

LEARNING REPRESENTATIONS USING COMPLEX-VALUED NETS

Andy M. Sarroff¹, Victor Shepardson² & Michael A. Casey^{1,2}

Departments of Computer Science¹ and Music²

Dartmouth College

Hanover, NH 03755, USA

sarroff@cs.dartmouth.edu

ABSTRACT

Complex-valued neural networks (CVNNs) are an emerging field of research in neural networks due to their potential representational properties for audio, image, and physiological signals. It is common in signal processing to transform sequences of real values to the complex domain via a set of complex basis functions, such as the Fourier transform. We show how CVNNs can be used to learn complex representations of real valued time-series data. We present methods and results using a framework that can compose holomorphic and non-holomorphic functions in a multi-layer network using a theoretical result called the Wirtinger derivative. We test our methods on a representation learning task for real-valued signals, recurrent complex-valued networks and their real-valued counterparts. Our results show that recurrent complex-valued networks can perform as well as their real-valued counterparts while learning filters that are representative of the domain of the data.

1 INTRODUCTION

There are many types of data for which complex-valued representations are natural and appropriate. For example wind measurements may use complex-valued data to represent joint measurements of magnitude and direction (Goh et al., 2006). Direction of arrival is naturally modeled in ultra-wideband communications using complex values (Terabayashi et al., 2014). It is also common to work with certain real-valued data, such as audio and EEG recordings, by first transforming them to complex-valued data in the frequency domain via a complex basis, as with the Fourier transform. Motivations behind complex-valued nets (CVNNs) are that they could be used with such real-to-complex transformed data, or that they may be used for learning complex-valued representations as alternatives to Fourier and related transforms.

Whilst research into CVNNs has developed in parallel with real-valued networks, there has been relatively little focus on CVNNs in deep learning and complex-valued representation learning. Most research targets highly-specific signal-processing domains such as communications and adaptive array processing. Several factors contribute to the slow adoption of CVNNs in applications outside of these domains: first, they are difficult to train because complex-valued activation functions cannot be simultaneously bounded and complex-differentiable; second, there are few if any known methods for regularization and hyper-parameter optimization specifically developed for CVNNs. Despite such obstacles, research on CVNNs is growing steadily, with new theoretical results (Zimmermann et al., 2011; Sorber et al., 2012; Hirose & Yoshida, 2012) appearing on the heels of comprehensive treatments in recent texts (Hirose, 2006; Mandic & Goh, 2009; Hirose, 2013).

Research on complex-valued activation functions and calculation of their derivatives for application to CVNNs is generally split between those composed exclusively from holomorphic activation functions and those composed exclusively from non-holomorphic activation functions. Holomorphic functions are complex differentiable at every point in a neighborhood of their domain. Non-holomorphic functions are not complex differentiable, but may be differentiable with respect to their real and imaginary parts. Each has advantages: Holomorphic activation functions may be more successful at jointly modeling phase and amplitude, however they are unbounded and therefore non-

holomorphic activation functions may be preferred at times. For example if one were to build a complex-valued Long-Short Term Memory network, the only suitable gating function in the complex domain would necessarily be non-holomorphic.

In this paper, we follow a more general framework (Amin et al., 2011; Amin & Murase, 2013) for building CVNNs, both deep and temporal, that allows for activation functions that are composed from combinations of both holomorphic and non-holomorphic functions. We do this by utilizing the mathematical conveniences of the Wirtinger derivative, which simplifies many of the computations that are required for gradient descent for complex-valued functions.

The remainder of this paper is organized as follows. In Section 2 we cover the background for CVNNs. Section 3 describes the Wirtinger derivative and how it is applied to back-propagation for gradient descent with complex-valued activation functions. We present experiments that illustrate the utility of the methods in Section 4 and we provide concluding remarks in Section 5.

