Recurrent Sparse MIMO Detection Network Based on Modified Projected Gradient Descent Method

Lanxin He and Tao Liu
College of Electronic and Information Engineering,
Nanjing University of Aeronautics and Astronautics, Nanjing, China
Email:{lanxin_he, tliu}@nuaa.edu.cn

Zheng Wang
School of Information Science and Engineering
Southeast University, Nanjing, China
Email: z.wang@ieee.org

***Abstract*—Deep learning (DL) has emerged as a powerful tool for signal detection in large-scale multiple-input multiple-output (MIMO) systems. In this paper, the recurrent sparse detection network (RS-Net) is proposed for performance improvement and complexity reduction. First of all, in order to reduce complexity, RS-Net unfolds the projected gradient descent (PGD) method in a modified way, which consists of a projection and a gradient descent (GD) part. Meanwhile, an RNN with the parameter-sharing structure is adopted to its projection, which significantly eases the training burden. Then, to improve the detection performance, we regularize RS-Net by introducing sparse representation. Besides, the step size of iterations in GD part is also investigated for better convergence efficiency. Finally, simulations demonstrate a better decoding trade-off between performance and complexity in RS-Net.**

***Index Terms*—large-scale MIMO detection, projected gradient descent, recurrent neural network, sparse representation.**

I. Introduction

Nowadays, massive MIMO constitutes a breakthrough technology in 6G communication systems owing to its benefits in spectral efficiency, power consumption and link reliability [1]. Nevertheless, the dramatic increase in the number of antennas at the base station and terminal imposes a stressing challenge on the uplink signal detection of massive MIMO systems. Meanwhile, by virtue of abundant data and increased computing power, DL technology has brought unprecedented boosting in various fields such as computer vision, natural language processing, e-commerce and so on [2], which also interests researchers in its applications on MIMO detection.

Established by unfolding PGD method, the DetNet proposed by Samuel [3] is able to achieve better performance than minimum mean square error (MMSE) detector. Besides, in [4] and [5], ScNet and FS-Net introduce sparse connectivity by substituting the Hadamard product for the matrix multiplication in DetNet to alleviate the complexity. BD-Net [6] employs distinct bidirectional long short-term Memory (BiLSTM) units in DetNet to enhance the performance. Furthermore, by unfolding conjugated gradient descent (CGD) method, [7] proposes LcgNet and [8] designs an OAMP-Net by adding some trainable parameters in orthogonal approximate message passing (OAMP) algorithm.

In this paper, for performance improvement and complexity reduction, we propose the recurrent sparse detection network, RS-Net. Specifically, we firstly establish it by unfolding a modified PGD method consisting of a projection and a GD part, to reduce the cost of learning its projection. Secondly, concerning this projection, a parameter-sharing RNN is introduced to further reduce complexity. Then, sparse representation is employed to enhance performance. After that, the optimal choice of the step size with respect to GD part is developed. Simulations demonstrate that RS-Net achieves a better decoding trade-off.

Corresponding author: Zheng Wang (e-mail: z.wang@ieee.org)

II. DL-based MIMO detector

Considering the standard linear MIMO system with N_t transmit and N_r receive antennas, let $\tilde{\mathbf{x}} \in \mathcal{X}^{N_t}$ denote the complex-valued transmit signal vector, and the corresponding received signal vector $\tilde{\mathbf{y}} \in \mathbb{C}^{N_r}$ is given by

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\tilde{\mathbf{x}} + \tilde{\mathbf{n}}. \tag{1}$$

Here, $\mathcal{X}^{N_t} \subset \{\mathbb{Z}^{N_t} + j\mathbb{Z}^{N_t}\}$ is the set for all possible transmitted M-QAM symbols, $\tilde{\mathbf{H}} \in \mathbb{C}^{N_r \times N_t}$ is the channel matrix, $\tilde{\mathbf{n}} \in \mathbb{C}^{N_r}$ is the additive white Gaussian noise (AWGN) vector with zero mean and variance σ^2. This accounts for a real-valued $2N_r \times 2N_t$ system

$$\begin{bmatrix} \mathscr{R}(\tilde{\mathbf{y}}) \\ \mathscr{I}(\tilde{\mathbf{y}}) \end{bmatrix} = \begin{bmatrix} \mathscr{R}(\tilde{\mathbf{H}}) & -\mathscr{I}(\tilde{\mathbf{H}}) \\ \mathscr{I}(\tilde{\mathbf{H}}) & \mathscr{R}(\tilde{\mathbf{H}}) \end{bmatrix} \begin{bmatrix} \mathscr{R}(\tilde{\mathbf{x}}) \\ \mathscr{I}(\tilde{\mathbf{x}}) \end{bmatrix} + \begin{bmatrix} \mathscr{R}(\tilde{\mathbf{n}}) \\ \mathscr{I}(\tilde{\mathbf{n}}) \end{bmatrix}, \tag{2}$$

which can be further expressed as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}. \tag{3}$$

