Leveraging Deep Neural Networks for Massive MIMO Data Detection

Ly V. Nguyen, Nhan T. Nguyen, Nghi H. Tran, Markku Juntti, A. Lee Swindlehurst, and Duy H. N. Nguyen

Abstract

Massive multiple-input multiple-output (MIMO) is a key technology for emerging next-generation wireless systems. Utilizing large antenna arrays at base-stations, massive MIMO enables substantial spatial multiplexing gains by simultaneously serving a large number of users. However, the complexity in massive MIMO signal processing (e.g., data detection) increases rapidly with the number of users, making conventional hand-engineered algorithms less computationally efficient. Low-complexity massive MIMO detection algorithms, especially those inspired or aided by deep learning, have emerged as a promising solution. While there exist many MIMO detection algorithms, the aim of this magazine article is to provide insight into how to leverage deep neural networks (DNN) for massive MIMO detection. We review recent developments in DNN-based MIMO detection that incorporate the domain knowledge of established MIMO detection algorithms with the learning capability of DNNs. We then present a comparison of the key numerical performance metrics of these works. We conclude by describing future research areas and applications of DNNs in massive MIMO receivers.

Introduction

As an integrated part of modern 5G and emerging 6G systems, massive MIMO offers several orders of magnitude enhancements in throughput and energy efficiency over conventional MIMO in existing 4G systems [1, 2]. Through the use of large antenna arrays with tens to thousands of elements, massive MIMO enables the design of extremely narrow spatial beams that boost the desired signal power, resulting in considerable performance gains in terms of user coverage and system throughput. However, the increase in the dimension of massive MIMO and the corresponding increase in the number of served users adversely impact the complexity in its signal processing pipeline. For example, optimal maximum likelihood (ML) detection comes with a complexity that is exponential in the number of users. Low-complexity and near-optimal detection is thus crucial to fully realize the potential of massive MIMO system performance targets.

For massive MIMO systems, in which a base station equipped with a large array of antennas serving a large number of users simultaneously, low-complexity detectors, such as zero-forcing (ZF) and linear minimum mean-squared error (LMMSE) may incur large performance gaps compared with the optimal ML detector. In contrast, near-optimal detection schemes, such as sphere decoding (SD), K-best SD (KSD), and fixed-complexity SD (FSD), may come at the cost of excessively high complexity [3]. The algorithm deficits of these conventional approaches prompt the interesting prospect of applying deep learning (DL) for massive MIMO detection [4], in which the computational complexity is shifted to an offline training phase, enabling faster run time in the online detection phase.

The application of DL in communications has recently gained much attention. Several model-based deep neural network (DNN) architectures have been proposed for massive MIMO detection. The pioneering detection network (DetNet), introduced by Samuel et al., [5], has showcased the power of DL for MIMO data detection. A fast-convergence sparsely-connected neural network (FS-Net) has been recently proposed in [6] as a simplified but optimized variant of DetNet. DetNet and FS-Net were both developed to mimic and optimize the iterative gradient descent algorithm. Another notable approach in DNN-based detection is based on the orthogonal approximate message passing (OAMP) algorithm [7], offering better performance, as well as lower computational complexity compared to gradient descent-based algorithms. In particular, He et al., introduced OAMP-Net2 [8] for data detection in both independent and identically distributed (i.i.d.) Gaussian and small-size correlated channels. Khani et al., [9] proposed MMNet targeting data detection in correlated MIMO channels, and showed that it significantly outperforms OAMP-Net2. We note that all these detection networks are based on the deep unfolding technique [10] and designed to optimize the free parameters of the underlying detection algorithms.

In this article, we present a holistic framework for leveraging DNNs in massive MIMO data detection. We review the conventional MIMO detection algorithms, and show how to incorporate the domain knowledge of these established algorithms into the development of DNN detectors, including DetNet, FSNet, OAMP-Net2, and MMNet. We then present numerical results comparing key per-

Ly V. Nguyen and Duy H. N. Nguyen (corresponding author) are with San Diego State University, USA;
Nhan Thanh Nguyen and Markku Juntti are with the University of Oulu, Finland; Nghi H. Tran is with the University of Akron, USA;
A. Lee Swindlehurst is with the University of California, USA.

formance metrics of these works, including symbol error rate (SER) and run time. We conclude by describing future research areas and applications of DNNs in massive MIMO communications.

