Received: 24 September 2019 | Revised: 4 June 2020 | Accepted: 28 September 2020

IET Communications

DOI: 10.1049/cmu2.12076

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

Early results on deep unfolded conjugate gradient-based large-scale MIMO detection

Messaoud Ahmed Ouameur | **Daniel Massicotte**

Department of Electrical and Computer Engineering, Laboratoire des Signaux et Systèmes Intégrés, Chaire de recherche sur les signaux et l'intelligence des systèmes hautes performances, Université du Québec à Trois-Rivières, Trois-Rivières, QC, Canada

Correspondence
Daniel Massicotte, Department of Electrical and Computer Engineering, Laboratoire des Signaux et Systèmes Intégrés, Université du Québec à Trois-Rivières, 3351, Boul. des Forges, Trois-Rivières, Québec, Canada
Email: daniel.massicotte@uqtr.ca

Abstract

Deep learning (DL) is attracting considerable attention in the design of communication systems. This paper derives a deep unfolded conjugate gradient (CG) architecture for large-scale multiple-input multiple-output detection. The proposed technique combines the advantages of a model-driven approach in readily incorporating domain knowledge and deep learning in effective parameters learning. The parameters are trained via back-propagation over a data flow graph inspired from the iterative conjugate gradient method. We derive the closed-form expressions for the gradients for parameters training and discuss early results on the performance in a statistically identical and independent distributed channel where the training overhead is considerably low. It is worth noting that the loss function is based on the residual error that is not an explicit function of the desired signal, which makes the proposed algorithm blind. As an initial framework, we will point to the inherent issues and future directions.

1 | INTRODUCTION

Propagation channel modelling, hardware imperfections and design of optimal signalling and detection schemes to ensure reliable communication links are becoming mature subjects in communication system design. In order to provide tangible benefits, any machine learning (ML)- or deep learning (DL)-based approach must pass a high bar of performance [1]. Zappone et al. [2] have provided a thorough discussion on ML-based approaches for wireless communication networks' design and operation, whereas Björnson et al. [3] have envisioned to use ML as an instrumental tool to enable a truly intelligent massive multiple-input multiple-output (MIMO). Both works agreed on the fact that the grand question is not whether ML will be integrated but rather how and when this integration will be implemented. Large-scale MIMO detection is one of the main disruptive technology directions for 5G [3, 4]. In fact, massive MIMO has now made its way to 5G as one of the means to substantially improve both spectral and energy efficiencies [3]. As a matter of fact, base stations (BSs) with 64 fully digital transceiver chains are commercially deployed and the key component of massive MIMO has made its way into the 5G standard [4, 5]. Nevertheless, Qu et al. [6] have pointed out that massive MIMO implementation continues to be at least as exciting as massive MIMO theory. Massive MIMO is a form of multiuser MIMO where the number of serving antennas at the base transceiver station (BS) is an order of magnitude larger than the number of user terminals served within each radio resource element. Given a large number of antennas, reliance on time division duplex (TDD) channel reciprocity is essential [3].

Because of its advantages in terms of very high spectral efficiency (sum rates), increased reliability and power efficiency, massive MIMO has been the subject of a large number of research activities [7]. Under favourable channel conditions and/or as the number of antennas increases, the users' channels are mutually orthogonal which makes linear processing based on maximum ratio combining (MRC), zero-forcing (ZF) detection or minimum mean squared error (MMSE) detection, a suitable and optimal choice [7–9]. Many works on linear and low complexity processing are proposed in [9–12], which raise the performance bar to pass even higher. The detection/precoding problem based on ZF or MMSE technique is an arithmetic operation with cubic computational complexity in the order of the matrix dimension. To reduce the implementation complexity, matrix inversion approximations such as Neumann series expansion (NSE) is proposed [9]. A technique based on Gauss-

Seidel (GS) was shown to outperform NSE due to its fast convergence at considerably low computational complexity [10]. However, this comes at the expense of higher latency and lower throughput [10]. It has actually been shown that the NSE performance degrades as the number of UTs increases [11]. To counter the load increase effect, GS can still afford using more iterations while maintaining lower computational complexity, albeit at the expense of reduced throughput [10]. It has, therefore, been argued to resort to exact matrix inversion [12]. On the other hand, it has also been argued that these centralized processing techniques still impose stringent constraints on the interconnects' bandwidth between the massive MIMO radio heads and the central processing unit. Distributed, or decentralized, massive MIMO processing has been introduced to overcome such limitations [13, 14]. On the other hand, to support ultra-reliable low-latency communications, low latency and high throughput processing is required. As such, we introduce a recursive Gram matrix inversion update method wherein the inversion of the Gram matrix is performed by exploiting matrix inversion update of a matrix in the form of $\mathbf{H}^H\mathbf{H}$ when a new column is added/updated to a complex-valued matrix $\mathbf{H}$ [12].

It is worth mentioning that these signal detection schemes can be readily used for downlink precoding in a single cell scenario. In a multi-cell scenario, the reader is referred to [15–18]. In [18], Kazemi et al. considered a multi-cell scenario with noisy CSI where a fully cooperative cellular structure is presented first, and then, to mitigate the overhead, a limited cooperation setting, where the amount of the exchanged CSI among the cells is significantly decreased, is proposed. On the other hand, a low overhead centralized construction for constant envelop precoding (CEP), which employs limited cooperation among the cells while providing higher system throughput, is discussed in [15]. To achieve the minimum feedback overhead, a distributed realization of CEP is proposed. Furthermore, a new optimization problem is solved to compensate for the effects of pilot contamination [15, 16].

Recent contributions seem to advocate for the potentials of using DL for communication system design [1, 19–21]. Even if most signal processing algorithms have solid well-established roots in statistics and information theory for tractable mathematical models, it remains that a practical system has many impairments and non-linearities, which can be roughly captured by such models [22]. For this reason, a DL-based communications system, which is tailored for a specific hardware configuration and channel, might be able to better optimize in the presence of such impairments. It has been shown that neural networks (NNs) are universal function approximators and has shown a notable ability for algorithmic learning [23]. As such, some initial insights and findings, using state-of-the-art DL tools, on signal compression [20] and channel decoding [21] are revealed. On the other hand, massively parallel processing architectures, such as graphic processing units, have shown to be very energy efficient with remarkable computational capabilities when fully exploited by concurrent algorithms [24].

So far, the goal in introducing DL is to either improve parts of existing algorithms or to completely replace them with an end-to-end approach [25, 26]. As an example, O'Shea and Hoydis [1] have discussed several promising new applications of DL to the physical layer. They have introduced a new way of addressing a communication system as an end-to-end reconstruction optimization task using autoencoders. On the other hand, two different deep architectures for point-to-point MIMO detection are introduced in [19] wherein the promising architecture relies on unfolding the iterations of a projected gradient descent algorithm into a network.

Despite the historical context and related works (refer to the introduction section of [1] and [19]), our primary approach is relying on the deep unfolding of existing iterative algorithms by mainly interpreting every iteration as a set of layers. The deep unfolding of existing iterative algorithms is discussed in [27]. It has been recently applied in the context of MIMO detection and channel decoding in [19] and [28], respectively.

There is a general consensus that ultra-large-scale MIMO, at the infrastructure level, and machine-learning aided physical layer algorithms, at the protocol/algorithmic level, are key enablers for the next generation of communication systems [7] and [32]. There is also a common interest to investigate different ML and DL architectures (such as deep NNs-DNN, deep unfolding, etc.) and framework (such as reinforcement learning, transfer learning, etc.) for the sole purpose to gain more insights on the potential of leveraging these tools and frameworks to outperform the baseline models [29, 33]. As a major contribution, we propose a basic deep unfolded (model-driven approach) conjugate gradient (CG) architecture wherein the parameters are learned via backpropagation. We first unfold the iterative CG method to infer the data flow graph over which the gradient is computed. We then adopted a loss function based on the residual error which is not an explicit function of the training data. As such, the architecture exhibits following key features.