Then, the problem of MIMO detection becomes

$$\widehat{\mathbf{x}} = \underset{\mathbf{x} \in \mathcal{A}^{2N_t}}{\arg\min} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2, \tag{4}$$

where $\mathcal{A} = \{\pm 1, \pm 3, ..., \pm\sqrt{M} - 1\}$ with $\sqrt{M}$ representing the modulation index of the corresponding real-valued ASK. Based on PGD method, which is essentially a GD method assisted with a projection operation, the DL technique is introduced to MIMO detection by Samuel [3]. Specifically, GD method performs the optimization in an iterative way as

$$\mathbf{x}_t = \mathbf{x}_{t-1} - \delta \nabla f_{\text{objective}}(\mathbf{x}_{t-1}). \tag{5}$$

Here, $\nabla f_{\text{objective}}$ is the gradient of the objective function, δ is the step size, t is the iteration index. Then, by letting $f_{\text{objective}} = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2$, the iteration with respect to $\mathbf{x}$ in (4) becomes

$$\mathbf{x}_t = \mathbf{x}_{t-1} + 2\delta(\mathbf{H}^T\mathbf{y} - \mathbf{H}^T\mathbf{H}\mathbf{x}_{t-1}), \tag{6}$$

2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall) | 978-1-6654-1368-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/VTC2021-FALL52928.2021.9625500

which achieves the same detection performance with the zero forcing (ZF) detector. Similarly, if $\mathbf{H}$ and $\mathbf{y}$ are augmented by $\underline{\mathbf{H}} = [\mathbf{H}, \sigma\mathbf{I}_{N_t}]^T$ and $\underline{\mathbf{y}} = [\mathbf{y}; \mathbf{0}_{N_t\times 1}]$ respectively, the performance of MMSE detector can be obtained [9].

Based on the GD iterations in (6), by employing a projection operation $\mathcal{P}$ to tune each iteration as

$$\mathbf{x}_t = \mathcal{P}(\mathbf{x}_{t-1} - \delta_{1t}\mathbf{H}^T\mathbf{y} + \delta_{2t}\mathbf{H}^T\mathbf{H}\mathbf{x}_{t-1}), \tag{7}$$

PGD method is introduced by Samuel. Here the step-sizes δ_{1t} and δ_{2t} are treated independently. Specifically, a deep neural network (DNN) named as DetNet is designed to find the projection $\mathcal{P}$ and the step-sizes δ_{1t}, δ_{2t} that minimize the following loss function

$$f_{\text{loss}} = \sum_{t=1}^{L} \log(t)\|\mathbf{x}_m - \mathbf{x}_t\|^2. \tag{8}$$

Here $\mathbf{x}_m$, also referred to as the label vector in DL, is the modulated transmitted signal, $\mathbf{x}_t$ is the point outputted by the projection in the t-th iteration and L is the total iteration in DetNet. The weighted structure in (8) is adopted to involve outputs as mush as possible in back-propagation (BP) procedure so that the vanishing gradient problem is less likely to occur. Typically, at t-th iteration, a GD step is firstly applied in DetNet to output a point $\mathbf{x}_t$. Based on it, then the estimated $\widehat{\mathbf{x}}_t$ is returned by

$$\widehat{\mathbf{x}}_t = \mathcal{P}(\mathbf{x}_t) = f_{oh}\left[\mathcal{T}_{2t}(\text{ReLU}(\mathcal{T}_{1t}([\mathbf{x}_t, \mathbf{h}_t])))\right] \tag{9}$$

with an auxiliary vector $\mathbf{h}$ updated as

$$\mathbf{h}_{t+1} = \mathcal{T}_{3t}\left[\text{ReLU}(\mathcal{T}_{1t}([\mathbf{x}_t, \mathbf{h}_t]))\right] \tag{10}$$

