Complex-valued neural networks for fully-temporal micro-Doppler classification

Daniel A Brooks, Olivier Schwander, Frédéric Barbaresco, Jean-Yves Schneider, Matthieu Cord

Complex-valued neural networks for fully-temporal micro-Doppler classification

Daniel A. Brooks*†, Olivier Schwander†, Frédéric Barbaresco*, Jean-Yves Schneider*, Matthieu Cord†

*Thales Air Systems, GBU LAS, Advanced Radar Concepts
Limours, FRANCE
email: daniel.brooks@lip6.fr

†Sorbonne Université, CNRS, LIP6 - Laboratoire d'Informatique de Paris 6, F-75005
Paris, FRANCE

Abstract: *Micro-Doppler analysis commonly makes use of the log-scaled, real-valued spectrogram, and recent work involving deep learning architectures for classification are no exception. Some works in neighboring fields of research directly exploit the raw temporal signal, but do not handle complex numbers, which are inherent to radar IQ signals. In this paper, we propose a complex-valued, fully temporal neural network which simultaneously exploits the raw signal and the spectrogram by introducing a Fourier-like layer suitable to deep architectures. We show improved results under certain conditions on synthetic radar data compared to a real-valued counterpart.*

1. Introduction

In the context of an evermore diverse crowd of various Unmanned Aircraft Vehicles (UAVs), the task of UAV Traffic Management (UTM) is faced with a challenge which conventional surveillance may not hold up well against. The success of future UTM methods may rely upon robust and flexible classifiers. A current trend to this purpose is the development of deep learning methods suited to micro-Doppler [4] classification [9] [3] [11] [5]. While the current art deals with real-valued spectrograms and their variants, this paper proposes to exploit the complex spectrogram in a complex-valued neural network, based on the theoretical developments introduced by [10] in the context of image classification. Our approach differs from the models studied in [10] as it sources from architectures taylored for the heavily structured micro-Doppler radar data. More specifically, we build upon the work in [3] which proposes a fully-temporal convolutional network (FTCN) on spectrograms, and in turn develop a complex-valued counterpart. Our contributions are as follows:

1. A complex-valued, fully-temporal neural network for micro-Doppler signal classification;
2. A Fourier-like convolutional layer suited for deep learning;
3. Extensive experimentation on synthetic radar data validating architectural choices specific to the proposed model.

978-3-7369-9860-5

2. 2D representation of complex numbers

It is at first tempting to handle complex numbers in convolutional neural networks (CNNs) by simply considering a 2-channeled input containing the real and imaginary parts. One should however take care of respecting the inherent structure of complex numbers; both channels are neither independent nor interchangeable. In this section we describe how complex numbers can indeed be handled in a 2-channel fashion given certain constraints and interactions.

2.1. $\mathbb{CR}$ calculus

First, we establish the formal equivalence between complex numbers and 2D real vectors as developed by Wirtinger in 1927 [12] and rediscovered in [2] and [1]. The context of these original developments was the generalization of the complex gradient to non-holomorphic complex functions, to which the proper conceptualisation of complex differentiability is usually limited to. As such, the following equations also establish the complex gradient operators for non-holomorphic functions, which in turn may be used in the backpropagation phase of subsequent neural networks. The key idea equates the Taylor expansions of a function $f\ :\ z \in \mathbb{C} \mapsto y \in \mathbb{R}$ with its counterpart, which by abuse of notation is also noted $f\ :\ (u,v)^T \in \mathbb{R}^2 \mapsto y \in \mathbb{R}$; both functions map to the same scalar y. The Taylor expansion for both forms is written:

$$f(z) \approx f(z_0) + \nabla_z f_{z_0}(z - z0) \tag{1}$$

$$f(u,v) \approx f(u_0,v_0) + \begin{bmatrix}\nabla_u f & \nabla_v f\end{bmatrix}_{(u_0,v_0)} \begin{bmatrix}u_0 \\ v_0\end{bmatrix} \tag{2}$$

We now introduce the 2×2 real-to-complex matrix T:

$$T := \begin{bmatrix}1 & j \\ 1 & -j\end{bmatrix} \text{ such that: } \quad T^H T = 2I = TT^H \quad \text{and} \quad T\begin{bmatrix}u \\ v\end{bmatrix} = \begin{bmatrix}u+jv \\ u-jv\end{bmatrix} = \begin{bmatrix}x \\ x^*\end{bmatrix} := \underline{\mathrm{x}} \tag{3}$$

