*, *41*(2), 458–479. doi:10.1121/1.1910357. (Cited on page 16)

Griffin, D., & Lim, J. (1984). Signal estimation from modified short-time Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *32*(2), 236–243. doi:10.1109/tassp.1984.1164317. (Cited on page 4)

Guberman, N. (2016, February 29). On complex valued convolutional neural networks. arXiv: 1602.09046v1 [cs.NE]. (Cited on page 27)

Gunawan, D., & Sen, D. (2005). Sinusoidal frequency estimation based on the time derivative of the STFT phase response. In *2005 5th international conference on information communications & signal processing*. doi:10.1109/icics.2005.1689299. (Cited on page 17)

Hahnloser, R. H. R., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., & Seung, H. S. (2000). Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, *405*(6789), 947–951. doi:10.1038/35016072. (Cited on page 27)

Han, Y., Kim, J., & Lee, K. (2017). Deep convolutional neural networks for predominant instrument recognition in polyphonic music. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *25*(1), 208–221. doi:10.1109/taslp.2016.2632307. (Cited on page 1)

Hawthorne, C., Elsen, E., Song, J., Roberts, A., Simon, I., Raffel, C., ... Eck, D. (2017, October 30). Onsets and frames: Dual-objective piano transcription. arXiv: 1710.11153v1 [cs.SD]. (Cited on page 5)

Haykin, S. S. (2014). *Adaptive filter theory* (5th edition). Upper Saddle River, New Jersey: Pearson. (Cited on page 29).

Hirose, A. (1992a). Continuous complex-valued back-propagation learning. *Electronics Letters*, *28*(20), 1854–1855. doi:10.1049/el:19921186. (Cited on pages 25, 27)

Hirose, A. (1992b). Dynamics of fully complex-valued neural networks. *Electronics Letters*, *28*(16), 1492–1494. doi:10.1049/el:19920948. (Cited on page 25)

Hirose, A. (1994). Applications of complex-valued neural networks to coherent optical computing using phase-sensitive detection scheme. *Information Sciences: Applications*, *2*(2), 103–117. doi:10.1016/1069-0115(94)90014-0. (Cited on page 5)

Hirose, A. (2011). Nature of complex number and complex-valued neural networks. *Frontiers of Electrical and Electronic Engineering in China*, *6*(1), 171–180. doi:10.1007/s11460-011-0125-3. (Cited on page 6)

Hirose, A., Higo, T., & Tanizawa, K. (2006a). Efficient generation of holographic movies with frame interpolation using a coherent neural network. *IEICE Electronic Express*, *3*(19), 417–423. doi:10.1587/elex.3.417. (Cited on page 5)

Hirose, A., Higo, T., & Tanizawa, K. (2006b). Holographic three-dimensional movie generation with frame interpolation using coherent neural networks. In *The 2006 IEEE international joint conference on neural network proceedings*. doi:10.1109/ijcnn.2006.246722. (Cited on page 5)

Hirose, A., & Kiuchi, M. (2000). Coherent optical associative memory system that processes complex-amplitude information. *IEEE Photonics Technology Letters*, *12*(5), 564–566. doi:10.1109/68.841287. (Cited on page 5)

Hirose, A., & Onishi, H. (1999). Proposal of relative-minimization learning for behavior stabilization of complex-valued recurrent neural networks. *Neurocomputing*, *24*(1-3), 163–171. doi:10.1016/s0925-2312(98)00093-9. (Cited on page 23)

Hirose, A., & Yoshida, S. (2012). Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(4), 541–551. doi:10.1109/tnnls.2012.2183613. (Cited on page 6)

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. doi:10.1162/neco.1997.9.8.1735. (Cited on page 5)

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, *4*(2), 251–257. doi:10.1016/0893-6080(91)90009-t. (Cited on page 5)

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366. doi:10.1016/0893-6080(89)90020-8. (Cited on page 24)

Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., ... Liu, H. H. (1998). The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *454*(1971), 903–995. doi:10.1098/rspa.1998.0193. (Cited on page 18)

Huang, R.-C., & Chen, M.-S. (2000). Adaptive equalization using complex-valued multilayered neural network based on the extended kalman filter. In *WCC 2000 - ICSP 2000. 2000 5th international conference on signal processing proceedings. 16th world computer congress 2000*. doi:10.1109/icosp.2000.894544. (Cited on page 5)

Huang, P.-S., Kim, M., Hasegawa-Johnson, M., & Smaragdis, P. (2014). Deep learning for monaural speech separation. In *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. doi:10.1109/icassp.2014.6853860. (Cited on pages 1, 59)