2 BACKGROUND

Complex numbers extend the concept of one-dimensional real numbers to two dimensions by expressing an ordered pair $(x, y) \in \mathbb{R}$ as a point $z \in \mathbb{C}$ in the complex plane, where $z = x + iy$ and $i = \sqrt{-1}$. Numbers in the complex domain provide a natural means for jointly expressing magnitude, $|z|$, and phase or direction, $\arg(z)$.

Suppose we wish to learn a function $f : \mathbb{C}^m \rightarrow \mathbb{C}^n$ by optimizing the squared error

$$\mathcal{L}(z) = |z|^2 = z\bar{z} \quad , \quad (1)$$

where $(\bar{\cdot})$ denotes the complex conjugate operator. Note that $z\bar{z} = (x + iy)(x - iy) = x^2 + y^2 \in \mathbb{R}$ and therefore the objective function is real-valued even though z is complex-valued.

Real-valued functions of complex variables are non-holomorphic and therefore their complex derivative is undefined. However if we denote $\mathcal{L}(z) = u(x, y) + iv(x, y)$ with $u : \mathbb{R} \rightarrow \mathbb{R}$ and $v : \mathbb{R} \rightarrow \mathbb{R}$ and u and v are real-analytic (u and v are differentiable) functions then it is possible to find a stationary point in the objective function. Stated more simply, we may perform gradient descent with a real-valued cost function of complex variables even though the function does not have a complex derivative.

In this paper we apply the Wirtinger derivative (Wirtinger, 1927) to compute the gradient (Brandwood, 1983). Doing so allows us to perform differentiation on functions that are not complex-analytic but are real-analytic. It also provides a means for easily composing a combination of holomorphic and non-holomorphic functions within the computational graph of a neural network. Finally, by taking advantage of basic properties of the Wirtinger derivative, we perform gradient descent using two Jacobians per computational node.

Due to space limitations the following summary is necessarily brief. A great overview of the core mechanics of complex-valued nets and the Wirtinger derivative is found in Mandic & Goh (2009). This and other literature are built on the theory developed in Brandwood (1983) and van den Bos (1994) for optimization of complex-valued nets using respectively first- and second-order derivatives with Wirtinger calculus. For a deeper discussion of Wirtinger calculus and optimization techniques we refer the reader to Kreutz-Delgado (2009); Li & Adali (2008). Finally Amin et al. (2011); Amin & Murase (2013) advocate a framework for composing holomorphic and non-holomorphic functions in complex-valued nets.

2.1 COMPLEX-VALUED AND REAL-VALUED NETS

The components of a complex-valued number can be represented as a bivariate real number, so it is natural to ask why a complex-valued representation may be preferred. A multiplication of values in the real domain yields scaling. A multiplication of complex values yields scaling and rotation. Hence if we wish to model magnitude and phase jointly, it may be more natural to do so by using a complex representation.

There are cases when we may wish to model real-valued processes in the complex domain. For instance one cannot determine the instantaneous frequency or amplitude of a real-valued periodic

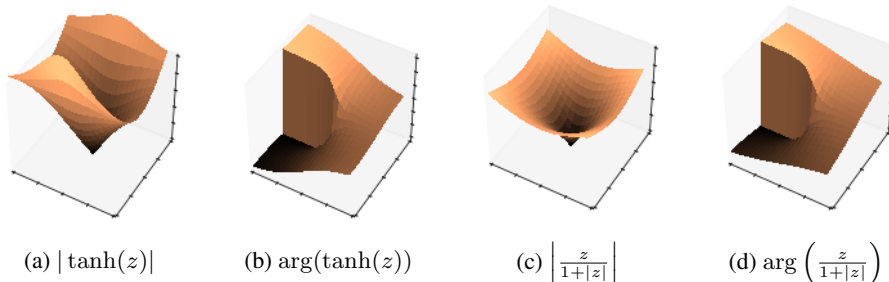


Figure 1: Fully complex elementary transcendental function (a,b) (Kim & Adali, 2003) and split phase-magnitude (c,d) (Georgiou & Koutsougeras, 1992) activation functions. Axes: $\Re(z)$, $\Im(z)$, $f(z)$.

waveform from a single sample. Applying the Hilbert transform yields a complex-valued waveform with the same positive frequency components. However we suggest that complex-valued networks may also learn important relationships on instantaneous frequency and amplitude.