In the equation above, $\cdot^*$ and $\cdot^H$ denote the complex conjugate and transconjugate. We can now rewrite the first-order moments of the vector function f as a function on complex values:

$$\begin{bmatrix}\nabla_u f & \nabla_v f\end{bmatrix}_{(u_0,v_0)} \begin{bmatrix}u_0 \\ v_0\end{bmatrix} = \left(\frac{1}{2}\begin{bmatrix}\nabla_u f & \nabla_v f\end{bmatrix}_{(u_0,v_0)} T^H\right)\left(T\begin{bmatrix}u_0 \\ v_0\end{bmatrix}\right) := \left(\nabla_{\underline{\mathrm{x}}} f_{\underline{\mathrm{x}}_0}\right)\left(\underline{\mathrm{x}} - \underline{\mathrm{x}}_0\right) \tag{4}$$

In the equation above, we defined the complex gradient:

$$\nabla_{\underline{\mathrm{x}}} f_{\underline{\mathrm{x}}_0} = \begin{bmatrix}\nabla_x f_{x_0} & \nabla_{x^*} f_{x_0}\end{bmatrix} = \begin{bmatrix}\frac{1}{2}(\nabla_u f - j\nabla_v f)_{(u_0,v_0)} & \frac{1}{2}(\nabla_u f + j\nabla_v f)_{(u_0,v_0)}\end{bmatrix} \quad (5)$$

The complex gradient is composed of the complex differential operator ∇_x and the complex conjugate differential operator ∇_{x^*}, which can then be inserted in the backpropagation framework for any neural network using the formal representation of complex numbers as 2D vectors. The following paragraph generalizes that correspondence to the convolution of multi-channel complex inputs with multi-channel complex filter banks.

2.2. Complex convolutional blocks

As such, a complex spectrogram can be represented as a 2-channeled real-valued image, provided we respect the corresponding structure of complex numbers in upcoming calculations. More generally, a C-channeled complex image x is represented as a $2C$-channeled block, the first and second halves respectively containing the real and imaginary part, noted $\Re x$ and $\Im x$. Furthermore, a C-channeled complex convolutional filter bank is represented as a couple of C-channeled real-valued convolutional filter banks, respectively containing the real part and imaginary part. Then, the proper convolutional operation for a complex-valued CNN on an input block $x \in \mathbb{R}^{(2C,H,W)}$ representing the complex z (batch size is omitted for clarity) of $2C$ channels (conceptually, C complex channels) and of dimension $(H*W)$, by C' (real/imaginary) couples of convolutional blocks $\Re W$ and $\Im W$ outputs $x' \in \mathbb{R}^{(N,2C',H',W')}$ representing the complex z' as follows:

$$\forall c' \leq C',\ z'_{c'} = \sum_{c\leq C} z * W \Rightarrow \begin{cases}\Re x'_{c'} = \sum_{c\leq C}\left(\Re x * \Re W - \Im x * \Im W\right)\\ \Im x'_{c'} = \sum_{c\leq C}\left(\Re x * \Im W + \Im x * \Re W\right)\end{cases} \quad (6)$$

By slight abuse of notation, we indifferently refer to $\Re z$ as $\Re x$. We name a network operating on this formalism, and composed of the complex layers described in the section below, a $\mathbb{CR}$Net.

3. Complex layers for neural networks

This section details the complex layers involved in a complex network; specifically, it addresses weight initialization, rectification and batch normalization.

3.1. Weight initialization

Stochasticity being a fundamental property of neural networks, random centered Gaussian weight initialization schemes are standard in any conventional architecture; [7] introduced the

Glorot criterion, which provides a reasonable scaling of the variance insuring layer's outputs and gradients to be of the same order of magnitude. Specifically, the criterion sets $Var[W] = \frac{2}{n_i+n_o}$, where n_i and n_o are the input and output number of channels. In the same line as [10], we initialize the complex convolution weights to a complex Gaussian distribution respecting the Glorot criterion, only in practice we require to know the corresponding distributions of the real and imaginary part to fit the Wirtinger-inspired modeling of the complex network described above. The first step to this computation is the knowledge that the modulus of a complex Gaussian distribution follows a Rayleigh distribution $\mathcal{R}(\cdot\,|\sigma)$, σ denoting the mode. Then, we can express the variance of W in function of the moments of $|W|$ as follows:

$$Var[W] = \mathbb{E}[WW^*] - \underbrace{(\mathbb{E}[W])^2}_{0} = Var[|W|] + (\mathbb{E}[|W|])^2 = \frac{4-\pi}{2}\sigma^2 + (\sigma\sqrt{\frac{\pi}{2}})^2 = 2\sigma^2 \tag{7}$$