Humphrey, E. J., Bello, J. P., & LeCun, Y. (2013). Feature learning and deep architectures: New directions for music informatics. *Journal of Intelligent Information Systems*, *41*(3), 461–481. doi:10.1007/s10844-013-0248-5. (Cited on page 2)

Jang, E., Gu, S., & Poole, B. (2016, November 3). Categorical reparameterization with Gumbel-Softmax. arXiv: 1611.01144v5 [stat.ML]. (Cited on page 61)

Jankowski, S., Lozowski, A., & Zurada, J. M. (1996). Complex-valued multistate neural associative memory. *IEEE Trans. Neural Networks*, *7*(6), 1491–1496. doi:10.1109/72.548176. (Cited on pages 22, 25)

Jansson, A., Humphrey, E. J., Montecchio, N., Bittner, R. M., Kumar, A., & Weyde, T. (2017). Singing voice separation with deep U-net convolutional networks. In S. J. Cunningham, Z. Duan, X. Hu, & D. Turnbull (Editors), *Proceedings of the 18th international society for music information retrieval conference* (Pages 745–751). Retrieved from https://ismir2017.smcnus.org/wp-content/uploads/2017/10/171_Paper.pdf. (Cited on page 5)

Jianping, D., Sundararajan, N., & Saratchandran, P. (2002). Communication channel equalization using complex-valued minimal radial basis function neural networks. *IEEE Transactions on Neural Networks*, *13*(3), 687–696. doi:10.1109/tnn.2002.1000133. (Cited on page 27)

Kalchbrenner, N., Elsen, E., Simonyan, K., Noury, S., Casagrande, N., Lockhart, E., ... Kavukcuoglu, K. (2018, February 23). Efficient neural audio synthesis. arXiv: 1802.08435v1 [cs.SD]. (Cited on pages 4, 61)

Kang, T. G., Kwon, K., Shin, J. W., & Kim, N. S. (2015). NMF-based target source separation using deep neural network. *IEEE Signal Processing Letters*, *22*(2), 229–233. doi:10.1109/lsp.2014.2354456. (Cited on page 61)

Kataoka, M., Kinouchi, M., & Hagiwara, M. (1998). Music information retrieval system using complex-valued recurrent neural networks. In *SMC'98 conference proceedings. 1998 IEEE international conference on systems, man, and cybernetics*. doi:10.1109/icsmc.1998.727520. (Cited on page 6)

Kechriotis, G., & Manolakos, E. S. (1994). Training fully recurrent neural networks with complex weights. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, *41*(3), 235–238. doi:10.1109/82.279210. (Cited on page 23)

Kim, M. S., & Guest, C. C. (1990). Modification of backpropagation networks for complex-valued signal processing in frequency domain. In *1990 IJCNN international joint conference on neural networks*. doi:10.1109/ijcnn.1990.137820. (Cited on page 23)

Kim, T., & Adalı, T. (2001). Complex backpropagation neural network using elementary transcendental activation functions. In *2001 IEEE international conference on acoustics, speech, and signal processing. proceedings*. doi:10.1109/icassp.2001.941159. (Cited on page 25)

Kim, T., & Adalı, T. (2002). Fully complex multi-layer perceptron network for nonlinear signal processing. *The Journal of VLSI Signal Processing*, *32*(1/2), 29–43. doi:10.1023/a:1016359216961. (Cited on page 25)

Kim, T., & Adalı, T. (2003). Approximation by fully complex multilayer perceptrons. *Neural Computation*, *15*(7), 1641–1666. doi:10.1162/089976603321891846. (Cited on pages 25, 27)

Kingma, D. P., & Ba, J. (2014, December 22). Adam: A method for stochastic optimization. arXiv: 1412.6980v9 [cs.LG]. (Cited on pages 21, 34, 51)

Kingma, D. P., & Welling, M. (2013, December 20). Auto-encoding variational bayes. arXiv: 1312.6114v10 [stat.ML]. (Cited on page 61)

Kingsbury, N. (2001). Complex wavelets for shift invariant analysis and filtering of signals. *Applied and Computational Harmonic Analysis*, *10*(3), 234–253. doi:10.1006/acha.2000.0343. (Cited on page 18)

Kinouchi, M., & Hagiwara, M. (1995). Learning temporal sequences by complex neurons with local feedback. In *Proceedings of ICNN'95 - international conference on neural networks*. doi:10.1109/icnn.1995.487291. (Cited on page 6)

Kinouchi, M., & Hagiwara, M. (1996). Memorization of melodies by complex-valued recurrent network. In *Proceedings of international conference on neural networks (ICNN'96)*. doi:10.1109/icnn.1996.549090. (Cited on page 6)

