

PAPER

An Efficient Mapping Scheme on Neural Networks for Linear Massive MIMO Detection

Lin LI[†], *Member* and Jianhao HU^{††a)}, *Nonmember*

SUMMARY For massive multiple-input multiple-output (MIMO) communication systems, simple linear detectors such as zero forcing (ZF) and minimum mean square error (MMSE) can achieve near-optimal detection performance with reduced computational complexity. However, such linear detectors always involve complicated matrix inversion, which will suffer from high computational overhead in the practical implementation. Due to the massive parallel-processing and efficient hardware-implementation nature, the neural network has become a promising approach to signal processing for the future wireless communications. In this paper, we first propose an efficient neural network to calculate the pseudo-inverses for any type of matrices based on the improved Newton's method, termed as the PINN. Through detailed analysis and derivation, the linear massive MIMO detectors are mapped on PINNs, which can take full advantage of the research achievements of neural networks in both algorithms and hardwares. Furthermore, an improved limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) quasi-Newton method is studied as the learning algorithm of PINNs to achieve a better performance/complexity trade-off. Simulation results finally validate the efficiency of the proposed scheme.

Key words: massive MIMO, linear detector, neural networks, map, improved Newton's method, L-BFGS method

1. Introduction

Massive multiple-input multiple-output (MIMO) has been regarded as a key technology in the fifth generation mobile communication systems (5G) for improving the system capacity and spectral efficiency by deploying hundreds of antennas at the base station (BS) to simultaneously serve tens of user equipments (UEs) at the same frequency band [1]. Unfortunately, while the large number of antennas brings great benefits, it also brings huge computational burden to the practical implementation [2]. Therefore, how to design an efficient massive MIMO detector is a challenging issue for the future communications [3].

The maximum likelihood (ML) detector is considered to be optimal, but its computational complexity grows exponentially with the number of the transmitting antennas and the modulation order, which is not feasible for massive MIMO systems [4]. In contrast, traditional linear detectors such as the zero forcing (ZF) detector and the minimum mean squared error (MMSE) detector, have been proved to be near-optimal with lower computational complexity, espe-

cially when the BS-to-user-antenna ratio (BUAR) is large [5]. However, such linear massive MIMO detectors inevitably involve the high-dimensional matrix inversion, which is computationally expensive to implement in practice [6].

To design an efficient linear massive MIMO detector, many efforts have been devoted to alleviating the computational burden of the exact matrix inversion, which can be roughly divided into the explicit methods (requiring a separate matrix inversion) and implicit methods (directly calculating the transmitted vector) [7]–[9]. The major advantage of the implicit detection methods is the fact that they require lower computational complexity than the explicit detection methods. However, the linear detectors using the explicit methods have the following advantages: (1) In the time-division duplexing (TDD) massive MIMO systems, the matrix inversion results obtained during the uplink transmission can be reused to perform precoding and beamforming for the downlink. (2) In the slow-fading channels or the flat-fading channels with low-delay spread, the inversion results can be reused for the consecutive symbols and the adjacent subcarriers. (3) Some specific calculations such as the LLR computation and the rapid matrix updating modification require the separate matrix inversion [10]. Hence, we focus on the study of the high-accuracy and low-complexity explicit methods. As one widely-employed explicit detection method, Neumann series approximation (NSA) which expands the matrix inversion into the accumulation of matrix-vector multiplications is proposed to reduce the complexity. However, NSA converges slowly, and when its term is larger than 2, the complexity may be higher than exact matrix inversion [11]. Newton iteration (NI) and Chebyshev iteration (CI) are proposed successively to accelerate convergence, whereas their complexity is even higher [12].

As the mainstream algorithm of artificial intelligence (AI), neural networks have been researched extensively due to the massive parallel-processing nature, strong learning ability, and efficient hardware implementation [13]. In addition, AI hardware accelerators have been developed rapidly, which achieve higher processing speed and energy efficiency compared with traditional processors such as CPUs and GPUs [14]. Thus, deploying neural networks in AI hardware accelerators using AI learning frameworks can be implemented conveniently and cost-efficiently. During recent decades, neural networks have been widely applied to the physical layer of wireless communications [15], e.g. channel estimation [16] and signal detection [17]. Especially for massive MIMO detection, [18] and [19] proposed the model-

Manuscript received October 26, 2022.

Manuscript revised February 17, 2023.

Manuscript publicized May 19, 2023.

[†]The author is with Qinghai Normal University, Xining, 810000 China.

^{††}The author is with University of Electronic Science and Technology of China, Chengdu, 611731 China.

a) E-mail: jhhu@uestc.edu.cn (Corresponding author)

DOI: 10.1587/transfun.2022EAP1132

driven deep neural networks (DNNs) constructed by unfolding existing iterative algorithms with some adjustable parameters to train for excellent detection performance. However, they ignore the high computational complexity and the hardware implementation difficulty. And the results may not be effective in realistic environments with the offline training and online working mode.

In this paper, we aim to fully leverage the advantages of neural networks to design a high-accuracy and low-complexity linear signal detector with explicit matrix inversion for massive MIMO. A single-layer feedforward neural network with improved Newton's method is first proposed to calculate pseudo-inverses of matrices, termed as the PINN. Then we map the classical linear massive MIMO detectors on the PINN, which is the first work to combine the linear detection with the proposed neural network model to the authors' best knowledge. Through the mapping, we can give full play to the advantage and potential of both AI learning algorithms and hardware accelerators for more efficient detectors. To achieve a better trade-off between performance and complexity, the limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) quasi-Newton method is further studied. Simulation results show that the proposed scheme can closely approach to the exact MMSE accuracy with lower complexity than the existing explicit linear detectors.

The remainder of this paper is organized as follows. Section 2 briefly introduces the system model and linear detection. Section 3 presents the PINN with improved Newton's method, the mapping scheme and the L-BFGS method for PINNs. Numerical simulation is discussed in Sect. 4. Section 5 concludes the entire paper.