Background

Signal Model and MIMO Detection Problem

We consider an uplink massive MIMO system, where the base station (BS) equipped with N antennas serves K single-antenna users. Note that the detectors presented in this article are also applicable to multi-antenna users. The propagation channel from the users to the BS is modeled by a matrix $\mathbf{H}$, in which each entry represents the channel between a user and a receive antenna. We denote by x the vector of K transmitted symbols associated with K users, and we assume that these symbols are drawn from a discrete alphabet $\mathbf{A}$. The input-output relationship of the considered system is modeled as:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n},$$

where $\mathbf{y}$ is a vector of the received signals at the N antennas of the BS, and n is a noise vector. Since the use of complex-valued parameters is uncommon in machine learning, we assume that all quantities in Eq. 1 are real-valued. This is also a matter of notational convenience, since a length-n complex-valued vector is isomorphic to a length-2n real-valued vector. In addition, a square complex-valued constellation of size n^2 (i.e., quadrature phase-shift keying (QPSK) and 16-quadrature amplitude modulation (16-QAM)) can be effectively represented by two independent real-valued alphabets of size n. The above model assumes a flat-fading or narrowband channel and the channel matrix is assumed to be known at the receiver. Our discussion can easily be extended to wideband channels using orthogonal frequency division multiplexing (OFDM).

The task of MIMO detection is to determine the transmitted symbol vector $\mathbf{x}$ based on the received vector $\mathbf{y}$. The detection error is minimized by classifying the most likely $\mathbf{x}$ with the ML criterion when no a priori information is available. That is equivalent to finding the solution to the optimization problem $\min_{\mathbf{x}\square\mathcal{A}^K}\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2$. MIMO ML data detection is a combinatorial problem, and its complexity grows exponentially with the number of users $\mathbf{K}$. Performing *joint* detection of the entire symbol vector $\mathbf{x}$ is computationally expensive even for a small-scale MIMO system, and even more so for massive MIMO systems. For example, the search space $\mathcal{A}^K$ grows to a set of 2^{32} candidates for a relatively *modest* large-scale MIMO system supporting 8 users with 16-QAM. Thus, there is a need for near-optimal and reduced-complexity data detection algorithms that scale well to massive MIMO systems. To this end, we first review two typical classes of massive MIMO data detection schemes, namely, linear and nonlinear detectors with a comprehensive review in [11].

Conventional MIMO/Massive MIMO Data Detectors

Linear Data Detectors: Linear data detectors with low complexity are practical candidates for massive MIMO systems [2]. These detection schemes detect one symbol at a time while treating all the other symbols transmitted from the other users as interference. The estimated symbol is obtained from a linear combination of the received signals, which is then projected into the nearest symbol in the alphabet $\mathbf{A}$. The simplest of these is the matched-filter (MF) detector which aims at maximizing the energy of the signal of interest.

A ZF detector targets elimination of the inter-user interference. Both schemes require relatively few computations, but they suffer from significant performance degradation due to the interference and/or noise enhancement. Unlike these two, an LMMSE detector tries to balance the enhancements in the signal of interest and the interference/noise. The LMMSE detector achieves the best performance among the three detectors, but it requires a matrix inversion, which can quickly result in excessive complexity for large-scale MIMO systems.

While relatively simple to implement, except the possible need for a matrix inversion, linear detectors can achieve good performance when the number of receive antennas is large enough compared to the number of users and the channel vectors from different users are independent [1, 2]. However, their performance deteriorates quickly when the number of users approaches the number of receive antennas or when the channel is ill-conditioned [11], prompting the need for more sophisticated nonlinear detectors.