1. The parameters are trained rather than being explicitly computed using the data flow graph based on unfolding the iterative CG algorithm. The closed-form expressions of the gradients of the loss function with respect to the parameters are derived. This enables (for future works) the use of state-of-the-art methods mainly used in transparent ML for interpreting and understanding such networks [35].
2. The loss function is set to be the squared norm of the residual error term, which is *not an explicit function of the training data*. Therefore, the approach is *blind* so that no explicit and dedicated training data is required. Nevertheless, the appendix addresses another approach where the loss function is an explicit function of the training data.
3. The architecture can be readily incorporated as part of an end-to-end communication system learning process where the propagation of the gradient is seamlessly supported (see our work [34]). This is in line with the visionary statement in [2] wherein it is argued that the optimal design of smart radio environments needs to be tackled by taking the benefits of both model-based and data-driven (or simulation-driven) approaches and by leveraging the concept of transfer learning [34]. In addition, the regularity of the network lends itself to transfer learning to update the last stages only to enable on-line training [33].

4. The deep unfolding can be seen as a means of incorporating domain knowledge experts *in order to aid in speeding up the training phase*. For instance, Nachmani et al. [29] used radio transformer network (RTN) as domain knowledge experts to compensate for the carrier frequency offset. It is argued that the benefit of such an arrangement may be a reduced complexity and more flexibility regarding imprecise knowledge about the channel [29].

Notations: This paper adopts the following notations: $(\cdot)^{\mathrm{H}}$ represents the Hermitian transpose operator while $(\cdot)^{\mathrm{T}}$ and $(\cdot)^{-1}$ represent the transpose and the matrix inverse operators, respectively. Matrices and column-vectors are denoted by boldface capital and boldface small letters, respectively. We convert a $K \times K$ complex-valued matrix $\bar{\mathbf{A}} \in \mathbb{C}^{K\times K}$ to a $2K \times 2K$ real-valued one $\mathbf{A} \in \Re^{2K\times 2K}$ using the following transformation $\mathbf{A} = \begin{bmatrix} \mathrm{Re}(\bar{\mathbf{A}}) & -\mathrm{Im}(\bar{\mathbf{A}}) \\ \mathrm{Im}(\bar{\mathbf{A}}) & \mathrm{Re}(\bar{\mathbf{A}}) \end{bmatrix} \in \Re^{2K\times 2K}$. Similarly, we convert a $K \times 1$ complex-valued vector $\bar{\mathbf{s}} \in \mathbb{C}^{K\times 1}$ to a $2K \times 1$ real-valued vector $\mathbf{s} \in \Re^{2K\times 1}$ using the following transformation $\mathbf{s} = \begin{bmatrix} \mathrm{Re}(\bar{\mathbf{s}}) \\ \mathrm{Im}(\bar{\mathbf{s}}) \end{bmatrix} \in \Re^{2K\times 1}$, where Re(•) and Im(•) denote the element-wise real and imaginary parts, respectively. These transformations are used in Section 2.2.

The paper is organized as follows: Section 2 presents the uplink signal model and the CG-based detection technique. Section 3 details the proposed deep unfolded CG method. Early performance results are discussed in Section 4. Finally, the conclusions are drawn and some future research directions are outlined in Section 5.

2 | SIGNAL MODEL AND CG-BASED DETECTION

2.1 | Signal model

We consider an uplink transmission where K single antenna users are communicating with a BS equipped with M antennas (where$M \gg K$) in TDD duplex mode using the OFDM modulation scheme. For the sake of simplicity, we consider a baseband equivalent channel and expressions per subcarrier where the subcarrier index is suppressed. The data signal of the kth user is denoted by $\bar{s}_k \in \mathbb{C}$ and is normalized to unit power. The vector $\bar{\mathbf{h}}_k \in \mathbb{C}^{M\times 1}$ represents the corresponding channel which is modelled, for simulation purposes, as a flat Rayleigh fading channel vector whose entries are assumed to be independent and identically distributed (i.i.d.) with zero mean and unit variance. We model the received signal at the BS as

$$\bar{\mathbf{y}} = \bar{\mathbf{H}}\bar{s} + \bar{\mathbf{n}} \tag{1}$$

where $\bar{\mathbf{y}} \in \mathbb{C}^{M\times 1}$, $\bar{\mathbf{H}} = [\bar{\mathbf{h}}_1\ \bar{\mathbf{h}}_2\ \cdots\ \bar{\mathbf{h}}_K]$ is the channel matrix and $\bar{\mathbf{s}} = [\bar{s}_1\ \bar{s}_2\ \cdots\ \bar{s}_K]^T$. $\bar{\mathbf{n}} \in \mathbb{C}^{M\times 1}$ represents the additive receiver noise vector whose entries have a zero mean and a variance equal to σ^2.

TABLE 1 CG-based detection applied to Equation (3)

1. **Inputs**: $\mathbf{y}_{MF}$ and $\mathbf{A}$
2. **Initialization**:
   2.1 $\mathbf{r}^{(0)} = \mathbf{y}_{MF}$
   2.2 $\mathbf{s}^{(0)} = [0\ \cdots\ 0]^T \in \Re^{2K\times 1}$
   2.3 $\mathbf{p}^{(0)} = \mathbf{s}^{(0)}$
   2.4 $\mathbf{q}^{(0)} = \mathbf{A}\mathbf{p}^{(0)}$
   2.5 $\mathbf{x}^{(0)} = \mathbf{A}\mathbf{r}^{(0)}$
   2.6 $\gamma^{(0)} = \|\mathbf{x}^{(0)}\|_2^2$
   2.7 $\alpha^{(0)} = \gamma^{(0)}/\|\mathbf{q}^{(0)}\|_2^2$
3. **For** $\ell = 1, 2, \ldots, L_{MAX}$
   3.1 $\mathbf{q}^{(\ell)} = \mathbf{A}\mathbf{p}^{(\ell-1)}$
   3.2 $\alpha^{(\ell)} = \gamma^{(\ell-1)}/\|\mathbf{q}^{(\ell)}\|_2^2$
   3.3 $\mathbf{s}^{(\ell)} = \mathbf{s}^{(\ell-1)} + \alpha^{(\ell)}\mathbf{p}^{(\ell-1)}$
   3.4 $\mathbf{r}^{(\ell)} = \mathbf{r}^{(\ell-1)} - \alpha^{(\ell)}\mathbf{q}^{(\ell)}$
   3.5 $\mathbf{x}^{(\ell)} = \mathbf{A}\mathbf{r}^{(\ell)}$
   3.6 $\gamma^{(\ell)} = \|\mathbf{x}^{(\ell)}\|_2^2$
   3.7 $\mathbf{p}^{(\ell)} = \mathbf{x}^{(\ell)} + (\gamma^{(\ell)}/\gamma^{(\ell-1)})\mathbf{p}^{(\ell-1)}$

   End for
4. **Output**: $\mathbf{s} = \mathbf{s}^{(L_{MAX})}$

The ZF detection technique applies $\bar{\mathbf{W}} = (\bar{\mathbf{H}}^H\bar{\mathbf{H}})^{-1}\bar{\mathbf{H}}^H = [\bar{\mathbf{w}}_1, \ldots, \bar{\mathbf{w}}_K] \in \mathbb{C}^{M\times K}$ to the received signal $\bar{\mathbf{y}}$ to estimate the users' transmitted signal $\bar{\mathbf{s}}$ as

$$\bar{\mathbf{s}} = \bar{\mathbf{W}}\bar{y} = \left(\bar{\mathbf{H}}^H\bar{\mathbf{H}}\right)^{-1}\bar{\mathbf{H}}^H\bar{\mathbf{y}} = \left(\bar{\mathbf{H}}^H\bar{\mathbf{H}}\right)^{-1}\bar{\mathbf{y}}_{MF} = \bar{\mathbf{A}}^{-1}\bar{\mathbf{y}}_{MF} \tag{2}$$

where $\bar{\mathbf{y}}_{MF} \overset{\Delta}{=} \bar{\mathbf{H}}^H\bar{\mathbf{y}}$.[1] Notice that the MRC technique considers this approximation$\bar{\mathbf{A}}^{-1} \cong (\mathrm{diag}(\bar{\mathbf{H}}^H\bar{\mathbf{H}}))^{-1}$.