Notations: The bold uppercase and lowercase denote the matrix and vector, respectively. The conjugate transpose, transpose, inverse, and pseudo-inverse of a matrix are denoted by $[\cdot]^H$, $[\cdot]^T$, $[\cdot]^{-1}$, and $[\cdot]^+$, respectively. \mathbb{R} and \mathbb{C} represent the set of real and complex numbers, respectively. $\mathbb{E}\{\cdot\}$, $\Re\{\cdot\}$ and $\Im\{\cdot\}$ denote the expectation of a random variable, the real and imaginary part of a complex argument, respectively. $\mathcal{CN}(\cdot, \cdot)$ represents the complex Gaussian distribution with the two arguments being the mean and covariance matrix, respectively. $\|\cdot\|_2$ and $\|\cdot\|_F$ denotes the 2-norm of a vector and Frobenius norm of a matrix, respectively. For any matrix \mathbf{A} , A_i refers to the i th column of \mathbf{A} , and a_{ij} refers to the (i, j) th entry of \mathbf{A} . \mathbf{I}_n denotes the $n \times n$ identity matrix.

2. Preliminaries

2.1 System Model

We consider an uplink massive MIMO system equipped with N_r antennas at the BS, which simultaneously serves N_t single-antenna UEs ($N_r \leq N_t$). The received signal $\bar{\mathbf{y}} \in \mathbb{C}^{N_r \times 1}$ is

$$\bar{\mathbf{y}} = \bar{\mathbf{H}}\bar{\mathbf{s}} + \bar{\mathbf{n}}, \quad (1)$$

where $\bar{\mathbf{H}} \in \mathbb{C}^{N_r \times N_t}$ denotes the Rayleigh fading channel

matrix, whose entry is the independent and identically distributed (i.i.d) Gaussian random variable with zero mean and unit variance. Without losing generality, we assume $\bar{\mathbf{H}}$ can be perfectly known at the BS. $\bar{\mathbf{s}} \in \mathbb{C}^{N_t \times 1}$ denotes the transmitted symbol vector with $\mathbb{E}\{\bar{\mathbf{s}}\bar{\mathbf{s}}^H\} = \mathbf{I}_{N_t}/N_t$. $\bar{\mathbf{n}} \in \mathbb{C}^{N_r \times 1} \sim \mathcal{CN}(0, \sigma_n^2 \mathbf{I}_{N_r})$ denotes the additive white Gaussian noise (AWGN) vector with σ_n^2 being the average noise power. The signal-to-noise ratio (SNR) in this circumstance is defined as $1/\sigma_n^2$.

For convenience in subsequent discussion, given $N = 2N_r$ and $K = 2N_t$, the complex-valued system model can be converted to the real domain by $\mathbf{y} = \begin{bmatrix} \Re\{\bar{\mathbf{y}}\} \\ \Im\{\bar{\mathbf{y}}\} \end{bmatrix} \in \mathbb{R}^{N \times 1}$, $\mathbf{s} = \begin{bmatrix} \Re\{\bar{\mathbf{s}}\} \\ \Im\{\bar{\mathbf{s}}\} \end{bmatrix} \in \mathbb{R}^{K \times 1}$, $\mathbf{n} = \begin{bmatrix} \Re\{\bar{\mathbf{n}}\} \\ \Im\{\bar{\mathbf{n}}\} \end{bmatrix} \in \mathbb{R}^{N \times 1}$, and

$$\mathbf{H} = \begin{bmatrix} \Re\{\bar{\mathbf{H}}\} & -\Im\{\bar{\mathbf{H}}\} \\ \Im\{\bar{\mathbf{H}}\} & \Re\{\bar{\mathbf{H}}\} \end{bmatrix} \in \mathbb{R}^{N \times K}.$$

Thus the real-valued model can be expressed as

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}. \quad (2)$$

2.2 Linear Detectors

The linear detector estimates the transmitted signal $\hat{\mathbf{s}}$ by multiplying the received signal \mathbf{y} with a filter matrix \mathbf{G} , which can be expressed as

$$\hat{\mathbf{s}} = \mathbf{G}\mathbf{y}. \quad (3)$$

Specifically, the ZF filter matrix can be written as

$$\mathbf{G}_{\text{ZF}} = \left(\mathbf{H}^T \mathbf{H}\right)^{-1} \mathbf{H}^T. \quad (4)$$

The ZF detector can eliminate the interference among the received signals, but it suffers from the noise amplification problem.

To address this problem, the noise item is included in the MMSE filter matrix, which can be expressed as

$$\mathbf{G}_{\text{MMSE}} = \left(\mathbf{H}^T \mathbf{H} + \sigma_n^2 \mathbf{I}_K\right)^{-1} \mathbf{H}^T, \quad (5)$$

where $\sigma_n^2 = 1/2\sigma_n^2$. The MMSE detector can achieve a good trade-off between the noise amplification and inference suppression.

As can be seen from (4) and (5), the linear detectors require the matrix inversion with the exact computational complexity being $\mathcal{O}(K^3)$.

3. Methodology

3.1 PINN Model

The structure of the PINN is shown in Fig. 1. Suppose the input vector $X_i = [x_{1i}, x_{2i}, \dots, x_{ni}]^T$, output vector $Y_i =$

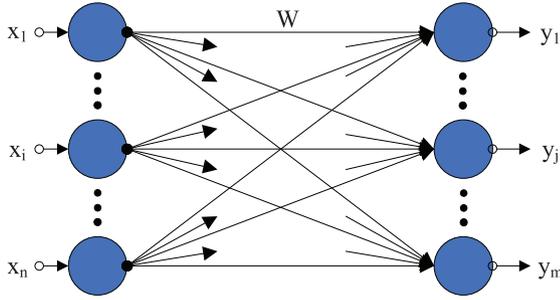


Fig. 1 The structure of PINN.