Nonlinear Detectors: SD is one of the most well-known nonlinear algorithms for MIMO detection. Similar to ML detection, SD attempts to find the optimal lattice point closest to $\mathbf{y}$. However, its search is limited to the points inside a hypersphere which is a subset of the feasible set $\mathbf{A}^K$ and determined by a given radius. Each time a point lying inside the hypersphere is found, the search is further restricted by shrinking the sphere. When there is only one point in the sphere, the point becomes the final solution. The better optimized the sphere radius is, the better performance and/or complexity reduction can be achieved by SD [3].

Approximate message passing (AMP) is a relatively low-complexity iterative signal recovery algorithm for large-scale linear systems. A variant of AMP, referred to as OAMP [7], has been exploited for MIMO data detection in recent papers. In OAMP, the recovered signal is updated via a non-linear transformation of the previous iterate, which includes a linear estimator and a nonlinear denoiser. OAMP can attain near-optimal performance in few iterations. Except for a highly complicated matrix inversion in the linear estimator, OAMP can be a promising technique for massive MIMO detection.

The conventional MIMO detectors discussed above, especially those originally proposed for conventional small-sized MIMO systems, lead to a challenging performance-complexity trade-off. Specifically, nonlinear detectors with near-optimal performance but high complexity may not be feasible for deployment in large-scale systems. On the other hand, the linear detectors with low complexity perform relatively poorly in large-scale systems where the numbers of users and receive antennas are comparable. This concern motivates recent research on DL for massive MIMO detection.

DNN Detector for Linear MIMO Systems

The task of MIMO detection is to determine the transmitted symbol vector $\mathbf{x}$ based on the received vector $\mathbf{y}$. The detection error is minimized by classifying the most likely $\mathbf{x}$ with the ML criterion when no a priori information is available.
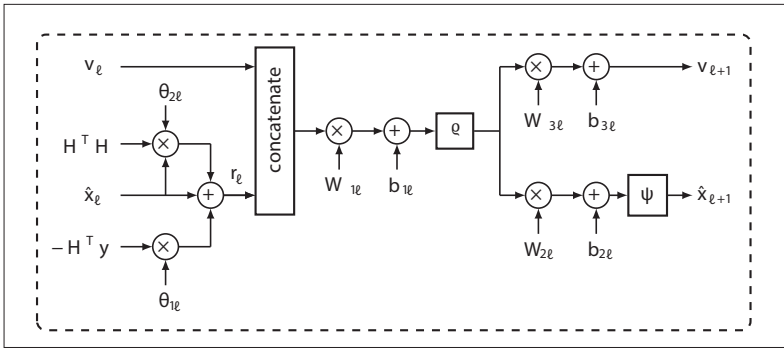
FIGURE 1. The ℓth layer of DetNet with trainable parameters $\theta_\ell = \{\mathbf{W}_{1\ell}, \mathbf{b}_{1\ell}, \mathbf{W}_{2\ell}, \mathbf{b}_{2\ell}, \mathbf{W}_{3\ell}, \mathbf{b}_{3\ell}, \theta_{1\ell}, \theta_{2\ell}\}$, a rectified linear unit (ReLU), ϱ and a soft quantizer ψ.

In this section, we provide an overview of the design of DNN detectors in MIMO and massive MIMO systems. We first focus on the fundamentals of a DNN detector. We then review and analyze recent developments of DNN detectors in the literature.

FUNDAMENTALS OF DNN-BASED DATA DETECTION

A DNN model can be trained efficiently to provide reliable prediction/approximation of the transmitted signal vectors. It accepts the received signals $\mathbf{y}$ and channel information $\mathbf{H}$ as inputs and outputs an estimate $\hat{\mathbf{x}}$ of the transmitted signal vector $\mathbf{x}$. In this respect, $\hat{\mathbf{x}}$ can be modeled as the target of a nonlinear mapping $f(\mathbf{H}, y; \theta)$, where θ consists of parameters pertaining to the neural network. The fidelity of the mapping $f(\cdot)$, also known as the *inference rule*, is measured by a cost function, which is defined as the mean squared error (MSE) between the estimate $\hat{\mathbf{x}}$ and the true transmitted signal $\mathbf{x}$. The goal of a DNN detector is to design $f(\cdot)$ via the optimization of the parameters θ to minimize this cost function. The data for training a DNN detector can be generated from the system model Eq. 1 with known prior distributions on the channel, the transmitted symbols, and the noise.

