Complex-Valued Neural Networks for Millimeter Wave FMCW-Radar Angle Estimations

Kevin Kaiser[1,4], Jonas Daugalas[2,4], Javier López-Randulfe[2], Alois Knoll[2], Robert Weigel[1], Fabian Lurz[3]
[1]Institute for Electronics Engineering, Friedrich-Alexander University Erlangen-Nuremberg, Erlangen, Germany
[2]Department of Informatics, Technical University of Munich, Munich, Germany
[3]Institute of High-Frequency Technology, Hamburg University of Technology, Hamburg, Germany
[4]Infineon Technologies, Munich, Germany
kevin.kaiser@fau.de

***Abstract* — Processing radar signals with neural networks has shown promising results in classification and regression tasks. While processed radar data is intrinsically complex-valued, most architectures using neural networks are comprised of real-values and their arithmetic. Previous work has found that keeping the complex-valued number system and extending it into the domain of neural networks can be beneficial. In this paper, we demonstrate that in two-dimensional direction-of-arrival (DoA) estimation, complex-valued neural networks (CVNNs) show better results than real-valued neural networks (RVNNs). Real-world recordings of ten different FMCW radar devices were used to train numerous models, varying in the computational complexity and varying in data properties. Over all models trained, the best CVNN surpassed the best RVNN by 14%. In terms of model complexity, CVNNs also showed better results, both per trainable parameter and per floating point operation (FLOP). Similarly, CVNNs surpass RVNNs, both when trained with decreased data quantity and decreased data quality.**

***Keywords* — neural networks, machine learning, computational complexity, millimeter wave radar.**

I. Introduction

The majority of fields using machine learning rely on data which is real-valued, the most prominent example being RGB images. This is not surprising, as many sensors and their hardware sample the world with real values. However, the signals sampled by various sensors frequently include wave phenomena, which are often resolved via transforms like the discrete Fourier transform or wavelet transforms. With these signal processing tools, important features of the sampled waves such as frequency, amplitude and phase can easily be extracted in a complex-valued representation.

Complex-valued neural networks (CVNNs) keep these quantities of phase and amplitude together by extending this number system to neural networks, as introduced in [1], [2]. With the number system of complex-values, CVNNs have been shown to be good with a wide variety of wave phenomena. The addition of prior knowledge about the data reduces the degrees of freedom of networks, which can result in a more meaningful generalization [3]. Similarly, the concept of locality in convolutions also reduces the degrees of freedom when compared to fully-connected layers.

In the case of frequency modulated continuous wave (FMCW) radar, the relative relationships of the properties of phase and magnitude throughout multiple antennas can be interpreted as metrics of position, velocity and angle of a target in reference to the sensor. While other applications of CVNNs have started earlier, first works with radar appeared in 2003 [4] and first works with generic direction-of-arrival (DoA) appeared in 1994 [5]. Although some recent publications working on FMCW radar neural networks have emphasized the concept of phase in data [6], a common approach is to input complex-valued data into real-valued neural networks (RVNNs), e.g. by simply concatenating or interleaving real and imaginary part in the channel dimension [7], [8].

In recent history, CVNNs have conquered a variety of different applications in radar, including imaging [9],[10], automotive scene classification [11], ego-velocity estimation [12], signal denoising [13] and human activity classification [14]. In these fields, some publications have shown better results for CVNNs with fewer parameters [9], fewer training data [13],[14] and lower signal-to-noise ratios (SNR) [14] than their RVNN counterparts. The performance in reference to parameter count, floating point operations (FLOPs), amount of training data and SNR are all important metrics, especially when targeting an embedded platform. In embedded scenarios, reduced computational and memory overhead can reduce size, energy consumption and cost of a microcontroller. Analogously, achieving higher performance for identical SNRs can allow for economical radar hardware to be able to meet the demands of more use-cases. While the need for less training data is not only interesting for embedded sensors, reducing the need for large amounts of training data can reduce the barrier of entry for new use-cases.