$[y_{1i}, y_{2i}, \dots, y_{mi}]^T$, and \mathbf{W} is the weight matrix, which is also often written in the vector form $\mathbf{w} = [w_{11}, w_{12}, \dots, w_{mn}]^T$. Set the activation function as the linear identity function, and the threshold vector as the zero vector. The forward propagation can be expressed as

$$Y_i = \mathbf{W}X_i. \quad (6)$$

Given a training set $\{(X_1, D_1), \dots, (X_p, D_p)\}$, where D_i is the expected output vector and p is the sample number. The mean squared error is defined as the cost function:

$$C(\mathbf{W}) = \frac{1}{2p} \sum_{i=1}^p \|D_i - Y_i\|_2^2. \quad (7)$$

It is well known that matrix inversion can be transformed into an optimization problem as to minimize

$$C(\mathbf{W}) = \|\mathbf{I} - \mathbf{W}\mathbf{X}\|_F^2. \quad (8)$$

And the optimal approximation \mathbf{W}^* is proved to be the pseudo-inverse of \mathbf{X} [20].

Referred to the PINN, we set the column vector of the identity matrix \mathbf{I}_m as the desired output, and the column vector of \mathbf{X} to be inverted as the input vector with the number of columns being the number of samples. The cost function can be expressed as

$$\begin{aligned} C(\mathbf{W}) &= \frac{1}{2m} \sum_{i=1}^m \|I_i - \mathbf{W}X_i\|_2^2 \\ &= \frac{1}{2m} \|\mathbf{I} - \mathbf{W}\mathbf{X}\|_F^2. \end{aligned} \quad (9)$$

Thus, we relate the calculation of pseudo-inverses to PINNs, and \mathbf{X}^+ can be obtained from the converged weight matrix \mathbf{W} . The key to solve pseudo-inverse problems becomes the training process of PINNs.

3.2 Training Strategy

Gradient descent (GD) method is often utilized as the learning algorithm of PINNs to train the weights [21]. Its updating formula of \mathbf{w} can be expressed as

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu \frac{\partial C}{\partial \mathbf{w}}, \quad (10)$$

where μ is the step length, and k is the k -th iteration. The advantage of the GD method is that it's very simple in calculations, only involving the multiplication and addition. However, it converges slowly, which is unsuitable for the real-time applications.

To overcome the shortcomings of the GD method, Newton's method leverage the second-order information is investigated and improved as the learning algorithm of the PINN for the first time in our work. The derivation and computation process of Newton's method applied to the PINN will be detailed below.

We define the error matrix $\mathbf{E} = \mathbf{D} - \mathbf{Y}$, and the error vector $\mathbf{e} = [e_{11}, e_{12}, \dots, e_{mm}]^T$ for the constructed PINN. Thus the cost function can be calculated as

$$C(\mathbf{W}) = \frac{1}{2m} \mathbf{e}^T * \mathbf{e}. \quad (11)$$

Subsequently, the key step is to calculate Jacobian matrix, and the calculation process is as below

$$\begin{aligned} \mathbf{J} &= \begin{bmatrix} \frac{\partial e_{11}}{\partial w_{11}} & \frac{\partial e_{11}}{\partial w_{12}} & \dots & \frac{\partial e_{11}}{\partial w_{mn}} \\ \frac{\partial e_{12}}{\partial w_{11}} & \frac{\partial e_{12}}{\partial w_{12}} & \dots & \frac{\partial e_{12}}{\partial w_{mn}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{mm}}{\partial w_{11}} & \frac{\partial e_{mm}}{\partial w_{12}} & \dots & \frac{\partial e_{mm}}{\partial w_{mn}} \end{bmatrix} \\ &= - \begin{bmatrix} \mathbf{X}^T & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{X}^T \end{bmatrix}_{mm \times mn}. \end{aligned} \quad (12)$$

It can be seen that Jacobian matrix can be directly obtained from the input matrix \mathbf{X} for PINNs without the complex calculation and massive storage space.

In the following, we calculate the gradient.

$$\mathbf{g} = \left[\frac{\partial C}{\partial w_{11}} \quad \frac{\partial C}{\partial w_{12}} \quad \dots \quad \frac{\partial C}{\partial w_{mn}} \right]^T = \frac{1}{m} \mathbf{J}^T \mathbf{e}. \quad (13)$$

To simplify the calculation, we rewrite the gradient in the matrix form as

$$\begin{aligned} \mathbf{G} &= \begin{bmatrix} \frac{\partial C}{\partial w_{11}} & \frac{\partial C}{\partial w_{12}} & \dots & \frac{\partial C}{\partial w_{1n}} \\ \frac{\partial C}{\partial w_{21}} & \frac{\partial C}{\partial w_{22}} & \dots & \frac{\partial C}{\partial w_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial C}{\partial w_{m1}} & \frac{\partial C}{\partial w_{m2}} & \dots & \frac{\partial C}{\partial w_{mn}} \end{bmatrix} \\ &= -\frac{1}{m} \mathbf{E} \mathbf{X}^T. \end{aligned} \quad (14)$$

Hessian matrix composed of the second-order partial derivatives of a multivariate function can be calculated as

$$\begin{aligned} \mathbf{F} &= \begin{bmatrix} \frac{\partial^2 C}{\partial w_{11} \partial w_{11}} & \frac{\partial^2 C}{\partial w_{11} \partial w_{12}} & \dots & \frac{\partial^2 C}{\partial w_{11} \partial w_{mn}} \\ \frac{\partial^2 C}{\partial w_{12} \partial w_{11}} & \frac{\partial^2 C}{\partial w_{12} \partial w_{12}} & \dots & \frac{\partial^2 C}{\partial w_{12} \partial w_{mn}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 C}{\partial w_{mn} \partial w_{11}} & \frac{\partial^2 C}{\partial w_{mn} \partial w_{12}} & \dots & \frac{\partial^2 C}{\partial w_{mn} \partial w_{mn}} \end{bmatrix} \\ &= \frac{1}{m} \begin{bmatrix} \mathbf{X} \mathbf{X}^T & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{X} \mathbf{X}^T \end{bmatrix}_{mn \times mn} \end{aligned}$$

$$= \frac{1}{m} \mathbf{J}^T \mathbf{J}. \quad (15)$$

We find out that Hessian matrix for PINNs can also be obtained directly from \mathbf{X} unlike the other cases where Hessian matrix is particularly difficult to calculate.