2.2 ACTIVATION FUNCTIONS

Activation functions that are bounded and differentiable are generally desirable for training neural networks. (The rectified linear unit is a notable exception for boundedness.) Due to Liouville’s theorem, the only entire (holomorphic over the entire complex domain) function that is bounded is a constant. Thus we must choose between boundedness and differentiability for complex nets.

Split-complex activation functions operate on the real and imaginary or phase and magnitude components independently and merge the outputs together. Such functions are not holomorphic. However it is easy to define a bounded split-complex activation function, for example Georgiou and Koutsougeras’ magnitude squashing activation function (Georgiou & Koutsougeras, 1992). It is suggested by Mandic & Goh (2009) that the split phase-magnitude and real-imaginary approaches are appropriate when we can assume rotational or cartesian symmetry of the data, respectively.

Alternatively we may choose to use fully complex activation functions that are bounded almost everywhere. Certain elementary transcendental functions have been identified which provide squashing-type nonlinear discrimination with well defined first-order derivatives (Kim & Adali, 2003). These functions have singularities, but with proper treatment of weights or using other regularization mechanisms singularities may be avoided. Figure 1 shows magnitude and phase surface plots for a complex tanh activation function and Georgiou and Koutsougeras’ activation function. The tanh activation has regularly spaced singularities along the imaginary axis beyond the limits of the plots.

3 WIRTINGER FRAMEWORK FOR GRADIENT DESCENT

This section outlines the routine for optimizing an arbitrary complex-valued neural network using the Wirtinger derivative and gradient descent. The network has a real-valued objective function of complex variables. It may have any combination of holomorphic and non-holomorphic activation functions. Wirtinger calculus (also known as $\mathbb{C}\mathbb{R}$ Calculus in some texts) facilitates defining a computational graph that can be modularized as in many popular deep learning libraries, thereby allowing the construction of deep or temporal networks having many layers.

In the following subsection, the core concepts of Wirtinger calculus are reviewed. The following subsection describes the framework for building a computational graph and performing gradient descent.

3.1 WIRTINGER DERIVATIVES

Define $z \in \mathbb{C}$ and $x, y \in \mathbb{R}$ with $f(z) = g(x, y) = u(x, y) + iv(x, y)$. We extend the definition of f to include the complex conjugate of its input variable so that

$$\begin{aligned} f(z) &= f(z, \bar{z}) = g(x, y) = u(x, y) + iv(x, y) \\ z &= x + iy \\ \bar{z} &= x - iy \end{aligned} \quad (2)$$

Using this definition, the \mathbb{R} -derivative and $\overline{\mathbb{R}}$ -derivative of f are defined as:

$$\left. \frac{\partial f}{\partial z} \right|_{\bar{z} \text{ is constant}} \quad \text{and} \quad \left. \frac{\partial f}{\partial \bar{z}} \right|_{z \text{ is constant}} \quad (3)$$

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

$$\frac{\partial f}{\partial z} = \frac{1}{2} \left(\frac{\partial f}{\partial x} - i \frac{\partial f}{\partial y} \right) \quad \text{and} \quad \frac{\partial f}{\partial \bar{z}} = \frac{1}{2} \left(\frac{\partial f}{\partial x} + i \frac{\partial f}{\partial y} \right) \quad (4)$$

We note 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 $f(z) = g(x, y) = u(x, y) + iv(x, y)$ to exist, the following identities must hold:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \quad \text{and} \quad \frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y} \quad (5)$$

If we expand the right hand side of the $\overline{\mathbb{R}}$ -derivative and substitute the Cauchy-Riemann equations 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. As an example, consider the loss function in Eq. (1). It is real-valued and therefore non-holomorphic and it clearly depends on the conjugate of its input variable, having \mathbb{R} - and $\overline{\mathbb{R}}$ -derivatives of \bar{z} and z , respectively.