to pass information within network. Here, $[\cdot]$ is the concatenation operation, f_{oh} is the mapping function to transform the one-hot vector into the scalar estimate, $\mathcal{T}_{it}$ is the i-th fully connected layer in the t-th iteration of DetNet. To be specific, these trainable layers can be expressed as

$$\mathcal{T}_{it}(\mathbf{x}) = \mathbf{W}_{it}\mathbf{x} + \mathbf{b}_{it}, \quad i = 1, 2, 3, \ t = 1, 2, ..., L, \tag{11}$$

where $\mathbf{W}_{it}$ and $\mathbf{b}_{it}$ are the weights and bias needed to quantify during the training phase. We denote the overall trainable parameters in DetNet as

$$\theta_{Det} = \{\mathbf{W}_{1t}, \mathbf{b}_{1t}, \mathbf{W}_{2t}, \mathbf{b}_{2t}, \mathbf{W}_{3t}, \mathbf{b}_{3t}, \delta_{1t}, \delta_{2t}\}_{t=1,2,...,L}. \tag{12}$$

In the training step of DetNet, BP algorithm tunes θ_{Det} to minimize the loss function in (8), achieving a better detection performance than MMSE.

III. Proposed RS-Net

Deriving from PGD, we now give the proposed recurrent sparse detection network (RS-Net). Similar to DetNet, RS-Net also contains a GD part and a projection part, but here we use RNN rather than DNN to implement the projection $\mathcal{P}$. The RNN is a classic DL model capturing the dynamics of sequences via cycles in the network. Unlike standard feedforward DNN, it retains a state $\mathbf{h}$ that can represent information from an arbitrarily long context window [10]. This state $\mathbf{h}$ can be utilized to preserve the potential important hidden information in an iterative procedure. Based on it, there are three main modifications in the proposed RS-Net.

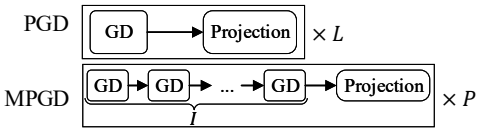


Fig. 1. PGD (upper) and MPGD (bottom) method

A. Modified PGD Method

To reduce the cost of learning and implementing the projection $\mathcal{P}$, RS-Net is unfolded in a modified PGD (MPGD) way. In detail, as shown in Fig.1, MPGD method only performs one-time projection for every I steps of GD and repeats this iteration up to P times. This modification comes from the fact that the point outputted by the projection $\mathcal{P}$ would be quite close to the constellations because of f_{loss} in (8). Therefore, as visualized in Fig.2, the frequent projections after each single step of GD actually might make little improvement. Meanwhile, simulation shows that DetNet unfolded in MPGD method has a comparable performance with the original but taking less iterations ($P < L$). Hence, MPGD is a more reasonable method that saves the computation by projection.

Note that this modification increases the complexity of implementing a single iteration, but we will point out in IV-B that this increment in complexity is negligible in some cases. Besides, with respect to the step-size δ in GD, we recommend to predefine it as a constant rather than train it like DetNet, so that RS-Net can focus on the task of predicting the projection $\mathcal{P}$. This is also a reduction on training burden and we will give it a detailed explanation in IV-A.

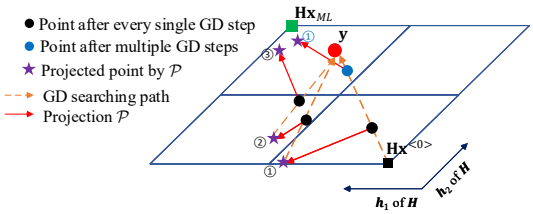


Fig. 2. Projecting after each single GD step takes three iterations to approach ML point, while projection after multiple GD steps needs only one iteration.

B. Adoption of RNN

With respect to the projection part in MPGD method, we propose to adopt an RNN cell that can share the parameters during iterations. In this way, all we need to predict is only the single projection operator $\mathcal{P}$ and the training burden can be further reduced. Specifically, as presented in Fig.3, given the current point $\mathbf{x}$ after GD steps and hidden state $\mathbf{h}$, the procedure in the RNN cell can be outlined as follows

$$\begin{aligned}
&\mathbf{z} = \mathcal{T}_h([\mathbf{x}, \mathbf{h}]) = \mathbf{W}_h[\mathbf{x}, \mathbf{h}] + \mathbf{b}_h,\\
&\mathbf{z}' = \text{ReLU}(\mathbf{z}),\\
&\mathbf{h} = \mathbf{z}',\\
&\widehat{\mathbf{x}} = \mathcal{T}_x(\mathbf{z}') = \mathbf{W}_x\mathbf{z}' + \mathbf{b}_x.
\end{aligned} \tag{13}$$