Thus, we set $\sigma = \sqrt{\frac{Var[W]}{2}} = \frac{1}{\sqrt{n_i+n_o}}$ and sample $|W|$ from the according Rayleigh distribution. Since the variance of W only depends on $|W|$, the phase θ is uniformly sampled in the periodic space. Finally, we can initialize the convolutional layer as follows:

$$\begin{cases} \Re W = |W|\; cos(\theta) \\ \Im W = |W|\; sin(\theta) \end{cases} \quad \text{, with} \quad \begin{cases} |W| \sim \mathcal{R}(\cdot\,|\frac{1}{\sqrt{n_i+n_o}}) \\ \theta \sim \mathcal{U}_{[-\pi,\pi)} \end{cases} \tag{8}$$

3.2. Rectified linear unit

Three rectification function inspired from the rectified linear unit [13] (ReLU) are explored in [10]: modReLU(z) = ReLU($|z|$ + b) $e^{i\theta_z}$, zReLU(z) = $z\mathbb{1}_{\theta_z\in[0,\frac{\pi}{2})}$, and $\mathbb{C}$ReLU(z) = ReLU($\Re z$) + i ReLU($\Im z$).

Experiments on various tasks proved the $\mathbb{C}$ReLU function to be vastly the most effective, which leads our decision to adopt it as well. Furthermore, its complex form induces the most simple implementation in the double-channel Wirtinger framework:

$$\mathbb{C}\text{ReLU}(x) = \text{ReLU}(x) \tag{9}$$

3.3. Batch normalization

Batch normalization [8] is a widely used regularization technique in modern networks; in essence, its purpose is the batch-wise centering and rescaling of data at each layer during training to reduce the impact of internal covariate shift, ie variations in scale and bias at each layer, which adversely affects training. The normalization of a batch is then followed by a learned re-scaling and re-shifting. The authors in [10] propose a batchnorm process for complex networks, respecting the underlying structure of complex numbers. The main difference with the real-valued case is the normalization by the 2×2 covariance of the real and imaginary parts of the batch elements, instead of the scalar covariance. In summary, the complex form of the algorithm is shown in algorithm 1:

Algorithm 1 Complex form of the batchnorm training scheme

Require: batch $\{x_i\}_{i \leq N}$ of N data points; $(\Gamma, \beta) \in \mathcal{S}_+^*(2) \times \mathbb{R}^2$ trainable parameters

1: $\mu_\mathcal{B} \leftarrow \frac{1}{N} \sum_{i \leq N} x_i$ ▷ batch norm

2: $\Sigma_\mathcal{B} \leftarrow \begin{bmatrix} Cov(\Re x, \Re x) & Cov(\Re x, \Im x) \\ Cov(\Re x, \Im x) & Cov(\Im x, \Im x) \end{bmatrix}$ ▷ batch covariance

3: $\forall i \leq N,\ \bar{x}_i \leftarrow \Sigma_\mathcal{B}^{-\frac{1}{2}} (x_i - \mu_\mathcal{B})$ ▷ batch normalization

4: $\forall i \leq N,\ y_i \leftarrow \Gamma \bar{x}_i + \beta$ ▷ batch re-scaling and re-shifting

5: **return** $\{y_i\}_{i \leq N}$

The inference phase, on the other hand, does not use batch-wise statistics, but rather dataset statistics. The most standard estimate of the latter are running mean and variance $\mu_\mathcal{D}$ and $\Sigma_\mathcal{D}$ with momentum α usually set to 0.9. $\mu_\mathcal{D}$ and $\Sigma_\mathcal{D}$ are initialized at 0 and $\frac{1}{\sqrt{2}} I_2$.

The rescaling parameter Γ is no longer scalar, but a learnt covariance matrix of the space of 2×2 symmetric positive definite matrices $\mathcal{S}_+^*(2)$, parameterized with three scalars $(\Gamma_{11}, \Gamma_{12}, \Gamma_{22})$ which are individually updated during training.

4. Experimental validation

In this section we validate the usage of a complex network rather than a real one. First we explore which architectural strategies and which data configuration seem to benefit from exploiting complex values, then give results on synthetic radar data.

4.1. Model exploration

The main goal of model exploration in the context of $\mathbb{CR}$Nets is the comparison with a real-valued conterpart. In practice, we use the fully-temporal convolutional neural network (FTCN) introduced in [3], and simply replace the real-valued layers with the complex ones while exploring the additional degrees of freedom and uncertainties induced by the complex nature of the model.