Most of the computational complexity of a DNN detector lies in the offline training phase. On the other hand, a DNN detector enables data detection with a much lower computational complexity at run time. This can be accomplished by performing the task in batch, offering polynomial time complexity in data detection based on simple matrix additions and multiplications. These operations are far simpler than the computationally expensive matrix inversions/pseudo-inversions or searching mechanisms that are performed in conventional linear or nonlinear detection algorithms. Furthermore, the DNN architectures, and their batch operations, are more natural for hardware implementation than hand-engineered algorithms, which is a critical distinction between the two.

An efficient DNN detector requires good designs across various aspects, including, but not limited to, the network architecture, input structure, and training strategy. In [5], it was shown that a generic fully-connected DNN with only the received signals and channel coefficients as inputs leads to poor detection performance. In contrast, DetNet [5], FS-Net [6], OAMP-Net2 and its predecessor OAMP-Net [8], and MMNet [9] can achieve excellent performance in MIMO detection by exploiting not only the learning ability of DL, but also the domain knowledge from hand-engineered data detection algorithms. All of these detectors follow an unfolding network architecture [10], allowing data detection to be performed in a layer-by-layer manner. The ingenuity of these architectures lies in the design of each layer, derived from well-developed data detection algorithms, leading to their differing performance and complexity.

GRADIENT DESCENT-BASED DNN DETECTORS

A gradient descent-based DNN detector incorporates the projected gradient descent (PGD) algorithm into the unfolding network architecture in an ingenious way. The network mimics the update process of the PGD algorithm and generates an estimated symbol vector at each layer. The operation at layer-ℓ is modeled by a nonlinear transformation $\hat{\mathbf{x}}_\ell = f_{gd}(\hat{\mathbf{x}}_{\ell-1} - \delta_\ell \mathbf{H}^T y + \delta_\ell \mathbf{H}^T \mathbf{H} \hat{\mathbf{x}}_{\ell-1})$, which accepts the output of the previous layer and information from the channels and the received signals as inputs. The network is trained to optimize the nonlinear transformation $f_{gd}(\cdot)$ and the step sizes δ_ℓ, motivating the developments of DetNet and FS-Net.

DetNet: To learn the nonlinear projection $f_{gd}(\cdot)$ of the original PGD method, DetNet employs a trainable parameter set, including the weights, biases, and step sizes. We illustrate the operation of the ℓth layer of DetNet in Fig. 1. In the DetNet architecture, a soft quantizer ψ is introduced at the end of the layer to perform a soft element-wise quantization of the output $\hat{\mathbf{x}}_{\ell+1}$. This ensures that the elements of $\hat{\mathbf{x}}_{\ell+1}$ are in an appropriate range specified by the modulation scheme. In DetNet, the initial solution $\hat{\mathbf{x}}_0$ is set to all-zero vector, which is then updated over L layers of the DNN to approach the true transmit signal vector by minimizing the loss function $\sum_{\ell=1}^{L} \log(\ell) \| x - \hat{\mathbf{x}}_\ell \|^2$. The final solution of DetNet is obtained by a hard quantization of the last layer's outputs (i.e., $\hat{\mathbf{x}}_L$) to the nearest symbols in the alphabet $\mathcal{A}$.

The network architecture and operations of DetNet exhibit the following potential issues:
- The value of the loss function of one layer in DetNet is added to the total loss of the network with a discounted weight, while the solution predicted in one layer is obtained using only the connections in that layer and the input passed from the previous layer. Sophisticated features cannot be extracted within one layer, implying that the loss function of DetNet limits the learning ability of multiple hidden layers in a general DNN. Furthermore, it is evident that this loss function only minimizes the total loss of all the layers. However, it does not minimize the number of required layers to accelerate the training and prediction [6].
- As seen in Fig. 1, an intermediate signal vector vl is concatenated with $\hat{\mathbf{x}}_\ell - \delta_{\ell+1}\mathbf{H}^T y + \delta_{\ell+1}\mathbf{H}^T\mathbf{H}\hat{\mathbf{x}}_\ell$ to form the inputs that are processed by the network connections. Although vl helps to overcome the limitations of the loss function [5], it enlarges the size of the input vector and additionally requires a sub-network associated with the trainable parameter set $\{\mathbf{W}_{3\ell}, \mathbf{b}_{3\ell}\}$. This makes DetNet computationally expensive. Moreover, the use of different step sizes (i.e., $\theta_{1\ell}, \theta_{2\ell}$) in DetNet is not clearly motivated or suggested by the PGD procedure in f_{gd}. We note that both $\mathbf{v}_\ell$ and $\theta_{2\ell}$ can be removed in