Here the transformation $\mathcal{T}_h : \mathbb{R}^{2N_t+KN_t} \to \mathbb{R}^{KN_t}$ represented by $\mathbf{W}_h$ and its bias $\mathbf{b}_h$, in addition to $\mathcal{T}_x : \mathbb{R}^{KN_t} \to \mathbb{R}^{2N_t}$ by $\mathbf{W}_x$ and $\mathbf{b}_x$ is the trainable component in the expected

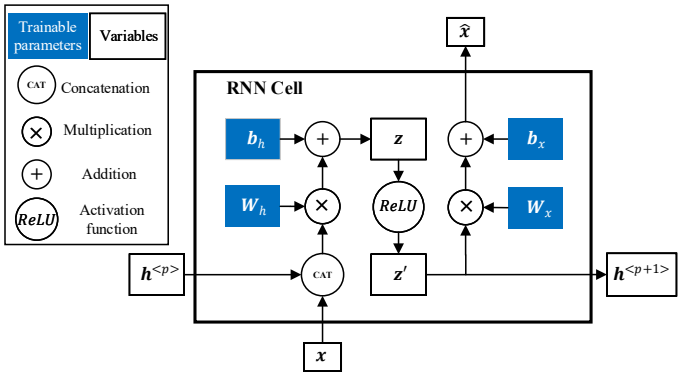


Fig. 3. Structure of the RNN cell in RS-Net.

$\mathcal{P}$. $\mathbf{z}$ as well as $\mathbf{z}'$ is the higher-dimension point in a new domain $\mathcal{I} \subset \mathbb{R}^{KN_t}$, where K is a hyper-parameter controlling the dimension of $\mathcal{I}$. In short, the unique projection operator $\mathcal{P}$ in RS-Net has been modified to

$$\mathcal{P}(\mathbf{x}) = \mathcal{T}_x(\text{ReLU}(\mathcal{T}_h([\mathbf{x}, \mathbf{h}]))) \tag{14}$$

with the hidden state $\mathbf{h}$ updated as

$$\mathbf{h} = \text{ReLU}(\mathcal{T}_h([\mathbf{x}, \mathbf{h}])). \tag{15}$$

Compared to (9) and (10) in DetNet, RS-Net implies a noticeable reduction in complexity since all the parameters required to quantify during training

$$\theta = \{\mathbf{W}_h, \mathbf{b}_h, \mathbf{W}_x, \mathbf{h}_x\}, \tag{16}$$

are shared within the recurrent network. Such a reduction in complexity inevitably results in the performance loss, but this parameter-sharing structure does provide a possible way to the realization of extremely simple architecture. What is more, it will be shown in simulation that the performance loss resulted from this utilization of a single RNN is acceptable, considering the noticeable reduced complexity by it. Meanwhile, simulation also shows that the point $\hat{\mathbf{x}}$ returned by the single RNN is quite close to constellations. Therefore the one-hot encoding and mapping function f_{oh} in (9) can also be further eliminated.

C. Sparse Representation

As for the detection performance, we regularize the RNN cell by introducing sparse representation. This regularization can further reduce the connections required to train, but more importantly, it would make the network more robust to noise and thus enhance the performance. According to [13], two computationally efficient modifications are required to our case. First, let $|\cdot|$ measure the total amount of elements and S denote the fraction of non-zero weights. Then only $S|\mathbf{W}_h|$ elements in $\mathbf{W}_x$ and $S|\mathbf{W}_x|$ in $\mathbf{W}_x$ are initialized non-zero and those zero-value connections are treated as non-existent during the training and detection. Second, the ReLU function is substituted by a k-winners layer in [13], which maintains only the top-k ($k = S|\mathbf{z}|$) active neurons in its output. Fig 4 visualizes this modification and the overall RS-Net is presented in Algorithm 1.