2.2 | CG technique

Equation (2) can be readily solved iteratively using CG techniques (refer to page 214 of [30]). For convenience, we first convert the baseband complex-valued problem to a real-valued one where we reformulate Equation (2) as

$$\mathbf{y}_{MF} = \begin{bmatrix} \mathrm{Re}\left(\bar{\mathbf{y}}_{MF}\right) \\ \mathrm{Im}\left(\bar{\mathbf{y}}_{MF}\right) \end{bmatrix} = \begin{bmatrix} \mathrm{Re}\left(\bar{\mathbf{A}}\right) & -\mathrm{Im}\left(\bar{\mathbf{A}}\right) \\ \mathrm{Im}\left(\bar{\mathbf{A}}\right) & \mathrm{Re}\left(\bar{\mathbf{A}}\right) \end{bmatrix} \begin{bmatrix} \mathrm{Re}\left(\bar{\mathbf{s}}\right) \\ \mathrm{Im}\left(\bar{\mathbf{s}}\right) \end{bmatrix} + \begin{bmatrix} \mathrm{Re}\left(\bar{\mathbf{H}}^H\bar{\mathbf{n}}\right) \\ \mathrm{Im}\left(\bar{\mathbf{H}}^H\bar{\mathbf{n}}\right) \end{bmatrix} = \mathbf{A}\mathbf{s} + \mathbf{z} \tag{3}$$

where $\mathbf{A} = \begin{bmatrix} \mathrm{Re}(\bar{\mathbf{A}}) & -\mathrm{Im}(\bar{\mathbf{A}}) \\ \mathrm{Im}(\bar{\mathbf{A}}) & \mathrm{Re}(\bar{\mathbf{A}}) \end{bmatrix} \in \Re^{2K\times 2K}$, $\mathbf{s} = \begin{bmatrix} \mathrm{Re}(\bar{\mathbf{s}}) \\ \mathrm{Im}(\bar{\mathbf{s}}) \end{bmatrix} \in \Re^{2K\times 1}$ and $\mathbf{z} = \begin{bmatrix} \mathrm{Re}(\bar{\mathbf{H}}^H\bar{\mathbf{n}}) \\ \mathrm{Im}(\bar{\mathbf{H}}^H\bar{\mathbf{n}}) \end{bmatrix} \in \Re^{2K\times 1}$. The CG technique is, therefore, summarized in Table 1.

[1] Subscripts MRC (maximum ratio combining) and MF (matched filter) are interchangeably used through this paper.

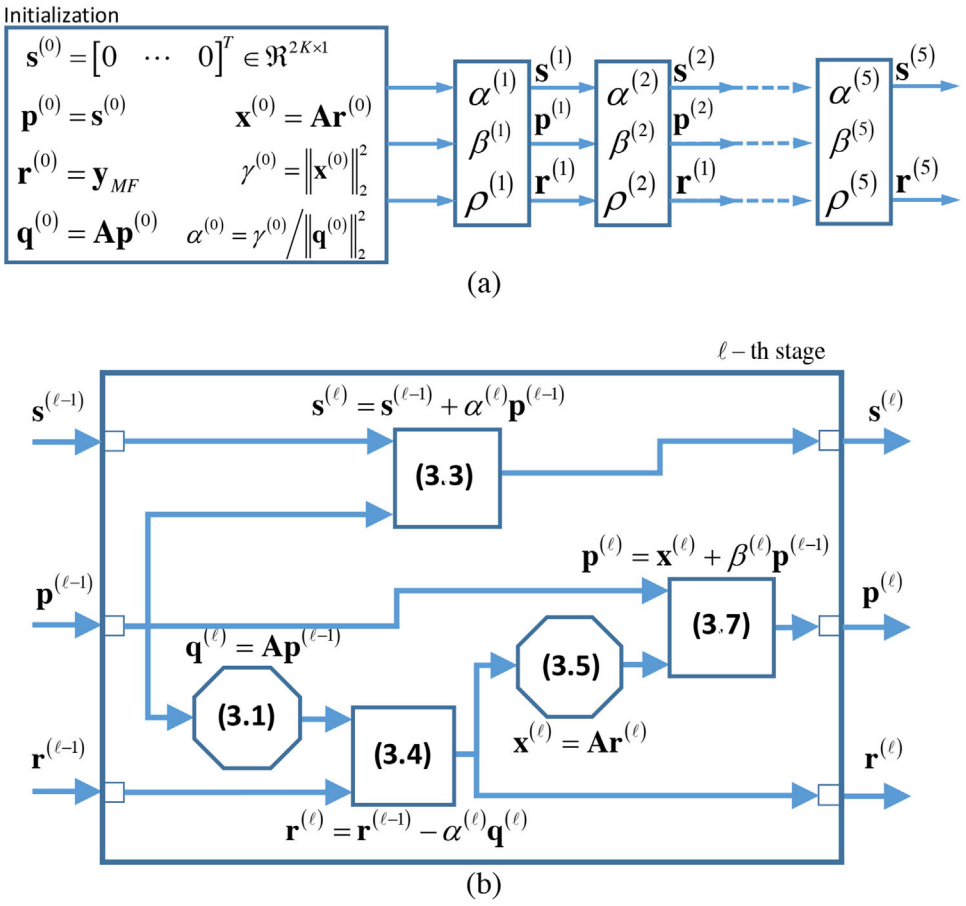


FIGURE 1 (a) CG-based data flow graph for 5 stages (layers) $L_{\mathrm{MAX}} = 5$ and b) the ℓth stage

3 | DEEP UNFOLDING CG

3.1 | Data flow graph for the CG algorithm

Prior to the deep unfolding transformation, Equation (3.7) in Table 1 is written as $\mathbf{p}^{(\ell)} = \mathbf{x}^{(\ell)} + \beta^{(\ell)}\mathbf{p}^{(\ell-1)}$ while Equation (3.2) is skipped so that the problem has $2L_{\mathrm{MAX}}$ tuning parameters to train, namely $\alpha^{(\ell)}$ and $\beta^{(\ell)}$ for $\ell = 1, 2, \ldots, L_{\mathrm{MAX}}$. In addition, the matrix $\mathbf{A}$ can be replaced with $\mathbf{B}^{(\ell)} = \mathbf{A} + \rho^{(\ell)}\mathbf{I}_{2K}$ to consider additional L_{MAX} tuning parameters $\rho^{(\ell)}$ for $\ell = 1, 2, \ldots, L_{\mathrm{MAX}}$. The parameters $\alpha^{(\ell)}$ and $\beta^{(\ell)}$ preserve the CG algorithm structure while $\rho^{(\ell)}$ can be viewed as a parameter that learns the inherent noise variance (as in the MMSE, problem formulation). For the sake of simplicity in introducing a simple canonical framework for deep unfolded CG-based detection method, we limit the current structure to these $3L_{\mathrm{MAX}}$ parameters. The algorithm structure can be enhanced to include more parameters and non-linear activation functions.