Kleijnen, J. P. C., & Rubinstein, R. Y. (1996). Optimization and sensitivity analysis of computer simulation models by the score function method. *European Journal of Operational Research*, *88*(3), 413–427. doi:https://doi.org/10.1016/0377-2217(95)00107-7. (Cited on page 61)

Kolbæk, M., Tan, Z.-H., & Jensen, J. (2017). Speech intelligibility potential of general and specialized deep neural network based speech enhancement systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *25*(1), 153–167. doi:10.1109/taslp.2016.2628641. (Cited on page 59)

Kreutz-Delgado, K. (2009, June 26). The complex gradient operator and the CR-calculus. arXiv: 0906.4835v1 [math.OC]. (Cited on pages 22, 29, 30)

Kuroe, Y., Hashimoto, N., & Mori, T. (2001a). Qualitative analysis of a self-correlation type complex-valued associative memories. *Nonlinear Analysis: Theory, Methods & Applications*, *47*(9), 5795–5806. doi:10.1016/s0362-546x(01)00823-9. (Cited on page 22)

Kuroe, Y., Hashimoto, N., & Mori, T. (2001b). Qualitative analysis of continuous complex-valued associative memories. In *Artificial neural networks — ICANN 2001* (Pages 843–851). doi:10.1007/3-540-44668-0_117. (Cited on page 22)

Kuroe, Y., & Taniguchi, Y. (2007). Models of orthogonal type complex-valued dynamic associative memories and their performance comparison. In *Lecture notes in computer science* (Pages 838–847). doi:10.1007/978-3-540-74690-4_85. (Cited on page 22)

Kuroe, Y., Yoshid, M., & Mori, T. (2003). On activation functions for complex-valued neural networks — existence of energy functions —. In *Artificial neural networks and neural information processing — ICANN/ICONIP 2003* (Pages 985–992). doi:10.1007/3-540-44989-2_117. (Cited on pages 24, 27)

Le Roux, J., Hershey, J. R., & Weninger, F. (2015). Deep NMF for speech separation. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. doi:10.1109/icassp.2015.7177933. (Cited on page 61)

Lee, Y.-S., Wang, C.-Y., Wang, S.-F., Wang, J.-C., & Wu, C.-H. (2017). Fully complex deep neural network for phase-incorporating monaural source separation. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. doi:10.1109/icassp.2017.7952162. (Cited on page 77)

Leung, H., & Haykin, S. (1991). The complex backpropagation algorithm. *IEEE Trans. Signal Processing*, *39*(9), 2101–2104. doi:10.1109/78.134446. (Cited on page 23)

Li, D., & Levinson, S. E. (2003). A bayes-rule based hierarchical system for binaural sound source localization. In *2003 IEEE international conference on acoustics, speech, and signal processing, 2003. proceedings. (ICASSP '03)*. doi:10.1109/icassp.2003.1200021. (Cited on page 48)

Li, H., & Adalı, T. (2008). Complex-valued adaptive signal processing using nonlinear functions. *EURASIP Journal on Advances in Signal Processing*, *2008*(1). doi:10.1155/2008/765615. (Cited on page 32)

Liang, F. T., Gotham, M., Johnson, M., & Shotton, J. (2017). Automatic stylistic composition of bach chorales with deep LSTM. In S. J. Cunningham, Z. Duan, X. Hu, & D. Turnbull

(Editors), *Proceedings of the 18th international society for music information retrieval conference* (Pages 449–456). Retrieved from https://ismir2017.smcnus.org/wp-content/uploads/2017/10/156_Paper.pdf. (Cited on page 5)

Liong, V. E., Lu, J., Wang, G., Moulin, P., & Zhou, J. (2015). Deep hashing for compact binary codes learning. In *2015 IEEE conference on computer vision and pattern recognition (CVPR)*. doi:10.1109/cvpr.2015.7298862. (Cited on page 61)

Little, G. R., Gustafson, S. C., & Senn, R. A. (1990). Generalization of the backpropagation neural network learning algorithm to permit complex weights. *Applied Optics*, *29*(11), 1591. doi:10.1364/ao.29.001591. (Cited on page 23)

Liu, H., Pang, C., & Zhang, J. (2015). Binaural sound source localization based on generalized parametric model and two-layer matching strategy in complex environments. In *2015 IEEE international conference on robotics and automation (ICRA)*. doi:10.1109/icra.2015.7139822. (Cited on page 48)

Liu, H., & Zhang, J. (2014). A binaural sound source localization model based on time-delay compensation and interaural coherence. In *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. doi:10.1109/icassp.2014.6853832. (Cited on page 48)