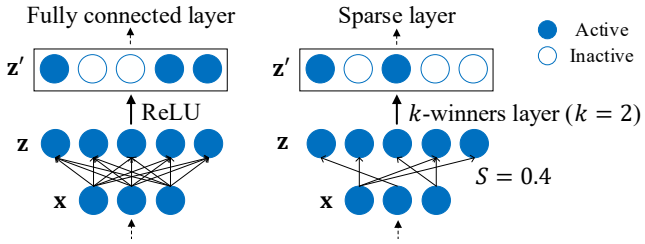


Fig. 4. Differences between a standard fully connected layer (left) and a sparse layer adopted in RS-Net (right).

Algorithm 1: RS-Net

Input I, P, $\mathbf{H}$, $\mathbf{y}$, S, δ, θ, k-$winners$ layer in [13]
Output $\hat{\mathbf{x}}$: estimated transmit signal

Initialize: set $\mathbf{x}^{<0>}$ as a random point in $\mathcal{A}^{2N_t}$; $\mathbf{h} = \mathbf{0}_{KN_t}$
for $p = 1, 2, ..., P$ **do**
  for $i = 1, 2, ..., I$ **do**
    $\mathbf{x}^{<i>} = \mathbf{x}^{<i-1>} + 2\delta(\mathbf{H}^T\mathbf{y} - \mathbf{H}^T\mathbf{H}\mathbf{x}^{<i-1>})$
  end for
  $\mathbf{h} = k\text{-}winners(\mathbf{W}_h[\mathbf{x}^{<I>}, \mathbf{h}] + \mathbf{b}_h)$
  $\hat{\mathbf{x}} = \mathbf{W}_x\mathbf{h} + \mathbf{b}_x$
  $\mathbf{x}^{<0>} = \hat{\mathbf{x}}$
end for

IV. Algorithm Analysis

A. Choice of step size

The success of GD depends on effective choice of the step size δ. Instead of obtaining step-sizes by training as in DetNet, we recommend to predefine δ as a constant, so that RS-Net is less likely to be disturbed by the changeable step-sizes during training. In order to develop a reasonable choice of δ, we explicate the role of it by induction over t in (6):

$$\begin{aligned}\hat{\mathbf{x}}^{<t>} &= (\mathbf{I} - 2\delta\mathbf{K})\mathbf{x}^{<t-1>} + 2\delta\mathbf{k} \\ &= (\mathbf{I} - 2\delta\mathbf{K})^t\mathbf{x}^{<0>} + \sum_{i=0}^{t-1}(\mathbf{I} - 2\delta\mathbf{K})^i(2\delta\mathbf{k}),\end{aligned} \tag{17}$$

where $\mathbf{K}$ and $\mathbf{k}$ denote $\mathbf{H}^T\mathbf{H}$ and $\mathbf{H}^T\mathbf{y}$ respectively. Now we consider $\mathbf{G} = \mathbf{I} - 2\delta\mathbf{K}$. According to the theory of matrix power series [11], for a $N_t \times N_t$ matrix $\mathbf{G}$, $\mathbf{G}^t$ converges to $\mathbf{0}_{N_t}$ when the spectral radius of $\mathbf{G}$, denoted by $\rho(\mathbf{G})$, is less than 1, which means the following inequality needs to be held:

$$\rho(\mathbf{G}) = \rho(\mathbf{I} - 2\delta\mathbf{K}) < 1. \tag{18}$$

In this case, (17) gives a simple form: $\hat{\mathbf{x}}^{<t>} = \mathbf{K}^{-1}\mathbf{k} = \mathbf{x}_{\text{ZF}}$, corresponding to the mentioned fact that GD method obtains the same detection performance with ZF detector. Moreover, satisfying (18) is equivalent to bound the absolute value of every eigenvalue of $\mathbf{G}$, $\lambda(\mathbf{G})$, under 1. Then (18) changes to

$$|1 - 2\delta\lambda(\mathbf{K})| < 1 \implies 0 < \delta\lambda(\mathbf{K}) < 1. \tag{19}$$