Deep unfolding the CG algorithm entails unfolding the L_{MAX} iterations into L_{MAX} stages (layers) as shown in Figure 1. Figure 1(a) depicts $L_{\mathrm{MAX}} = 5$ data flow graph which comprises nodes corresponding to the different operations in the iterative CG algorithm, and the directed edges corresponding to the data flow between the operations. In this case, the ℓth iteration of the CG algorithm corresponds to the ℓth stage of the data flow graph. In each stage of the graph, there are five steps as shown in Figure 1(b).

3.2 | Network training and gradient computation

The loss function is based on the L_2 norm applied on $\mathbf{r}^{(L_{\mathrm{MAX}})}$ which represents the mean squared error so that

$$\mathrm{Loss}\left(\Theta;\mathbf{A},\mathbf{y}_{\mathrm{MF}}\right) = \left\|\mathbf{r}^{(L_{\mathrm{MAX}})}\right\|_2^2 \tag{4}$$

where $\Theta = \{\alpha^{(\ell)}, \beta^{(\ell)}, \rho^{(\ell)}\}_{\ell=1}^{L_{\mathrm{MAX}}}$. The loss function Loss $(\Theta;\mathbf{A},\mathbf{y}_{\mathrm{MF}})$ is minimized over $\Theta = \{\alpha^{(\ell)}, \beta^{(\ell)}, \rho^{(\ell)}\}_{\ell=1}^{L_{\mathrm{MAX}}}$ parameters. The gradient of the loss function is computed with respect to every parameter using backpropagation over the deep network of Figure 1. In the forward pass, we process the data in the ℓthstage as shown in Figure 1(b) while the gradients are computed in the reverse direction (the backward pass). For the sake of simplicity, we compute the gradient for $L_{MAX} = 5$ and then generalize for any ℓthstage.

The gradient of the loss function $\text{Loss}(\Theta;\mathbf{A},\mathbf{y}_{\text{MF}})$w.r.t $\alpha^{(5)}$ is

$$\frac{\partial \text{Loss}(\Theta;\mathbf{A},\mathbf{y}_{\text{MF}})}{\partial \alpha^{(5)}} = \frac{\partial \left\|\mathbf{r}^{(5)}\right\|_2^2}{\partial \alpha^{(5)}} = \frac{\partial \left\|\mathbf{r}^{(5)}\right\|_2^2}{\partial \mathbf{r}^{(5)}} \frac{\partial \mathbf{r}^{(5)}}{\partial \alpha^{(5)}}. \\ = \left(\mathbf{r}^{(5)}\right)^T \left(-\mathbf{q}^{(5)}\right) \tag{5}$$

Applying the chain rule one can compute the gradient w.r.t $\alpha^{(\ell)}$ as

$$\frac{\partial \text{Loss}\left(\Theta;\mathbf{A},\mathbf{y}_{\text{MF}}\right)}{\partial \alpha^{(\ell)}} = -\left(\mathbf{r}^{(L_{MAX})}\right)^T \mathbf{q}^{(\ell)}. \tag{6}$$

Similarly, the gradient w.r.t $\beta^{(4)}$ is computed as follows[2]

$$\begin{aligned} \frac{\partial \text{Loss}(\Theta;\mathbf{A},\mathbf{y}_{\text{MF}})}{\partial \beta^{(4)}} &= \frac{\partial \left\|\mathbf{r}^{(5)}\right\|_2^2}{\partial \beta^{(4)}} = \frac{\partial \left\|\mathbf{r}^{(5)}\right\|_2^2}{\partial \mathbf{r}^{(5)}} \frac{\partial \mathbf{r}^{(5)}}{\partial \beta^{(4)}} \\ &= \frac{\partial \left\|\mathbf{r}^{(5)}\right\|_2^2}{\partial \mathbf{r}^{(5)}} \left(\frac{\partial \mathbf{r}^{(5)}}{\partial \mathbf{q}^{(5)}} \frac{\partial \mathbf{q}^{(5)}}{\partial \mathbf{p}^{(4)}} \frac{\partial \mathbf{p}^{(4)}}{\partial \beta^{(4)}}\right) \\ &= \left(\mathbf{r}^{(5)}\right)^T \left(-\alpha^{(5)}\right) \mathbf{B}^{(4)} \mathbf{p}^{(3)} \\ &= \left(\mathbf{r}^{(L_{MAX})}\right)^T \left(-\alpha^{(L_{\text{MAX}})}\right) \mathbf{B}^{(L_{\text{MAX}}-1)} \mathbf{p}^{(L_{\text{MAX}}-1-1)}. \end{aligned} \tag{7}$$

Applying the chain rule one can compute gradient w.r.t $\beta^{(\ell)}$, for $\ell < L_{\text{MAX}} - 1$, as

$$\frac{\partial \text{Loss}\left(\Theta;\mathbf{A},\mathbf{y}_{\text{MF}}\right)}{\partial \beta^{(\ell)}} \\ = -\alpha^{(L_{\text{MAX}})} \left(\mathbf{r}^{(L_{\text{MAX}})}\right)^T \mathbf{B}^{(\ell)} \mathbf{p}^{(\ell-1)} \prod_{n=L_{\text{MAX}}-1}^{\ell+1} \beta^{(n)}. \tag{8}$$

Finally, the gradient w.r.t. $\rho^{(\ell)}$ is computed as

$$\frac{\partial \text{Loss}\left(\Theta;\mathbf{A},\mathbf{y}_{\text{MF}}\right)}{\partial \rho^{(\ell)}} = -\alpha^{(\ell)} \left(\mathbf{r}^{(L_{\text{MAX}})}\right)^T \mathbf{p}^{(\ell-1)}. \tag{9}$$

Note that the loss function (Equation (4)) is not an explicit function of the desired signal $\mathbf{s} = \begin{bmatrix}\text{Re}(\bar{\mathbf{s}}) \\ \text{Im}(\bar{\mathbf{s}})\end{bmatrix} \in \mathfrak{R}^{2K\times 1}$ which makes the proposed architecture blind. It would not be the case if the loss function is defined as $\text{Loss}(\Theta;\mathbf{A},\mathbf{y}_{\text{MF}},\mathbf{s}) = \|\mathbf{s}^{(L_{\text{MAX}})} - \mathbf{s}\|_2^2$, which is an explicit function of the desired signal. In fact, our first attempts were based on computing the gradients of$\text{Loss}(\Theta;\mathbf{A},\mathbf{y}_{\text{MF}},\mathbf{s})$. It ended up having a similar performance using the loss function in Equation (4). The computation of the gradients $\text{Loss}(\Theta;\mathbf{A},\mathbf{y}_{\text{MF}},\mathbf{s})$ is straightforward using the data flow graph in Figure 1(a) and (b) and reported in the appendix.

The initialization of the parameters is based on the first forward pass using steps 2.5 and 2.6 in Table 1.

[2] The gradient w.r.t $\beta^{(5)}$ is not required for the final stage.

Many works have addressed the matrix inversion issue inherent in the MMSE or ZF based detection [10–12]. It is also well known that the CG technique [31, p. 214] is a low complexity iterative method compared to a direct matrix inversion using Cholesky decomposition. Wu et al. [11] and [12] have studied this matter in detail, this is the reason why we did not consider it in this paper. Note that the algorithm in Table 1 does not involve any matrix–matrix multiplications. The matrix–vector multiplications in (3.1) and (3.5) are equivalent to the detection phase in MMSE and ZF after an implicit matrix inversion.

3.3 | Discussion and potential future

Even though the proposed architecture might seem simple but it paves the way towards exploring the features outlined at the end of the introduction section. Prior to outlining the future works, let's infer a few key points from the closed forms solution of the gradient.