Liu, H., Zhang, J., & Fu, Z. (2014). A new hierarchical binaural sound source localization method based on interaural matching filter. In *2014 IEEE international conference on robotics and automation (ICRA)*. doi:10.1109/icra.2014.6907065. (Cited on page 48)

Maddison, C. J., Mnih, A., & Teh, Y. W. (2016, November 2). The Concrete distribution: A continuous relaxation of discrete random variables. arXiv: 1611.00712v3 [cs.LG]. (Cited on pages 61, 64)

Maezawa, A. (2017). Fast and accurate: Improving a simple beat tracker with a selectively-applied deep beat identification. In S. J. Cunningham, Z. Duan, X. Hu, & D. Turnbull (Editors), *Proceedings of the 18th international society for music information retrieval conference* (Pages 309–315). Retrieved from https://ismir2017.smcnus.org/wp-content/uploads/2017/10/9_Paper.pdf. (Cited on page 5)

Mandel, M. I. (2010). *Binaural model-based source separation and localization* (Doctoral dissertation, Columbia University). (Cited on page 48).

Mandic, D. P., & Goh, V. S. L. (2009). *Complex valued nonlinear adaptive filters*. doi:10.1002/9780470742624. (Cited on pages 23, 24, 37)

Mathes, R. C., & Miller, R. L. (1947). Phase effects in monaural perception. *The Journal of the Acoustical Society of America*, *19*(5), 780–797. doi:10.1121/1.1916623. (Cited on page 16)

McDermott, J. H. (2013). Audition. In K. Ochsner & S. M. Kosslyn (Editors), *The oxford handbook of cognitive neuroscience* (Chapter 8, Volume 1, Pages 135–170). Oxford University Press. (Cited on page 1).

Mehri, S., Kumar, K., Gulrajani, I., Kumar, R., Jain, S., Sotelo, J., ... Bengio, Y. (2016, December 22). SampleRNN: An unconditional end-to-end neural audio generation model. arXiv: 1612.07837v2 [cs.SD]. (Cited on pages 4, 13)

Middlebrooks, J. C., & Green, D. M. (1991). Sound localization by human listeners. *Annual Review of Psychology*, *42*(1), 135–159. doi:10.1146/annurev.ps.42.020191.001031. (Cited on pages 3, 47)

Minami, M., & Hirose, A. (2003). Phase singular points reduction by a layered complex-valued neural network in combination with constructive Fourier synthesis. In *Artificial neural networks and neural information processing — ICANN/ICONIP 2003* (Pages 943–950). doi:10.1007/3-540-44989-2_112. (Cited on page 5)

Minin, A. (2012). *Modeling of dynamical systems with complex valued recurrent neural networks* (Doctoral dissertation, Technische Universität München, München). (Cited on page 29).

Minin, A., Chistyakov, Y., Kholodova, E., Zimmermann, H., & Knoll, A. (2012). Complex valued open recurrent neural network for power transformer modeling. *International Journal of Applied Mathematics and Informatics*, *6*(1), 41–48. (Cited on page 5).

Minsky, M., & Papert, S. (1969). *Perceptrons*. MIT press. (Cited on page 23).

Miyajima, T., Baisho, F., Yamanaka, K., Nakamura, K., & Masahiro, A. (2000). A phasor model with resting states. *IEICE Transactions on Information and Systems*, *83*(2), 299–301. (Cited on page 25).

Miyauchi, M., & Seki, M. (1992). Interpretation of optical flow through neural network learning. In *Singapore ICCS/ISITA '92*. doi:10.1109/iccs.1992.255060. (Cited on page 5)

Miyauchi, M., Seki, M., Watanabe, A., & Miyauchi, A. (1993). Interpretation of optical flow through complex neural network. In *New trends in neural computation* (Pages 645–650). doi:10.1007/3-540-56798-4_215. (Cited on page 5)

Müezzinoğlu, M. K., Güzeliş, C., & Zurada, J. M. (2003). A new design method for the complex-valued multistate hopfield associative memory. *IEEE Transactions on Neural Networks*, *14*(4), 891–899. doi:10.1109/tnn.2003.813844. (Cited on page 22)

Narayanan, A., & Wang, D. (2013). Ideal ratio mask estimation using deep neural networks for robust speech recognition. In *2013 IEEE international conference on acoustics, speech and signal processing*. doi:10.1109/icassp.2013.6639038. (Cited on page 59)

Nemoto, I., & Kubono, M. (1996). Complex associative memory. *Neural Networks*, *9*(2), 253–261. doi:10.1016/0893-6080(95)00004-6. (Cited on page 22)