Newton's method is considered the most fundamental in the second-order learning algorithms due to the fact that many other efficient second-order algorithms can be derived from it [22]. Its algebraic principle is to use Taylor expansion to find the approximate solution. The weight updating formula with Newton's method can be expressed as

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mathbf{F}^{-1} \mathbf{g}. \quad (16)$$

According to the previous calculation results above, (16) can easily be calculated as

$$\mathbf{w}(k+1) = \mathbf{w}(k) - (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \mathbf{e}. \quad (17)$$

Obviously, for PINNs, Newton's method is identical to the Gauss-Newton method [23].

Referring to (17), we improve Newton's method from two aspects. The first is to reduce the redundant computational complexity caused by the high-dimensional Jacobian matrix \mathbf{J} . In order to simplify the calculation, we modify the updating formula of the weight vector to the matrix form, and (17) can be deduced as

$$\mathbf{W}(k+1) = \mathbf{W}(k) - \mathbf{G} \left(\frac{1}{m} \mathbf{X} \mathbf{X}^T \right)^{-1}. \quad (18)$$

By further processing, (18) can be expressed as

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \mathbf{E} \mathbf{X}^T (\mathbf{X} \mathbf{X}^T)^{-1}. \quad (19)$$

This is an important achievement of our work and will be analyzed in the sequel for different types of \mathbf{X} .

Assuming that \mathbf{X} is a nonsingular square or row full-rank rectangular matrix, then \mathbf{H} is the positive definite matrix. Thus, the cost function is quadratic, and the constructed PINN can achieve global convergence by one epoch with Newton's method.

If \mathbf{X} is a column full-rank rectangular matrix, \mathbf{X}^T is row full-rank, which can be set as the input matrix. Then (19) can be changed to

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \mathbf{E} \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1}. \quad (20)$$

Its transpose can be written as

$$\mathbf{W}^T(k+1) = \mathbf{W}^T(k) + (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{E}. \quad (21)$$

In the case of that \mathbf{X} is rank deficient, \mathbf{H} is the positive semi-definite matrix. We need to further improve Newton's method by introducing a positive definite diagonal matrix to guarantee the convergence, which can be expressed as:

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \mathbf{E} \mathbf{X}^T (\mathbf{X} \mathbf{X}^T + \lambda \mathbf{I})^{-1}, \quad (22)$$

where λ is zero approached through positive real values. It is similar to the Levenberg-Marquardt algorithm in form, except that λ is limited to a positive value close to zero without adjustments [24].

To prove the convergence of the proposed PINN with the improved Newton's method, the theoretical solutions of pseudo-inverses are presented by the below two lemmas [25].

Lemma 1: Suppose $\mathbf{A} \in \mathbb{R}^{m \times n}$ is full rank, i.e., $\text{rank}(\mathbf{A}) = \min\{m, n\}$, $\mathbf{A} \mathbf{A}^T$ ($m < n$) or $\mathbf{A}^T \mathbf{A}$ ($m > n$) is nonsingular. If $\text{rank}(\mathbf{A}) = m$, $\mathbf{A}^+ = \mathbf{A}^T (\mathbf{A} \mathbf{A}^T)^{-1}$; if $\text{rank}(\mathbf{A}) = n$, $\mathbf{A}^+ = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$.

Lemma 2: For any matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{A}^+ = \lim_{\lambda \rightarrow 0} \mathbf{A}^T (\mathbf{A} \mathbf{A}^T + \lambda \mathbf{I})^{-1}$.

Referring to (20), (21) and (22), if the initial weight matrix $\mathbf{W}(0)$ is set to be zero matrix, the error matrix \mathbf{E} will be the identity matrix correspondingly. Then it's obvious that the converged weight matrix after one epoch is consistent with the theoretical solution of the pseudo-inverse. This conclusion demonstrates the convergence of the proposed PINN model and its correctness of calculating pseudo-inverses.

To sum up, the PINN with improved Newton's method can be used to calculate the pseudo-inverse for any type of matrices. Moreover, its convergence speed is greatly accelerated compared with the GD method, and its computational complexity is greatly reduced compared with Newton's method.

3.3 Mapping Scheme

Based on the conclusion of the previous subsection, we can map the linear MIMO detection on the proposed PINN model. The mapping scheme is shown in Fig. 2. The proposed PINN works in training and inference stages, respectively. We assume that the ideal channel matrix \mathbf{H} and noise power σ_n^2 can be estimated by the channel estimator as the input of the PINN in the training process. The pseudo-inverse of \mathbf{H}^T can be calculated using the PINN model. Initialize the weight matrix to the zero matrix, and we can deduce the following equations from (21) and (22):

$$\mathbf{W}^T = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T, \quad (23)$$

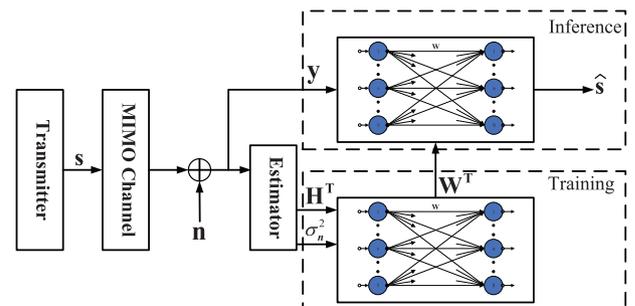


Fig. 2 The block diagram of the mapping scheme.

$$\mathbf{W}^T = \left(\mathbf{H}^T \mathbf{H} + \lambda \mathbf{I} \right)^{-1} \mathbf{H}^T. \quad (24)$$

Obviously, (23) is identical to the ZF filter matrix. Assuming $\lambda = \sigma_n^2$, (24) is consistent with the MMSE filter matrix. After one epoch training, the convergent weight matrix can be obtained. Then we deployed it as the weight matrix in the inference process of the PINN. Finally, we take the received signal \mathbf{y} as the input vector, and the output vector is the estimation of the transmitted signal \mathbf{s} , which can be expressed as

$$\hat{\mathbf{s}} = \mathbf{W}^T \mathbf{y}. \quad (25)$$

Up to now, we have mapped the ZF and MMSE linear detectors on the proposed PINN model. The benefits of this mapping scheme are significant: (1) Through the mapping, the linear detectors can be easily deployed on the AI hardware accelerator with higher processing speed and energy efficiency, and the efficiency of the linear detectors in the practical massive MIMO systems can be improved accordingly. (2) We can make full use of the evolving learning algorithms of neural networks so as to improve the detection algorithms. In order to avoid Hessian matrix inversion in improved Newton's method, the quasi-Newton method is always applied [26].

