ABOUT THE EQUIVALENCE BETWEEN COMPLEX-VALUED AND REAL-VALUED FULLY CONNECTED NEURAL NETWORKS - APPLICATION TO POLINSAR IMAGES

J. A. Barrachina[⋆†] *C. Ren*[⋆] *G. Vieillard*[†] *C. Morisseau*[†] *J.-P. Ovarlez*[⋆†]

[⋆] SONDRA, CentraleSupélec, Université Paris-Saclay, 91192 Gif-sur-Yvette, France
[†] DEMR, ONERA, Université Paris-Saclay, 91120 Palaiseau, France

ABSTRACT

In this paper we provide an exhaustive statistical comparison between Complex-Valued MultiLayer Perceptron (CV-MLP) and Real-Valued MultiLayer Perceptron (RV-MLP) on Oberpfaffenhofen Polarimetric and Interferometric Synthetic Aperture Radar (PolInSAR) database. In order to compare both networks in a fair manner, the need to define the equivalence between the models arises. A novel definition for an equivalent Real-Valued Neural Network (RVNN) is proposed in terms of its *real-valued trainable parameters* that maintain the *aspect ratio* and analyze its dynamics. We show that CV-MLP gets a slightly better statistical performance for classification on the PolInSAR image than a capacity equivalent RV-MLP.

Index Terms— Complex-Valued Neural Network, Real-Valued Neural Network, Polarimetric and Interferometric Synthetic Aperture Radar.

1. INTRODUCTION

Real-Valued Neural Network (RVNN) are widely used in the literature for Polarimetric Synthetic Aperture Radar (PolSAR) applications [1–3]. However, these networks are fed with the amplitude information of the PolSAR image while neglecting the phase data. In contrast, the signal processing community has a greater interest in developing techniques that can deal with complex-valued data [4], which is the principal characteristic of signals encountered in most radar applications, such as our case of study, PolSAR images. CVNNs appear as a natural choice to process and to learn from these complex-valued features since the operation performed at each layer of CVNNs can be interpreted as complex filtering. Notably, CVNNs are more adapted to extract phase information [5], which could be helpful, *e.g.*, for retrieving Doppler frequency in radar signals, classifying Polarimetric Synthetic Aperture Radar (PolSAR) data [6, 7], radio frequency signal processing in wireless communications, etc [4].

The authors would like to thank the Délégation Générale de l'Armement (DGA) for funding.

Applications of Complex-Valued Neural Network (CVNN) on PolSAR images are recent [8]. Furthermore, limited research has been done to compare Complex-Valued MultiLayer Perceptron (CV-MLP) with Real-Valued MultiLayer Perceptron (RV-MLP) on PolSAR applications. In references [9] and [10], the authors tested CV-MLP on a PolSAR database but did not provide a comparison with RV-MLP. In [6], a comparison was performed for both types of networks but did not offer a confidence interval and used a different input representation for each model. Although, in [11], the same authors suggested giving the same amount of input representation to get a more precise comparison between the models.

A criterion to create Complex-Valued Neural Network (CVNN) and Real-Valued Neural Network (RVNN) with equivalent capacities remains missing. This results in an unbalanced comparison, as will be discussed in Section 2. Although [12] added some neurons to the RV-MLP compared to the CV-MLP which made it an acceptable comparison, they do not specify the criteria used and it was not yet enough to make it equivalent to the complex network by neither of the two criteria to be analyzed in Section 2, which are the equal number of *real-valued trainable parameters* (tp) or *real-valued neurons parameters* (np). In [12], even though CV-MLP performed better than RV-MLP, confidence intervals intersect, leaving room for doubt about CV-MLP outperformance.

Works using complex Convolutional Neural Network (CNN) have been published for PolSAR applications. Ref. [13] compares a CV-CNN with the real-valued equivalent but lacking confidence intervals. Other recent works [7, 14, 15] use a CV-CNN for PolSAR applications but without comparing its result with a real-valued model.

