Deep Complex-Valued Neural Networks for Massive MIMO Signal Detection

Isayiyas Nigatu Tiba[1(✉)] and Mao youhong[2]

[1] Jimma University Institute of Technology, 378, Jimma, Ethiopia
isayiyas.tiba@ju.edu.et
[2] Xidian University, Xi'an, People's Republic of China

Abstract. In the fifth generation (5G) and future mobile networks, the design of efficient detectors for massive multiple-input multiple-output (MIMO) is essential. The main challenge in designing detectors is the trade-off between performance and computational complexity; that is, efficient detectors incur higher computational costs, while computationally cheaper detectors have lower efficiency. Recently, many deep learning-based detectors have been proposed in the literature to fill in such gaps. However, most of the existing MIMO detectors work only with real-valued parameters. First, they transform the complex received MIMO signal into an equivalent real-valued parameter by concatenating the real and imaginary parts and then train a network based on the real-valued data. Such an approach has several disadvantages. On one hand, the number of trainable parameters will be doubled; on the other hand, the phase information, which is important in the communication signals, might be lost or distorted. In this work, we aim to investigate the application of complex-valued neural networks for MIMO signal detection based on Wirtinger Calculus. To do so, we propose a simple feedforward architecture that directly works with the complex-valued QPSK and 16-QAM modulation signals. Our method is simple and computationally cheaper. Simulation results show that the proposed approach can improve the performance of the existing detectors while providing a lower computational cost.

Keywords: Complex-valued neural networks · MIMO detector · Wirtinger calculus

1 Introduction

Massive multiple-input multiple-output (MIMO) is a key technology for 5G and future mobile networks. One of the issues that must be addressed in order to make the promising benefits of this technology a reality is signal detection. Signal detection is the process of recovering information that has been transmitted through a noisy channel in a wireless communication system [1]. Due to interference, noise, and fading channel conditions, detection becomes increasingly

B. H. Woldegiorgis et al. (Eds.): ICAST 2022, LNICST 455, pp. 239–251, 2023.
https://doi.org/10.1007/978-3-031-28725-1_15

difficult in massive MIMO systems. Furthermore, because of the high number of antennas, computational complexity is one of the key challenges in this technology [2,3].

As a result, different categories of traditional detectors are proposed in the literature to realize the promises of massive MIMO technology [4–6]. Besides the traditional ones, neural network-based detectors are also a promising class of MIMO detectors [7–9]. Deep neural network (DNN) detectors can operate in data-driven, model-driven, or hybrid modes. The performance of these detectors has shown a significant improvement compared with their traditional counterparts. However, one of these detectors' limitations is that they only work with real-valued parameters. Most existing DNN-based applications are tuned to work with real-valued parameters in order to take advantage of the computing resources available in the existing DNN libraries [10].

One of the key reasons behind the success of supervised deep learning (DL) in several tasks is its ability to learn a useful hierarchical representation of data [11]. Eventually, it has been used in difficult scenarios where tractable mathematical models cannot characterize a problem. In such cases, the DL architecture is represented by a black box and optimized through learning from data to solve a specific problem. In various domains of applications, such as computer vision, such a representation has surpassed human-level performance [12]. Since the nature-made signals (RGB images, videos, ...) appear as real-valued signals, most DNN libraries are built to process real-valued signals.

When it comes to engineering applications, including wireless communication, the situation is rather different. On the one hand, the information signals are synthetically generated by humans. Hence, applying the DL to these signals (man-made) doesn't guarantee the best performance as that of nature-made signals [13]. On the other hand, unlike most nature-made signals, the communication signals are represented in a complex baseband form, in which the phase information is as important as the magnitude. As such, it is critical to resolve these issues in order to effectively use the DL in wireless communications. In the first case, the work in [13] has analyzed the applications of DL in the physical layer of communication systems. In that work, it is shown that DL can be efficiently applied to signal detection tasks. Nevertheless, the dataset used for the analysis was transformed from complex to real.

In this work, we address the second issue by investigating the use of complex-valued neural networks for MIMO signal detection. The use of complex numbers when working with communication signals allows for precise representation of both magnitude and phase [14]. The main challenge in complex analysis, however, is the diffentiability problem, which is described by the Cauchy-Riemann equations [15]; i.e., in the DL, the partial derivatives of a real-valued cost function with respect to the complex-valued parameters in the back-propagation do not exist. As such, we resort to Wirtinger Calculus, which allows us to fully exploit the power of complex-valued signal processing [16]. To the best of our knowledge, this is the first study to apply complex-valued neural networks to the MIMO detection problem.