To implement the networks in this work, the TensorFlow based library of [15] was used, which showed higher accuracy, smaller variance and less overfitting of CVNNs for simulated data. In this paper, the two-dimensional DoA estimation performance of CVNNs and RVNNs is compared with real-world FMCW radar data of a corner target. In this domain, we show that CVNNs outperform RVNNs when trained under variations of the following characteristics:

Complexity of models:
- Trainable parameters
- FLOPs

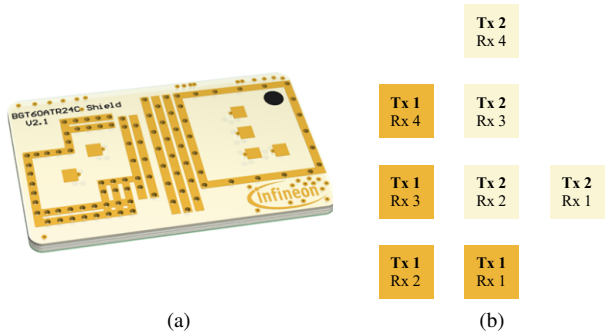
Reduction of Data:
- Quality
- Quantity

II. Dataset and Radar System

A. Radar System

The two-dimensional DoA estimation was carried out with Infineon's 60 GHz FMCW radar sensor BGT60ATR24C. This device has two transmit (Tx) and four receive (Rx) antennas. The device used for recording, along with its virtual array configuration, is depicted in Fig. 1.

With this antenna configuration, each transmit antenna emits sequentially in time, also called time-division multiplexing. The signal resulting from mixing the received with the original chirp is digitized by an analog-to-digital converter (ADC) for each combination of transmit and receive antenna. This signal is output in a 4D array of dimensions F x C x D x R, where F is the index of the frame captured, C the index of the virtual array antennas, D the index in slow-time and R the index in fast time.



(a) (b)

Fig. 1. Infineon BGT60ATR24C (a) an image of the antenna shield; (b) the virtual antenna array

B. Corner Target Measurement Setup

The data for DoA estimation was captured in an anechoic chamber, with a target at a fixed position. In a distance of 1.4 m, a total of ten different radar devices were fixed to a pan-and-tilt gimbal as depicted in Fig. 2. For each device, different positions in azimuth and elevation were traversed in a grid like pattern. At the origin of both angles, the corner target is perpendicular to the front of the radar devices. The entire grid spans from -50° to 50° in 5° steps in both azimuth and elevation angles. For each position, multiple frames were acquired. Because the position in both angles is known during acquisition, the labels for later DoA estimation can be saved.

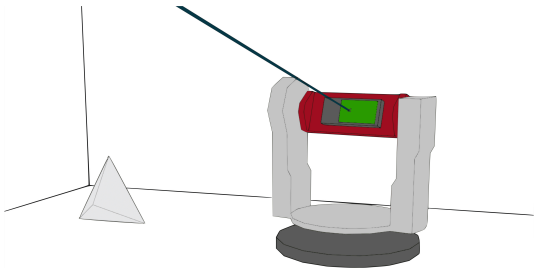


Fig. 2. Recording setup with radar sensor fixed to pan-and-tilt gimbal, to record a corner target at fixed distance from multiple angles of azimuth and elevation.

III. Baseline Network Architectures

A. Architecture and Hyperparameters

Between CVNNs and RVNNs, a comparable base architecture was derived. Each architecture consists of multiple hyperparameters and is separated into two parts, feature extraction and regression. The feature extraction is composed of a variable number of Conv2D layers, *Conv2D_1* to *Conv2D_5*. These are sequentially added to the network, until the value of *num_conv_layers* is reached. Each of these convolutional layers also have a variable number of filters, as listed in Table 1. Each layer is followed by an activation function, which is fixed model-wide with the hyperparameter *act_func*. To limit the amount of hyperparameters in the final study, the activation functions were fixed to *ReLU* for RVNNs and *cart_ReLU* for CVNNs after empirical analysis showed no improvements with other activation functions.

In the second half of the model, dimensionality is reduced and angles are regressed. First, reduction is achieved by the means of a Conv2D Layer with a singular output filter, activated by *act_func*. The output thereof is flattened and followed by a fully connected layer with a variable number of neurons *num_dense*. After *act_func* is applied, the final layer of the network is another fully-connected layer with two neurons as output.

Table 1. Hyperparameters for models of RVNNs and CVNNs to be trained for DoA estimation. The Conv2D based layers are added sequentially, until the selected number of *num_conv_layers* is reached. Activation functions *act_func* are applied after every layer.