Nemoto, I., & Saito, K. (2002). A complex-valued version of Nagumo–Sato model of a single neuron and its behavior. *Neural Networks*, *15*(7), 833–853. doi:10.1016/s0893-6080(02)00066-7. (Cited on page 25)

Nitta, T. (1993a). A back-propagation algorithm for complex numbered neural networks. In *Proceedings of 1993 international conference on neural networks (IJCNN-93-nagoya, japan)*. doi:10.1109/ijcnn.1993.716968. (Cited on page 23)

Nitta, T. (1993b). A complex numbered version of the back-propagation algorithm. In *Proceedings of the world congress on neural networks* (Volume 3, Pages 576–579). (Cited on page 23).

Nitta, T., & Buchholz, S. (2008). On the decision boundaries of hyperbolic neurons. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. doi:10.1109/ijcnn.2008.4634216. (Cited on page 27)

Nitta, T., & Furuya, T. (1991). A complex back-propagation learning. *Transactions of Information Processing Society of Japan*, *32*(10), 1319–1329. In Japanese. (Cited on page 23).

Nix, J., & Hohmann, V. (2006). Sound source localization in real sound fields based on empirical statistics of interaural parameters. *The Journal of the Acoustical Society of America*, *119*(1), 463–479. doi:10.1121/1.2139619. (Cited on page 48)

Noest, A. J. (1987). Phasor neural networks. In *Neural information processing systems* (Pages 584–591). Retrieved from http://papers.nips.cc/paper/90-phasor-neural-networks. (Cited on page 22)

Noest, A. J. (1988a). Associative memory in sparse phasor neural networks. *Europhysics Letters (EPL)*, *6*(5), 469–474. doi:10.1209/0295-5075/6/5/016. (Cited on pages 22, 25)

Noest, A. J. (1988b). Discrete-state phasor neural networks. *Physical Review A*, *38*(4), 2196–2199. doi:10.1103/physreva.38.2196. (Cited on page 22)

Al-Nuaimi, A. Y. H., Amin, M. F., & Murase, K. (2012). Enhancing MP3 encoding by utilizing a predictive complex-valued neural network. In *The 2012 international joint conference on neural networks (IJCNN)*. doi:10.1109/ijcnn.2012.6252535. (Cited on page 6)

Oppenheim, A. V., & Lim, J. S. (1981). The importance of phase in signals. *Proceedings of the IEEE*, *69*(5), 529–541. doi:10.1109/proc.1981.12022. (Cited on page 14)

Oramas, S., Nieto, O., Barbieri, F., & Serra, X. (2017, July 16). Multi-label music genre classification from audio, text, and images using deep features. arXiv: 1707.04916v1 [cs.IR]. (Cited on page 5)

Pascual, S., Bonafonte, A., & Serrà, J. (2017). SEGAN: Speech enhancement generative adversarial network. In *Interspeech 2017*. doi:10.21437/interspeech.2017-1428. (Cited on page 59)

Pearson, J. K. (1995). *Clifford networks*. University of Kent at Canterbury. (Cited on page 27).

Peterson, G. E., & Lehiste, I. (1960). Duration of syllable nuclei in English. *The Journal of the Acoustical Society of America*, *32*(6), 693–703. doi:10.1121/1.1908183. (Cited on page 4)

Picinbono, B. C. (1997). On instantaneous amplitude and phase of signals. *IEEE Trans. Signal Processing*, *45*(3), 552–560. doi:10.1109/78.558469. (Cited on page 18)

Rahman, A. F. R., Howells, W. G. J., & Fairhurst, M. C. (2001). A multiexpert framework for character recognition: A novel application of Clifford networks. *IEEE Transactions on Neural Networks*, *12*(1), 101–112. doi:10.1109/72.896799. (Cited on page 5)

Raspaud, M., Viste, H., & Evangelista, G. (2010). Binaural source localization by joint estimation of ILD and ITD. *IEEE Transactions on Audio, Speech, and Language Processing*, *18*(1), 68–77. doi:10.1109/tasl.2009.2023644. (Cited on page 48)

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386–408. doi:10.1037/h0042519. (Cited on page 21)

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. doi:10.1038/323533a0. (Cited on page 23)

Sainath, T. N., Weiss, R. J., Senior, A. W., Wilson, K. W., & Vinyals, O. (2015). Learning the speech front-end with raw waveform CLDNNs. In *INTERSPEECH 2015, 16th annual conference of the international speech communication association* (Pages 1–5). Retrieved from http://www.isca-speech.org/archive/interspeech_2015/i15_0001.html. (Cited on page 13)