Related Works: Several DNN-based detectors for massive MIMO have been proposed [17]. We chose two detectors with superior performance in [7] and [9], for ease of generation of training data. However, because of the aforementioned complex-analysis issues, the data in these detectors must first be transformed to the real-valued equivalent. This means that an n-dimensional complex-valued vector will be converted to a $2n$-dimensional real-valued vector: i.e., $\mathbb{C}^n \mapsto \mathbb{R}^{2n}$. As a result, in those works, the network parameters are fully trained using real-valued calculus and the output is converted back to complex. This method has two major drawbacks: first, besides increased dimensionality (doubling the number of trainable parameters), it may be ineffective for nonlinear functions because the functional form may not be easily separated into real and imaginary parts; second, phase information may be lost or distorted.

The method we extend, however, can easily compute complex-valued gradients without requiring a transform into the real equivalent. For scalar quantities, Brandwood [18] and Adali [16] show that the derivitive of the real-valued cost function with respect to the complex-valued parameters exists. We show that the same approach is valid for complex-valued vectors as well. Furthermore, we present a design of a simple feedforward DL architecture for MIMO signal detection that can be trained fully in a complex domain. We will demonstrate through simulation that complex-valued neural networks can be efficiently designed for detecting MIMO signals.

The remainder of this paper is organized as follows. Section 2 provides preliminary knowledge and operations that will be useful in the subsequent sections. Section 3 describes the system model and problem formulation. Section 4 presents the proposed network architecture and offline training procedures. Section 5 discusses several numerical results demonstrating the validity of the complex-valued DNN for MIMO detection problems, and Sect. 6 concludes this work.

2 Preliminaries

2.1 Complex Matrix Multiplication

Suppose that $\mathbf{W} = \mathbf{W}_r + j\mathbf{W}_i$ and $\mathbf{b} = \mathbf{b}_r + j\mathbf{b}_i$ are complex matrix and vector, respectively, Where the elements in $\mathbf{W}_r, \mathbf{W}_i, \mathbf{b}_r, \mathbf{b}_i$ are real quantities, and $j = \sqrt{-1}$. Then a complex matrix multiplication which results in a vector $\mathbf{h}$ can be written as

$$\mathbf{h} = \mathbf{W}\mathbf{b} \tag{1a}$$

$$= \mathbf{W}_r\mathbf{b}_r - \mathbf{W}_i\mathbf{b}_i + j(\mathbf{W}_i\mathbf{b}_r + \mathbf{W}_r\mathbf{b}_i), \tag{1b}$$

The formulation in (1b) is a basis for various signal processing models that deal with complex-valued parameters. Several algorithms, including the DL applications, use this approach to transform the results of their computation from complex to real-domain and vice versa. Now notice that (1b) can be represented as a real vector by concatenating its real and imaginary parts, which produces a double dimensionality $\mathbf{h}' = [\mathbf{h}_r, \mathbf{h}_i]^T$. In this work, we utilize the compact complex form as in (1a).

2.2 Wirtinger Calculus

Let $f(\omega) = u(\omega_r, \omega_i) + v(\omega_r, \omega_i)$ is a complex function, where $\omega = \omega_r + j\omega_i$. If the continuous partial derivatives $u(\omega_r, \omega_i)$, $v(\omega_r, \omega_i)$ with respect to ω_r and ω_i exist, then a generalized complex derivatives can be defined as [16]

$$\frac{\partial f}{\partial \omega} = \frac{1}{2}\left(\frac{\partial f}{\partial \omega_r} - j\frac{\partial f}{\partial \omega_i}\right), \quad \frac{\partial f}{\partial \omega^*} = \frac{1}{2}\left(\frac{\partial f}{\partial \omega_r} + j\frac{\partial f}{\partial \omega_i}\right) \tag{2}$$

Formally, the complex derivative can be calculated by considering f to be a bivariate function $f(\omega, \omega^*)$ and ω and ω^* to be independent variables. The Wirtinger Calculus can generalize the standard complex calculus, as discussed in [16]. As such, instead of taking partial derivatives with respect to the real and imaginary components, it is more efficient to use the approach described above: Rewrite $f(\omega)$ as $f(\omega, \omega^*)$ and differentiate with respect to one by treating the other as a constant.

3 System Model and Problem Formulation

3.1 System Model

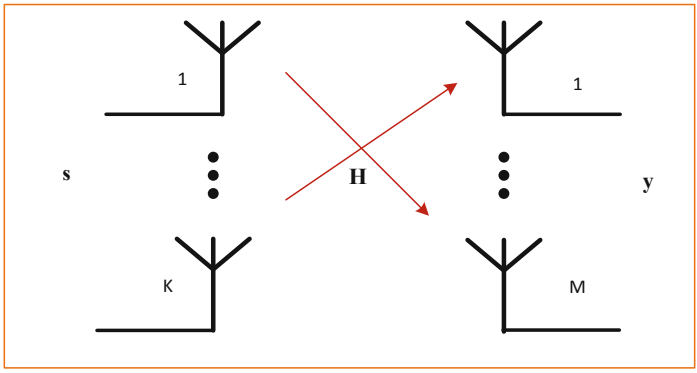


Fig. 1. A MIMO system with **s** transmit symbols, and **y** received signal vector over a channel matrix **H**.