Looking at Equations (3.3) $\mathbf{s}^{(\ell)} = \mathbf{s}^{(\ell-1)} + \alpha^{(\ell)}\mathbf{p}^{(\ell-1)}$ and (3.4) $\mathbf{r}^{(\ell)} = \mathbf{r}^{(\ell-1)} - \alpha^{(\ell)}\mathbf{q}^{(\ell)}$, it is clear that the data signal update is in the opposite direction compared to the residual error update. The proportion of the update is given by Equation (5) for the last layer and by Equation (6) for the subsequent layers. Substituting $\mathbf{q}^{(\ell)} = \mathbf{A}\mathbf{p}^{(\ell-1)}$ into Equation (6), the parameter update for $\alpha^{(\ell)}$ can be expressed as

$$\frac{\partial \text{Loss}\left(\Theta;\mathbf{A},\mathbf{y}_{\text{MF}}\right)}{\partial \alpha^{(\ell)}} = -\left(\mathbf{r}^{(L_{\text{MAX}})}\right)^T \mathbf{A}\mathbf{p}^{(\ell-1)}. \tag{10}$$

For interpretation purposes, let's assume that $\beta^{(\ell)} = 0$. Therefore Equation (10) can be reduced to

$$\begin{aligned} \frac{\partial \text{Loss}(\Theta;\mathbf{A},\mathbf{y}_{\text{MF}})}{\partial \alpha^{(\ell)}} &= -\left(\mathbf{r}^{(L_{\text{MAX}})}\right)^T \mathbf{A}\mathbf{x}^{(\ell-1)} \\ &= \left(\mathbf{r}^{(L_{\text{MAX}})}\right)^T \mathbf{A}\left(\mathbf{A}\mathbf{r}^{(\ell-1)}\right) \\ &= \left(\mathbf{A}\mathbf{r}^{(L_{\text{MAX}})}\right)^T \left(\mathbf{A}\mathbf{r}^{(\ell-1)}\right). \end{aligned} \tag{11}$$

The second line of Equation (11) follows from Equation (3.1) and the last line follows from the fact that $\mathbf{A}$ is a symmetrical matrix. This shows that the parameter update for $\alpha^{(\ell)}$ is the per-layer modified norm of the projected residual error. The plot of the residual error will be depicted in Figure 5 in Section 4.4.

For a non-zero value of $\beta^{(\ell)}$, the second term in $\mathbf{p}^{(\ell)} = \mathbf{x}^{(\ell)} + \beta^{(\ell)}\mathbf{p}^{(\ell-1)}$ plays the role of a smoothing/averaging term.

Similarly, the update term for $\beta^{(\ell)}$ in Equation (8) depends on the per-layer modified norm of the projected residual error which is, in turn, weighted by the product of the past layers' parameters $\prod_{n=L_{\text{MAX}}-1}^{\ell+1}\beta^{(n)}$. One can perceive this term as a factor that controls the trade-off between stability and the convergence speed. This is an intuitive interpretation of the interaction between the parameters' update (gradients), the data signal and the residual errors updates. As future works, we envisage adopting an in-depth look at the deep CG architecture via adopting state-of-the-art method and tools mainly used in trans-

parent ML [35] for interpreting and understanding such networks.

It shall be noted that the proposed deep unfolded architecture is simple enough as only a few parameters are trained but it can be augmented by considering these points for future works.

1. Adding extra parameters such as replacing a single-parameter equation $\mathbf{p}^{(\ell)} = \mathbf{x}^{(\ell)} + \beta^{(\ell)}\mathbf{p}^{(\ell-1)}$ by $\mathbf{p}^{(\ell)} = \mathbf{V}^{(\ell)}\begin{bmatrix}\mathbf{x}^{(\ell)}\\ \mathbf{p}^{(\ell-1)}\end{bmatrix}$ where $\mathbf{V}^{(\ell)} \in \mathfrak{R}^{2K\times 4K}$ is a higher dimension weight matrix operating on a stacked real-valued $4K \times 1$ vector made by concatenating $\mathbf{x}^{(\ell-1)}$ and $\mathbf{p}^{(\ell-1)}$. Such a proposal will increase the algorithm learning degree of freedom close to what the current DL architecture (such as deep NN-DNN) is using.
2. Adding non-linear activation functions to enable the algorithm structure to capture the channel and the hardware non-linearities.
3. Now that the deep unfolded architecture exposes the values of the gradients at every layer, one can adopt state-of-the-art method and tools mainly used in transparent ML framework [35] for interpreting and understanding such networks. To the authors best knowledge, no work has been conducted to infer the optimal size of deep unfolded networks, to discuss analysis frameworks such as activation maximization and sensitivity to identify the most important input features via the relevance scores [35] or even consider other backward propagation techniques such as layer-wise relevance propagation which incorporate filtering to form a separate explanation for (1) what is specifically relevant to a given task (think of learning the modulation and radio resources assignment as a learning task) and (2) what is commonly relevant to all tasks.

4 | PERFORMANCE RESULTS AND ANALYSIS

This section discusses the performance of the proposed deep unfolded CG technique under i.i.d. channel with zero mean and unit variance. The simulation results cover (i) the number of the required training symbols overhead (where an explicit knowledge of the symbols themselves are not needed) and (ii) the performance in terms of error vector magnitude (EVM) as a function of the number of users (K) and SNR. We adopt EVM instead of bit error rate (BER) as a performance metric as the initial simulation platform is designed to evaluate transmit precoding. However, there is a direct link between EVM and BER [31]. The system is operating in a massive MIMO regime where the BS is equipped with 128 antennas while the number of served single-antenna users is an order of magnitude lower than the number of the antennas at the BS. Any QAM based modulation can be used.[3] Herein 64-QAM modulation (i.e., $M_{\text{QAM}} = 64$) is used unless otherwise stated. The

[3] So far only QAM-based modulation is tested.

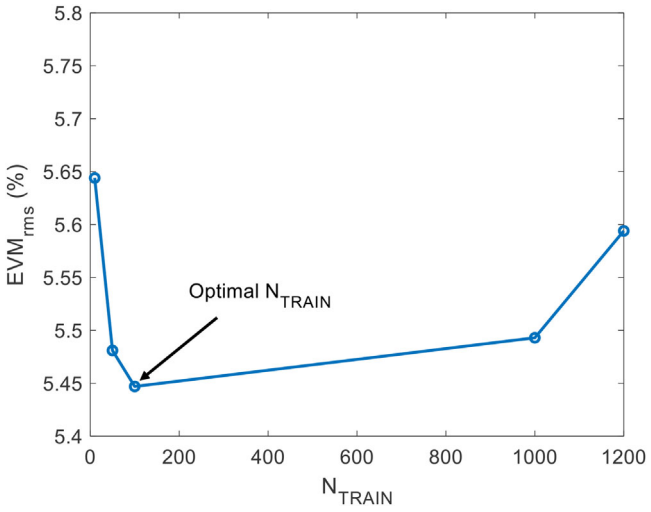
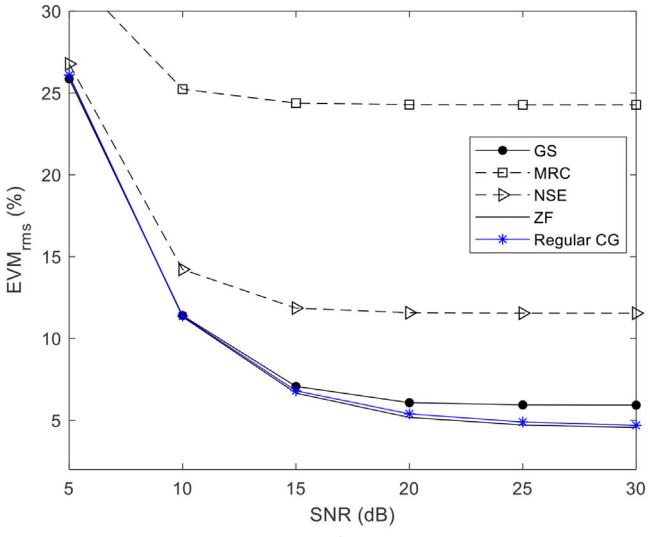