3.4 L-BFGS Method

The essence of the quasi-Newton method is to construct a symmetric positive definite matrix to approximate the inverse of Hessian matrix for the lower complexity and efficient implementation [27]. Thereinto, the BFGS method is regarded as the most effective one, and the L-BFGS method is its modification that only stores a certain number of correction vector pairs for the approximate computation without storing the Hessian approximation explicitly [28]. Because of its lower complexity and storage, the L-BFGS method is studied in this paper to further simplify the improved Newton's method.

The outline of the L-BFGS method used for the PINN is summarized as follows:

Step 1: Initialize the initial point $\mathbf{w}(0)$, the threshold ϵ , the correction number l , the approximate Hessian inversion $\mathbf{B}(0)$, the search direction $\mathbf{d}(0)$ and set $k = 0$;

Step 2: If $\|\mathbf{g}(k)\| \leq \epsilon$, stop;

Step 3: Compute the exact step length such that $\mu = \arg \min_{\mu > 0} f(\mathbf{w}(k) + \mu \mathbf{d}(k))$, and obtain the new iteration $\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \mathbf{d}(k)$;

Step 4: Compute the new gradient $\mathbf{g}(k+1) = \mathbf{J}^T \mathbf{e}$;

Step 5: Update the pair $\mathbf{p}(k) = \mathbf{w}(k+1) - \mathbf{w}(k)$ and $\mathbf{q}(k) = \mathbf{g}(k+1) - \mathbf{g}(k)$;

Step 6: If $k > l$, discard the pair \mathbf{p}_{k-l} , \mathbf{q}_{k-l} from memory storage;

Step 7: Compute the new search direction $\mathbf{d}(k+1) = -\mathbf{B}(k+1)\mathbf{g}(k+1)$ using Algorithm 1;

Step 8: $k := k + 1$ and go to Step 2.

Algorithm 1 L-BFGS two-loop recursion

Input: $\mathbf{g}(k+1)$, $\mathbf{p}(i)$ and $\mathbf{q}(i)$, and $\rho(i) = 1/(\mathbf{p}(i)^T \mathbf{q}(i))$ where $i = k-l+1, \dots, k$;

```

1:  $\mathbf{d}(k+1) = -\mathbf{g}(k+1)$ ;
2: for  $i = k$  to  $k-l+1$  do
3:    $\alpha(i) = \rho(i)\mathbf{p}(i)^T \mathbf{d}(k+1)$ ;
4:    $\mathbf{d}(k+1) = \mathbf{d}(k+1) - \alpha(i)\mathbf{q}(i)$ ;
5: end for
6:  $\mathbf{d}(k+1) = \mathbf{B}(0)\mathbf{d}(k+1)$ ;
7: for  $i = k-l+1$  to  $k$  do
8:    $\beta = \rho(i)\mathbf{q}(i)^T \mathbf{d}(k+1)$ ;
9:    $\mathbf{d}(k+1) = \mathbf{d}(k+1) + \mathbf{p}(i)(\alpha(i) - \beta)$ ;
10: end for

```

Output: $\mathbf{d}(k+1)$.

The two-loop recursive procedure of L-BFGS is proposed to compute $-\mathbf{B}(k+1)\mathbf{g}(k+1)$ without storing $\mathbf{B}(k+1)$ explicitly, which is shown as below.

Since the L-BFGS method avoids the inverse operation of Newton's method and the large storage space required by the BFGS method, it is computationally efficient, requiring only about $4lnm$ operations (one multiplication and one addition) in the two-loop recursion. In addition, we should point out that we have proved in our previous work that $l = 1$ and $\mu = 1$ are feasible for the linear detectors [29].

4. Simulation Results

4.1 PINN Convergence Performance

In order to illustrate the convergence performance of the proposed PINN model, we compare the convergence speed and accuracy of the GD method, BFGS method and Newton's method by calculating the inverse matrix of a nonsingular matrix $\mathbf{A} \in \mathbb{R}^{4 \times 4}$. For fairness and ease of illustration, MATLAB neural network toolbox is used here to simulate the proposed PINN with some specific settings [30].

The convergence behaviors are shown in Fig. 3. We set the goal of the mean squared error being 10^{-10} . It can be observed that Newton's method can achieve the best accuracy with the least number of epochs. Although the computational complexity of the GD method is the lowest for each epoch, it leads to the largest number of epochs and the worst accuracy. BFGS method is a compromise between the GD method and Newton's method in terms of the computational complexity and convergence performance.

