Navigating the Complex Plane: A Curious Exploration of Complex-Valued Activation Functions in Neural Networks

Nicklas Holmberg
nholmber@ucsd.edu
2022

Purpose

This exploratory paper covers some of the key factors involved in selecting an efficient activation function for Complex Valued Neural Networks (CVNNs). This effort is driven out of curiosity and a desire to understand the underlying principles that make CVNNs a promising new direction in neural network research. My primary intention here is to introduce the problem, take initial steps in testing, and provide some numerical insights.

This exploration is an example of the interests I wish to pursue further in graduate school. Many of my side projects, like this one, have been attempts to explore topics at the intersection of mathematics and computer science, particularly within machine learning. Hopefully, these endeavors reflect my curiosity and a growing passion for understanding subjects within the field, and I hope to continue this journey through more structured and in-depth studies in the future.

Abstract

Complex-valued neural networks process data that includes both magnitude and phase information, accommodating systems where data is inherently complex-valued. Designing such a neural network involves choosing an activation function, whose primary role is to introduce non-linearities into the model, allowing the network to capture patterns and relationships that would be unattainable with linear transformations alone. This paper addresses the issue of developing effective such functions and explores the underlying analytical characteristics and challenges of the complex domain relevant to the design and efficacy of activation functions in complex-valued neural networks. Additionally, this study covers the theory of some of the previously proposed activation functions, and evaluates their influence on model accuracy, convergence speed, and robustness to noise. The technical evaluation is done using a Python framework that extends current Tensorflow support for complex-valued data [1]. As anticipated, it was concluded that the choice of activation function indeed has a clear impact on model accuracy and loss metrics, with certain functions significantly enhancing the network's performance in complex data environments.

Index Terms

neural network, machine learning, complex number, signal processing, activation function.

I. Introduction

In the rapidly evolving field of neural networks, the introduction of Complex-Valued Neural Networks (CVNNs) has opened new possibilities for processing and analyzing complex-valued data. Historically, neural networks have largely centered around real-valued data, with frameworks designed for floating point or integer numbers. However, many domains, from signal processing to bioinformatics, are sometimes better expressed using complex data. Complex numbers, by definition, consist of two parts: a magnitude (or amplitude) and a phase (or angle). In scenarios where the data naturally has both magnitude and phase, merely projecting complex data to a real domain can risk losing its inherent richness and nuance. In signal processing for instance, the phase of a signal can dictate crucial aspects like timing or synchronization. This is why developing neural networks capable of efficiently handling complex data is becoming increasingly essential in modern computing.

An essential component of developing any neural network (real-valued or otherwise) involves choosing an activation function, whose primary role is to introduce non-linearities into the model. These non-linearities allow the network to capture patterns and relationships in data that would be unattainable with linear transformations alone. While the main goal of an activation function is to introduce non-linearities, there are other considerations that can be crucial for the stable training of neural networks. For instance, bounded activation functions are desirable because they prevent extreme activations, helping in stabilizing the learning process by ensuring values don't explode or vanish as they pass through layers, thereby aiding convergence during training. In the real-valued domain, functions like the sigmoid is often used and is defined as

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

which is analytic and bounded while being non-linear [2]. However, directly applying functions like these to the complex case in CVNNs is not straightforward. Complicating the landscape for CVNNs is Liouville's theorem, which posits that any function that is holomorphic (analytic) throughout the complex plane and remains bounded must be a constant function— which, as mentioned, is a non-starter for neural networks. The appeal of analytic functions in CVNNs is rooted in their smooth, differentiable nature, which can foster predictable behavior and smoother optimization landscapes. Given these constraints, the challenge is to create activation functions for CVNNs that are non-linear, maintain a strong degree of holomorphic behavior, stay bounded, and thereby enable efficient learning dynamics. This underscores one of the many distinct challenges associated with CVNNs when contrasted with their real-valued counterparts.