Number of parameters Intrinsicly, a complex network will have twice as many parameters as its real counterpart; in practice, it is not obvious how this increase would affect performance. For instance, neural networks tend to better generalize in the case of a large dataset when allowed

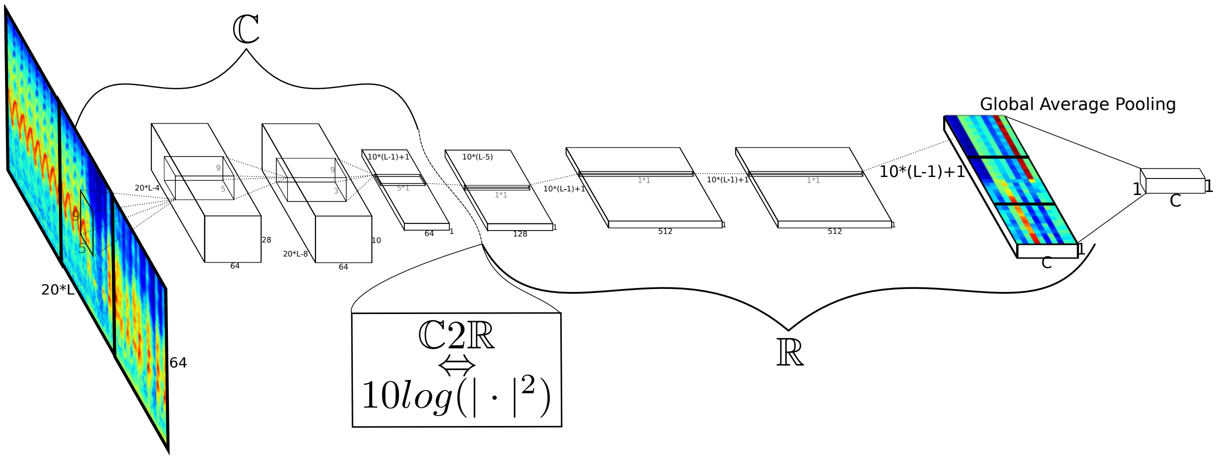

Figure 1: Illustration of the proposed partially complex ℂℝNet architecture. The first 1D convolutional layer is omitted, as well as activations and normalization layers, and the real-valued spectrogram is shown for visual clarity, although the complex spectrogram serves as true input.

more parameters, but may also suffer from overfitting when a sufficient amount of diversified data is not met. A reasonable way to experiment on this interrogation is to allow half as many channels in the ℂℝNet'S convolutional blocks and focussing experiments on small amounts of data. Results show that keeping the same number of channels as in the real network still performs better, which is a conclusive statement as, while the practical number of parameters has doubled, the network did not suffer even when presented with few data. As a sanity check, doubling the number of channels in the complex network performs the worst of all cases.

Signal scaling and complex representation Raw radar data, along with their Fourier transforms, often exhibit major variations in scale, due to different intervening physical phenomena operating in a variety of scales. This translates to the practical habit of converting spectrograms to a logarithmic scale, most often decibels, whether it be for visualization or further analysis. A real-valued network benefits from this rescaling from the start as the inputs are the decibel-spectrograms. In the ℂℝNet however, the log-scale is ambiguously defined for complex values, which allows potentially harmful variations in scale to propagate within. Proper weight initialization and batchnorm explicitly combat this issue, but formally fail to recover a log-scale as they remain linear transformations. To this end, we propose a partially complex network for which the output complex representation is log-scaled after the passage to absolute value, and heuristically study the impact on performance of the complex-to-real ($\mathbb{C}2\mathbb{R}(x) = 10log(|x|^2|)$) function's position in the layer hierarchy. The conclusion is conceptually satisfying as it places the $\mathbb{C}2\mathbb{R}$ right after the final temporal representation layer, ie right before the convolutionalized fully-connected layers [6], as represented in figure 1; in practice the ante-penultimate convolutional layer of the network proposed in [3]. This result leads to a rather natural interpretation: while the complex spectral representation of the signal in a real-valued network stops at the Fourier transform, the latter in a ℂℝNet explores a hierarchy of further filter banks in addition