Hyperparameter	**RVNN**	**CVNN**
act_func	*cart_ReLU*	*ReLU*
num_conv_layers	[1,2,3,4,5]	
Conv2D_1	[12,16,20]	[8,10,12]
Conv2D_2	[6,8,10]	[6,8]
Conv2D_3	[6,8]	[6,8]
Conv2D_4	[6]	[4]
Conv2D_5	[4]	[4,2]
num_dense	[6,8,12,16]	[6,8,12]

While the base architecture is similar, some functionality had to be adapted between CVNNs and RVNNs. While both types can be normalized per frame, the normalization factor for CVNNs is calculated by the magnitude of the complex-values. Similar layers, such as a convolution for RVNNs and a complex convolution for CVNNs, have a different amount of total trainable parameters per filter of each layer. To overcome this difference, the hyperparameters were empirically adapted to approximately match each other in total trainable parameter count. The output angles of the network are determined in two distinctive ways. For RVNNs, the values of the last layers were used while in the case of CVNNs, the phase of the two output layers was used.

B. Training Procedure

For a comparison between RVNNs and CVNNs, a random subsampling of all possible combinations of parameters

was carried out. The resulting models of these parameter combinations were trained until the validation loss stopped improving. To decrease resource consumption, parameters such as learning rate or batch size were fixed after empirical evaluation.

The regression loss during training was calculated with the mean-squared-error (MSE) of all angle estimations of a batch. After training, other metrics of mean-absolute-error (MAE) were calculated. In the spherical coordinates of this recording setup, elevation angles converge at the poles with increasing value. To overcome artifacts of the coordinate system at higher angles and to have a more meaningful metric, both angles of azimuth and elevation were converted to a single, direct angle between labels and predictions.

C. Data Pipeline

Before the dataset is passed to the network, the raw radar data was pre-processed. First, a DC offset was removed, and a Fourier transform in both slow- and fast-time was applied. The resulting range-doppler-map (RDM) was both calculated with and without a previous Blackman–Harris window function, for separate models respectively. After traditional radar signal processing, a complex-valued, additive white Gaussian noise (AWGN) was added dynamically during training for all networks to reduce overfitting. The dataset was split into training, validation and test subsets along unique radar devices. The total of ten devices were thereby split into six for training, two for validation and two for testing. Because all the following evaluations rely on the same processed data, the already processed subsets were loaded during training. While the evaluation of model complexity required the calculation of FLOPs and parameters, the subsequent evaluations of data quantity and quality required two additional transforms to the dataset. As these dynamic transforms of the data introduce additional hyperparameters, a subset of the network parameters was chosen to save computational resources.

1) Reduction in Dataset Training Size

For an accurate comparison, both CVNNs and RVNNs were trained on the same subsets of the training dataset. We chose to compare the entire training dataset to five different percentages of itself: 10%, 20%, 33%, 50% and 75%. For evaluation, however, the entire validation dataset was used.

2) Reduction in Dataset Signal Quality

As above, the base architecture was retained, while the input training data was modified. The reduction of signal quality was achieved by re-using the same data from before, but adding noise dynamically. This dynamic transform avoids overfitting to static noise. To avoid the significant computational overhead of pre-processing during training, noise was added after this step. With no windowing present, a SNR can be calculated traditionally. With the signal of a sample known, an additional AWGN term can be calculated to create a predefined SNR. In the case of the windowed data, additional caution is necessary. If a window function is applied in time domain, the resulting signal in a RDM is spread over multiple bins. In this case, only a modified SNR can be calculated, in which the peak of the windowed signal divided by the new noise: $SNR_{modified} = Signal_{Windowed}/Noise$. This joins both the processing friendly noise addition after pre-processing with the beneficial presence of windowing

IV. Experiments and Results

In the following, the performance of all fully trained neural network models of CVNNs and RVNNs were evaluated in terms of the metric of MAE. The first part compares MAE in reference to the number of trainable parameters and FLOPs of various models. The second part of this section is based on a subset of models. There, the MAE of different architectures are compared when trained on fewer training data and with lower signal quality data.