In this project, the focus is on testing the performance of complex activation functions that have been highlighted in past research and their efficacy on Fourier-transformed image data, resembling standard signal processing datasets. The existing literature offers a variety of promising approaches for complex activation functions. Some researchers advocate for functions that have strong holomorphic properties, given their potential benefits. At the same time, there's growing interest in "phasor networks", where the activation functions focus on maintaining magnitude while adjusting phase, with their outputs radiating from the origin to the boundaries of the unit circle [3].

Hopefully this gives a brief overview of the background and reasoning behind this project. As the debate between holomorphic and nonholomorphic complex activation functions is still ongoing, the endeavor here is to learn about and document their behavior.

II. Activation functions of interest

The activation functions of interest in this paper represent a diverse set of mathematical properties and behaviors within the context of CVNNs. The Rectified Linear Unit (ReLU) serves as a baseline for comparison, given its widespread use in traditional real-valued neural networks. The ReLU function introduces non-linearity by allowing positive values to pass through unchanged while setting negative values to zero, defined mathematically as:

$$\text{ReLU(z)} = \max(0, \text{R(z)})$$

As can be seen, the ReLU activation function is not directly applicable to complex numbers as the imaginary part is completely overlooked. Yet, it is a cornerstone in deep learning, having contributed to state-of-the-art performances in various real-valued tasks including image classification, natural language processing, and speech recognition [4]. However, as previously stated, ReLU is conventionally defined for real numbers, and its application to complex-valued neural networks require adaptation or generalization. In this experiment however, ReLU is used as a measuring stick against the other activation functions being tested.

Given its success in real-valued networks, there have been suggestions to adapt the ReLU activation function for the complex domain. A well-known example of this is the modified ReLU (modReLU), defined as:

$$\sigma_{\text{modReLU}}(z) = \begin{cases} (|z|+b)\frac{z}{|z|} & \text{if } |z|+b \geq 0 \\ 0 & \text{if } |z|+b < 0 \end{cases}$$

where b and c are constants [4]. Just as ReLU allows positive numbers to pass while setting negative values to zero, modReLU allows complex numbers with magnitudes above a certain threshold to pass, and sets others to zero, offering a similar "filtering" effect for complex values.

The Complex Leaky ReLU was developed as another complex-valued generalizations of ReLU [5]. It is an extension of the traditional Leaky ReLU activation function designed specifically for complex-valued inputs. It's defined as:

$$f(z) = \{\text{LeakyReLU}\}(\text{Re}(z)) + i \cdot \{\text{LeakyReLU}\}(\text{Im}(z))$$

where the Leaky ReLU function is given by

$$\{\text{LeakyReLU}\}(x) = \begin{cases} x & \{\text{if }\} x > 0 \\ \alpha x & \{\text{if }\} x \leq 0 \end{cases}$$

and α is a constant that governs the slope of the function for negative values of x. In contrast to the standard ReLU, where negative values are set to zero and thus have no gradient, the Leaky ReLU assigns a non-zero slope to the negative part. The Complex Leaky ReLU operates on both the real and imaginary components separately. If either the real or imaginary part of the complex input is negative, that component is multiplied by α (typically a value close to zero). This ensures that even negative components maintain a non-zero gradient, aiding in gradient propagation during training and preventing the 'dying ReLU' problem. This approach retains the essence of "leakiness" in the complex domain, allowing neural networks to better retain information and potentially enhancing their learning capabilities.

Lastly, I examine a variant of a complex cardioid functions, a novel activation function tailored for complex-valued data [6]. A salient feature of this function is its magnitude transformation, which aligns closely with the ReLU function when considered on the real axis. The distinguishing factor of the complex cardioid is its focus on input phase over magnitude. The function's output magnitude is modulated by the input phase, yet the input phase remains consistent in the output. Mathematically, this is represented as:

$$f(z) = \frac{(1 + \cos(\angle z)) \cdot z}{2}$$

Under this function, inputs on the positive real axis are scaled by a factor of one, those on the negative real axis by zero, and values with a non-zero imaginary component undergo a scaled transformation from one to zero as their phase transitions from the positive to the negative real axis. Notably, when limited to real inputs, the cardioid function converges to the familiar ReLU function. This inherent link between real-valued and complex-valued activations emphasizes its potential significance in our assessment of complex activation functions in CVNNs.