The paper's first contribution is to give a new definition of an equivalent-RVNN with respect to any given CVNN architecture (Section 2), in order to get equitable comparisons between networks. Second, a thorough analysis is performed, involving at least one hundred independent trials for each network, in order to infer appropriate errors and statistics. Furthermore, we will test on several MLP model architectures to assert that there is a global trend for the difference between

978-1-7281-6338-3/21/$31.00 ©2021 European Union

the network's performance and that it is no dependant on a specific model. In Section 3, discussions on the database and model architecture choices are detailed. Finally, results are analyzed in Section 4.

The real equivalent definition proposed in Section 2 is already added as a feature to the open-source Python library [16] that enables CVNN implementations.

2. REAL EQUIVALENT NETWORK

To assess whether a Complex-Valued Neural Network (CVNN) is actually of interest, it is necessary to compare it with a Real-Valued equivalent network. However, the equivalence between both networks is not straightforward. Considering that complex plane $\mathbb{C}$ is isomorphic to $\mathbb{R}^2$, one complex parameter ($p_{\mathbb{C}}$) is equivalent to two real parameters ($p_{\mathbb{R}}$) so that $p_{\mathbb{C}} = 2p_{\mathbb{R}}$. The superscript $\mathbb{C}$ and $\mathbb{R}$ indicate whether it corresponds to the CVNN or RVNN respectively.

A natural way to keep the same amount of hidden representation consists of doubling the size of each hidden layer of the RV-MLP with respect to CV-MLP so that both models will have an equivalent number of *real-valued neurons parameters* (np) per layer [9, 17, 18]. However, this definition leads to a higher number of *real-valued trainable parameters* (tp) for the RVNN, as it has been pointed out in reference [19]. Indeed, an example of a CVNN with two consecutive hidden layers of size 10 each will result in 100 (10x10) complex weights for connecting them without bias which is equivalent to a total of $tp_{\mathbb{C}} \triangleq 200$. The above approach will propose an equivalent RVNN with two consecutive hidden layers of size 20 each, needing a total of 400 weights to connect them and, therefore, $tp_{\mathbb{R}} \triangleq 400$. So, the RVNN has a higher value of tp, and thus the latter has potentially a higher capacity if this method is followed.

For a general CV-MLP and RV-MLP with K hidden layers, the global numbers of tp are given by the formula [19]:

$$\begin{aligned} tp_{\mathbb{C}} &= 2\,N_0^{\mathbb{C}}\,N_1^{\mathbb{C}} + 2\sum_{i=1}^{K-1} N_i^{\mathbb{C}}\,N_{i+1}^{\mathbb{C}} + 2N_K^{\mathbb{C}}\,N_L^{\mathbb{C}}\,, \\ tp_{\mathbb{R}} &= N_0^{\mathbb{R}}\,N_1^{\mathbb{R}} + \sum_{i=1}^{K-1} N_i^{\mathbb{R}}\,N_{i+1}^{\mathbb{R}} + N_K^{\mathbb{R}}\,N_L^{\mathbb{R}}\,. \end{aligned} \tag{1}$$

where N_i is the number of neurons for layer $i \in 1, ..., K$. N_0 corresponds to the number of features or input size and N_L to the output size.

Note that the input and output sizes of the real network are directly defined by the task to solve so that $N_0 = N_0^{\mathbb{R}} = 2\,N_0^{\mathbb{C}}$ and

$$N_L = N_L^{\mathbb{R}} = \begin{cases} 2\,N_L^{\mathbb{C}}, & \text{regression task} \\ N_L^{\mathbb{C}}, & \text{classification task} \end{cases}. \tag{2}$$