Since $\mathbf{K}$ is positive definite [11] with positive eigenvalues, this convergence only requires $\delta > 0$ and $\delta\lambda_{max}(\mathbf{K}) < 1$. Meanwhile it is derived in [12] that as N_r grows, the largest

eigenvalues of the complex Wishart matrix $\mathbf{K}$ converge *in probability* to

$$\lambda_{max}(\mathbf{K}) \xrightarrow{p} N_r(1+1/\sqrt{\alpha})^2, \tag{20}$$

where $\alpha = N_r/N_t$ is the antenna ratio. Therefore, δ can be utilized to guarantee the convergence:

$$0 < \delta < \frac{1}{N_r(1+1/\sqrt{\alpha})^2}. \tag{21}$$

It is worth noticing that (21) is only the necessary condition, since $\lambda_{max}(\mathbf{K})$ may fluctuate if N_r is not large enough. To put it concisely, it is highly recommended that the step-size δ be set as close to the upper bound in (21) as possible, as long as the exploding gradient is not observed during training (the real-time loss value increases from the beginning and even overflows). Furthermore, we point out that in MPGD method, those GD procedures separated by the projection can be treated independently without considering the projection part as we do. Also, if a trainable step-size vector $\boldsymbol{\delta}$ as in DetNet is expected, (21) can still be used to initialize $\boldsymbol{\delta}$ following the uniform distribution $\mathcal{U}(0, \frac{1}{N_r(1+1/\sqrt{\alpha})^2})$. Whereas it will be shown in our simulation that this approach makes the model more sensitive to the training data and less robust.

B. Complexity Analysis

In the sequel we present the training burden, or the space complexity, of RS-Net and its time complexity analysis. The trainable parameters in RS-Net stem from the single RNN cell,

$$\begin{aligned}|\theta| &= \underbrace{KN_t \times (2N_t + KN_t)}_{\mathbf{W}_h} + \underbrace{KN_t}_{\mathbf{b}_h} + \underbrace{2N_t \times KN_t}_{\mathbf{W}_x} + \underbrace{2N_t}_{\mathbf{b}_x} \\ &= \mathcal{O}(K^2N_t^2).\end{aligned} \tag{22}$$

While for DetNet [3], also letting K denote the coefficient in the size of its higher dimension, the total is

$$\begin{aligned}|\theta_{Det}| = \big[&\underbrace{KN_t \times (2N_t + KN_t)}_{\mathbf{W}_{1t}} + \underbrace{KN_t}_{\mathbf{b}_{1t}} + \underbrace{2N_t|\mathcal{A}| \times KN_t}_{\mathbf{W}_{2t}} \\ &+ \underbrace{2N_t|\mathcal{A}|}_{\mathbf{b}_{2t}} + \underbrace{KN_t \times KN_t}_{\mathbf{W}_{3t}} + \underbrace{KN_t}_{\mathbf{b}_{3t}} + \underbrace{2}_{\delta_t}\big]L \\ &= \mathcal{O}(K^2N_t^2L),\end{aligned} \tag{23}$$

where $|\mathcal{A}|$ measures the cardinality of $\mathcal{A}$. It conveys that in RS-Net, since the amount of trainable weights is independent of iterations, the training burden is significantly eased. Also note that $|\theta|$ in RS-Net is actually even lower than (22) owing to sparse representation, where those zero-value connectivities are excluded during training. For instance, the DetNet shown in Fig. 7 has 217950 parameters to quantify while RS-Net ($K = 8$, $KS = 3.2$) has only 9990.

The time complexity of RS-Net is as follows. The complexity of calculating a GD step in (6) is of order $\mathcal{O}(N_rN_t)$, while in the projection part, the complexity required to compute (13) is of order $\mathcal{O}(K^2N_t^2)$. In total, the complexity of RS-Net is $\mathcal{O}(K^2N_t^2)$. This is acceptable under the background of massive MIMO, where the influence of hidden-size-coefficient K is less significant compared to the large-scale amount of antennas. Besides, we point out that under some circumstances, MPGD method has nearly the same complexity with PGD. For simplicity we assume $N_t = N_r$. Then, when I in MPGD is less than K^2, the order of implementing I GD steps, $\mathcal{O}(IN_t^2)$, is still lower than that of applying one-time projection, $\mathcal{O}(K^2N_t^2)$. Therefore, it can be concluded that if $I < K^2$, MPGD method is no more complex than PGD.

V. Numerical Results

A. Training Details

In accordance with the analysis under backpropagation through time (BPTT) in [14], we adopt the same weighted loss function as (8) to involve each iteration in BP process as much as possible and make the vanishing gradient problem less likely to occur. If not specified, we use $I = 10$ and $P = 4$ for RS-Net and all GD baselines are carried out by $I \times P = 40$ iterations with $\underline{\mathbf{H}}$ and $\underline{\mathbf{y}}$. Besides, we set $\alpha = 0.1$ and $\beta = 1.5$ in k-winners layer. The training procedure works on the DL library PyTorch. We draw on 150,000 noise-free data ($\mathbf{y} = \mathbf{Hx} + \mathbf{0}$) and train RS-Net with Adam Optimizer [15] using a batch size of 100. For other competing networks, the noise variance is randomly generated within Eb/N0 $\sim \mathcal{U}([\text{Eb/N0}]_{\min} - 1, [\text{Eb/N0}]_{\max} + 1)$, where $[\cdot]_{\min}$ and $[\cdot]_{\max}$ take the minimal and maximal values used in detection. The learning rate is set as 0.001 and would decay by 0.97 after each epoch.