It is further shown by Brandwood that if $f : \mathbb{C} \rightarrow \mathbb{R}$ is a real-valued function, either $\frac{\partial f}{\partial z} = 0$ or $\frac{\partial f}{\partial \bar{z}} = 0$ is a necessary and sufficient condition for f to have a stationary point. By extension if $f : \mathbb{C}^N \rightarrow \mathbb{R}$ is a real-valued function of a complex vector $\mathbf{z} = (z_1, z_2, \dots, z_N)^T$ and we define the cogradient and conjugate cogradient

$$\nabla_{\mathbf{z}} = (\partial/\partial z_1, \partial/\partial z_2, \dots, \partial/\partial z_N)^T \quad (6)$$

$$\nabla_{\bar{\mathbf{z}}} = (\partial/\partial \bar{z}_1, \partial/\partial \bar{z}_2, \dots, \partial/\partial \bar{z}_N)^T \quad (7)$$

then $\nabla_{\mathbf{z}} f = 0$ or $\nabla_{\bar{\mathbf{z}}} f = 0$ are necessary and sufficient to determine a stationary point. Finally, Brandwood uses Schwarz's inequality to show that the maximum rate of change of f is in the direction of the conjugate cogradient $\nabla_{\bar{\mathbf{z}}} f$. Using these definitions, we can perform gradient descent with the conjugate cogradient operator.

3.2 THE COMPUTATIONAL GRAPH

We wish to perform gradient descent on a computational graph having a real-valued cost function and an arbitrary composition of holomorphic and non-holomorphic functions. Performing back-propagation on such a graph can be unwieldy if we choose to repeatedly switch between complex and real-valued representations of the graph. If we remain in the complex domain for all computations and use Wirtinger calculus, it is easier to build a modular framework that is useful for deep networks.

Consider a complex-valued function,

$$\mathbf{F}(\mathbf{z}, \bar{\mathbf{z}}) = [f_1(\mathbf{z}, \bar{\mathbf{z}}), f_2(\mathbf{z}, \bar{\mathbf{z}}), \dots, f_M(\mathbf{z}, \bar{\mathbf{z}})]^T \quad \text{with} \quad (8)$$

$$\mathbf{z} = [z_1, z_2, \dots, z_N]^T \quad \text{and} \quad (9)$$

$$\bar{\mathbf{z}} = [\bar{z}_1, \bar{z}_2, \dots, \bar{z}_N]^T \quad . \quad (10)$$

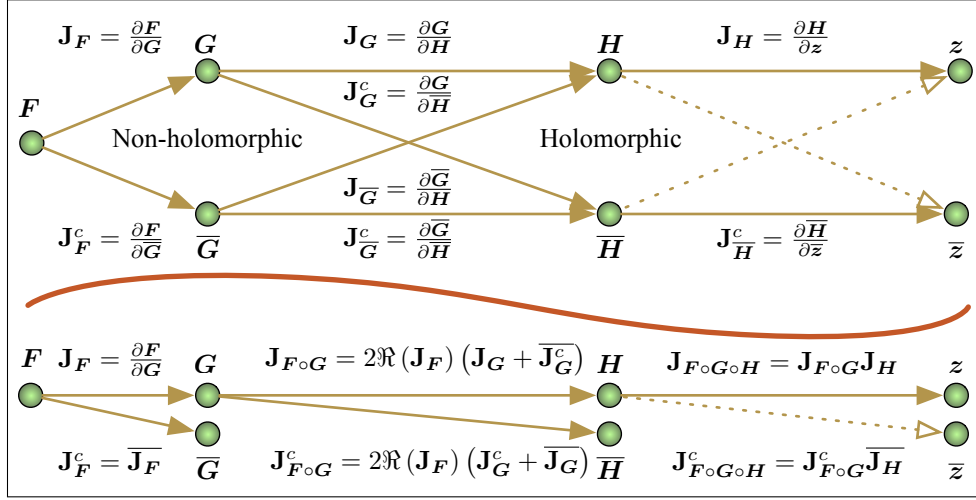


Figure 2: Computational dependency graph for a composition of functions $(F \circ G \circ H)(z, \bar{z})$ with F real and H holomorphic. Top: naive computation requires keeping track of up to four dependencies per function. Bottom: using properties of the complex derivative we need only keep track of two dependencies per function.