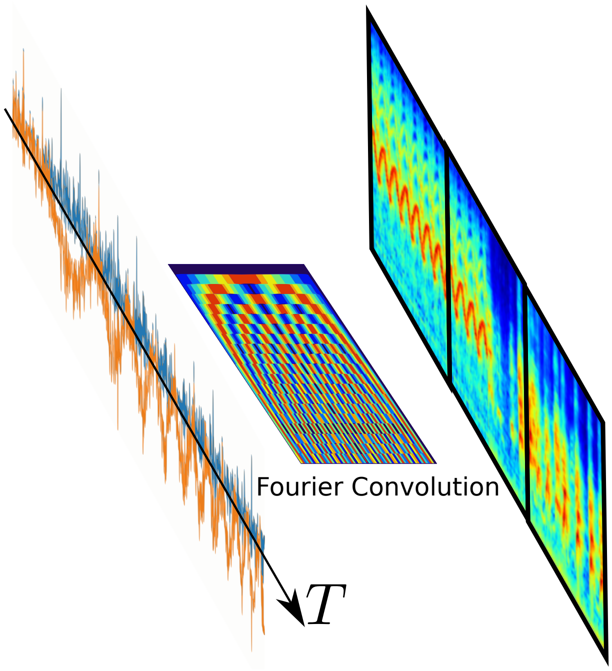

Figure 2: Illustration of the proposed Fourier-like convolutional layer. The Fourier atoms are represented as sine waves of increasing frequency.

to the Fourier filtering.

Fourier convolution parameters The first layer of a CNN on spectrograms is conceptually preceded by a windowed Fourier transform, which remains a fixed pre-processing. The $\mathbb{CR}$Net however directly handles the raw complex data, and as such, its first layer is a 1D convolution. While conventional initialization schemes such as [7] can be applied, we may benefit from exploiting the spectral properties of the radar signal. Indeed, since the Fourier transform is essentially a convolution, we can initialize the filter bank weights to the n Fourier atoms $(e^{-2i\pi k\frac{\cdot}{n}})_{k\leq n}$, where n represents the windowing applied to the signal, and corresponds to the 1D filter size; such a layer is represented in figure 2. Experiments show a consistent improvement when using such a Fourier-like convolutional layer. Similarly, the window overlap percentage or hop length of the Fourier transform corresponds to the convolution stride. In the context of learning the 1D filter banks, a low stride (set to 1 in the experiments, ie maximum overlap) proved paramount to the network's performance, regardless of initialization. On the other hand, real-valued counterparts seemed much more robust to this hyperparameter. One interpretation of this phenomenon is that the passage from raw complex data to real-valued spectrograms averages through coherent integration any potential added information from a higher overlap, while keeping both amplitude and phase sensitizes further processing to this added information. We call FourierNet a $\mathbb{CR}$ whose first convolutional layer is initialized with the Fourier atoms.

Table 1: Performance comparison of complex and real networks on radar data on various amount of noisy data.

Train size	100%	20%	5%
TFCN	67.2 ± 0.27	65.1 ± 0.39	$\mathbf{63.5 \pm 0.46}$
$\mathbb{CR}$Net	68.8 ± 0.17	65.1 ± 0.50	59.3 ± 1.51
FourierNet	$\mathbf{70.8 \pm 0.22}$	$\mathbf{67.8 \pm 0.40}$	62.1 ± 0.90

Quality and amount of data Throughout conducted experiments, a general trend seemed to emerge: complex networks overpowered real networks when presented with a large yet complicated dataset. Specifically, we observed improvement for SNRs on the IQ data close to zero or in the negatives, positive SNRs leading to insignificant improvements. Furthermore, when the amount of training data was kept relatively small (in our scenario, less than 5 minutes), $\mathbb{CR}$Nets performed poorly to worse than their real-valued counterpart.

4.2. Results

In this section we show experimental results on synthetic radar data, issued by the simulator introduced in [3]. Approximately 20 minutes of signal are generated for each of 3 different classes of drones; signals are passed through the models 35ms at a time. The simulation configurations are set to an extremely noisy case, where the raw data is $5dB$ below noise ($SNR = -5dB$). For reference, a coherent integration of 20 timesteps (which corresponds to the filter size of the first convolutional layer) would bring the spectrum $8dB$ above noise. A PRF of $4kHz$ is used; at this frequency and with the considered drones, Doppler ambiguity is omnipresent. As stated above, we voluntarily chose a large amount of data in a very challenging configuration. We also give performance results for the models when trained on a fraction of the data to quantify the robustness of the models to lack of data. All models are run in a 5-fold cross-validation, holding out 50% of data for validation, using standard gradient descent. Three models are put to testing: the real-valued TFCN, the corresponding $\mathbb{CR}$Net and the equivalent FourierNet. The architectural choices described above are included in the complex models. Results are presented in table 1.