To further compare the computational complexity of the GD method and improved Newton's method, we employ the multiplication operation that dominates the overall complexity to make a rough estimate. For the GD method, the required number of multiplication operations is $m^2n + mn$ per epoch, i.e., its computational complexity is $O(k(m^2n + mn))$, where $n \times m$ is the dimension of the matrix to be inverted and k is the epoch number. Accordingly, improved Newton's method has a computational complexity of $O((m+n)^2n)$, since its epoch number is 1. We simulate the inversion of randomly generated matrices with 4, 8, 16 dimensions under the goal accuracy 10^{-3} to compare the epoch number and

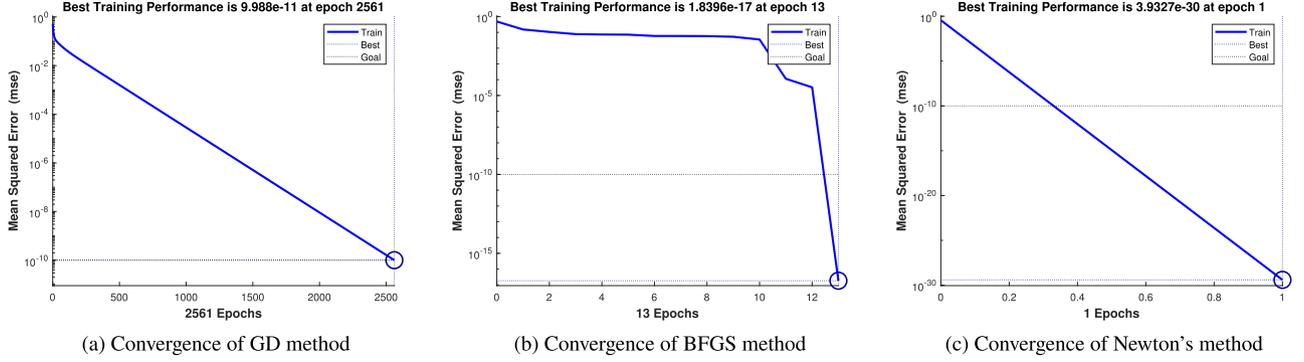


Fig. 3 Comparison of the convergence performance of different learning algorithms in a simple example.

Table 1 Comparison of the epoch number.

| Dimension | GD Method | Proposed Method |
|----------------|-----------|-----------------|
| 4×4 | 1448 | 1 |
| 8×8 | 5148 | 1 |
| 16×16 | 23279 | 1 |

runtime of the two algorithms.

As can be seen in Table 1, the GD method converges much more slowly than the improved Newton’s method. The epoch number of the GD method increases rapidly with the increase of matrix scales, while it is always 1 for the proposed Newton’s method. Since k is much bigger than the matrix scale in the simulation, we can conclude that the PINN with improved Newton’s method has a lower total computational cost than the GD-based neural network for matrix inversion.

4.2 PINN Detection Performance

We consider an uplink massive MIMO system with the BS antennas $N_r = 128$ and UE antennas $N_t = 128$ under the i.i.d. Rayleigh fading channel [31]. Fig. 4(a) and Fig. 4(b) compare the symbol error rate (SER) performance of ZF, MMSE, and the proposed PINN detectors under QPSK and 16QAM modulations. We can see that the detection performance using the proposed PINN model is consistent with that of the traditional linear detectors. In addition, results also show the correctness of the PINN model in calculating the pseudo-inverses of large-scale square matrices.

SER performance in the $N_r = 128$ and $N_t = 64$ antenna configuration is also provided, which exhibits the channel-hardening phenomenon and the validity of the proposed PINN model to solve the rectangular matrix. As seen in Fig. 4(c) and Fig. 4(d), the linear detectors achieve near-optimal detection performance, and the performance difference between the ZF detector and MMSE detector decreases. The proposed PINN model can still achieve the same detection performance with the traditional linear detectors. Results also show the correctness of the PINN model in calculating the pseudo-inverses of rectangular matrices.

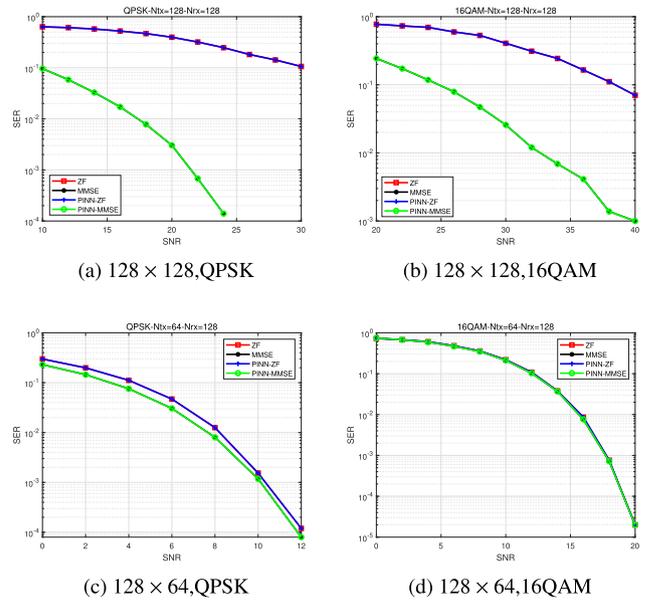


Fig. 4 Detection performance comparison under different simulation conditions.

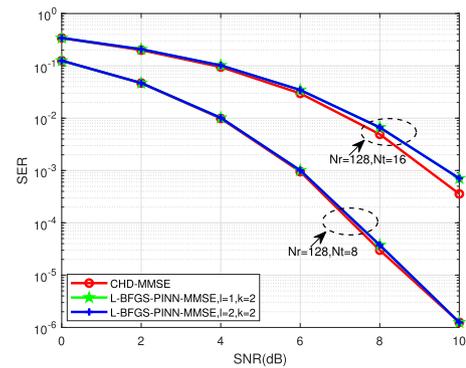


Fig. 5 L-BFGS-PINN SER performance with different l .