Consider a MIMO communication system model shown in Fig. 1. We assume that the transmitter side contains K single antenna user terminals (UT)s, and the receiver is a base station (BS) equipped with M antennas. We denote the spatially-multiplexed UTs transmitted symbols vector as $\mathbf{s} = [s_1, s_2, \cdots, s_K]^T$ and its corresponding received signal vector as $\mathbf{y} = [y_1, y_2, \cdots, y_M]^T$, where s_k, y_j represent the transmitted symbol from the k-th UT, and the received signal by the j-th BS antenna, respectively. Each s_k is drawn from a set $\mathcal{S}$

of rectangular complex QPSK, and 16-QAM alphabets. We assume that the complex propagation channel $\mathbf{H} \in \mathbb{C}^{M\times K}$, is a flat fading and Rayleigh distributed, whose entries are independent and identically distributed (i.i.d) with $\mathcal{CN} \sim (0, 1)$. Then, the received signal vector $\mathbf{y}$ in a matrix form can be written as

$$\mathbf{y}^{(i)} = \mathbf{H}\mathbf{s}^{(i)} + \boldsymbol{\upsilon}^{(i)}, \quad i = 1, 2, \cdots, m, \tag{3}$$

where $\boldsymbol{\upsilon} \in \mathbb{C}^M$, a complex additive white Gaussian noise (C-AWGN), and m is the number of transmission instants. Here, we also assume that the channel state information is known to the transmitter.

In the existing works the complex-valued parameters in the above model is transformed into equivalent real model [9]

$$\mathbf{y}_r^{(i)} = \mathbf{H}_r\mathbf{s}_r^{(i)} + \boldsymbol{\upsilon}_r^{(i)}, \quad i = 1, 2, \cdots, m, \tag{4}$$

where

$$\begin{aligned} \mathbf{s}_r^{(i)} &= \begin{bmatrix} \mathrm{Re}(\mathbf{s}^{(i)}) \\ \mathrm{Im}(\mathbf{s}^{(i)}) \end{bmatrix} \in \mathcal{S}_r^{2K}, \ \mathbf{H}_r^{(i)} = \begin{bmatrix} \mathrm{Re}(\mathbf{H}^{(i)}) & -\mathrm{Im}(\mathbf{H}^{(i)}) \\ \mathrm{Im}(\mathbf{H}^{(i)}) & \mathrm{Re}(\mathbf{H}^{(i)}) \end{bmatrix} \in \mathbb{R}^{2M\times 2K}, \\ \mathbf{y}_r^{(i)} &= \begin{bmatrix} \mathrm{Re}(\mathbf{y}^{(i)}) \\ \mathrm{Im}(\mathbf{y}^{(i)}) \end{bmatrix} \in \mathbb{R}^{2M}, \boldsymbol{\upsilon}_r = \begin{bmatrix} \mathrm{Re}(\boldsymbol{\upsilon}^{(i)}) \\ \mathrm{Im}(\boldsymbol{\upsilon}^{(i)}) \end{bmatrix} \in \mathbb{R}^{2M}, \quad i = 1, 2, \cdots, m. \end{aligned} \tag{5}$$

Notice that the transformation in (5) follows the formulation in (1b). It is clear from (5) that the dimension of each parameter is doubled when we attempt to work with complex-valued parameters in the equivalent real. As discussed above, the main issue in such an approach is that besides computational burden in the network, the phase information will be lost or easily distorted, since the network only optimizes the real parameters. In this work, however, we ignore the transformation in (5) and directly work with the complex parameters in (3).

3.2 Problem Setup

Given (3), the goal of a MIMO detector is to recover the transmitted symbols $\mathbf{s}^{(i)}$ from the received signal $\mathbf{y}^{(i)}$ relying on the knowledge of the channel at each transmit instant i. Now, assuming all the information symbols in $\mathcal{S}$ are chosen (transmitted) with equal probability, the optimum MIMO detector known as maximum likelihood (ML) can be written as

$$\mathbf{s}_{\mathrm{ML}}^{(i)} = \underset{\mathbf{s}^{(i)}\in\mathcal{S}^K}{\operatorname{argmin}} \|\mathbf{y}^{(i)} - \mathbf{H}^{(i)}\mathbf{s}^{(i)}\|_2^2, \quad i = 1, 2, \cdots, m. \tag{6}$$