A. Complexity of Models

All fully trained models are compared along the number of trainable parameters and FLOPs of all layers within the network. These values are plotted against the MAE in Fig. 3 (a) for trainable parameters and (b) for FLOPs. For easier evaluation, the cumulative minimum over all models of each network type is visualized as a step plot. The best performing models of CVNNs seem to have significantly lower values of MAE than RVNNs over the entire range of parameter counts. From all trained models, the best CVNN outperformed the best RVNN by 14%. The difference of the best MAE between architectures decreases with an increasing number of parameters. This agrees with the findings of [9], which showed

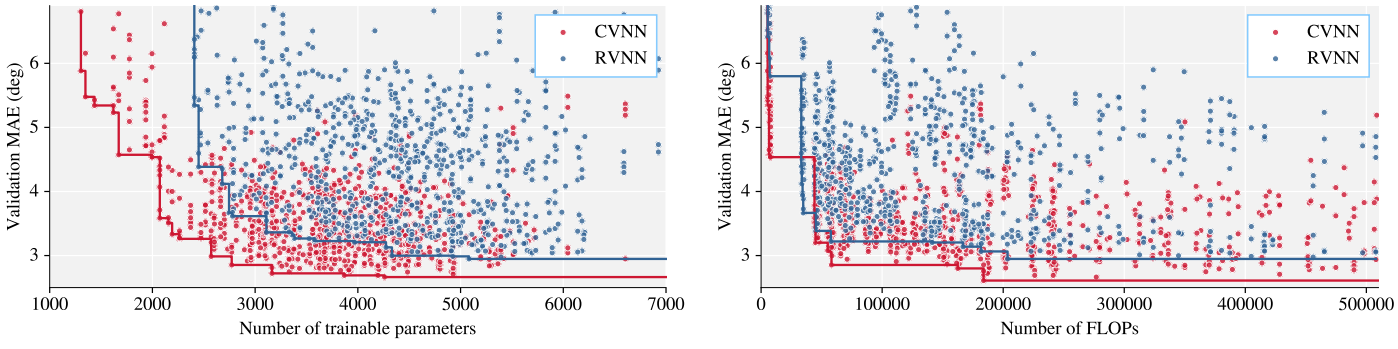


Fig. 3. Fully-trained models for DoA estimation with a wide variety of hyperparameters. Model complexity comparison of fully trained CVNN (red) and RVNN (blue) models plotted with MAE against the number of: (a) trainable parameters; (b) FLOPs. The solid lines depict the cumulative minimum per architecture.

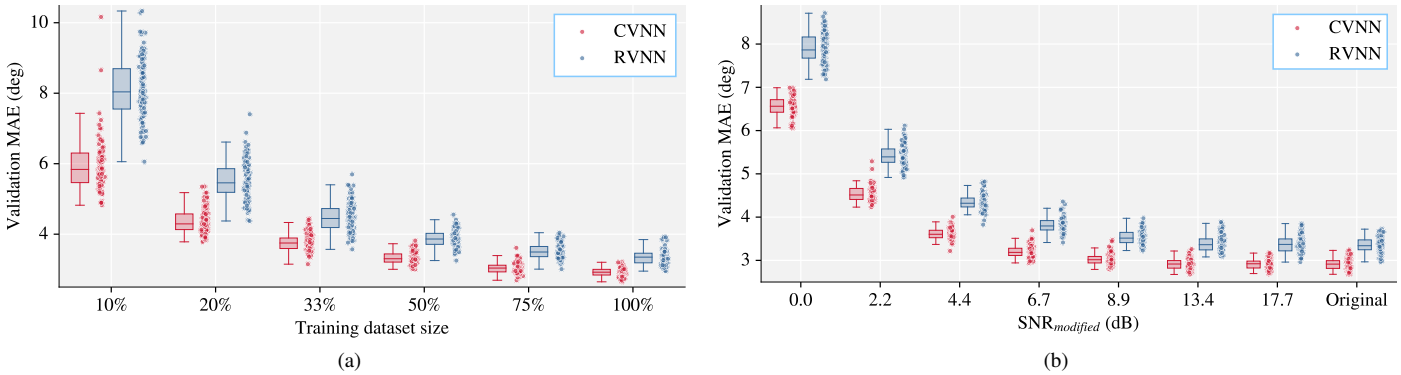


Fig. 4. Fully-trained models for DoA estimation with a wide variety of hyperparameters. Data reduction comparison of CVNN (red) and RVNN (blue) models with MAE against: (a) training dataset size; (b) $SNR_{modified}$. All models of a reduction step are plotted as box plots (left) and their respective values (right).

better radar imaging results with fewer parameters. When the same models are evaluated against the number of FLOPs, a similar, but slightly weaker trend of CVNNs showing better values of MAE is visible. This smaller gap can be attested to analogous layers between architectures being comprised of more operations in CVNNs.

