WIDELY LINEAR KERNELS FOR COMPLEX-VALUED KERNEL ACTIVATION FUNCTIONS

Simone Scardapane[⋆] *Steven Van Vaerenbergh*[†] *Danilo Comminiello*[⋆] *Aurelio Uncini*[⋆]

[⋆] DIET Department, Sapienza University of Rome, Italy
[†]Department of Communications Engineering, University of Cantabria, Spain

ABSTRACT

Complex-valued neural networks (CVNNs) have been shown to be powerful nonlinear approximators when the input data can be properly modeled in the complex domain. One of the major challenges in scaling up CVNNs in practice is the design of complex activation functions. Recently, we proposed a novel framework for learning these activation functions neuron-wise in a data-dependent fashion, based on a cheap one-dimensional kernel expansion and the idea of kernel activation functions (KAFs). In this paper we argue that, despite its flexibility, this framework is still limited in the class of functions that can be modeled in the complex domain. We leverage the idea of widely linear complex kernels to extend the formulation, allowing for a richer expressiveness without an increase in the number of adaptable parameters. We test the resulting model on a set of complex-valued image classification benchmarks. Experimental results show that the resulting CVNNs can achieve higher accuracy while at the same time converging faster.

Index Terms— Complex-valued neural network, activation function, kernel method

1. INTRODUCTION

Inference in the complex domain is a fundamental task in both signal processing [1] and machine learning [2]. Among the approaches proposed over the years, complex-valued neural networks (CVNNs) are gaining a large interest [3–6], as they promise to replicate the recent breakthroughs in (real-valued) deep learning to complex-valued problems, such as forecasting and control of complex signals. Working in the complex domain, however, poses a range of unique problems arising from the properties of complex algebra. Foremost among them is the design of complex activation functions [5]: even extending the rectified linear unit (ReLU) has been shown to be highly non-trivial, with multiple proposals being made over the last two years [3, 7]. Several works end up using naive *split* formulations, wherein the real and imaginary parts of the activation are processed independently, with a loss in terms of expressiveness [8].

In [5] we proposed a different approach, where the activation functions are *learned* in the complex domain via a simple mono-dimensional parameterization. The idea, based on the concept of kernel activation functions (KAFs) originally developed in [9] for the real domain, is to model each function as an independent one-dimensional kernel model, whose mixing weights are adapted through back-propagation, while the dictionary of the kernel matrix is fixed in advance by sampling the complex plane. Despite the empirical performance shown in [5] on multiple benchmarks problems, in this paper we argue that the expressiveness of each KAF, as defined in [5], is still limited when working in the complex domain. In particular, very recently it was shown that the standard formulation of complex-valued kernel methods (which is also adopted in the KAF) is insufficient to model a large set of signals, because more than a single kernel is needed to model the statistics of a complex signal [10, 11]. This leads to the concept of *pseudo-kernels* and to *widely linear kernel* methods.

Contribution of the paper: in this paper we combine the ideas of [5] and [10] and we propose a widely linear KAF (WL-KAF) model, a non-parametric activation function defined directly in the complex domain with no constraints on its expressiveness (as opposed to [5]). We experiment with different choices for the kernel and pseudo-kernel, showing definite improvements on a series of image classification benchmarks in the complex domain, with higher accuracy and faster convergence during optimization.

Organization of the paper: in Sections 2 and 3 we recall the formulation of CVNNs and complex-valued activation functions. Section 4 describes the proposed WL-KAF. Then, we empirically validate its performance in Section 5, before concluding in Section 6 with some remarks on future lines of research.