However, solving ML problem (6) globally is NP-hard since it requires $|\mathcal{S}|^K$ evaluations, where $|\cdot|$ denotes cardinality of a set. Because the computational complexity in ML detection is prohibitively expensive, we resort to a suboptimal solution that achieves near-ML performance while requiring less computational complexity. In this work, we propose a complex-valued neural network architecture as a supervised learning model that can take the complex received signal

vector $\mathbf{y}$ as an input and return an estimate of the transmitted symbol $\mathbf{s}$. Mathematically, this can be written as a machine learning problem

$$\hat{\mathbf{s}}_{DL}^{(i)} = \underset{f(\cdot,\boldsymbol{\theta})}{\operatorname{argmin}} \, \mathbb{E}\big(\mathcal{L}(f(\mathbf{y}^{(i)};\boldsymbol{\theta}), \mathbf{s}^{(i)})\big), \tag{7}$$

where $\mathcal{L}(\cdot)$ is some specific loss function, $\mathbb{E}(\cdot)$, is expectation operator, and $\boldsymbol{\theta}$ is a set of trainable parameters. In (7), our goal is, through exploiting DNN, to learn an optimal function $f(\cdot, \boldsymbol{\theta})$ and its corresponding parameter $\boldsymbol{\theta}$ in the sense that it can minimize the objective function $\mathcal{L}(\cdot)$.

4 Complex-Valued Feed-Forward Networks

In this section, we describe the proposed complex-valued feed-forward network or a multilayer perceptron (MLP), which from now on will be referred to as "cFFDnet."

4.1 Forward Propagation

The cFFDnet, like its real counterpart, MLP, performs feature mapping during forward propagation. There are two basic operations: linear and non-linear mappings, which are briefly described as follows. The network takes the noisy received signal $\mathbf{y}$ which is denoted as $\mathbf{a}^0$ as an input, and returns the estimated symbol $\hat{\mathbf{s}}$ as an output.

Suppose that N denotes the number of units (neurons), and ℓ denotes the number of layers. Let $\mathbf{W}^\ell \in \mathbb{C}^{N^\ell \times N^{\ell-1}}, \mathbf{b}^\ell \in \mathbb{C}^{N^\ell}$ denote the complex weight matrix and bias term respectively. Then the linear and non-linear mapping at each layer can be written as [19]

$$\mathbf{z}^\ell = \mathbf{W}^\ell \mathbf{a}^{\ell-1} + \mathbf{b}^\ell, \tag{8a}$$

$$\mathbf{a}^\ell = \sigma(\mathbf{z}^\ell), \quad \ell = 1, 2, \cdots, L, \tag{8b}$$

where $\sigma(\cdot)$ is the non-linearity (activation function). For notation consistency, we consider the input as a 0-th layer; i.e., $\mathbf{a}^0 = \mathbf{y}$.

At each layer, the linear mapping in (8a) computes a complex-matrix multiplication to obtain a vector $\mathbf{z}^\ell \in \mathbb{C}^{N^\ell}$. Then it goes through $\sigma(\cdot)$, a non-linear mapping. Here, $\sigma(\cdot)$ is a pointwise activation function that brings non-linearity to the network. There are several classes of activation functions proposed for complex-valued architectures. In this work, we utilize the modified tanh [20]

$$\sigma(\mathbf{z}) = \tanh\big(||\mathbf{z}||\big) \odot \frac{\mathbf{z}}{||\mathbf{z}||}, \tag{9}$$

where $||\cdot||$ denotes magnitude, and $\odot$ denotes a point-wise multiplication. The main advantage of this activation function is that it keeps the phase information. The corresponding set of trainable parameters can be defined as

$$\boldsymbol{\Theta} = \{\mathbf{W}^1, \mathbf{b}^1, \cdots, \mathbf{W}^L, \mathbf{b}^L\}. \tag{10}$$

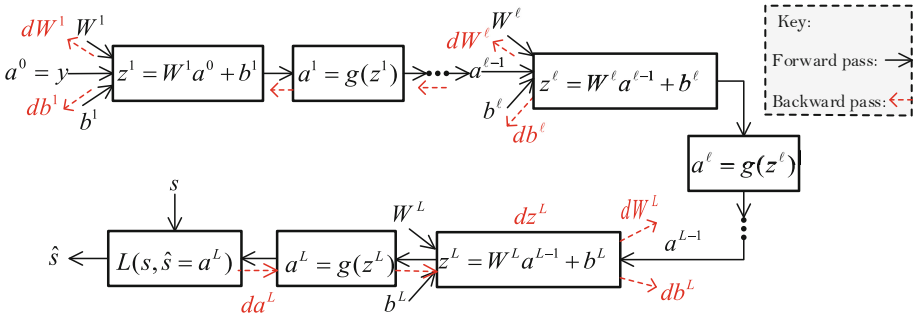


Fig. 2. A Computational graph representation of the proposed cFFDnet for the MIMO detection.