B. Reduction of Data

When a subset of the previous models was trained with altered training data, CVNNs again show better ability of DoA estimation. The values of reduced training data and reduced $SNR_{modified}$ are plotted against MAE in Fig. 4 (a) and (b) respectively. In both metrics, CVNNs outperform their real-valued counterparts not only in the best performing models, but also in the mean of their distributions. When looking at the best models of each architecture per training data reduction increment, CVNNs outperform RVNNs in our scenario by 8-26% overall. In the same manner, the best models per signal quality reduction step show an improvement of 17-22% over RVNNs. Another remarkable behavior is the concise distribution of CVNNs compared to the wider distributions in the real-valued case.

V. Conclusion

This paper investigated the use of CVNNs for DoA estimation for FMCW radar systems. We showed CVNNs to be superior to RVNNs when evaluated with data from ten unique radar devices. When compared under both model complexity (trainable parameters and FLOPs) and reduced training data (quantity and quality), CVNNs showed better results than RVNNs. In the case of the best of all models trained, CVNNs show lower estimation error than RVNNs, exceeding them in performance by 14%. Surprisingly, CVNNs show better performance per FLOP than RVNNs most of the time, even though analogous layers require more operations for CVNNs.

The presented CVNN DoA estimation is well suited for embedded radar, as it is capable of working with lower quality signals and fewer computational resources for edge computing.

References

[1] A. Hirose, "Dynamics of fully complex-valued neural networks," *Electronics letters*, vol. 28, no. 16, pp. 1492–1494, 1992.

[2] 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.

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

[4] A. B. Suksmono and A. Hirose, "Adaptive interferometric radar image processing by using complex-valued neural network," in *Complex-Valued Neural Networks: Theories and Applications*. World Scientific, 2003, pp. 277–301.

[5] W.-H. Yang, K.-K. Chan, and P.-R. Chang, "Complex-valued neural network for direction of arrival estimation," *Electronics Letters*, vol. 30, no. 7, pp. 574–575, 1994.

[6] E. Hayashi, J. Lien, N. Gillian, L. Giusti, D. Weber, J. Yamanaka, L. Bedal, and I. Poupyrev, "Radarnet: Efficient gesture recognition technique utilizing a miniature radar sensor," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–14.

[7] J. Fuchs, M. Gardill, M. Lübke, A. Dubey, and F. Lurz, "A machine learning perspective on automotive radar direction of arrival estimation," *IEEE Access*, 2022.

[8] D. Brodeski, I. Bilik, and R. Giryes, "Deep radar detector," in *2019 IEEE Radar Conference (RadarConf)*. IEEE, 2019, pp. 1–6.

[9] J. Gao, B. Deng, Y. Qin, H. Wang, and X. Li, "Enhanced radar imaging using a complex-valued convolutional neural network," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 1, pp. 35–39, 2018.

[10] H. Jing, S. Li, K. Miao, S. Wang, X. Cui, G. Zhao, and H. Sun, "Enhanced millimeter-wave 3-d imaging via complex-valued fully convolutional neural network," *Electronics*, vol. 11, no. 1, p. 147, 2022.

[11] M. Meyer, G. Kuschk, and S. Tomforde, "Complex-valued convolutional neural networks for automotive scene classification based on range-beam-doppler tensors," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–6.

[12] H.-W. Cho, S. Choi, Y.-R. Cho, and J. Kim, "Complex-valued channel attention and application in ego-velocity estimation with automotive radar," *IEEE Access*, vol. 9, pp. 17 717–17 727, 2021.

[13] A. Fuchs, J. Rock, M. Toth, P. Meissner, and F. Pernkopf, "Complex-valued convolutional neural networks for enhanced radar signal denoising and interference mitigation," in *2021 IEEE Radar Conference (RadarConf21)*. IEEE, 2021, pp. 1–6.

[14] X. Yao, X. Shi, and F. Zhou, "Human activities classification based on complex-value convolutional neural network," *IEEE Sensors Journal*, vol. 20, no. 13, pp. 7169–7180, 2020.

[15] 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)*. IEEE, 2021, pp. 2990–2994.