III. Experiment Overview

In the following experiment, the influence of the mentioned complex-valued activation functions on an associated neural network is tested. Utilizing the MNIST dataset—a cornerstone benchmark in image recognition consisting of 70,000 labeled 28x28 grayscale images of handwritten digits—the models are tasked with classifying digits from 0 to 9. This dataset is well-regarded for its balance between complexity and manageability, providing a robust platform for rapid algorithm testing and performance benchmarking. In a novel approach to this classic problem, the experiment introduces a preprocessing step where Fourier Transforms are applied to the images, effectively transitioning the data into the frequency domain. This transformation is pivotal as it allows the neural networks to potentially leverage frequency patterns for image classification, which could offer insights into the effectiveness of complex-valued activation functions like ModReLU, Complex Leaky ReLU, and Complex Cardioid in a domain that differs from the traditional spatial analysis of image data.

The preprocessing step, including formatting the images and applying Fourier Transforms is done as shown in the figure below:

```python
# Load MNIST dataset containing grayscale images of handwritten digits
(X_train, y_train), (X_test, y_test) = mnist.load_data()

# Apply Fourier Transform to the images, transforming them from spatial domain to frequency domain
X_train = np.fft.fft2(X_train)
X_test = np.fft.fft2(X_test)

# Reshape the data to include a channel dimension, making it suitable for convolutional layers
X_train = X_train.reshape((-1, 28, 28, 1))
X_test = X_test.reshape((-1, 28, 28, 1))

# Convert labels to one-hot encoding, making them suitable for categorical cross-entropy loss
y_train = to_categorical(y_train, num_classes=constants.NUM_CLASSES)
y_test = to_categorical(y_test, num_classes=constants.NUM_CLASSES)
```

This transformation into the frequency domain is intended to reveal insights into the behavior of the activation functions under test, particularly in how they process and interpret the phase information inherent in complex numbers. Additionally, the experiment aims to establish a comparative baseline for the effectiveness of these activation functions in a domain that closely mirrors real-world complex data scenarios.

More information about the exact Python framework used for this experiment is available in the same GitHub repository as this document [1]. Without going into too much detail, the framework is developed using TensorFlow and Keras, well-known deep learning frameworks. For each activation function, a model is trained over the course of 100 epochs and subsequently evaluated to compare their effectiveness. The framework extends Tensorflow support for complex-valued data using a custom layer tailored for complex-valued computations to efficiently handling both their real and imaginary parts.

The evaluation of the activation functions is based on the following performance metrics, chosen to capture various aspects of each model's performance:

- **Validation Accuracy**: This is a metric that represents the model's performance on a separate dataset not used during training, typically reflecting the model's ability to generalize. If validation accuracy increases over 100 epochs, it suggests the model is improving and effectively learning from the training data; if it plateaus or decreases, it may indicate overfitting or a need for further tuning.
- **Confusion Matrix**: A 10-channel confusion matrix is a grid that visualizes the accuracy of a classification model across 10 distinct classes, which in the context of this experiment corresponds to 10 separate integers. Each row of the matrix represents the true class, while each column represents the predicted class, with the diagonal showing the number of correct predictions, and the off-diagonal cells indicating misclassifications; this allows for a detailed analysis of the model's performance, revealing which specific classes are being confused with others.
- **Training Loss**: This is a measure of the model's prediction error on the training dataset, with lower values indicating better performance. Over 100 epochs, a decreasing trend in training loss demonstrates the model's learning and convergence, while a plateau or increase may indicate issues such as overfitting or underfitting.