4.2 Back Propagation

In this subsection, we describe the back propagation process that characterizes the learning steps of cFFDnet.

Let $\mathcal{Y} = \mathbb{C}^M$ denotes M-dimensional feature space and $\mathcal{S}' = [s_1, s_2, \cdots, s_K]^T$ denotes the label space. Then the training dataset can be described as $D = \left\{\left(\mathbf{a}^{0(i)}, \mathbf{s}^{(i)}\right)\right\}_{i=1}^{m}$, where $\mathbf{a}^0 \in \mathcal{Y}$, $\mathbf{s} \in \mathcal{S}'$, are the input feature and true label, respectively, and m is the number of training examples. Then, the detection task in the cFFDnet can be defined as a mapping function $\mathrm{f}_\Theta : \mathcal{Y} \mapsto \mathcal{S}'$. In other words, the f_Θ is a regression function that minimizes a loss function to find the best parameter $\boldsymbol{\Theta}$ that will be used estimate the transmit symbol $\mathbf{s}$. The loss function is defined here as the mean-squared-error (MSE) between the true label $\mathbf{s}$ and the estimated network output $\mathbf{a}^L$ (see Fig. 2). To do so, let us define the error corresponding to each symbol as

$$e_k = s_k - a_k^L, \quad k = 1, 2, \cdots, N^L = K. \tag{11}$$

Then, the MSE loss produced by the network is

$$\mathcal{L} = \frac{1}{K} \sum_{k=1}^{K} e_k e_k^*, \tag{12}$$

where $*$ denotes a complex conjugate operation. Alternatively, in vector notation, it can be written as

$$\begin{aligned} \mathcal{L} &= \frac{1}{K}\Big((\mathbf{s} - \mathbf{a}^L)^\dagger (\mathbf{s} - \mathbf{a}^L)\Big), \\ &= \frac{1}{K}\Big(\mathbf{s}^\dagger \mathbf{s} - \left(\mathbf{a}^L\right)^\dagger \mathbf{s} - \mathbf{s}^\dagger \mathbf{a}^L + \left(\mathbf{a}^L\right)^\dagger \mathbf{a}^L\Big), \end{aligned} \tag{13}$$

where $\dagger$ denotes hermitian operation (transpose of a complex conjugate), and $\mathbf{a}^L \in \mathbb{C}^K$ represents the estimated output of the network. The superscript (i) is ignored for notational simplicity; i.e., the loss function is defined for a single

training example. Now notice that the loss function (13) is a real quantity ($\mathcal{L}: \mathbb{C}^K \times \mathbb{C}^K \mapsto \mathbb{R}^K$), while the network parameters are complex. As discussed in the previous sections, the Wirtinger Calculus can lay the groundwork for computing complex gradients with respect to the real-valued loss function. A modification required here is that the generalization of dimensionality since the results of Wirtinger Calculus are provided for scalar quantities. Proposition 1 discusses this extension.

Proposition 1. *Let $f: \mathbb{C}^{n_1} \mapsto \mathbb{R}^{n_2}$ be a real valued vector function of a complex vector $\mathbf{w}$. Let $f(\mathbf{w}) = \mathbf{g}(\mathbf{w}, \mathbf{w}^*)$, where $g: \mathbb{C}^{n_1} \times \mathbb{C}^{n_1} \mapsto \mathbb{R}^{n_2}$ is a real-valued vector function of two complex vector variables. Then computing $\frac{\partial g}{\partial \mathbf{w}^*} = 0$ is a necessary and sufficient condition to determine the stationary point of f.*

See Appendix A for the proof. A complex-valued network can be effectively trained utilizing the result in proposition 1 and the chain rule in the Wirtinger Calculus [21]. The gradient computation in the back propagation starts at the output (L-th layer) by computing the loss function (the error between true label $\mathbf{s}$ and the estimated symbol vector $\hat{\mathbf{s}}$.