Reference [19] argue that a real-valued equivalent model must have the same tp capacity as the complex one: $tp_{\mathbb{C}} = tp_{\mathbb{R}} = tp$. To accomplish this, they propose to alternate between doubling or not the number of neurons of the real-valued model hidden layers with respect to the complex-valued model. This method, however, only works when there is an even number of hidden layers for classification tasks and an odd number of hidden layers for regressions tasks. To solve this issue, we propose to select one hidden layer to be:

$$N_i^{\mathbb{R}} = 2\frac{N_{i-1}^{\mathbb{C}} + N_{i+1}^{\mathbb{C}}}{N_{i-1}^{\mathbb{R}} + N_{i+1}^{\mathbb{R}}}\,N_i^{\mathbb{C}}\,. \tag{3}$$

Nevertheless, this solution will not maintain the same *aspect ratio* for both models. For example, performing classification with a CV-MLP with 2 hidden layers of sizes 100 and 50 will be converted to a RV-MLP where both hidden layer sizes are 100. This means to change a network where the first hidden layer doubles the size of the second, to one where both hidden layers are the same size. Another proposition in [19] is to make all layers the same size but the issue of the *aspect ratio* remains.

We propose to maintain the same *aspect ratio* and therefore to satisfy the following equation:

$$N_i^{\mathbb{R}} = r\,N_i^{\mathbb{C}},\ \forall i \in 1, ..., K \tag{4}$$

where r is a positive constant real value. Replacing (4) in (1) we obtain the following second-order polynomial equation in the variable r:

$$tp = r\,N_0\,N_1^{\mathbb{C}} + \sum_{i=1}^{K-1} r^2\,N_i^{\mathbb{C}}\,N_{i+1}^{\mathbb{C}} + r\,N_K^{\mathbb{C}}\,N_L\,. \tag{5}$$

Since r should be positive as well as all parameters tp, $N_i^{\mathbb{R}}$, N_L and N_0, the only possible solution to our problem is therefore:

$$r = \frac{-b + \sqrt{b^2 - 4\,a\,(-tp)}}{2\,a}\,, \tag{6}$$

where $a = \sum_{i=1}^{K-1} N_i^{\mathbb{C}}\,N_{i+1}^{\mathbb{C}}$, $b = N_0\,N_1^{\mathbb{C}} + N_k^{\mathbb{C}}\,N_L$.

In conclusion, there are two possible definitions for an equivalent-RVNN. Either by setting the same *real-valued trainable parameters* (tp) or by its *real-valued neurons parameters* (np) per hidden layer. The former can be done by creating a RVNN where each hidden layer size is given by equation (4) with r being defined by (6); this will result in an equivalent-RVNN in terms of the *real-valued training parameters* that maintain the same aspect ratio that the CVNN hidden layers.

To give an interval for r, we first assume $r < 1$,

$$\begin{aligned} &a\,r^2 + b\,r - tp = 0 < r\,a + r\,b - tp, \\ \Rightarrow\ &0 < r\,a + r\,b - 2\,a - \beta \le a\,(r-2) + b\,(r-1), \end{aligned} \tag{7}$$

where $tp = 2\,a + \beta$ with $b \le \beta = 2\,N_0^{\mathbb{C}}\,N_1^{\mathbb{C}} + 2N_k^{\mathbb{C}}\,N_L^{\mathbb{C}} < 2b$ (equation (1)). As both a and b are positive, equation 7 is absurd, which is expected as it implies that real-valued models

will never have fewer neurons than the complex-valued models. On the other hand, for $r \geq 1$:

$$\begin{aligned} & a\,r^2 + b\,r - tp = 0 \geq r\,a + r\,b - tp, \\ \Rightarrow\ & 0 \geq a\,(r-2) + b\,r - \beta > a\,(r-2) + b\,(r-2)\,. \end{aligned} \tag{8}$$

Again, as a and b are positive, equation (8) is absurd if $r \geq 2$. Because of inequalities (7) and (8) we conclude that $1 \leq r < 2$, meaning that the equivalent-RVNN should have at least the same dimension than CVNN and at most double. In particular, $r = 2$ corresponds to the case for the same value of np. This proves that it is not possible to reach both conditions at the same time and one must choose between setting an equal value for np or tp.