We define the Jacobian matrices,

$$\mathbf{J}_F \triangleq \frac{\partial F(z, \bar{z})}{\partial z} \quad (11)$$

$$\mathbf{J}_F^c \triangleq \frac{\partial F(z, \bar{z})}{\partial \bar{z}} \quad (12)$$

A deep neural network is constructed from a composition of several non-scalar functions. Suppose we have a composition of functions $(F \circ G \circ H)(z, \bar{z})$, with F being a real-valued (non-holomorphic cost function), G being a non-holomorphic complex-valued function, and H being a holomorphic function. We would like to compute the gradient of F with respect to \bar{z} . Figure 2 shows the dependency graph for back-propagating the gradient. The top part of the figure shows the Jacobian matrices for each stage of back-propagation. The naive method requires keeping track of four dependencies for every function in the graph.

We need keep track of only two partial derivatives for each function, as shown in the bottom part of Figure 2 (Amin et al., 2011; Li & Adali, 2008). Keeping in mind that F is a real-valued function of complex variables and that H is holomorphic s.t. $\frac{\partial H}{\partial \bar{z}} = 0$, we apply the chain rule to the Jacobian matrices:

$$\begin{aligned} \mathbf{J}_{F \circ G}^c &= \mathbf{J}_F \mathbf{J}_G^c + \mathbf{J}_F^c \bar{\mathbf{J}}_G \\ &= 2\Re(\mathbf{J}_F)(\mathbf{J}_G^c + \bar{\mathbf{J}}_G) \end{aligned} \quad (13)$$

$$\mathbf{J}_{F \circ G \circ H}^c = \mathbf{J}_{F \circ G}^c \bar{\mathbf{J}}_H \quad (14)$$

More generally, given arbitrary functions F and G in the computational graph, we compose their Jacobians in the following way (Kreutz-Delgado, 2009):

$$\begin{aligned} \mathbf{J}_{F \circ G} &= \mathbf{J}_F \mathbf{J}_G + \mathbf{J}_F^c \bar{\mathbf{J}}_G^c \\ \mathbf{J}_{F \circ G}^c &= \mathbf{J}_F \mathbf{J}_G^c + \mathbf{J}_F^c \bar{\mathbf{J}}_G \end{aligned}$$

4 EXPERIMENTS

The Discrete Fourier Transform of N regularly-sampled points on a waveform yields complex coefficients of N orthogonal complex sinusoids. However the Fourier representation may not be the best

transform for a given learning task. Deep networks are regularly trained to learn data representations that are more suitable than hand-picked features. In this experiment, we generate real and complex-valued waveforms having wide-band spectral components at multiple phases and magnitudes. We train recurrent complex- and real-valued models to predict the N -th frame of a waveform given the first $N - 1$ frames. In the following subsections we detail the data, models, and results.

4.1 DATA

We generated four synthetic datasets having wide-band frequency spectra with random phases: Sawtooth-Like, Sawtooth-Like (Analytic), Inharmonic, and Inharmonic (Analytic). Each dataset had unique training, validation, and testing partitions. The training sets consisted of 10,000 observations split into 10 batches. The validation and test sets each had 1 batch of 1,000 observations.

Datasets were generated as described below. Each observation (waveform) has 1024 samples with a Nyquist frequency denoted Ω . The waveform was split into four non-overlapping rectangular-windowed frames of 256 samples. The first three frames were used as input to the model and the remaining frame is reserved as ground truth for inference.