The first observation is the improvement of the two complex networks over the real counterpart when given all 10 minutes of training data (the 50% training split of the total 20 minutes), the FourierNet being superior to the $\mathbb{CR}$Net. Given 20% of available training data (2 minutes), the FourierNet still outperforms all models, but the $\mathbb{CR}$Net starts decreasing towards the TFCN's performance. Given only 5% of training data (30 seconds), all complex models begin to perform worse than the TFCN. Finally, we repeat the experiments on a cleaner dataset, by changing the SNR from $-5dB$ to $5dB$ (we limit ourselves to the FTCN and FourierNet). Results shown in table 2 naturally exhibit better performances overall, but the FourierNet struggles to outperform the FTCN, which supports the argument of complex networks working noticeably better in challenging configurations only.

Table 2: Performance comparison of complex and real networks on radar data on various amount of less noisy data.

Train size	100%	20%	5%
TFCN	98.6 ± 0.34	94.3 ± 0.57	$\mathbf{91.6 \pm 0.98}$
FourierNet	$\mathbf{99.0 \pm 0.07}$	$\mathbf{94.4 \pm 0.14}$	88.7 ± 1.12

5. Conclusion

In conclusion, we have developed a fully-temporal, partially-complex convolutional neural network combining previous works on complex-valued neural networks on the one hand, and fully-temporal networks for radar classification on the other hand. We have furthermore introduced a Fourier-like convolutional layer, which harvests the advantages of both the Fourier transform and of learning filter banks on the raw data, an intuition proved to be consistently true in practice. We performed extensive experimentation on synthetic data to isolate the cases where performance benefitted from complex values. The main conclusions obtained were, that above a certain amount of observed data (a couple of minutes for our datasets), in challenging configurations (under $5dB$ of SNR in our scenarios), complex-valued networks significantly outperformed their real counterparts. These results initiate a hopeful stance on introducing complex values in deep learning-based classification methods on micro-Doppler radar data.

References

[1] A. v. d. Bos. Complex gradient and Hessian. *IEE Proceedings - Vision, Image and Signal Processing*, 141(6):380–383, Dec. 1994.

[2] D. H. Brandwood. A complex gradient operator and its application in adaptive array theory. *IEE Proceedings H - Microwaves, Optics and Antennas*, 130(1):11–16, Feb. 1983.

[3] D. A. Brooks, O. Schwander, F. Barbaresco, J. Schneider, and M. Cord. Temporal Deep Learning for Drone Micro-Doppler Classification. In *2018 19th International Radar Symposium (IRS)*, pages 1–10, June 2018.

[4] V. C. Chen, F. Li, S.-S. Ho, and H. Wechsler. Micro-Doppler effect in radar: phenomenon, model, and simulation study. *IEEE Transactions on Aerospace and electronic systems*, 42(1):2–21, 2006.

[5] J. J. M. De Wit, R. I. A. Harmanny, and P. Molchanov. Radar micro-Doppler feature extraction using the singular value decomposition. In *Radar Conference (Radar), 2014 International*, pages 1–6. IEEE, 2014.

[6] R. Girshick. Fast R-CNN. *arXiv:1504.08083 [cs]*, Apr. 2015. arXiv: 1504.08083.

[7] 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*, pages 249–256, Mar. 2010.

[8] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167 [cs]*, Feb. 2015. arXiv: 1502.03167.

[9] P. Molchanov, R. I. Harmanny, J. J. de Wit, K. Egiazarian, and J. Astola. Classification of small UAVs and birds by micro-Doppler signatures. *International Journal of Microwave and Wireless Technologies*, 6(3-4):435–444, June 2014.

[10] 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:1705.09792 [cs]*, May 2017. arXiv: 1705.09792.

[11] R. P. Trommel, R. I. A. Harmanny, L. Cifola, and J. N. Driessen. Multi-target human gait classification using deep convolutional neural networks on micro-doppler spectrograms. In *2016 European Radar Conference (EuRAD)*, pages 81–84, Oct. 2016.

[12] W. Wirtinger. Zur formalen Theorie der Funktionen von mehr komplexen Vernderlichen. *Mathematische Annalen*, 97(1):357–375, Dec. 1927.

[13] B. Xu, N. Wang, T. Chen, and M. Li. Empirical Evaluation of Rectified Activations in Convolutional Network. *arXiv:1505.00853 [cs, stat]*, May 2015. arXiv: 1505.00853.