Savitha, R., Suresh, S., & Sundararajan, N. (2013). Projection-based fast learning fully complex-valued relaxation neural network. *IEEE Transactions on Neural Networks and Learning Systems*, *24*(4), 529–541. doi:10.1109/tnnls.2012.2235460. (Cited on page 28)

Scardapane, S., Vaerenbergh, S. V., Hussain, A., & Uncini, A. (2018, February 22). Complex-valued neural networks with non-parametric activation functions. arXiv: 1802.08026v1 [cs.NE]. (Cited on page 27)

Scarpiniti, M., Vigliano, D., Parisi, R., & Uncini, A. (2008). Generalized splitting functions for blind separation of complex signals. *Neurocomputing*, *71*(10-12), 2245–2270. doi:10.1016/j.neucom.2007.07.037. (Cited on page 5)

Schmidt, R. (1986). Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, *34*(3), 276–280. doi:10.1109/tap.1986.1143830. (Cited on page 46)

Schreier, P. J., & Scharf, L. L. (2010). *Statistical signal processing of complex-valued data: The theory of improper and noncircular signals*. Cambridge University Press. (Cited on page 23).

Shan, C., Zhang, J., Wang, Y., & Xie, L. (2017, July 22). Attention-based end-to-end speech recognition on voice search. arXiv: 1707.07167v3 [cs.CL]. (Cited on page 1)

Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., ... Wu, Y. (2017, December 16). Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions. arXiv: 1712.05884v2 [cs.CL]. (Cited on page 1)

Smaragdis, P., Raj, B., & Shashanka, M. (2007). Supervised and semi-supervised separation of sounds from single-channel mixtures. In *Independent component analysis and signal separation* (Pages 414–421). doi:10.1007/978-3-540-74494-8_52. (Cited on page 61)

Solazzi, M., Uncini, A., Claudio, E. D. D., & Parisi, R. (2001). Complex discriminative learning Bayesian neural equalizer. *Signal Processing*, *81*(12), 2493–2502. doi:10.1016/s0165-1684(01)00129-3. (Cited on page 5)

Soltau, H., Liao, H., & Sak, H. (2016, October 31). Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition. arXiv: 1610.09975v1 [cs.CL]. (Cited on page 1)

Søndergaard, P., & Majdak, P. (2013). The auditory modeling toolbox. In J. Blauert (Editor), *The technology of binaural listening* (Pages 33–56). Berlin, Heidelberg: Springer. (Cited on page 56).

Sonoda, S., & Murata, N. (2017). Neural network with unbounded activation functions is universal approximator. *Applied and Computational Harmonic Analysis*, *43*(2), 233–268. doi:10.1016/j.acha.2015.12.005. (Cited on page 5)

Suksmono, A. B., & Hirose, A. (2002). Adaptive noise reduction of InSAR images based on a complex-valued MRF model and its application to phase unwrapping problem. *IEEE Transactions on Geoscience and Remote Sensing*, *40*(3), 699–709. doi:10.1109/tgrs.2002.1000329. (Cited on page 5)

Suksmono, A. B., & Hirose, A. (2003). Adaptive beamforming by using complex-valued multi layer perceptron. In *Artificial neural networks and neural information processing — ICANN/ICONIP 2003* (Pages 959–966). doi:10.1007/3-540-44989-2_114. (Cited on page 5)

Sutherland, J. G. (1990). A holographic model of memory, learning and expression. *International Journal of Neural Systems*, *01*(03), 259–267. doi:10.1142/s0129065790000163. (Cited on page 23)

Takeda, M., & Kishigami, T. (1992). Complex neural fields with a Hopfield-like energy function and an analogy to optical fields generated in phase-conjugate resonators. *Journal of the Optical Society of America A*, *9*(12), 2182. doi:10.1364/josaa.9.002182. (Cited on page 25)

Takeda, M., & Kishigami, T. (1993). Dynamics of complex neural fields with an analogy to optical fields generated in a phase-conjugate resonator. In R. Roy (Editor), *Chaos in optics*. doi:10.1117/12.164775. (Cited on page 25)

Talagala, D. S., Wu, X., Zhang, W., & Abhayapala, T. D. (2014). Binaural localization of speech sources in the median plane using cepstral HRTF extraction. In *Proceedings of the 22nd european signal processing conference (EUSIPCO)* (Pages 2055–2059). (Cited on page 48).