IV. Results & Observations

Figure 1 shows a comparison of validation accuracy for four different neural network activation functions — ReLU, ModReLU, Complex Leaky ReLU, and Complex Cardioid — over 100 epochs. Each line represents the mean validation accuracy per epoch for a model using the corresponding activation function. As can see be seen, the ReLU activation function exhibits the lowest accuracy, stabilizing at around 50 %. This is somewhat expected as the complex-valued portion of the signal is completely unaccounted for in our model. The ModReLU activation functions shows significantly improved accuracy compared to ReLU, stabilizing at around 86 % validation accuracy. As previously stated, ModReLU introduces a parameter that allows negative values to be modulated, which can help overcome the "dying ReLU" issue. Its better performance compared to ReLU suggests that the modulation of negative values indeed helps in capturing more useful features during training. The Complex Leaky

ReLU activation function and the Complex Cardioid activation function exhibit the highest validation accuracy, stabilizing at around 97%. For the Complex Leaky ReLU function, this suggests that introducing a "leaky" component that allows small gradients when the unit is not active can be very effective in complex domains. For the Complex Cardioid function, this suggest that the added phase-sensitive complex extension of the ReLU functions is effective and adds significant accuracy to the model compared to the traditional ModReLU function.

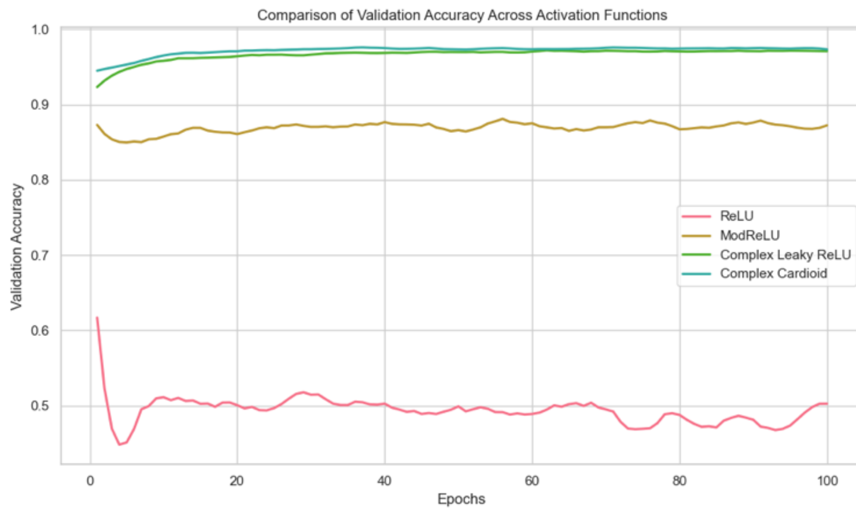

Fig 1. Validation Accuracy across 100 epochs

Figure 2 shows a confusion matrix for each model using the 4 activation functions being tested. Each matrix displays the number of predictions for each actual digit (0-9) made by the respective models. Diagonal cells represent correctly classified instances, with darker shades indicating higher counts. Non-diagonal cells indicate misclassifications. ReLU clearly exhibits the least accurate predictions, where most digits varying extents (especially 2 & 3) were commonly falsely identified as the number 8. Intuitively, this could hint at the fact that most numbers have the make-up of the number 8, suggesting that the model may be overfitting to characteristics of the number 8 or that the internal representation learned by the model is not sufficiently discriminative for the unique features of each digit. This could be due to the topology of the digit 8, which contains loops and curves that are partial components of many other digits, leading to a higher rate of confusion. As for the other three activation functions, ModReLU corrects most of the errors made by the model using the ReLU function. Similarly, the models using Complex Leaky ReLU and Complex Cardioid demonstrate an even improved performance with uniform correct classifications across all digits.