4.3 L-BFGS-PINN Detection Performance

We now provide simulation results to verify the effectiveness

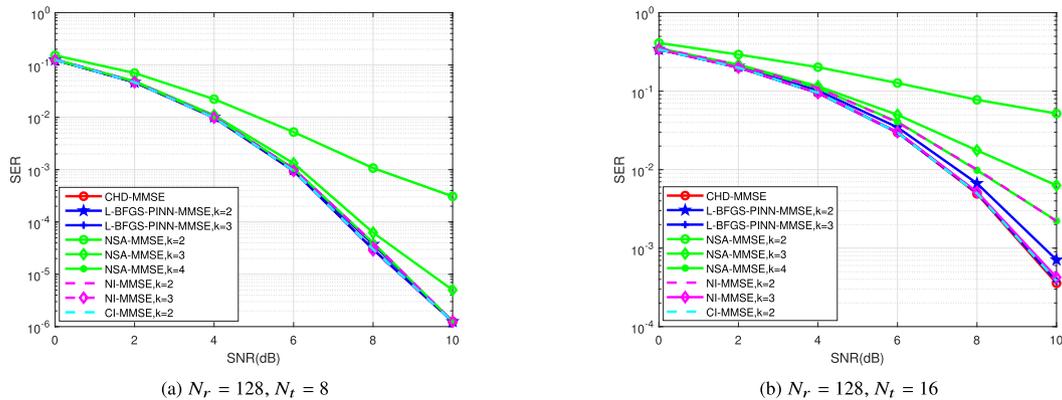


Fig. 6 Detection performance comparison of the proposed method with existing methods under 16QAM modulation for different antenna configurations.

of the PINN based on the L-BFGS method (L-BFGS-PINN) for MMSE detection, the performance of which is compared to the conventional explicit detection methods (namely NSA, NI, and CI) against different values of SNRs with $N_r = 128$ and $N_t = 8$ or 16 under 16QAM modulation. Cholesky decomposition (CHD) method for the exact linear MMSE detector is used as the baseline [32].

The effect of the correction vector pairs number l to be stored in L-BFGS method on the detection performance is first discussed. Fig. 5 shows the detection performance of L-BFGS-PINN at $l = 1$ and $l = 2$ with different antenna configurations under the fixed iteration number $k = 2$. It can be seen that the L-BFGS-PINN based MMSE detector achieves the identical performance at $l = 1$ and $l = 2$ and is very close to the exact CHD based MMSE detector. Thus, we set $l = 1$ in the subsequent performance simulations to reduce complexity.

To verify the performance of the L-BFGS-PINN based detector, we compare it to NSA, NI, CI based detectors. Fig. 6 shows the SER performance with the $N_r = 128, N_t = 8$ and $N_r = 128, N_t = 16$ antenna configurations, respectively. It can be seen that the L-BFGS-PINN based detector is superior to the NSA based detector in performance under all simulation conditions. Although the NI based detector performs as well as the proposed detector when BUAR is small, it performs relatively poorly when BUAR is large. The convergence of the CI based detector is the fast of all. For $N_r = 128, N_t = 8$, the performance of the proposed detector is almost identical to the exact MMSE detector when its number of iterations as low as 2. For $N_r = 128, N_t = 16$, the performance of the proposed detector loses about 0.3dB at $SER = 10^{-3}$ when the number of iterations is 2, and is consistent with the performance of exact MMSE detector when the number of iterations is up to 3.

We use the number of multiplications to evaluate the computational complexity, and the comparison results are shown in Fig. 7. It can be observed that when the NSA term is larger than 2 and the iteration number k of NI and CI is more than 1, their computational complexity is even higher

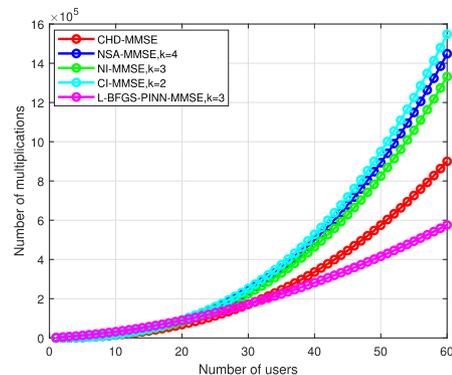


Fig. 7 Complexity comparison against the number of UEs.

than the CHD-based exact detector. For the proposed L-BFGS-PINN based detector, its complexity is lower than the CHD-based detector with $k = 3$ especially when the number of UEs is large, and its cost is lower than the other classical linear detectors. To sum up, the proposed PINN with the L-BFGS method can adjust the performance and complexity through the number of iterations for a better trade-off.

5. Conclusion

In this paper, we propose an efficient neural network to calculate the pseudo-inverse for any type of matrices, i.e. the PINN with the improved Newton's method, which performs better than that with the GD method in terms of convergence and complexity. Then linear massive MIMO detectors are mapped on the proposed PINN, which can give full play to the advantage and potential of AI learning algorithms and accelerators. To solve the Hessian matrix inversion involved in the improved Newton's method, L-BFGS method is further adopted as the learning algorithm of the PINN with reduced complexity. In our future work, we will design the efficient hardware correspondingly.