Talagala, D. S., Zhang, W., Abhayapala, T. D., & Kamineni, A. (2014). Binaural sound source localization using the frequency diversity of the head-related transfer function. *The Journal of the Acoustical Society of America*, *135*(3), 1207–1217. doi:10.1121/1.4864304. (Cited on page 48)

Tanaka, G., & Aihara, K. (2009). Complex-valued multistate associative memory with nonlinear multilevel functions for gray-level image reconstruction. *IEEE Transactions on Neural Networks*, *20*(9), 1463–1473. doi:10.1109/tnn.2009.2025500. (Cited on page 5)

Tay, C. S., Tanizawa, K., & Hirose, A. (2007). Error reduction in holographic movies using a hybrid learning method in coherent neural networks. In *Lecture notes in computer science* (Pages 884–893). doi:10.1007/978-3-540-74690-4_90. (Cited on page 5)

Tieleman, T., & Hinton, G. (2012). Lecture 6.5-RMSProp: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, *4*(2), 26–31. Retrieved from https://www.coursera.org/learn/neural-networks/lecture/YQHki/rmsprop-divide-the-gradient-by-a-running-average-of-its-recent-magnitude. (Cited on page 35)

Trabelsi, C., Bilaniuk, O., Zhang, Y., Serdyuk, D., Subramanian, S., Santos, J. F., ... Pal, C. J. (2017, May 27). Deep complex networks. arXiv: 1705.09792v4 [cs.NE]. (Cited on pages 6, 23, 27, 50, 62, 68, 77)

Tsuzuki, H., Kugler, M., Kuroyanagi, S., & Iwata, A. (2013). An approach for sound source localization by complex-valued neural network. *IEICE Transactions on Information and Systems*, *E96.D*(10), 2257–2265. doi:10.1587/transinf.e96.d.2257. (Cited on pages 6, 46, 53, 54, 56)

Uncini, A., Vecci, L., Campolucci, P., & Piazza, F. (1999). Complex-valued neural networks with adaptive spline activation function for digital-radio-links nonlinear equalization. *IEEE Transactions on Signal Processing*, *47*(2), 505–514. doi:10.1109/78.740133. (Cited on page 27)

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., … Kavukcuoglu, K. (2016, September 12). WaveNet: A generative model for raw audio. arXiv: 1609.03499v2 [cs.SD]. (Cited on pages 4, 60)

van den Oord, A., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., … Hassabis, D. (2017, November 28). Parallel WaveNet: Fast high-fidelity speech synthesis. arXiv: 1711.10433v1 [cs.LG]. (Cited on page 4)

van den Bos, A. (1994). Complex gradient and hessian. *IEE Proceedings - Vision, Image, and Signal Processing*, *141*(6), 380. doi:10.1049/ip-vis:19941555. (Cited on pages 23, 29)

Vincent, E., Barker, J., Watanabe, S., Le Roux, J., Nesta, F., & Matassoni, M. (2013). The second 'CHiME' speech separation and recognition challenge: Datasets, tasks and baselines. In *2013 IEEE international conference on acoustics, speech and signal processing*. doi:10.1109/icassp.2013.6637622. (Cited on page 67)

Vitagliano, F., Parisi, R., & Uncini, A. (2003). Generalized splitting 2D flexible activation function. In *Neural nets* (Pages 85–95). doi:10.1007/978-3-540-45216-4_9. (Cited on page 27)

von Helmholtz, H. (1863). *Die lehre von den tonempfindungen als physiologische grundlage für die theorie der musik*. Braunschweig: Friedrich Vieweg. (Cited on page 16).

Vu, T. T., Bigot, B., & Chng, E. S. (2016). Combining non-negative matrix factorization and deep neural networks for speech enhancement and automatic speech recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. doi:10.1109/icassp.2016.7471725. (Cited on pages 59, 61)

Wang, Z., & Sha, F. (2014). Discriminative non-negative matrix factorization for single-channel speech separation. In *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. doi:10.1109/icassp.2014.6854302. (Cited on page 61)

Weninger, F., Erdogan, H., Watanabe, S., Vincent, E., Le Roux, J., Hershey, J. R., & Schuller, B. (2015). Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR. In *Latent variable analysis and signal separation* (Pages 91–99). doi:10.1007/978-3-319-22482-4_11. (Cited on pages 60, 61)

Weninger, F., Le Roux, J., Hershey, J. R., & Watanabe, S. (2014). Discriminative NMF and its application to single-channel source separation. In H. Li, H. M. Meng, B. Ma, E. Chng, & L. Xie (Editors), *INTERSPEECH 2014, 15th annual conference of the international speech communication association* (Pages 865–869). ISCA. Retrieved from http://www.isca-speech.org/archive/interspeech_2014/i14_0865.html. (Cited on page 61)

Wenzel, E. M., Arruda, M., Kistler, D. J., & Wightman, F. L. (1993). Localization using nonindividualized head-related transfer functions. *The Journal of the Acoustical Society of America*, *94*(1), 111–123. doi:10.1121/1.407089. (Cited on pages 3, 45)

Widrow, B., & Hoff, M. E. (1960). *Adapting switching circuits* (technical report Number 4). IRE WESCON Convention Record. (Cited on page 21).