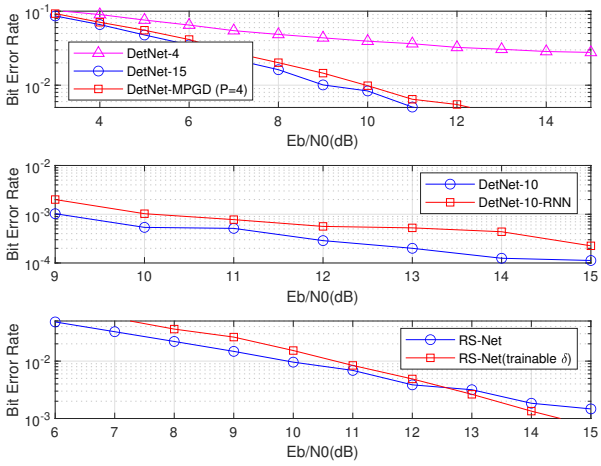


Fig. 5. Performance comparison under QPSK scheme of size 16×16 (upper) and 30×20 (middle and bottom).

B. BER Performance

The BER performance is presented with all the experiments conducted assuming i.i.d Rayleigh MIMO channels. Fig. 5 shows that our modifications in DetNet are reasonable. On one hand, if the PGD method in a 4-iterations DetNet ($L = 4$) is replaced by MPGD ($I = 10, P = 4$), a significant performance gain can be obtained owing to the increment of GD steps. Meanwhile, if the iteration of DetNet is increased to 15 ($L = 15$), the MPGD-based DetNet still shows a comparable performance with less projections ($P < L$). On the other hand, substituting a single RNN for the DNN in DetNet only leads

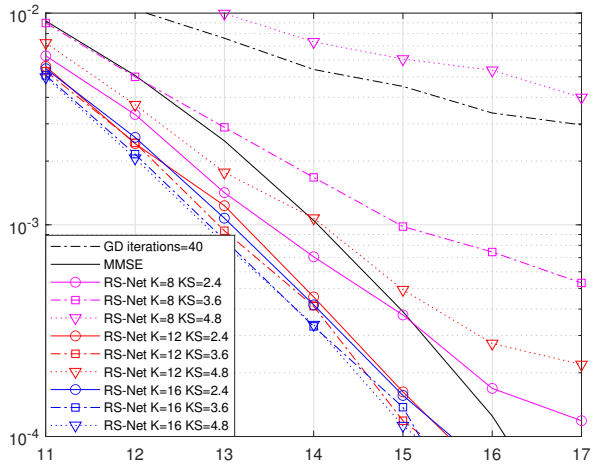


Fig. 6. Comparison of BER performance for 30×20 MIMO system using 16QAM, all related step sizes δ are set as 0.008.

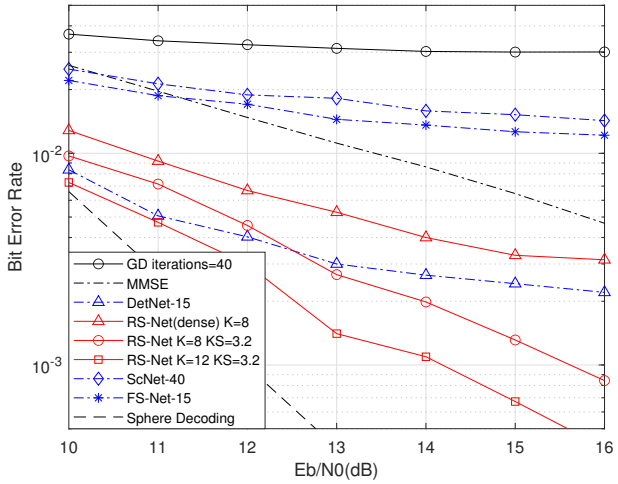
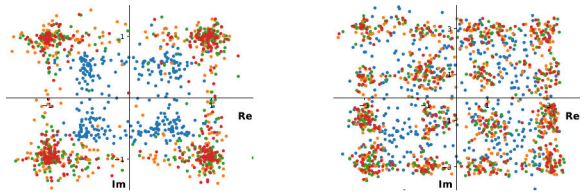