For single hidden layer models, $a = 0$ and therefore, r will be:

$$r = \frac{\beta}{b} = 2\frac{N_0^{\mathbb{C}} + N_L^{\mathbb{C}}}{N_0 + N_L}\,. \tag{9}$$

As it can be derived from (9), $r = 1$ for regressions tasks while for classifications tasks, $1 < r < 2$ depending on the relationship between N_0 and N_L. Finally, as the number of hidden neurons gets bigger with respect to the input and output, or in other words, $a >> b$, it will tend $r \to \sqrt{2}$.

Note that, the extra terms $2\sum_{i=1}^{K} N_i^{\mathbb{C}}$ and $\sum_{i=1}^{K} N_i^{\mathbb{R}}$ should be added to equation (1) in order to take into account the bias. This extra term will lead to a slight variation of r by changing the value of b but does not change its boundary $1 \leq r < 2$.

3. EXPERIMENTAL SETUP

3.1. PolInSAR image

Polarimetric and Interferometric Synthetic Aperture Radar (PolInSAR) classification algorithms generally make use of signal coherence (or equivalently phase and local phase variance) existing on any two co-registered single look complex data channels S_1 and S_2 measured in the horizontal (H) and vertical (V) transmit/receive polarimetric channels:

$$S^{(i)} = \left(S_{HH}^{(i)}, \sqrt{2}\,S_{HV}^{(i)}, S_{VV}^{(i)}\right)^T \quad i \in [1,2]\,. \tag{10}$$

For each pixel of the Synthetic Aperture Radar (SAR) image, these two backscattering vectors are usually expressed in the Pauli basis and are vectorized onto one single complex vector $\boldsymbol{k} = [\boldsymbol{k}^{(1)T}, \boldsymbol{k}^{(2)T}]^T \in \mathbb{C}^6$ where

$$\boldsymbol{k}^{(i)} = \frac{1}{\sqrt{2}}\left(S_{HH}^{(i)} + S_{VV}^{(i)}, S_{HH}^{(i)} - S_{VV}^{(i)}, 2\,S_{HV}^{(i)}\right)^T\,. \tag{11}$$

The Hermitian coherency matrix is then formally built according to $\boldsymbol{T} = \frac{1}{n}\sum_{j}^{n} \boldsymbol{k}_j\,\boldsymbol{k}_j^H$, where the operator ^{H} stands for complex conjugate operation and where n is the number of pixels chosen in a boxcar located in each local area of the SAR image.

(a) Visualization of data (b) Ground Truth

Fig. 1. PolInSAR data of Oberpfaffenhofen, Germany. **A** Built-up Area; **B** Woodland; **C** Open Area.

The experiments were run over the well-known Oberpfaffenhofen database that can be downloaded from the European Space Agency (ESA) website. The coherency matrix is provided as data. Because the diagonal is real-valued, it was treated as a complex-valued number with the imaginary part equal to zero. Since matrix $\boldsymbol{T}$ is Hermitian, the lower triangle of the matrix, excluding the diagonal, was discarded as it provided no additional information; this led to a 21 complex input vector for CV-MLP. For the real model, two methods were tested, either by using the real and imaginary parts (conventional RV-MLP) or by using the amplitude and phase as input (polar-RV-MLP). It can be seen in Fig. 1-b the ground truth for 3 different classes (built-up areas, woodland and open areas). These labels were obtained from [13].