Widrow, B., McCool, J., & Ball, M. (1975). The complex LMS algorithm. *Proceedings of the IEEE*, *63*(4), 719–720. doi:10.1109/proc.1975.9807. (Cited on page 22)

Wightman, F. L., & Kistler, D. J. (1989a). Headphone simulation of free-field listening. I: Stimulus synthesis. *The Journal of the Acoustical Society of America*, *85*(2), 858–867. doi:10.1121/1.397557. (Cited on page 3)

Wightman, F. L., & Kistler, D. J. (1989b). Headphone simulation of free-field listening. II: Psychophysical validation. *The Journal of the Acoustical Society of America*, *85*(2), 868–878. doi:10.1121/1.397558. (Cited on page 3)

Wightman, F. L., & Kistler, D. J. (1992). The dominant role of low-frequency interaural time differences in sound localization. *The Journal of the Acoustical Society of America*, *91*(3), 1648–1661. doi:10.1121/1.402445. (Cited on page 3)

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement learning* (Pages 5–32). doi:10.1007/978-1-4615-3618-5_2. (Cited on page 61)

Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, *1*(2), 270–280. doi:10.1162/neco.1989.1.2.270. (Cited on page 23)

Williamson, D. S., Wang, Y., & Wang, D. (2016). Complex ratio masking for monaural speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *24*(3), 483–492. doi:10.1109/taslp.2015.2512042. (Cited on pages 60, 62)

Wirtinger, W. (1927). Zur formalen Theorie der Funktionen von mehr komplexen Veränderlichen. *Mathematische Annalen*, *97*(1), 357–375. doi:10.1007/bf01447872. (Cited on pages 22, 29, 51)

Wisdom, S., Powers, T., Hershey, J., Le Roux, J., & Atlas, L. (2016). Full-capacity unitary recurrent neural networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, & R. Garnett (Editors), *Advances in neural information processing systems 29* (Pages 4880–4888). Curran Associates, Inc. (Cited on page 5).

Woodruff, J., & Wang, D. (2012). Binaural localization of multiple sources in reverberant and noisy environments. *IEEE Transactions on Audio, Speech, and Language Processing*, *20*(5), 1503–1512. doi:10.1109/tasl.2012.2183869. (Cited on page 48)

Wu, X., Talagala, D. S., Zhang, W., & Abhayapala, T. D. (2015). Binaural localization of speech sources in 3-D using a composite feature vector of the HRTF. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. doi:10.1109/icassp.2015.7178452. (Cited on page 48)

Wu, X., Talagala, D. S., Zhang, W., & Abhayapala, T. D. (2016). Spatial feature learning for robust binaural sound source localization using a composite feature vector. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. doi:10.1109/icassp.2016.7472893. (Cited on page 48)

Xu, Y., Du, J., Dai, L.-R., & Lee, C.-H. (2015). A regression approach to speech enhancement based on deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *23*(1), 7–19. doi:10.1109/taslp.2014.2364452. (Cited on page 59)

Xu, Y., Du, J., Huang, Z., Dai, L., & Lee, C. (2015). Multi-objective learning and mask-based post-processing for deep neural network based speech enhancement. In *INTERSPEECH 2015, 16th annual conference of the international speech communication association* (Pages 1508–1512). ISCA. Retrieved from http://www.isca-speech.org/archive/interspeech_2015/i15_1508.html. (Cited on page 59)

Zeiler, M. D. (2012, December 22). ADADELTA: An adaptive learning rate method. arXiv: 1212.5701v1 [cs.LG]. (Cited on page 35)

Zemel, R. S., Williams, C. K. I., & Mozer, M. (1995). Lending direction to neural networks. *Neural Networks*, *8*(4), 503–512. doi:10.1016/0893-6080(94)00094-3. (Cited on page 25)

Zhang, H., & Mandic, D. P. (2016). Is a complex-valued stepsize advantageous in complex-valued gradient learning algorithms? *IEEE Transactions on Neural Networks and Learning Systems*, *27*(12), 2730–2735. doi:10.1109/tnnls.2015.2494361. (Cited on page 23)

Zwicker, E., & Fastl, H. (2010). *Psychoacoustics: Facts and models*. Berlin; Heidelberg: Springer. (Cited on page 1).