4.1.1 SAWTOOTH-LIKE

Each waveform has a fundamental frequency drawn uniformly from the range $[0, \Omega)$. There are harmonics $n = (2, \dots, N)$ above the fundamental frequency, with all harmonic frequencies being less than Ω and each harmonic having an amplitude of $1/n$. All sinusoidal components have a random phase drawn from a uniform distribution $[0, 1)$. Each real-valued waveform is made complex by adding a zero-valued imaginary component.

We refer to these waveforms as ‘‘Sawtooth-Like’’ because they have the same spectral components of a band-limited sawtooth waveform. However since the phases of the spectral components are scrambled, the time-amplitude waveforms do not look like sawtooth waveforms. Each frame of an observation has a number of waveform periods in the range of $[0, 128]$. The expected number of periods per frame is 64.

4.1.2 SAWTOOTH-LIKE (ANALYTIC)

Waveforms were generated as above, but with the following modification. For each frequency component with frequency ω and phase ϕ , a sinusoidal component is added to the imaginary axes having frequency ω and phase $\phi - \pi/2$. An analytic signal encodes instantaneous magnitude and phase. In cases where a real-valued network was trained with this dataset, the real and imaginary parts of the data were split and hence there were twice the number of inputs and outputs as other experiments.

4.2 INHARMONIC

Inharmonic waveforms were generated with five spectral components, each having a frequency drawn from a uniform distribution in the range $[0, \Omega)$, a phase drawn from a uniform distribution in the range $[0, 1)$, and an amplitude of $1/5$. Hence the phases of the individual components are random but not drawn from the full available range of $[0, 2\pi)$. These waveforms are unlikely to exhibit periodicity.

4.3 INHARMONIC (ANALYTIC)

Analytic waveforms were generated as above using the same methodology as for the Sawtooth-Like (Analytic) dataset.

4.4 MODELS

We trained real- and complex-valued neural networks having one hidden recurrent layer of size 256. The input and output layers had 256 units each, with the exception of real-valued networks trained on the analytic datasets; these had 512 inputs and outputs. All models had weights and biases. Hence models had either 197,376 or 328,704 trainable parameters. Models were trained with a tanh activation function on the hidden layer and linear activation on the output layer.

Table 1: Test Error

Dataset	Complex	Real
Sawtooth-Like	0.1179	0.1060
Sawtooth-Like (Analytic)	0.3497	0.1937
Inharmonic	0.1664	0.1376
Inharmonic (Analytic)	0.2011	0.1999

4.5 TRAINING

Training was performed for exactly 1000 epochs using mini-batch stochastic gradient descent with a momentum of 0.9 and the mean squared cost function. We employed a learning rate with power scheduling decay (Senior et al., 2013).

We and other authors have found that complex-valued networks are extremely sensitive to initial conditions and learning rates (Zimmermann et al., 2011). In order to facilitate finding a good setting of hyperparameters, we performed hyperparameter optimization using Spearmint (Snoek et al., 2012) for the following parameters: initial weight scaling, learning rate, and learning rate decay half life. For each dataset, 100 real- and complex-valued models were trained with unique hyperparameter settings and initial weights. The final model was chosen using the best performance on the validation set.

4.6 RESULTS

4.6.1 OVERALL COMPARISON

Each dataset was trained, validated, and tested on a complex- and real-valued network. We had hoped that complex-valued networks would outperform real-valued nets. In most cases, the final error between complex and real nets was comparable. However in all experiments, the real-valued networks had a lower final test error. Table 1 shows that both real and complex valued networks perform best on the Sawtooth-Like dataset. We were not surprised by this result. Considering that this dataset consists of only harmonically related spectral components, we presume that this dataset is easier to learn than the Inharmonic datasets.

We were surprised that both real and complex-valued networks had difficulty learning the Analytic datasets. These datasets encode instantaneous frequency and phase, and we therefore expected that they would work well with the complex valued network. It is possible that the fully complex tanh activation function is inappropriate for this dataset since instantaneous frequency does not change between inputs and outputs. In future work we will examine the performance of other activation functions on this dataset.