All unlabeled pixels were removed and the remaining pixels were randomly shuffled. As the image is very large, previous works used a small percentage of pixels for training to speed up training and validation results. References [6] and [9] used about 2% of the image pixels for training whereas [2] and [20] used 5%. In [21], the authors adopted 10%. Finally, reference [13] tested different sampling rates and proposed, based on the results, to use 10% sampling rate for both train and validation set together. Therefore, we have chosen to use 8% as training set and only 2% as validation set which corresponds to 104928 and 26232 pixels respectively. The remaining 90% was used for the test set. Both train and validation sets had the same amount of examples per class as, regardless of the class occurrences, the application does not prioritize one class over another. This may result in different accuracies for validation and test sets.

3.2. Model Architecture

Most of the model architecture hyper-parameters were chosen based on reference [22] findings and results.

Categorical cross-entropy loss function was used for both complex and real-valued models. Uniform Glorot [23] weight

initialization was used and the bias was initialized as zero since those are *Tensorflow*'s current (v2.4) default initialization methods for dense layers. The adaptation for complex-valued weights initialization is described in [24, p. 6].

The network shape was determined heuristically. A two hidden layers architecture was chosen for the CV-MLP with 100 complex-valued neurons for the first layer and 50 for the second layer. The RV-MLP and polar-RV-MLP shape was dimensioned to have the same amount of tp as explained in Section 2. Dropout at 50% rate was used to prevent overfitting and it was verified that training accuracy falls but validation accuracy was higher than when not dropout was used.

Two cases of hidden layer activation functions were tested. Rectified Linear Unit (ReLU) and hyperbolic tangent (tanh). For the complex-valued model, these activation functions are separately applied to both real and imaginary parts as in [22, 25] and the output layer activation function being a *softmax* [26] that is directly applied to the absolute value.

Stochastic Gradient Descent (SGD) [27] was used as the optimizer with a learning rate of 0.01 and without momentum as they are Tensorflow's v2.4 default values.

The combination of two possible activation functions and three types of input format leads to a total of 6 different experiment setups. Each experiment was evaluated over 100 Monte-Carlo trials for all CV-MLP, RV-MLP and polar-RV-MLP. Each trial had 300 epochs and a batch size of 100.

4. EXPERIMENTAL RESULTS

Statistical indicators of the classification accuracy are summarized in Table 1 for the 6 experimental combinations. The median error was computed as in [28, 29]; if median intervals do not overlap, there is a 95% confidence that their values differ. The confidence interval of the mean is calculated for a confidence level of 99%. For all cases, RV-MLP obtained better performance than polar-RV-MLP; therefore, all discussions will be done with the former case to favor the best-case scenario for RV-MLP.

From Table 1, the use of ReLU hidden activation function led to the higher accuracy of all models. CV-MLP outperformed RV-MLP by an accuracy difference of $\sim 0.5\%$. Although this differences may be considered small, CV-MLP out-performance is statistically justified as the confidence intervals remain very far apart. Even the maximum registered accuracy of RV-MLP is under CV-MLP median. The use of tanh activation function led to similar conclusions but with lower accuracy for all models and a higher difference between them (around 1%). Furthermore, RV-MLP maximum accuracy lies more than 0.5% below CV-MLP lower quartile, meaning that more than 75% of CV-MLP models outperformed the best of the 100 RV-MLP trained models.

Fig. 2 helps to visualize Table 1 results, showing clearly that the minimum RV-MLP result was actually an outlier. It also helps appreciate that CV-MLP merits are statistically justified.

ReLU	CV-MLP	RV-MLP	polar-RV-MLP
median	$\mathbf{90.00 \pm 0.07}$	89.45 ± 0.06	84.93 ± 0.04
mean	$\mathbf{89.92 \pm 0.09}$	89.40 ± 0.13	84.93 ± 0.04
IQR	$89.73 - 90.17$	$89.30 - 89.45$	$84.93 - 85.03$
range	$88.87 - 90.52$	$85.32 - 89.94$	$84.50 - 85.32$
tanh	CV-MLP	RV-MLP	polar-RV-MLP
median	$\mathbf{87.93 \pm 0.08}$	86.43 ± 0.07	81.41 ± 0.05
mean	$\mathbf{87.88 \pm 0.10}$	86.38 ± 0.10	81.30 ± 0.13
IQR	$87.65 - 88.16$	$86.18 - 86.60$	$78.99 - 82.07$
range	$86.84 - 88.56$	$85.47 - 87.07$	$78.99 - 82.07$