2. COMPLEX-VALUED NEURAL NETWORKS

A CVNN is defined analogously to its real-valued counterpart as the composition of L layers [12]:

$$f(\mathbf{x}) = \left(f^L \circ f^{L-1} \circ \ldots \circ f^1\right)(\mathbf{x}), \tag{1}$$

S. Van Vaerenbergh is supported by the Ministerio de Economía, Industria y Competitividad (MINECO) of Spain under grant TEC2016-81900-REDT (KERMES).

where $\mathbf{x} \in \mathbb{C}^F$ is the input to the network. Each layer is composed of an adaptable linear projection followed by an element-wise nonlinearity g:

$$f^i(\mathbf{h}) = g\left(\mathbf{W}_i\mathbf{h} + \mathbf{b}_i\right) . \tag{2}$$

where $\mathbf{W}_i$ and $\mathbf{b}_i$ are a matrix and a vector that contain (complex-valued) adaptable parameters. While we focus on feedforward networks, we note that by replacing (2) with more elaborate formulations one can obtain complex equivalents of other types of NNs, e.g., convolutional or recurrent networks [3, 4, 6]. Given N training pairs $\{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^N$ we train the network by minimizing a regularized loss:

$$J(\mathbf{w}) = \sum_{n=1}^{N} l(\mathbf{y}_n, f(\mathbf{x}_n)) + C \cdot \|\mathbf{w}\|^2 , \tag{3}$$

where all adaptable parameters are collected in $\mathbf{w} \in \mathbb{C}^Q$, $l(\cdot,\cdot)$ is a loss function, and C a real-valued scalar (chosen by the user) weighting the regularization term. An example of complex loss is the squared one:

$$l(\mathbf{y}, \hat{\mathbf{y}}) = (\mathbf{y} - \hat{\mathbf{y}})^H (\mathbf{y} - \hat{\mathbf{y}}) , \tag{4}$$

where $(\cdot)^H$ is the Hermitian transpose of the vector. Since (3) is non-analytic, CR-calculus [1, 13] can be used to define proper complex derivatives for use in any optimization algorithm.

3. COMPLEX-VALUED ACTIVATION FUNCTIONS

As we stated in the introduction, the design of $g(\cdot)$ in the complex domain is more challenging when compared to the real-valued one, mostly due to Liouville's theorem [2].[1] It is common for example to work in a split fashion [14]:

$$g(z) = g_R(\Re\{z\}) + i g_R(\Im\{z\}) , \tag{5}$$

where z is a single (scalar) activation, $\Re\{z\}, \Im\{z\}$ are the real and imaginary components of z, and g_R a generic real-valued activation function. Alternative approaches involve phase-amplitude functions acting on the magnitude of the activations, e.g. [15]:

$$g(z) = \tanh\{|z|\} \exp\{i\phi(z)\} , \tag{6}$$

where $\phi(z)$ is the phase of z. As mentioned in Section 1, other authors have also proposed the use of fully complex trigonometric functions, or different variants of the ReLU (commonly used in the real-valued case) [4]. We refer to [5] for a more general overview on the topic. Generally speaking, none of these approaches clearly outperform the others in practice, making it an open research field.

[1] We only consider the choice of $g(\cdot)$ for the hidden layers, while the choice of the activation function in the outer layer depends on the task (see also Section 5).

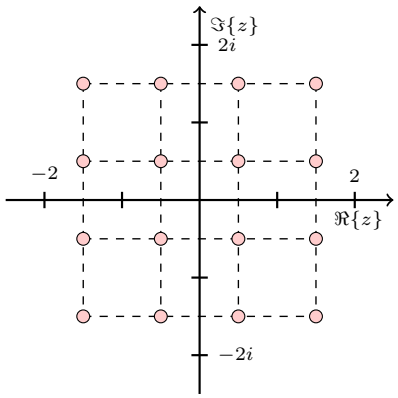

Fig. 1. Example of dictionary sampling in the complex plane, with $D = 16$ elements sampled in $[-2, +2]$ on both axes.

3.1. Kernel activation functions

In [5] we proposed to alleviate the problem of designing complex activation functions by *learning* their shape directly in the complex domain. To this end, we model each activation function (separately for every neuron) with a small number of complex-valued adaptable parameters, representing the linear coefficients in a kernel-based expansion. To introduce the model, we start by sampling the complex space uniformly around 0, with a resolution chosen by the user, as shown pictorially in Fig. 1. The resulting D elements $\mathbf{d} = [d_1, \ldots, d_D]^T$ will form our *dictionary*. Given this fixed dictionary, a kernel activation function (KAF) in the complex domain is defined as:[2]

$$g(z) = \sum_{n=1}^{D} \alpha_n \kappa(z, d_n) = \mathbf{k}^T \boldsymbol{\alpha} , \tag{7}$$

where κ is a valid kernel function over complex inputs, $\mathbf{k}$ is a column vector containing the D kernel values computed between z and the dictionary $\mathbf{d}$, and the parameters $\{\alpha_n\}_{n=1}^D$ are adapted independently for every neuron, together with the linear weights in (2), via standard back-propagation. Fixing the dictionary in advance allows for an extremely efficient (vectorized) implementation of (7) [5].