4.6.2 OPTIMIZATION

The left pane of Figure 3 shows the sorted error across hyperparameter settings employed with the Sawtooth-Like dataset. We find it notable that most settings perform relatively poorly. There were only a few settings for both types of networks that achieved optimal performance. This figure underscores how sensitivities both types of networks are to hyperparameter settings.

The right pane shows the validation error across epochs for the Sawtooth-Like dataset. Note the discontinuity in the error curve for the complex-valued net. The complex valued nets are quite difficult to train and can easily approach regions of instability. We believe this is due to the singularities of the tanh function.

4.6.3 FILTERS

We examined the input-to-hidden weights of the models. We found that despite the worse performance of complex-valued networks, they learned filters that are easily relatable to the datasets. Figure 4 shows the magnitude frequency responses of the first three input-to-hidden weights for the Sawtooth-Like (Analytic) (left) and Inharmonic (right) datasets. Observe that the frequency re-

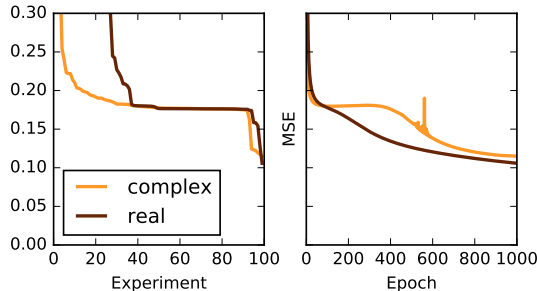


Figure 3: Left: Sorted validation error for hyperparameter optimization. Right: Validation error for best-performing hyperparameter setting. Both figures associated with Sawtooth-Like dataset.

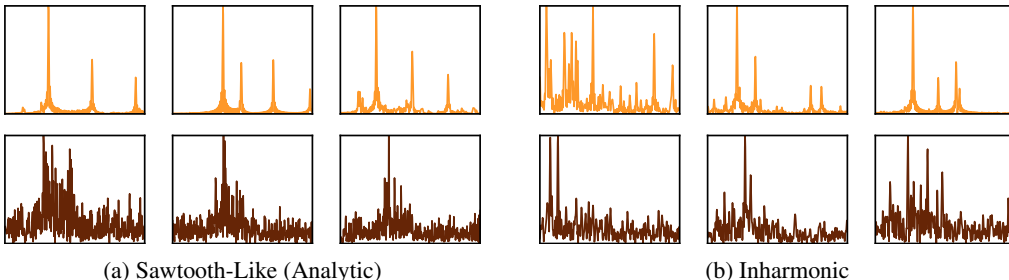


Figure 4: Magnitude frequency response of first three filters for complex (top) and real (bottom) valued nets. The x axis shows frequency index and the y axis shows magnitude.

sponse of the complex model for the Sawtooth-Like dataset exhibits harmonically spaced peaks in the spectrum. The filters from the real-valued network are much noisier and it is difficult to discern any harmonic spacing. The filters of the complex model trained on the Inharmonic dataset also show high selectivity for a few spectral peaks, whereas, the filters learned by the real-valued model show selectivity but to a more limited degree.

5 CONCLUSIONS

Despite potentially widespread applicability to machine learning tasks, the deep learning and representational learning communities have not fully embraced complex-valued nets. We argue that the mathematical conveniences of Wirtinger calculus offer a means for building a modular library for training complex-valued nets. Towards this end, we composed several synthetic datasets and compared the performance of complex- and real-valued nets. We found that complex-valued nets performed about as well as, but not better than, real-valued counterparts. We highlighted the fact that training complex-valued nets brings different challenges, including difficulties of boundedness and singularities in the activation functions. Finally we showed that despite poorer performance, complex-valued nets learn filter representations that are adapted to the domain of the data.

It is obvious that there are many challenges to successfully training complex-valued nets. We must find good methods for avoiding the singularities in holomorphic cost functions. There is no complex equivalent to the rectified linear unit. The models are extremely sensitive to initial conditions of the weights and to the learning rate. We will continue to explore these topics in future work. Our experiments were conducted on GPUs using a modified branch of the Chainer deep learning framework.¹. As we continue to investigate complex-valued networks, we intend to develop our framework further and release it to the community.