Table 1. Validation accuracy results (%)

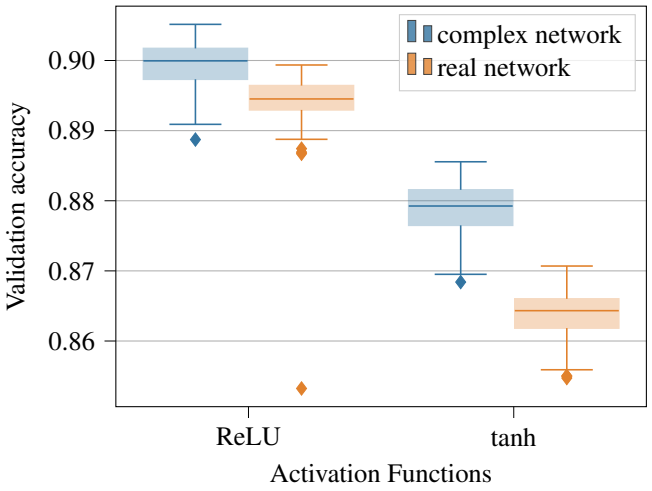


Fig. 2. Box Plot comparison between activation functions on the Oberpfaffenhofen dataset

Fig. 3 plots the overall accuracy and the loss of CV-MLP and RV-MLP on validation set with respect to epochs. As we can see, CV-MLP maintained a higher performance during the ensemble of the training phase. The accuracy and loss seem stabilized on average after 200 epochs and it is logical to assume that this behavior will be maintained regardless of the number of epochs.

Table 2 highlights the accuracy for each class showing that built-up areas were the hardest to classify, whereas open areas could be easily classified in comparison. The distinction between open areas and wood land was highly accurate for both models with a confusion under 1%. CV-MLP outperformed RV-MLP for every class.

Figure 4 shows a randomly selected predicted image from both CV-MLP and RV-MLP. Contrary to the balanced validation dataset, the majority of the test dataset pixels are la-

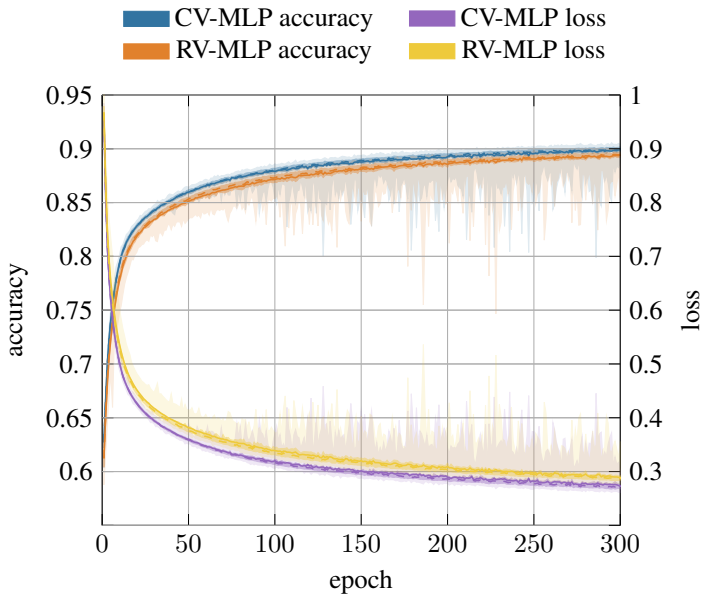
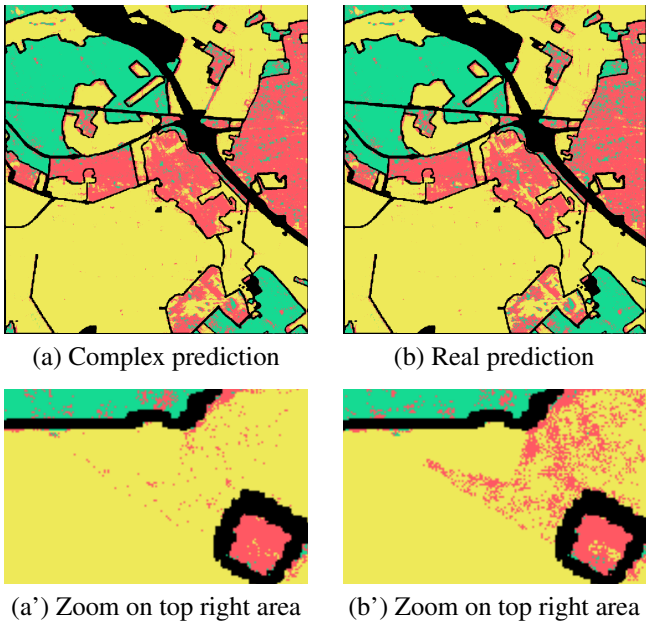