The choice of κ can leverage over a large body of literature on complex reproducing kernel Hilbert spaces [16, 17]. In particular, in [5] we performed experiments with a complex-valued extension of the classical Gaussian kernel:

$$\kappa(z, d) = \exp\left\{-\gamma (z - d^*)^2\right\} , \tag{8}$$

where γ is a hyper-parameter, and the independent kernel proposed in [17]:

[2] [5] also considers a split version of the standard KAF. We focus here on the fully complex extension.

$$\kappa(z,d) = \kappa_{\mathbb{R}}(\Re\{z\},\Re\{d\}) + \kappa_{\mathbb{R}}(\Im\{z\},\Im\{d\}) + i\left(\kappa_{\mathbb{R}}(\Re\{z\},\Im\{d\}) - \kappa_{\mathbb{R}}(\Im\{z\},\Re\{d\})\right). \quad (9)$$

where $\kappa_{\mathbb{R}}$ is a generic real-valued kernel (chosen as the standard Gaussian in [5]). In the experiments for this paper we will consider a more recent proposal from [10], a real-valued Gaussian kernel with complex inputs given by:

$$\kappa(z,d) = \exp\left\{-\gamma(z-d)^*(z-d)\right\}. \quad (10)$$

4. PROPOSED WIDELY LINEAR KAF

The key motivation for this paper is that the model in (7) is limited in the kind of complex-valued function it can approximate, an observation first made in [10]. To see this, note that one can express the complex function $g(z)$ in terms of a kernel method with two outputs, namely, the real and imaginary parts $g_r(z)$, $g_i(z)$. According to the theory of vector-valued kernel methods [18], the corresponding kernel is now *matrix-valued* and the output can be written as:

$$g(z) = \begin{bmatrix} g_r(z) \\ g_i(z) \end{bmatrix} = \begin{bmatrix} \mathbf{k}_{rr}^T & \mathbf{k}_{ri}^T \\ \mathbf{k}_{ir}^T & \mathbf{k}_{ii}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_r \\ \boldsymbol{\alpha}_i \end{bmatrix}, \quad (11)$$

where we now have *four* column vectors $\{\mathbf{k}_{rr}, \mathbf{k}_{ri}, \mathbf{k}_{ir}, \mathbf{k}_{ii}\}$ corresponding to the four outputs of the kernel , and two sets of linear weights $\boldsymbol{\alpha}_r$ and $\boldsymbol{\alpha}_i$. Substituting (7) into (11) shows that (7) forces the constraints $\mathbf{k}_{rr} = \mathbf{k}_{ii}$ and $\mathbf{k}_{ri} = -\mathbf{k}_{ir}$, limiting the expressiveness of the overall model. A solution to this is the adoption of widely linear kernel methods [10].

Following this, we propose an extension of the complex-valued KAF adopting widely linear kernels, that we term widely linear KAF (WL-KAF):

$$g(z) = \mathbf{k}^T\boldsymbol{\alpha} + \widetilde{\mathbf{k}}^T\boldsymbol{\alpha}^*, \quad (12)$$

where $\widetilde{\mathbf{k}} = [\widetilde{\kappa}(z,d_1),\ldots,\widetilde{\kappa}(z,d_D)]$, and $\widetilde{\kappa}$ is called the 'pseudo-kernel'. $\boldsymbol{\alpha}^*$ is the complex conjugate of $\boldsymbol{\alpha}$. The model in (12) does not impose the previously discussed limitations, and it can be shown that:

$$\mathbf{k} = 0.5\left[\mathbf{k}_{rr} + \mathbf{k}_{ii} + i\left(\mathbf{k}_{ir} - \mathbf{k}_{ri}\right)\right], \quad (13)$$

$$\widetilde{\mathbf{k}} = 0.5\left[\mathbf{k}_{rr} - \mathbf{k}_{ii} + i\left(\mathbf{k}_{ir} + \mathbf{k}_{ri}\right)\right]. \quad (14)$$