The main objective of learning in this work is to minimize the loss function (13) with respect to the complex set of parameters $\boldsymbol{\Theta}$. To briefly describe this procedure, let us use da, dz, dW, db to denote the gradients corresponding to

$$\frac{\partial \mathcal{L}}{\partial \mathbf{a}^{*L}}, \frac{\partial \mathcal{L}}{\partial \mathbf{z}^{*L}}, \frac{\partial \mathcal{L}}{\partial \mathbf{W}^{*L}}, \frac{\partial \mathcal{L}}{\partial \mathbf{b}^{*L}},$$

respectively. As shown in the computation graph in fig.2, at the output layer, the network first computes the gradient da^L and step by step goes back to compute gradients in each layer. For instance, the gradient with respect to the linear activation dz^L can be computed by applying the chain rule as

$$dz^L = \frac{\partial \mathcal{L}}{\partial \mathbf{z}^{*L}} = \frac{\partial \mathcal{L}}{\partial \mathbf{a}^L} \frac{\partial \mathbf{a}^L}{\partial \mathbf{z}^{*L}} + \frac{\partial \mathcal{L}}{\partial \mathbf{a}^{*L}} \frac{\partial \mathbf{a}^{*L}}{\partial \mathbf{z}^{*L}}. \tag{14}$$

Similarly, during back propagation, other gradients will be computed and updated. For the learning, we use the standard stochastic gradient descent (SGD) method.

5 Numerical Results and Discussion

In this section, we present different numerical results that can demonstrate the effectiveness of the proposed methods.

5.1 Implementation Details

All detectors are implemented in python 3.6 using TensorFlow 2.4 library, and Table 1 lists the basic parameters.

Table 1. A list of network parameters.

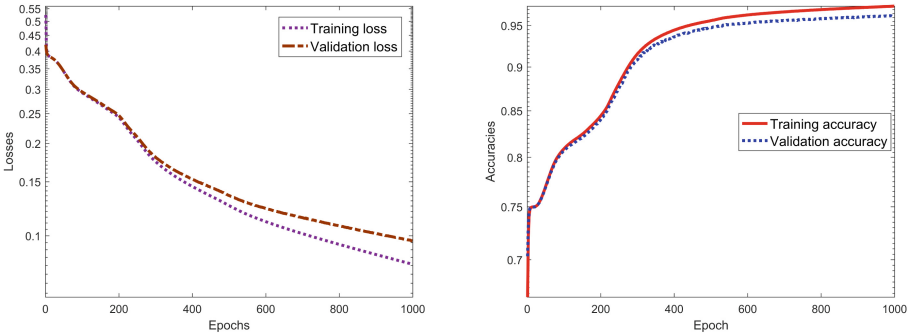
Parameter	Quantity
Number of hidden layers	5
Average epoch	3,000
Batch size	1024
Training data size	90,000
Validation data size	20,000
Maximum number of units in each layer	512
Maximum number of trainable parameters	1.3 million
ℓ_2 regularization factor	0.0001
Dropout ratio (max.)	0.3

Given the complex system model in 3, the dataset is generated as follows: In all realizations (offline training and online detection phases), the channel is chosen to be the classical i.i.d. Rayleigh fading, and fixed. The transmit symbols are drawn from the QPSK, and 16-QAM constellations. The noise variance is randomly sampled from complex AWGN, and the signal to noise ratio (SNR) in each training sample computed as $\text{SNR} = 10 \log \frac{||\mathbf{Hs}||_2}{||\boldsymbol{\nu}||_2}$. Then, the dataset $\mathbf{D}$ is generated as the definition in Sect. 4.2.

After the network is successfully trained (minimum overfitting and underfitting is obtained), it will be tested in online detection by using newly generated data that follows the same distribution as the training dataset. In the testing stage (online detection), the generalization ability of the network (detector) will be evaluated compared with the existing detectors. For this purpose, we employ the classical symbol-error-rate (SER) versus SNR performance analysis by running several Monte Carlo simulations. The numerical results for offline training and online detection are presented in the following subsections.

5.2 Learning History: Offline Training Phase

We begin our discussion of the proposed methods' performance analysis by assessing the offline learning characteristics. Figures 3(a),(b) show the basic training history parameters, training/validation loss and accuracy. We can see from these figures that the training loss is decreasing smoothly (Fig. 3(a)) or that the training accuracy is increasing continuously (Fig. 3(b)). This implies that the network is learning efficiently from the given dataset. On the other hand, we can see that the validation loss declines very closely to the training loss, or that



(a) Training and validation losses. (b) Training and validation accuracy.

Fig. 3. The offline learning histories for a QPSK modulated, $(M = 16, K = 4)$ MIMO system.

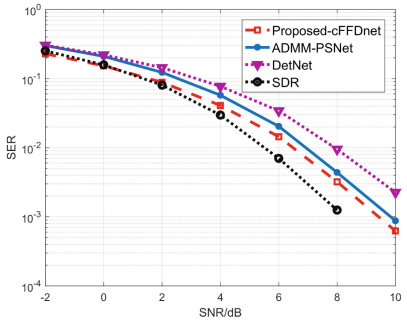
the validation accuracy increases consistently with the training accuracy until it reaches a maximum ($\geq 95\%$). This demonstrates that the network was less overfitted during offline training, implying that generalization ability in online detection will be improved.