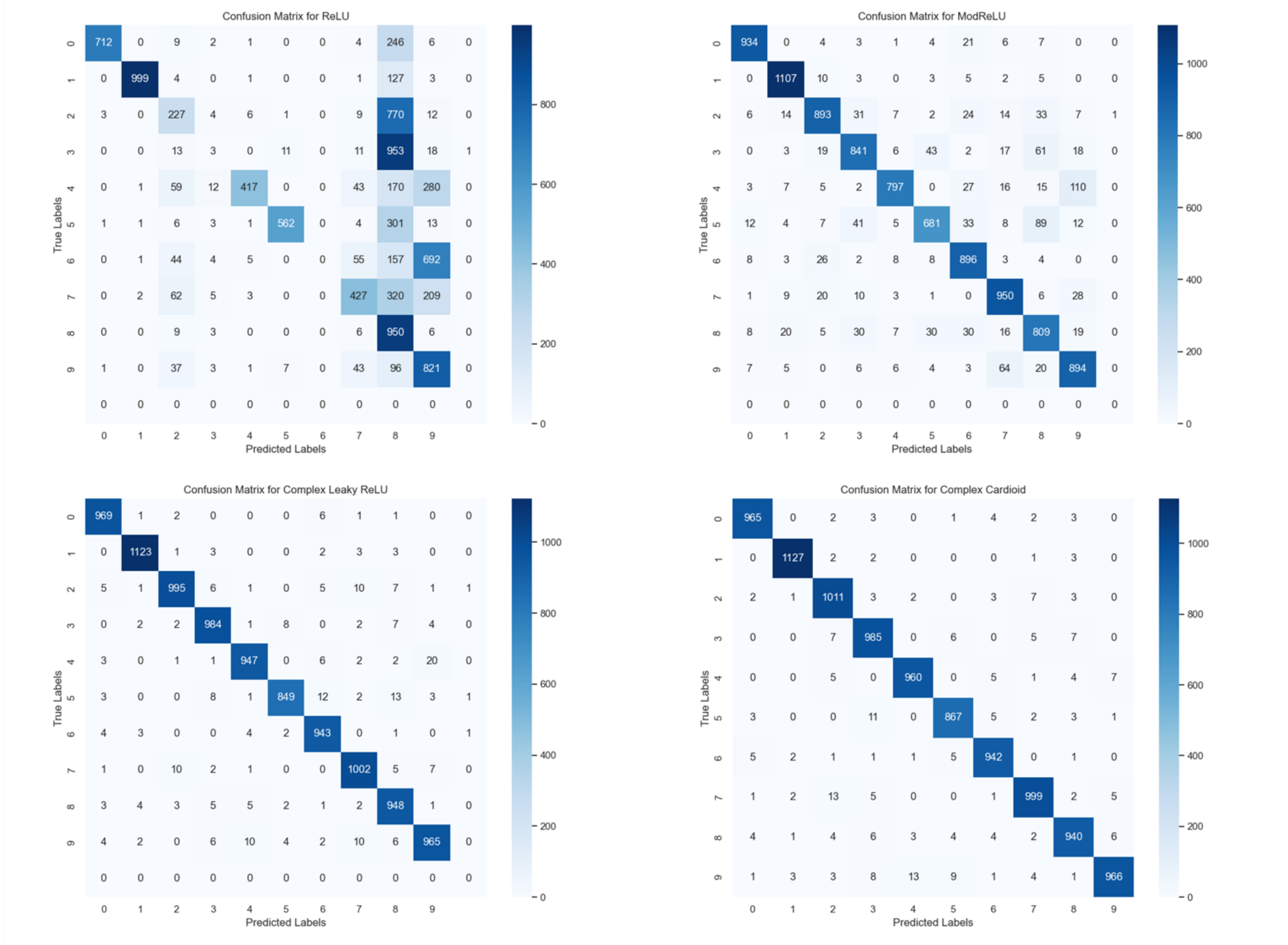

Fig 2. Confusion Matrix for each model

Figure 3 illustrates the training loss trajectories for each neural network models utilizing the same 4 activation functions being tested. Initially, all models exhibit a steep decline in training loss, indicating rapid learning from the dataset. As the epochs progress, the loss for each model stabilizes, suggesting that the models are converging and learning plateaus. Notably, the ModReLU activation function starts at a higher loss but quickly aligns with the others, while the remaining activation functions show a similarly smooth and stable convergence to a low training loss.