Fig. 7. Comparison of BER performance for 16×16 MIMO system using QPSK, all related step sizes δ are set as 0.012; RS-Net(dense) represents the dense version of RS-Net with $S = 1$.

to an insignificant performance loss. Moreover, the model using trainable step-sizes performs only well when test data is similar to the training set, which is undesirable in most cases. The performance of models with different hidden sizes and sparsities are shown in Fig. 6, where we simulate three-level-scale networks, $K = 8$, $K = 12$ and $K = 16$ with three different numbers of active connections, $KS = 2.4$, $KS = 3.6$ and $KS = 4.8$. It can be concluded that BER drops rapidly as the underlying dimensionality KN_t increases or as the model turns more sparse, but this improvement would saturate.

Fig. 7 demonstrates the decoding trade-off in RS-Net under QPSK scheme. It suggests that the utilization of sparse representation results in a noticeable enhancement compared to dense version, and shows comparable performance with a 15-iterations DetNet. In addition, with the active connectivity (KS) in RS-Net fixed, a further performance improvement can be obtained if K increases to 12. What is more, we simulate another two detection networks, ScNet [4] and FS-Net [5] with 40 and 15 iterations respectively. Both of them reduce their complexity by applying only element-wise product but



(a) QPSK (b) 16QAM

Fig. 8. Projection methods in a batch, where blue, orange, green and red dots stand for the first, second, third and forth projection, respectively.

both perform less competitively than RS-Net. The performance of nonlinear sphere decoding is presented as well. Fig. 8 visualizes the distributions of outputs after each projection, which validates that points can be nearly transformed onto the constellations by projection. Note that the bad behavior in the first projection (blue dots) is from its direct absence in loss function (the weight before the first projection in (8) is $\log(1) = 0$) and has negligible influence on final results.

VI. Acknowledgment

This work was supported in part by National Natural Science Foundation of China under Grants No. 61801216, Natural Science Foundation of Jiangsu Province under Grant No. BK20180420, State Key Laboratory of Integrated Services Networks (Xidian University) under Grant No. ISN21-31.

References

[1] E. G. Larsson, F. Tufvesson, O. Edfors, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 186–195, 2014.

[2] F. Emmert-Streib, Z. Yang, H. Feng, S. Tripathi, and M. Dehmer, "An introductory review of deep learning for prediction models with big data," *Frontiers in Artificial Intelligence*, vol. 3, p. 4, 2020.

[3] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2554–2564, 2019.

[4] G. Gao, C. Dong, and K. Niu, "Sparsely connected neural network for massive MIMO detection," in *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, 2018, pp. 397–402.

[5] N. T. Nguyen and K. Lee, "Deep Learning-Aided Tabu Search Detection for Large MIMO Systems," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 4262-4275, 2020.

[6] J. Yang and H. Li, "Deep learning based detection and channel tracking for mimo systems," in *2020 5th International Conference on Computer and Communication Systems (ICCCS)*, 2020, pp. 635–639.

[7] Y. Wei, M.-M. Zhao, M. Hong, M.-J. Zhao, and M. Lei, "Learned conjugate gradient descent network for massive mimo detection," *IEEE Transactions on Signal Processing*, vol. 68, pp. 6336–6349, 2020.

[8] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Model-driven deep learning for mimo detection," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1702–1715, 2020.

[9] J. Nocedal and S. Wright, "Numerical optimization," *Springer*, 2006.

[10] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint*, arXiv:1506.00019, 2015.

[11] R. A. Horn and C. R. Johnson, "Topics in Matrix Analysis," *Cambridge University Press*, 1991.

[12] A. Edelman, "Eigenvalues and condition numbers of random matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 9, no. 4, pp. 543–560, 1988.

[13] S. Ahmad and L. Scheinkman, "How can we be so dense? the benefits of using highly sparse representations," *CoRR*, vol. abs/1903.11257, 2019. [Online]. Available: http://arxiv.org/abs/1903.11257

[14] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," *International conference on machine learning*, pp. 1310–1318, 2013.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980, 2014.