Depending on the choice of the kernel and pseudo-kernel, the resulting model has a larger amount of expressiveness compared to the standard one. In the context of KAFs and CVNNs, the model has two additional properties to its favor. Firstly, as we will see shortly, since the dictionary is fixed the kernel and pseudo-kernel can generally share a large amount of computation, making the modification extremely cheap in terms of speed. Secondly, the use of widely linear models does not increase the number of adaptable parameters, since in our case we are only adapting the mixing coefficients $\boldsymbol{\alpha}$. Following [10], in the experiments we consider two different choices for the kernel and pseudo-kernel.

Case 1: if we assume that the real and imaginary parts of $g(z)$ are independent, the off-diagonal blocks in (11) cancel and we are left with:

$$\mathbf{k} = 0.5\left[\mathbf{k}_{rr} + \mathbf{k}_{ii}\right], \quad (15)$$

$$\widetilde{\mathbf{k}} = 0.5\left[\mathbf{k}_{rr} - \mathbf{k}_{ii}\right]. \quad (16)$$

In this case we use (10) with two separate parameters γ for $\mathbf{k}_{rr}$ and $\mathbf{k}_{ii}$. More specifically, both bandwidths in our experiments are initialized following the rule of thumb taken from [9], but are subsequently adapted via back-propagation independently for every neuron.

Case 2: in the case where the real and imaginary parts are not assumed independent, we can exploit the theory of separable kernels and mixed effect regularizers introduced for vector-valued kernels [18]. In our case we obtain, for an hyper-parameter Q chosen by the user [10]:

$$\kappa(z,d) = \sum_{q=1}^{Q}\kappa^q(z,d), \quad (17)$$

$$\widetilde{\kappa}(z,d) = 2i\sum_{q=1}^{Q}\omega^q\widetilde{\kappa}^q(z,d), \quad (18)$$

with all the kernels κ^q and $\widetilde{\kappa}^q$ being real-valued in output, and $0 < \omega^q < 1$. As before, one can exploit different Gaussian kernels as in (10), letting the different bandwidths adapt via back-propagation.

5. EXPERIMENTAL EVALUATION

We evaluate the two proposed WL-KAFs on a series of complex-valued image classification benchmarks extended from [5]. We consider four problems:

- **MNIST**,[3] composed of 60000 28×28 images belonging to ten digit classes.
- **Fashion MNIST** (F-MNIST) [19]: a variant of MNIST where classes are clothing items, with the same dimensionality and size as MNIST.
- **Extended MNIST** (EMNIST) [20]: we use the 'Digits' extension, having 240 thousand images of handwritten digits.
- **Latin OCR** [21]: an OCR problem of handwritten Latin characters extracted from manuscripts of the Vatican secret archives. There are 12000 images and 23 classes.

[3] http://yann.lecun.com/exdb/mnist/

Table 1. Test accuracy (mean and standard deviation) for the complex-valued image classification benchmarks (see main discussion for the preprocessing phase). First two rows are taken from [5]. The best results for each dataset are highlighted in bold.

Model	**MNIST**	**F-MNIST**	**E-MNIST**	**Latin OCR**
Real-valued NN	92.39 ± 0.10	71.08 ± 0.45	92.78 ± 1.25	39.01 ± 3.42
KAF	97.18 ± 0.27	81.94 ± 0.91	98.11 ± 2.04	71.79 ± 2.40
Proposed WL-KAF (Case 1)	$\mathbf{97.50 \pm 0.41}$	77.29 ± 2.43	98.46 ± 0.12	$\mathbf{74.57 \pm 0.80}$
Proposed WL-KAF (Case 2)	96.22 ± 0.74	$\mathbf{82.89 \pm 1.09}$	$\mathbf{99.03 \pm 1.01}$	72.53 ± 0.36

To convert these to complex-valued problems, we adopt the procedure from [22] and preprocess each image with a fast Fourier transform (FFT), then rank the coefficients of the FFT in terms of significance (by considering their mean absolute value), keeping only the 100 most significant coefficients as input to the models.