FIGURE 2 RMS EVM as a function of the number of the training symbols with $K = 20$ users and SNR = 20 dB

frame length is $N_{\text{Frame}} = 1200 \times 14$ symbols, whereas the number of the training symbols is set to $N_{\text{Train}} = 100$ per frame (see the simulation results in Section 4.1). Note that such a choice can be interpreted as follows: $N_{\text{Frame}} = 1200 \times 14$ symbols represent the number of effective subcarriers in a 20 MHz LTE signal frame while $N_{\text{Train}} = 100$ is roughly equivalent to one resource block (RB) in the first OFDM symbol only. The model-based CG (regular CG) and the basic deep unfolded CG (Deep CG) techniques use the same $L_{\text{MAX}} = \log_2(M_{\text{QAM}})K$ iterations and stages, respectively. This choice of the number of iterations/stages, as a function of the modulation depth M_{QAM} and the number of users K, is discussed in Section 4.3

4.1 | How many training symbols are needed?

Based on the simulation parameters stated above, the SNR and the number of users are fixed to 20 dB and 20 users, respectively. By varying $N_{\text{Train}} \in \{10, 50, 100, 500, 1000\}$, Figure 2 shows that the optimal training size is as low as 100 symbols in such i.i.d. (favourable) channel conditions. This is quite encouraging and enables effective training of the basic structure's parameters using one RB in the first OFDM symbol only. Depending on the channel coherence time, this represents a sub 1% training overhead in slow time varying channel scenarios. Therefore, it is worth noting that the overall computation complexity, due to backpropagation processing, of the basic deep-unfolded CG is not substantial. This, in fact, supports the expectation in [29] wherein incorporating domain knowledge experts is a key in reducing the training overhead. We attribute the high EVM at the higher number of the training symbols to the overfitting and a potential numerical divergence.



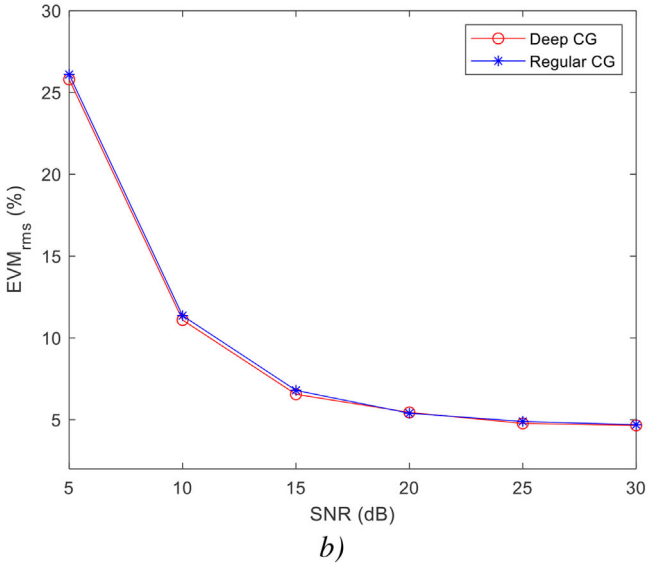


FIGURE 3 (a) RMS EVM as a function of SNR for model-based CG in comparison with ZF (using direct matrix inversion), NSE, GS and MRC. (b) RMS EVM as a function of SNR for model-based CG (Regular CG) and basic deep unfolded CG (Deep CG) with $K = 20$ users and $M_{\text{QAM}} = 64$. The training is done at SNR = 20 dB and a fixed channel realization

4.2 | Model-based CG versus basic deep-unfolded CG

As one would expect, the model-based CG detection is almost optimal in i.i.d. channel conditions. In the model-based CG, the parameters are explicitly computed for every symbol over L_{MAX} iterations. These parameters differ from one symbol to another. Whereas in deep-unfolded CG these parameters are computed during the training phase and kept constant over the rest of the transmission frame. Figure 3(a) compares the model-based CG with state-of-the-art methods such as NSE [9] and GS [10] and other reference methods such as ZF with direct matrix inversion and MRC.

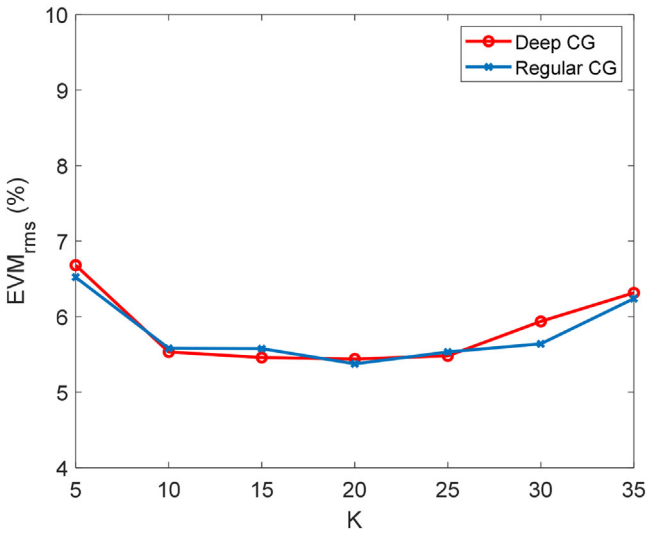


FIGURE 4 RMS EVM as a function of the number of users with the network dimensions scaled as function of the number of users and modulation depth (i.e. as$L_{\text{MAX}} = \log_2(M_{\text{QAM}})K$). Training is performed at a fixed SNR = 20 dB and a fixed channel realization for a given number of users while the simulations are performed at different channel realizations

Figure 3(b) shows that the basic deep-unfolded CG is a good implementation alternative to the iterative model-based CG. The simulation results depict the RMS EVM for 20 users transmitting 64 QAM symbols. Both the model-based CG and deep-unfolded CG are performing equally well in such favourable i.i.d. channel conditions. It is worth noting that the training is done at a fixed SNR of 20 dB and one i.i.d channel realization per training symbol whereas the simulations run over a wide range of SNR values and channel realizations.

4.3 | Does the basic deep-unfolded CG depend on the number of antennas as the system load increases?

Figure 4 depicts the EVM as a function of the number of users where $L_{\text{MAX}} = \log_2(M_{\text{QAM}})K$ for both deep CG and regular CG algorithms. The extensive simulations revealed that dimensioning the architecture as a function of the modulation depth and the number of users preserve the performance within a certain EVM threshold. The reason why Figure 4 shows an almost flat EVM performance (RMS EVM within +/- 0.5%) as far as $L_{\text{MAX}} = \log_2(M_{\text{QAM}})K$. One can, therefore, state that to preserve the system performance as the system load (K) increases, only the number of iterations/layers need to be scaled according to $L_{\text{MAX}} = \log_2(M_{\text{QAM}})K$ while the number of serving antennas at the BS is kept unchanged.