References

- [1] T.L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol.9, no.11, pp.3590–3600, 2010.
- [2] F. Rusek, D. Persson, B.K. Lau, E.G. Larsson, T.L. Marzetta, O. Edfors, and F. Tufvesson, "Scaling up MIMO: Opportunities and challenges with very large arrays," *IEEE Signal Process. Mag.*, vol.30, no.1, pp.40–60, 2013.
- [3] S. Yang and L. Hanzo, "Fifty years of MIMO detection: The road to large-scale MIMOs," *IEEE Commun. Surveys Tuts.*, vol.17, no.4, pp.1941–1988, 2017.
- [4] X. Zhu and R.D. Murch, "Performance analysis of maximum likelihood detection in a MIMO antenna system," *IEEE Trans. Commun.*, vol.50, no.2, pp.187–191, 2002.
- [5] L. Lu, G.Y. Li, A.L. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive MIMO: Benefits and challenges," *IEEE J. Sel. Topics Signal Process.*, vol.8, no.5, pp.742–758, 2014.
- [6] M. Wu, B. Yin, G. Wang, C. Dick, J.R. Cavallaro, and C. Studer, "Large-scale MIMO detection for 3GPP LTE: Algorithms and FPGA implementations," *IEEE J. Sel. Topics Signal Process.*, vol.8, no.5, pp.916–929, 2017.
- [7] H. Wang, Y. Ji, Y. Shen, W. Song, M. Li, X. You, and C. Zhang, "An efficient detector for massive MIMO based on improved matrix partition," *IEEE Trans. Signal Process.*, vol.69, pp.2971–2986, 2021.
- [8] X. Tan, J. Jin, K. Sun, Y. Xu, M. Li, Y. Zhang, Z. Zhang, X. You, and C. Zhang, "Enhanced linear iterative detector for massive multiuser MIMO uplink," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol.67, no.2, pp.540–552, 2020.
- [9] R. Guo, X. Li, W. Fu, and H. Yongqiang, "Low-complexity signal detection based on relaxation iteration method in massive MIMO systems," *China Commun.*, vol.12, no.1, pp.1–8, 2015.
- [10] M. Wu, B. Yin, K. Li, C. Dick, J.R. Cavallaro, and C. Studer, "Implicit vs. explicit approximate matrix inversion for wideband massive MU-MIMO data detection," *J. Sign. Process. Syst.*, vol.90, pp.1311–1328, 2018.
- [11] M. Wu, B. Yin, A. Vosoughi, C. Studer, J.R. Cavallaro, and C. Dick, "Approximate matrix inversion for high-throughput data detection in the large-scale MIMO uplink," 2013 IEEE International Symposium on Circuits and Systems (ISCAS), pp.2155–2158, IEEE, 2013.
- [12] S. Hashima and O. Muta, "Fast matrix inversion methods based on Chebyshev and Newton iterations for zero forcing precoding in massive MIMO systems," *EURASIP J. Wireless Com. Network.*, vol.2020, no.1, pp.1–12, 2020.
- [13] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol.61, pp.85–117, 2015.
- [14] C.B. Wu, Y.C. Hsueh, C.S. Wang, and Y.C. Lai, "High throughput hardware implementation for deep learning AI accelerator," 2019 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW), IEEE, 2019.
- [15] R. Li, Z. Zhao, X. Zhou, G. Ding, and Y. Chen, "Intelligent 5G: When cellular networks meet artificial intelligence," *IEEE Wireless Commun.*, vol.24, no.5, pp.175–183, 2017.
- [16] C.J. Chun, J.M. Kang, and I.M. Kim, "Deep learning-based channel estimation for massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol.8, no.4, pp.1228–1231, 2019.
- [17] M. Khani, M. Alizadeh, J. Hoydis, and P. Fleming, "Adaptive neural signal detection for massive MIMO," *IEEE Trans. Wireless Commun.*, vol.19, no.8, pp.5635–5648, 2020.
- [18] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Trans. Signal Process.*, vol.67, no.10, pp.2554–2564, 2019.
- [19] H. He, C.K. Wen, S. Jin, and G.Y. Li, "Model-driven deep learning for MIMO detection," *IEEE Trans. Signal Process.*, vol.68, pp.1702–1715, 2020.
- [20] I. Dokmanić and R. Gribonval, "Concentration of the Frobenius norm of generalized matrix inverses," *SIAM Journal on Matrix Analysis and Applications*, vol.40, no.1, pp.92–121, 2019.
- [21] X. Lv, L. Xiao, Z. Tan, Z. Yang, and J. Yuan, "Improved gradient neural networks for solving Moore–Penrose inverse of full-rank matrix," *Neural Process. Lett.*, vol.50, no.2, pp.1993–2005, 2019.
- [22] Y.J. Wang and C.T. Lin, "A second-order learning algorithm for multilayer networks based on block Hessian matrix," *Neural Networks*, vol.11, no.9, pp.1607–1622, 1998.
- [23] R.W.M. Wedderburn, "Quasi-likelihood functions, generalized linear models, and the Gauss-Newton method," *Biometrika*, vol.61, no.3, pp.439–447, 1974.
- [24] J.J. More, "The Levenberg-Marquardt algorithm: Implementation and theory," *Lecture Notes in Mathematics*, vol.630, pp.105–116, 1978.
- [25] A. Ben-Israel and T. Greville, "Generalized inverses: Theory and applications," *Journal of the Royal Statistical Society Series A (General)*, vol.139, no.1, 1976.
- [26] J. Nocedal, "Updating quasi-Newton matrices with limited storage," *Mathematics of Computation*, vol.35, no.151, pp.773–782, 1980.
- [27] M. Al-Baali, "Improved Hessian approximations for the limited memory BFGS method," *Numerical Algorithms*, vol.22, no.1, pp.99–112, 1999.
- [28] D.C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol.45, no.1, pp.503–528, 1989.
- [29] L. Li and J. Hu, "An efficient linear detection scheme based on L-BFGS method for massive MIMO systems," *IEEE Commun. Lett.*, vol.26, no.1, pp.138–142, 2022.
- [30] Y. Zhang, D. Guo, and Z. Li, "Common nature of learning between back-propagation and hopfield-type neural networks for generalized matrix inversion with simplified models," *IEEE Trans. Neural Netw. Learn. Syst.*, vol.24, no.4, pp.579–592, 2013.
- [31] X. Yang, W. Lu, N. Wang, K. Nieman, C.K. Wen, C. Zhang, S. Jin, X. Mu, I. Wong, Y. Huang, and X. You, "Design and implementation of a TDD-based 128-antenna massive MIMO prototype system," *China Commun.*, vol.14, no.12, pp.162–187, 2017.
- [32] L. Li and J. Hu, "Fast-converging and low-complexity linear massive MIMO detection with L-BFGS method," *IEEE Trans. Veh. Technol.*, vol.71, no.10, pp.10656–10665, 2022.



Lin Li received the B.E. and M.S. degrees in electronic information engineering and circuits and systems from XUT in 08 and 11, respectively. She received the Ph.D. degree in communication systems from UESTC in 22. From 11 to 14, she joined Nationz Technologies Inc. as an IC design engineer. She has been a lecturer at Qinghai Normal University since 15. Her research interests include wireless communication, high-speed DSP technology with VLSI, and artificial intelligence applications.



Jianhao Hu received the B.E. and Ph.D. degrees in communication systems from UESTC in 1993 and 1999 respectively. He joined the City University of Hong Kong from 1999 to 2000 as a postal doctor. From 00 to 04, he served as a senior system engineer at the 3G research center of the University of Hong Kong. He has been a professor of the national key lab. of UESTC since 05. His research interesting area includes high-speed DSP technology with VLSI, NoC and software radio.