The results in [5] are taken as a baseline, to which we add two CVNNs of the same dimensionality as [5] (three hidden layers of 100 neurons each) exploiting the proposed WL-KAF. We use a dictionary by sampling 8 points on each axis equispaced in $[-2, +2]$. For the case 2 in (18), as in [10], we use $Q = 1$, $\omega^1 = 0.3$, and the Gaussian kernel in (10) for the two kernels. As stated before, in all cases the kernel bandwidth γ in (10) is initialized with the rule of thumb in [9] and then adapted independently for every kernel via backpropagation. The KAFs are applied only to intermediate layers, while the output $\mathbf{h}$ of the last linear projection is fed to a softmax-like function to compute the class probabilities:

$$\text{softmax}_n(\mathbf{h}) = \frac{\exp\left\{\Re\{h_n\}^2 + \Im\{h_n\}^2\right\}}{\sum_{t=1}^{C} \exp\left\{\Re\{h_t\}^2 + \Im\{h_t\}^2\right\}}, \quad (19)$$

We minimize a regularized cross-entropy over the training data, where the amount of regularization is found through grid search as in [5]. We use a version of the Adagrad algorithm on random mini-batches of 40 images to perform optimization. We further employ an early stopping procedure, stopping the optimization whenever the accuracy computed over the validation split of the dataset is not improving for 1000 iterations of optimization.

The results of the experiments are provided in Table 1. "Real-valued NN" is a NN having the same dimensionality as the others, but treating real and imaginary parts of the input vector as separate inputs. "KAF" is the KAF in (7) using the independent kernel in (9). As can be seen, CVNNs with the proposed WL-KAFs can achieve in all cases a superior performance, without introducing additional parameters compared to the standard complex-valued KAF. This increase in performance translates to faster convergence, an example of which (on the Latin OCR dataset) is shown in Fig. 2.

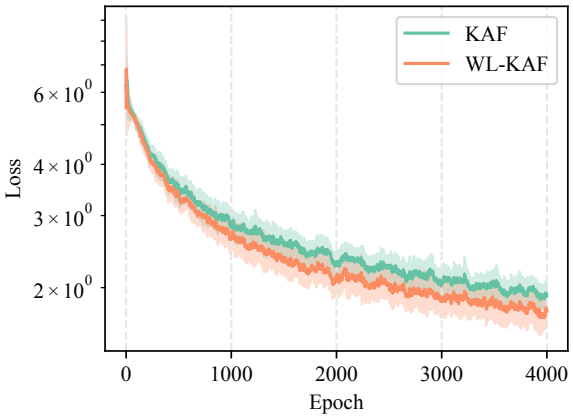

Fig. 2. Convergence of KAF and WL-KAF (case 1) on the Latin OCR dataset. Standard deviation is shown with a lighter color, while the plot is zoomed on the first 4000 iterations.

6. CONCLUSION

In this paper we proposed a new model for learning activation functions for complex-valued neural networks. The model extends the idea of kernel activation functions (KAFs), by incorporating recent ideas from the field of widely linear kernel approximation. Compared to the standard KAF, the widely linear KAF does not require additional trainable parameters while possessing increased flexibility. On a set of complex-valued image classification benchmarks, it achieves better accuracy in all problems while at the same time being faster in terms of optimization. Future work will consider a formal analysis of the generalization properties of the proposed KAFs, and their evaluation in more elaborate complex benchmarks. For the latter, we plan a more comprehensive evaluation of kernels over complex spaces, along with the definition of proper strategies for finding complex hyperparameters (e.g., complex-valued learning rates in the optimization procedure [23]).

7. REFERENCES

[1] P. J. Schreier and L. L. Scharf, *Statistical signal processing of complex-valued data: the theory of improper and noncircular signals*, Cambridge University Press, 2010.