5.3 SER: Online Detection Phase

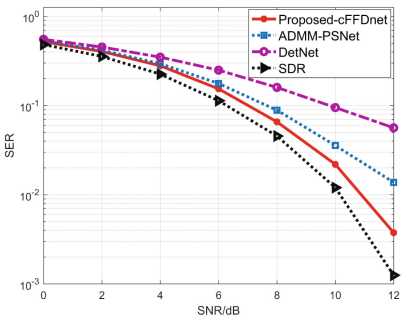
Since running simulation results for the optimal ML detector is very difficult for larger MIMO sizes, we use the classical suboptimal detector SDR [4] as a reference. Furthermore, we compare our proposed method with efficient existing detectors that are based on the model-driven deep learning approach.

Figures 4(a)–(d) show the online detection performance in terms of SER versus SNR for QPSK and 16-QAM modulations and different MIMO sizes. As shown in the Fig. 4(a), the proposed method has outperformed the existing model-driven methods (DetNet, ADMM-PSNet) in a significant range of SNR. It is evident from this figure that the proposed cFFDnet has achieved a performance very close to SDR particularly in the lower SNR regime.

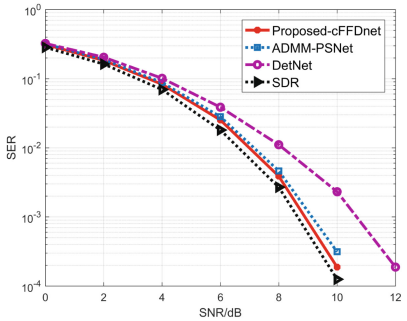
To further investigate the performance of the proposed method, we increased the modulation to 16-QAM with different MIMO sizes. The corresponding results of this scenario are depicted in Figs. 4(b)–(d). As it is clear from the figures, the performance of the proposed method is consistently improved under the given scenarios. In Fig. 4(d), it is also evident that even though the proposed detector achieved improved performance compared to the existing detectors, it seems to lag behind the SDR with a noticable range of SNR. This is due to the fact that as K increases, training the network with the same parameter settings results in limited efficiency. Because the size of the input data is limited (M = 64), it is difficult to achieve the required level of training accuracy. However, it is clear from the above results that once the network is properly trained, it can achieve significantly improved performance near the sub-optimal detector.



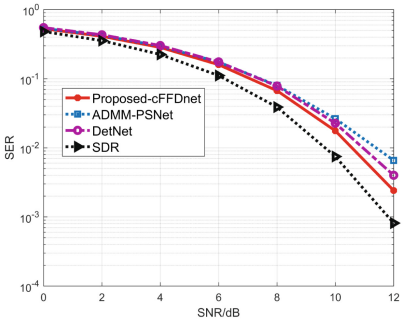
(a) QPSK, $(M = 16, K = 4)$ MIMO system.



(b) 16QAM, $(M = 32, K = 8)$ MIMO system.



(c) $(M = 64, K = 8)$ MIMO system.



(d) $(M = 64, K = 16)$ MIMO system.

Fig. 4. SER performance comparison between the proposed and the existing detectors in different modulations.

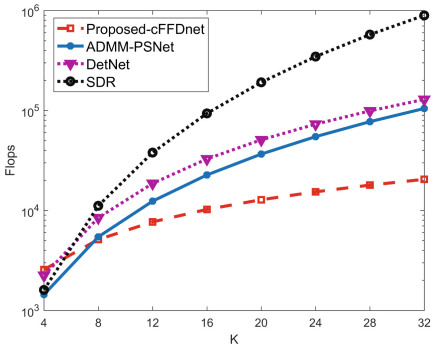


Fig. 5. Computational complexity comparison between the proposed and existing detectors for $M = 64$, and varying K MIMO sizes.

5.4 Computational Complexity

Apart from the SER performance improvement, the proposed detection method can also provide a lower computational cost compared to the existing methods. To show this, we use the standard number of floating point operations (Flops) as a performance measure.

Figure 5 illustrates the computational complexity comparison between the proposed and existing detectors. The results in this figure are obtained by counting the number of complex-vector multiplications for different MIMO sizes with $M = 64$, and varying K. As it is clear from the figure, the computational complexity of the proposed detector is significantly reduced compared to the existing model-driven detectors for a wide range of K. The main reason for such an improvement is that the proposed method is a simple feedforward network. During the training phase, since the network involves forward and backward propagation, its computational cost is higher (it takes longer time to learn). In the online detection, however, we only pass information forward, in which the network performs simple computations such as matrix-vector multiplication and pointwise activation. As a result, once properly trained, it has a lower computational cost.