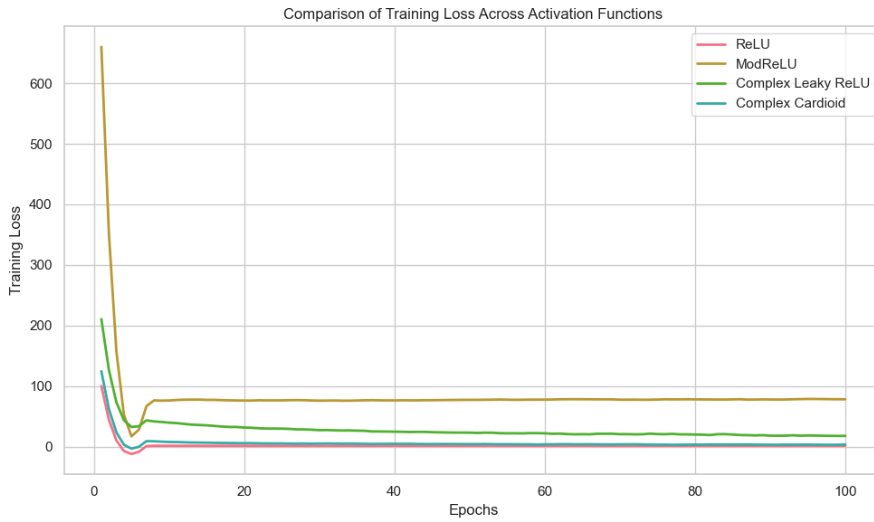

Fig 3. Training Loss across epochs

V. Conclusion

Upon reviewing the findings, it is evident that the choice of activation function plays a crucial role in model performance. Functions like the Complex Leaky ReLU and the Complex Cardioid clearly outperformed both the ReLU and the ModReLU functions, emphasizing the importance of ongoing research in this domain. The stark contrast in results between the functions suggests that even established methods can be improved when we tailor activation functions specifically to the nuances of the problem at hand.

In short, the behaviors of different activation functions, whether related to smoothness, boundary actions, or holomorphic properties, are clearly central to their effectiveness. A more granular breakdown and a series of systematic experiments would be necessary to truly discern the cause-and-effect relationship governing the performance differences across various complex activation functions. My primary intention here is not to present an exhaustive account, but rather to introduce the problem, take initial steps in testing, and provide some preliminary numerical insights. As I hopefully soon start my graduate studies, I hope to delve deeper into this area, seeking more answers and refining my understanding further.

Footnotes:

[1] Holmberg, Nicklas, ComplexValuedNeuralNetworks. Github Repository. Available at https://github.com/NicklasHolmberg/ComplexValuedNeuralNetworks. 2022
[2] Pratiwi, Heny, et al. "Sigmoid Activation Function in Selecting the Best Model of Artificial Neural Networks." *J. Phys.: Conf. Ser.*, vol. 1471, 2020, p. 012010.
[3] Bassey, Joshua, Xiangfang Li, and Lijun Qian. "A Survey of Complex-Valued Neural Networks." *Texas A&M University System,* 2021.
[4] M. Arjovsky et al., "Unitary Evolution Recurrent Neural Networks," 2016.
[5] "Empirical Evaluation of Rectified Activations in Convolutional Network," arXiv:1505.00853v2 [cs.LG], 2015.
[6] Virtue, Patrick, Stella X. Yu, and Michael Lustig. "Better than Real: Complex-valued Neural Nets for MRI Fingerprinting." *University of California, Berkeley / International Computer Science Institute,* 2017.