4.4 | Convergence behaviour through residual error

To complement our discussion in Section 3.3, Figure 5 depicts the residual error plot as a function of the number of layers

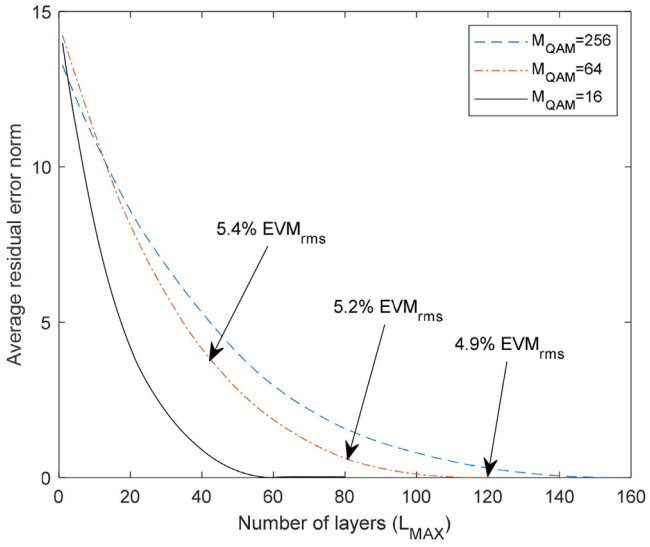


FIGURE 5 Residual error plot as a function of the number of layers L_{MAX} for 16, 64 and 256 QAM modulation at 25 dB SNR

L_{MAX} for 16, 64 and 256 QAM modulation at 25 dB SNR. The fast decaying residual error curves demonstrate the fast convergence behaviour of the learning process which depends, among other system parameters, on the modulation type. The arrows point to the iteration number that achieves a given RMS EVM so that the system engineer can infer the optimal number of layers for a give RMS EVM value.

4.5 | A note on the computational complexity

The regular CG method implements Equations (3.1)–(3.7) iteratively over L_{MAX} iterations. However, the proposed method does not compute (3.2), (3.6) and the division in (3.7), which amount to $4K$ multiplications/additions and 2 divisions per iteration, respectively. This is a total of $4KN_{Frame}$ multiplications/additions and $2N_{Frame}$ divisions per iteration over a frame of N_{Frame} symbols. On the other hand, the proposed method computes Equations (6) and (8) to determine the fixed parameters per layer. This amounts to $4K$ multiplications/additions for (6) and (8). Note that the matrix-vector multiplication $\mathbf{B}^{(\ell)}\mathbf{p}^{(\ell-1)}$ in (8) is explicitly computed in (3.1). We, therefore, expect a total of $8KN_{Train}$ multiplications/additions per layer. This results in a computational reduction by a factor of $N_{Frame}/(8N_{Train})$. Using our simulation parameters $N_{Frame} = 1200 \times 14$ and $N_{Train} = 100$, the complexity reduction factor is 21.

5 | CONCLUSION

This paper has proposed a basic deep unfolded implementation of the iterative model-based CG wherein the parameters are trained via backpropagation. We derived the closed-form expressions of the gradients of the loss function w.r.t. the parameters. The loss function is based on the squared norm of the residual error which is not an explicit function of the desired transmitted symbols. The simulation results reveal interesting insights; (i) the training overhead is very low which makes the deep unfolded CG a good implementation alternative to iterative model-based CG and (ii) the basic deep unfolded structure does not depend on the number of antennas as the load (the number of users) increases as far as the network structure is dimensioned based on the number of users and modulation depth. We do not attempt to outperform the iterative model-based CG in favourable i.i.d. channel. However, the following insights can be deduced: (i) the deep unfolded CG structure can be incorporated in an end-to-end communication system learning process [2-ZAP19] as the structure readily backpropagates the gradient for the parameters' optimization. The algorithm structure is preserved by the deep unfolded structure as domain knowledge expert which, in turn, explains the low number of training samples (fast training phase). This is in line with what the literature expects from incorporating domain knowledge and RTNs [30-DOR18].

As future work, one can envisage extending the architecture to consider more realistic time-varying channels and hardware impairments (and non-linearities). As in many DL approaches, the performance is sensitive to the initial values of the parameters (which we noted through extensive simulations), which seems to depend on the system parameters such as SNR, number of users and stages. It, therefore, needs an in-depth investigation.

ACKNOWLEDGEMENTS

This work was supported by the Natural Sciences and Engineering Research Council of Canada under Grant number RGPIN-2015-03674 and the authors would like to thank the team of Chaire de recherche sur les signaux et l'intelligence des systèmes haute performance for technical supports.

ORCID

Messaoud Ahmed Ouameur https://orcid.org/0000-0003-1095-8012

Daniel Massicotte https://orcid.org/0000-0002-7807-7919

REFERENCES

1. O'Shea, T., Hoydis, J.: An Introduction to Deep Learning for the Physical Layer. IEEE Transactions on Cognitive Communications and Networking. 3(4), 563–575 (2017)
2. Zappone, A., et al.: Wireless networks design in the era of deep learning: Model-based, Ai-based, or Both? IEEE Trans. Commun. 67(10), 7331–7376 (2019)
3. Björnson, E., et al.: Massive MIMO is a reality—What is next?: Five promising research directions for antenna arrays. Digit. Signal Process. 94, 3–20 (2019)
4. Björnson, E., A look at an LTE-TDD Massive MIMO product. (2018), http://ma-mimo.ellintech.se/2018/08/27/
5. Von Butovitsch, P., et al.: Advanced antenna systems for 5G networks. Ericsson White Paper, (2018)
6. Qu, Y., et al.: Nine communications technology trends for 2019. Communication society technology news. https://www.comsoc.org/publications/ctn/nine-communicationstechnology-trends-2019
7. Rajatheva, N., et al.: White paper on broadband connectivity in 6G. https://arxiv.org/abs/2004.14247. Accessed 20 May 2020