¹<http://docs.chainer.org/en/stable/index.html>

REFERENCES

- Amin, Md. Faijul and Murase, Kazuyuki. Learning algorithms in complex-valued neural networks using Wirtinger calculus. In *Complex-Valued Neural Networks: Advances and Applications*, pp. 75–102. John Wiley & Sons, Inc., 2013.
- Amin, Md.Faijul, Amin, MuhammadIlias, Al-Nuaimi, A.Y.H., and Murase, Kazuyuki. Wirtinger calculus based gradient descent and levenberg-marquardt learning algorithms in complex-valued neural networks. In Lu, Bao-Liang, Zhang, Liqing, and Kwok, James (eds.), *Neural Information Processing*, volume 7062 of *Lecture Notes in Computer Science*, pp. 550–559. Springer Berlin Heidelberg, 2011.
- Brandwood, D.H. A complex gradient operator and its application in adaptive array theory. *Communications, Radar and Signal Processing, IEE Proceedings F*, 130(1):11–16, February 1983.
- Georgiou, G.M. and Koutsougeras, C. Complex domain backpropagation. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 39(5):330–334, May 1992.
- Goh, S. L., Chen, M., Popović, D. H., Aihara, K., Obradovic, D., and Mandic, D. P. Complex-valued forecasting of wind profile. *Renewable Energy*, 31(11):1733–1750, 9 2006.
- Hirose, A. *Complex-Valued Neural Networks*. Studies in computational intelligence. Springer-Verlag, 2006.
- Hirose, A. *Complex-Valued Neural Networks: Advances and Applications*. John Wiley & Sons, Inc., 2013.
- Hirose, A. and Yoshida, S. Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence. *Neural Networks and Learning Systems, IEEE Transactions on*, 23(4):541–551, April 2012.
- Kim, T and Adali, T. Approximation by fully complex multilayer perceptrons. *Neural Computation*, 15(7):1641–1666, July 2003.
- Kreutz-Delgado, K. The Complex Gradient Operator and the CR-Calculus. *ArXiv e-prints*, June 2009.
- Li, Hualiang and Adali, Tülay. Complex-valued adaptive signal processing using nonlinear functions. *EURASIP J. Adv. Signal Process*, 2008:1–9, January 2008.
- Mandic, Danilo P and Goh, Vanessa Su Lee. *Complex valued nonlinear adaptive filters: noncircularity, widely linear and neural models*. Wiley, Chichester, U.K., 2009.
- Senior, A., Heigold, G., Ranzato, M., and Yang, Ke. An empirical study of learning rates in deep neural networks for speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 6724–6728, May 2013.
- Snoek, Jasper, Larochelle, Hugo, and Adams, Ryan P. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pp. 2951–2959, 2012.
- Sorber, Laurent, Barel, Marc Van, and Lathauwer, Lieven De. Unconstrained optimization of real functions in complex variables. *SIAM Journal on Optimization*, 22(3):879–898, 2012.
- Terabayashi, K., Natsuaki, R., and Hirose, A. Ultrawideband direction-of-arrival estimation using complex-valued spatiotemporal neural networks. *Neural Networks and Learning Systems, IEEE Transactions on*, 25(9):1727–1732, Sept 2014.
- van den Bos, A. Complex gradient and hessian. *IEE Proceedings - Vision, Image and Signal Processing*, 141:380–382(2), December 1994.
- Wirtinger, W. Zur formalen Theorie der Funktionen von mehr komplexen Ver anderlichen. *Mathematische Annalen*, 97(1):357–375, 1927.
- Zimmermann, Hans-Georg, Minin, Alexey, and Kusherbaeva, Victoria. Comparison of the complex valued and real valued neural networks trained with gradient descent and random search algorithms. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, Bruges, Belgium, April 27–29 2011.