Fig. 3. ReLU validation loss and accuracy per epoch, solid line represents the mean.



(a) Complex prediction (b) Real prediction

(a') Zoom on top right area (b') Zoom on top right area

Fig. 4. Comparison between CV-MLP and RV-MLP predictions
A Built-up Area; **B** Wood Land; **C** Open Area.

	CV-MLP			RV-MLP		
	A	**B**	**C**	**A**	**B**	**C**
A	**81.25**	8.62	10.13	**81.01**	9.19	9.80
B	6.86	**92.83**	0.30	7.33	**92.39**	0.28
C	4.11	0.21	**95.67**	4.97	0.24	**94.79**

Table 2. Confusion matrix of mean accuracy (%) for ReLU activation function

beled as open areas, which presented higher accuracy than the other classes (see Table 2). This meant that the test accuracy was higher for both networks being 91.63% for CV-MLP and 90.91% for RV-MLP.

5. CONCLUSIONS

We proposed a novel definition of an equivalent-RVNN with respect to the CVNN to provide an equitable comparison between them. Despite this equivalence, the classification performance on Oberpfaffenhofen PolInSAR database statistically indicates a slight superiority of CV-MLP over RV-MLP. This merits can be explained by the complex structure of the PolInSAR data in which the phase information matter to enhance the classification accuracy.

6. REFERENCES

[1] S. Chen, H. Wang, F. Xu, and Y.-Q. Jin, "Target classification using the deep convolutional networks for SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4806–4817, 2016.

[2] B. Hou, H. Kou, and L. Jiao, "Classification of polarimetric SAR images using multilayer autoencoders and superpixels," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 7, pp. 3072–3081, 2016.

[3] Y. Zhou, H. Wang, F. Xu, and Y.-Q. Jin, "Polarimetric SAR image classification using deep convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 12, pp. 1935–1939, 2016.

[4] Joshua Bassey, Lijun Qian, and Xianfang Li, "A survey of complex-valued neural networks," *arXiv preprint arXiv:2101.12249*, 2021.

[5] A. Hirose and S. Yoshida, "Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence," *IEEE Transactions on Neural Networks and learning systems*, vol. 23, no. 4, pp. 541–551, 2012.

[6] R. Hänsch and O. Hellwich, "Classification of polarimetric SAR data by complex valued neural networks,"

in *ISPRS workshop high-resolution earth imaging for geospatial information*, 2009, vol. 38, pp. 4–7.

[7] J. Zhao, M. Datcu, Z. Zhang, H. Xiong, and W. Yu, "Contrastive-regulated CNN in the complex domain: A method to learn physical scattering signatures from flexible PolSAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 12, pp. 10116–10135, 2019.