6 Conclusion

The use of complex numbers plays a vital role in processing communication signals. When it comes to MIMO signal detection, several DL-based detectors have been proposed in the literature. However, they failed to work with complex-valued parameters. In this work, we have analyzed the application of complex-valued neural networks for MIMO signal detection for QPSK and 16-QAM signals. Numerical results have shown that the proposed approach can improve the performance of the existing detectors. Moreover, the computational complexity of the proposed approach is very low compared with the existing methods.

Appendix A Proof of Proposition 1

Proof. By considering an element wise operation on the elements of the real vector f, the proof is straightforward from [18]. Further, since g is a real function, $\frac{\partial g}{\partial \mathbf{w}}$, $\frac{\partial g}{\partial \mathbf{w}^*}$ are conjugates of each other; i.e., $\frac{\partial g}{\partial \mathbf{w}^*} = \left(\frac{\partial g}{\partial \mathbf{w}}\right)^*$. Hence, it is sufficient to compute the term $\frac{\partial g}{\partial \mathbf{w}^*}$.

References

1. Shannon, C.E.: A mathematical theory of communication. ACM SIGMOBILE Mobile Comput. Commun. Rev. **5**(1), 3–55 (2001)
2. Marzetta, T.L.: Noncooperative cellular wireless with unlimited numbers of base station antennas. IEEE Trans. Wireless Commun. **9**(11), 3590–3600 (2010)

3. Yang, S., Hanzo, L.: Fifty years of MIMO detection: the road to large-scale MIMOs. IEEE Communi. Surveys Tutorials **17**(4), 1941–1988 (2015)
4. Wai, H.T., Ma, W.K., So, A.M.C.: Cheap semidefinite relaxation mimo detection using row-by-row block coordinate descent. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3256–3259 (2011)
5. Shahabuddin, S., Juntti, M., Studer, C.: Admm-based infinity norm detection for large mu-mimo: Algorithm and vlsi architecture. In: IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–4 (2017)
6. Wu, Z., Zhang, C., Xue, Y., Xu, S., You, X.: Efficient architecture for soft-output massive mimo detection with gauss-seidel method. In: 2016 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1886–1889 (2016)
7. Samuel, N., Wiesel, A., Diskin, T.: Learning to detect. IEEE Trans. Signal Process. **67**(10), 2554–2564 (2019)
8. Xiaosi, T., et al.: Improving massive MIMO message passing detectors with deep neural network. IEEE Trans. Veh. Technol. **69**(2), 1267–1280 (2020)
9. Tiba, I.N., Zhang, Q., Jiang, J., Wang, Y.: A low-complexity ADMM-based massive MIMO detectors via deep neural networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4930–4934 (2021)
10. Tiba, I.N., Kulimushia, B.B., Kajuna, C.K.: Massive MIMO data detection using 1-dimensional convolutional neural network. In: IEEE/CIC International Conference on Communications in China (ICCC), pp. 483–488 (2020)
11. Tishby, N., Zaslavsky, N.: Deep learning and the information bottleneck principle. In: 2015 IEEE Information Theory Workshop (ITW), 26 Apr, pp. 1–5. IEEE
12. He, K., Zhang, X., Ren, S., et al.: Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1026–1034 (2015)
13. Bjornson, E., Giselsson, P.: Two applications of deep learning in the physical layer of communication systems [lecture notes]. IEEE Signal Process. Mag. **37**(5), 134–140 (2021)
14. Adalı, T., Li, H., Haykin, S: Complex-valued adaptive signal processing. Adapt. Signal Process. Next Gener. Solutions (2010)
15. Ablowitz, M.J., Fokas, A.S.: Complex Variables. Cambridge University Press, Cambridge (2003)
16. Adali, T., Schreier, P.J., Scharf, L.L.: Complex-valued signal processing: the proper way to deal with impropriety. IEEE Trans. Signal Process. **59**(11), 5101–5125 (2011)
17. Albreem, M.A., Juntti, M., Shahabuddin, S.: Massive MIMO detection techniques: a survey. IEEE Commun. Surv. Tutorials **21**(4), 3109–3132 (2019)
18. Brandwood, D.H.: A complex gradient operator and its application in adaptive array theory. In: IEE Proceedings H-Microwaves, Optics and Antennas, pp. 11–16 (1983)
19. Sun, R.: Optimization for deep learning: theory and algorithms. arXiv preprint arXiv:1912.08957 (2019)
20. Pfeifenberger L., Zöhrer, M., Pernkopf, F.: Deep complex-valued neural beamformers. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2902–2906 (2019)
21. Amin, M., Amin, M.I., Al-Nuaimi, A.Y., Murase, K.: Wirtinger calculus based gradient descent and Levenberg-Marquardt learning algorithms in complex-valued neural networks. In: International Conference on Neural Information Processing, pp. 550–559 (2011)