8. Marzetta, T.L., Noncooperative cellular wireless with unlimited numbers of base station antennas. IEEE Trans. Wireless Commun. 9(11), 3590–3600 (2010)
9. Ngo, H.Q., Massive MIMO: Fundamentals and System Designs, PhD. Thesis, Linköping University Electronic Press, 2015
10. Wu, M., et al.: Large-scale MIMO detection for 3GPP LTE: Algorithms and FPGA implementations. IEEE J. Sel. Top. Sig. Proc. 8(5), 916–929 (2014)
11. Wu, Z., et al.: Efficient architecture for soft-output massive MIMO detection with Gauss-Seidel method. In: IEEE International Conference on Circuits and Systems, pp. 1886–1889, (2016)
12. Wu, M., et al.: High-throughput data detection for massive MU-MIMO-OFDM using coordinate descent. IEEE Trans. Circuits Syst. Regul. Pap. 63(12), 2357–2367 (2016)
13. Ouameur, M.A., et al.: Performance evaluation and implementation complexity analysis framework for ZF based linear massive MIMO detection. Wireless Netw. J. 26, 4079–4093, (2020)
14. Jeon, C., et al.: Decentralized equalization with feedforward architectures for massive MUMIMO. IEEE Trans. Signal Process. 67(17), 4418–4432, (2019)
15. Ouameur, M.A., Massicotte, D.: Efficient distributed processing for large scale MIMO detection. In: 27th European Signal Processing Conference, A Coruna, Spain, 2019, pp. 1–5
16. Shahabi, S.M., et al.: Low-overhead constant envelope precoding in multi-cell massive MIMO systems with pilot contamination. IET Commun. 13(7), 926–933, (2019)
17. Shahabi, S.M., et al.: Constant envelope precoding in multi-cell massive MIMO systems: A high-throughput pilot contamination aware scheme. In: 2018 Wireless Advanced (WiAd). IEEE, New York, pp. 1–6 (2018)
18. Kazemi, M., et al.: Discrete-phase constant envelope precoding for massive MIMO systems. IEEE Trans. Commun. 65(5), 2011–2021 (2017)
19. Shahabi, S.M., et al.: Constant envelope precoding in multi-cell massive MIMO systems with channel uncertainty. Phys. Commun. 34(6), 203–209 (2019)
20. Samuel, N., et al.: Deep MIMO detection. In: IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications, Sapporo, 1–5 (2017)
21. O'Shea, T.J., et al.: Unsupervised representation learning of structured radio communication signals. In: Proc. IEEE Int. Workshop Sensing, Processing and Learning for Intelligent Machines, pp. 1—5, (2016)
22. Gruber, T., et al.: On deep learning based channel decoding. In: Proc. IEEE 51st Annu. Conf. Inf. Sciences Syst. (CISS), 1–6 (2017)
23. Schenk, T., RF imperfections in high-rate wireless systems: Impact and digital compensation. Springer Science & Business Media, Berlin, Germany (2008)
24. Hornik, K., et al.: Multilayer feedforward networks are universal approximators. Neural Netw. 2(5), 359–366 (1989)
25. Chen, Y.-H., et al.: Eyeriss: An energy efficient reconfigurable accelerator for deep convolutional neural networks. IEEE J. Solid-State Circ. 52(1), 127–138 (2017)
26. Raj, V., Kalyani, S.: Backpropagating through the air: Deep learning at physical layer without channel models. IEEE Commun. Lett. 22(11), 2278–2281 (2018)
27. Ye, H., et al.: Power of deep learning for channel estimation and signal detection in OFDM systems. IEEE Wireless Commun. Lett. 7(1), 114–117 (2018)
28. Hershey, J.R., et al.: Deep unfolding: Model-based inspiration of novel deep architectures. arXiv preprint arXiv:1409.2574, (2014)
29. Nachmani, E., et al.: Learning to decode linear codes using deep learning. In: Proc. IEEE Annu. Allerton Conf. Commun., Control, Computing, pp. 341–346, (2016)
30. Dörner, S., et al.: Deep learning based communication over the air. IEEE J. Sel. Topics Signal Process. 12(1), 132–143 (2018)
31. Gentle, J.E., Matrix algebra: Theory, computations and applications in statistics, Springer, New York (2007)
32. Shafik, R.A., et al.: On the extended relationships among EVM, BER and SNR as performance metrics. In: International Conference on Electrical and Computer Engineering, Dhaka, pp. 408–411 (2006)
33. Ali, S., et al.: 6G white paper on machine learning in wireless communication networks. https://arxiv.org/abs/2004.13875 Accessed 20 May 2020
34. Cammerer, S., et al.: Trainable communication systems: Concepts and prototype. https://arxiv.org/abs/1911.13055 Accessed 20 May 2020
35. Lipton, Z.C.:, The mythos of model interpretability. ICML Workshop on Human Interpretability in Machine Learning, New York, NY. Available at: https://arxiv.org/abs/1606.03490v3
36. Ouameur, M.A., et al.: Model-aided distributed shallow learning for OFDM receiver in IEEE 802.11 channel model. Wireless Netw. J. 26, 5427–5436 (2020), revised on May 2020
37. Montavon, G., et al.: Methods for interpreting and understanding deep neural networks. Digit. Signal Process. 73, 1–15 (2018)

How to cite this article: OuameurMassicotte M, Ouameur D. Early results on deep unfolded conjugate gradient-based large-scale MIMO detection. *IET Commun.* 2021;15:435–444. https://doi.org/10.1049/cmu2.12076.

APPENDIX

Another alternative for the loss function is

$$\text{Loss}\left(\Theta;\mathbf{A},\mathbf{y}_{\text{MF}},\mathbf{s}\right)=\left\|\mathbf{s}^{(I_{\text{MAX}})}-\mathbf{s}\right\|_2^2. \tag{A.1}$$

Based on the structure in Figure 1 (a) and (b) where we assume that$\rho^{(\ell)}=0$, the gradient w.r.t $\alpha^{(5)}$ is

$$\begin{aligned}\frac{\partial\text{Loss}\left(\Theta;\mathbf{A},\mathbf{y}_{\text{MF}},\mathbf{s}\right)}{\partial\alpha^{(5)}}&=\frac{\partial\left\|\mathbf{s}^{(5)}-\mathbf{s}\right\|_2^2}{\partial\alpha^{(5)}}=\frac{\partial\left\|\mathbf{s}^{(5)}-\mathbf{s}\right\|_2^2}{\partial\mathbf{s}^{(5)}}\frac{\partial\mathbf{s}^{(5)}}{\partial\alpha^{(5)}}\\&=2\left(\mathbf{s}^{(5)}-\mathbf{s}\right)^T\left(\mathbf{p}^{(4)}\right).\end{aligned} \tag{A.2}$$

Applying the chain rule one can compute the gradient w.r.t $\alpha^{(\ell)}$ as

$$\frac{\partial\text{Loss}\left(\Theta;\mathbf{A},\mathbf{y}_{\text{MF}},\mathbf{s}\right)}{\partial\alpha^{(\ell)}}=2\left(\mathbf{s}^{(I_{\text{MAX}})}-\mathbf{s}\right)^T\mathbf{p}^{(\ell-1)}. \tag{A.3}$$

Similarly, the gradient w.r.t $\beta^{(4)}$ is computed as follow[4]

$$\begin{aligned}\frac{\partial\text{Loss}\left(\Theta;\mathbf{A},\mathbf{y}_{\text{MF}},\mathbf{s}\right)}{\partial\beta^{(4)}}&=\frac{\partial\left\|\mathbf{s}^{(5)}-\mathbf{s}\right\|_2^2}{\partial\beta^{(4)}}=\frac{\partial\left\|\mathbf{s}^{(5)}-\mathbf{s}\right\|_2^2}{\partial\mathbf{s}^{(5)}}\frac{\partial\mathbf{s}^{(5)}}{\partial\beta^{(4)}}\\&=\frac{\partial\left\|\mathbf{s}^{(5)}-\mathbf{s}\right\|_2^2}{\partial\mathbf{r}^{(5)}}\left(\frac{\partial\mathbf{s}^{(5)}}{\partial\mathbf{p}^{(4)}}\frac{\partial\mathbf{p}^{(4)}}{\partial\beta^{(4)}}\right)\\&=2\left(\mathbf{s}^{(5)}-\mathbf{s}\right)^T\alpha^{(5)}\mathbf{p}^{(3)}\\&=2\left(\mathbf{s}^{(I_{MAX})}-\mathbf{s}\right)^T\alpha^{(I_{\text{MAX}})}\mathbf{p}^{(I_{\text{MAX}}-1-1)}.\end{aligned} \tag{A.4}$$

[4] The gradient w.r.t $\beta^{(5)}$ is not required for the final stage.

Applying the chain rule on can compute gradient w.r.t $\beta^{(\ell)}$, for $\ell < L_{\mathrm{MAX}} - 1$, as

$$\frac{\partial \mathrm{Loss}\left(\Theta; \mathbf{A}, \mathbf{y}_{\mathrm{MF}}, \mathbf{s}\right)}{\partial \beta^{(\ell)}} = -2\left(\mathbf{s}^{(L_{\mathrm{MAX}})} - \mathbf{s}\right)^{T} \alpha^{(L_{\mathrm{MAX}})} \mathbf{p}^{(\ell-1)} \prod_{n=L_{\mathrm{MAX}}-1}^{\ell+1} \beta^{(n)}. \quad \text{(A.5)}$$

Note that the gradient w.r.t $\beta^{(\ell)}$ in (A.5) does not involve any matrix-vector multiplication compared to Equation (8), which may seem to reduce the computational complexity at the expense of an explicit knowledge of the training symbols**s**. However, the term $\mathbf{B}^{(\ell)}\mathbf{p}^{(\ell-1)}$ in Equation (8) is already explicitly computed in Equation (3.1).