[8] A. Hirose, *Complex-valued neural networks: Advances and applications*, vol. 18, John Wiley & Sons, 2013.

[9] R. Hänsch and O. Hellwich, "Complex-valued convolutional neural networks for object detection in PolSAR data," in *8th European Conference on Synthetic Aperture Radar*. VDE, 2010, pp. 1–4.

[10] S. De, L. Bruzzone, A. Bhattacharya, F. Bovolo, and S. Chaudhuri, "A novel technique based on deep learning and a synthetic target database for classification of urban areas in PolSAR data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 1, pp. 154–170, 2017.

[11] R. Hänsch, "Complex-valued multi-layer perceptrons - an application to polarimetric SAR data," *Photogrammetric Engineering & Remote Sensing*, vol. 76, no. 9, pp. 1081–1088, 2010.

[12] Y. Cao, Y. Wu, P. Zhang, W. Liang, and M. Li, "Pixel-wise PolSAR image classification via a novel complex-valued deep fully convolutional network," *Remote Sensing*, vol. 11, no. 22, pp. 2653, 2019.

[13] Z. Zhang, H. Wang, F. Xu, and Y.-Q. Jin, "Complex-valued convolutional neural network and its application in polarimetric SAR image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 12, pp. 7177–7188, 2017.

[14] Q. Sun, X. Li, L. Li, X. Liu, F. Liu, and L. Jiao, "Semi-supervised complex-valued GAN for polarimetric SAR image classification," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2019)*. IEEE, 2019, pp. 3245–3248.

[15] J. Zhao, M. Datcu, Z. Zhang, H. Xiong, and W. Yu, "Learning physical scattering patterns from PolSAR images by using complex-valued CNN," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2019)*. IEEE, 2019, pp. 10019–10022.

[16] J Agustin Barrachina, "Negu93/cvnn: First official release," Oct. 2020.

[17] A. Hirose, *Complex-valued neural networks*, vol. 400, Springer Science & Business Media, 2012.

[18] A. Hirose, "Complex-valued neural networks: The merits and their origins," in *2009 International Joint Conference on Neural Networks*, 2009, pp. 1237–1244.

[19] N. Mönning and S. Manandhar, "Evaluation of complex-valued neural networks on real-valued classification tasks," *CoRR*, vol. abs/1811.12351, 2018.

[20] Licheng Jiao and Fang Liu, "Wishart deep stacking network for fast polsar image classification," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3273–3286, 2016.

[21] Yanhe Guo, Shuang Wang, Chenqiong Gao, Danrong Shi, Donghui Zhang, and Biao Hou, "Wishart rbm based dbn for polarimetric synthetic radar data classification," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2015, pp. 1841–1844.

[22] J. A. Barrachina, C. Ren, C. Morisseau, G. Vieillard, and J.-P. Ovarlez, "Complex-valued vs. real-valued neural networks for classification perspectives: An example on non-circular data," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 2990–2994.

[23] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[24] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks," *arXiv preprint arXiv:1705.09792*, 2017.

[25] Y. Kuroe, M. Yoshid, and T. Mori, "On activation functions for complex-valued neural networks: existence of energy functions," in *Artificial Neural Networks and Neural Information Processing, ICANN/ICONIP 2003*, pp. 985–992. Springer, 2003.

[26] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016, http://www.deeplearningbook.org.

[27] Jack Kiefer, Jacob Wolfowitz, et al., "Stochastic estimation of the maximum of a regression function," *The Annals of Mathematical Statistics*, vol. 23, no. 3, pp. 462–466, 1952.

[28] R. McGill, J. W Tukey, and W. A. Larsen, "Variations of box plots," *The American Statistician*, vol. 32, no. 1, pp. 12–16, 1978.

[29] J. M. Chambers, *Graphical methods for data analysis*, CRC Press, 2018.