[2] A. Hirose, *Complex-valued neural networks: theories and applications*, vol. 5, World Scientific, 2003.

[3] N. Guberman, "On complex valued convolutional neural networks," *arXiv preprint arXiv:1602.09046*, 2016.

[4] C. Trabelsi, O. Bilaniuk, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks," *35th International Conference on Machine Learning (ICML)*, 2018.

[5] S. Scardapane, S. Van Vaerenbergh, A. Hussain, and A. Uncini, "Complex-valued neural networks with nonparametric activation functions," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2019, in press.

[6] Izhak Shafran, Tom Bagby, and RJ Skerry-Ryan, "Complex evolution recurrent neural networks (cernns)," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5854–5858.

[7] M. Arjovsky, A. Shah, and Y. Bengio, "Unitary evolution recurrent neural networks," in *33rd International Conference on Machine Learning (ICML)*, 2016, pp. 1120–1128.

[8] H. Leung and S. Haykin, "The complex backpropagation algorithm," *IEEE Transactions on Signal Processing*, vol. 39, no. 9, pp. 2101–2104, 1991.

[9] S. Scardapane, S. Van Vaerenbergh, S. Totaro, and A. Uncini, "Kafnets: kernel-based non-parametric activation functions for neural networks," *Neural Networks*, vol. 110, pp. 19–32, 2019.

[10] R. Boloix-Tortosa, J. J. Murillo-Fuentes, I. Santos, and F. Pérez-Cruz, "Widely linear complex-valued kernel methods for regression," *IEEE Transactions on Signal Processing*, vol. 65, no. 19, pp. 5240–5248, 2017.

[11] R. Boloix-Tortosa, J. J. Murillo-Fuentes, F. J. Payán-Somet, and F. Pérez-Cruz, "Complex Gaussian processes for regression," *IEEE Transactions on Neural Networks and Learning Systems*, 2018.

[12] T. Kim and T. Adalı, "Approximation by fully complex multilayer perceptrons," *Neural Computation*, vol. 15, no. 7, pp. 1641–1666, 2003.

[13] K. Kreutz-Delgado, "The complex gradient operator and the CR-calculus," *arXiv preprint arXiv:0906.4835*, 2009.

[14] T. Nitta, "An extension of the back-propagation algorithm to complex numbers," *Neural Networks*, vol. 10, no. 8, pp. 1391–1415, 1997.

[15] G. M. Georgiou and C. Koutsougeras, "Complex domain backpropagation," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 39, no. 5, pp. 330–334, 1992.

[16] I. Steinwart, D. Hush, and C. Scovel, "An explicit description of the reproducing kernel Hilbert spaces of Gaussian RBF kernels," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4635–4643, 2006.

[17] P. Bouboulis and S. Theodoridis, "Extension of Wirtinger's calculus to reproducing kernel Hilbert spaces and the complex kernel LMS," *IEEE Transactions on Signal Processing*, vol. 59, no. 3, pp. 964–978, 2011.

[18] M. A. Alvarez, L. Rosasco, and N. D. Lawrence, "Kernels for vector-valued functions: A review," *Foundations and Trends® in Machine Learning*, vol. 4, no. 3, pp. 195–266, 2012.

[19] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[20] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: an extension of MNIST to handwritten letters," *arXiv preprint arXiv:1702.05373*, 2017.

[21] D. Firmani, P. Merialdo, E. Nieddu, and S. Scardapane, "In Codice Ratio: OCR of handwritten latin documents using deep convolutional networks," in *11th International Workshop on Artificial Intelligence for Cultural Heritage (AI*CH 2017)*. CEUR Workshop Proceedings, 2017, pp. 9–16.

[22] P. Bouboulis, S. Theodoridis, C. Mavroforakis, and L. Evaggelatou-Dalla, "Complex support vector machines for regression and quaternary classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 6, pp. 1260–1274, 2015.

[23] H. Zhang and D. P. Mandic, "Is a complex-valued stepsize advantageous in complex-valued gradient learning algorithms?," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 12, pp. 2730–2735, 2016.