refined versions of DetNet.

Despite the above limitations, DetNet offers several performance advantages. In simulations for a massive MIMO system with 60 receive and 30 transmit antennas and binary phase-shift keying (BPSK), DetNet exhibits a 2-dB performance gain over the ZF detector and performs very close to the SD scheme with much lower complexity. This justifies the potential of DNNs in estimating the symbols transmitted via fading channels and observed by a noisy receiver.

FS-Net: FS-Net is proposed in [6] to overcome the limitations of DetNet. It achieves not only a considerable complexity reduction thanks to a simple network architecture (Fig. 2), but also significant performance improvement compared to DetNet [6]. These gains are obtained thanks to the following improvements:

• FS-Net does not require the intermediate vector $\mathbf{v}_\ell$ and only uses one trainable step size. Clearly, this simplifies the network structure and reduces the number of trainable parameters approximately by half, facilitating a better training while also being more computationally efficient than the original DetNet.

• In FS-Net, pair-wise connections between the input and output nodes are deployed instead of full connections as in DetNet. This is motivated by the fact that in f_{gd}, an element of the output $\hat{\mathbf{x}}_{\ell+1}$ only depends on the corresponding element of $\hat{\mathbf{x}}_\ell$. The pair-wise connections significantly reduce the number of trainable parameters.

• Finally, FS-Net employs an optimized loss function to accelerate the convergence in the training phase. The new loss function takes into account the correlation between the output of each layer and the label (i.e., the true transmitted signal vectors), thus ensuring that $\hat{\mathbf{x}}_\ell$ can reach $\mathbf{x}$ with fewer layers, compared to DetNet.

APPROXIMATE MESSAGE PASSING-BASED DNN DETECTORS

In this section, we review another prominent group of DNN detectors, consisting of OAMP-Net2 [8] and MMNet [9]. Similar to their PGD-based counterparts, OAMP-Net2 and MMNet follow the unfolding technique [10]. However, the major distinction between these two groups is the domain knowledge leveraged for constructing the layered architecture. The DNN detectors in this group are based on the iterative OAMP signal recovery algorithm.

The OAMP framework sequentially invokes a linear estimator and a nonlinear denoiser to refine the recovered signal. At iteration ℓ, it computes a linear estimate $\mathbf{r}_\ell = \hat{\mathbf{x}}_\ell + \mathbf{W}_\ell(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}_\ell)$, using the estimated signal from the previous iteration and a linear estimator $\mathbf{W}_\ell$. The linear estimate is then passed through a nonlinear denoiser $\eta(\mathbf{x}\,|\,\mathbf{r}_\ell, \tau_\ell)$ that provides a *divergence-free* estimate $\hat{\mathbf{x}}_{\ell+1}$. This nonlinear denoiser is an affine function of the posterior mean $\tilde{\eta}(\mathbf{x}\,|\,\mathbf{r}_\ell, \tau_\ell) = \mathbb{E}[\mathbf{x}\,|\,\mathbf{r}_\ell = \mathbf{x} + \tau_\ell\mathbf{z}]$, where τ_ℓ^2 is treated as the error variance and $\mathbf{z}$ is an i.i.d. standard Gaussian distributed error vector after the linear estimation. To improve the performance of OAMP as a data detection algorithm, OAMP-Net2 and MMNet were proposed to leverage the learning ability of DNNs for optimizing the free parameters in the linear estimator and the nonlinear denoiser.
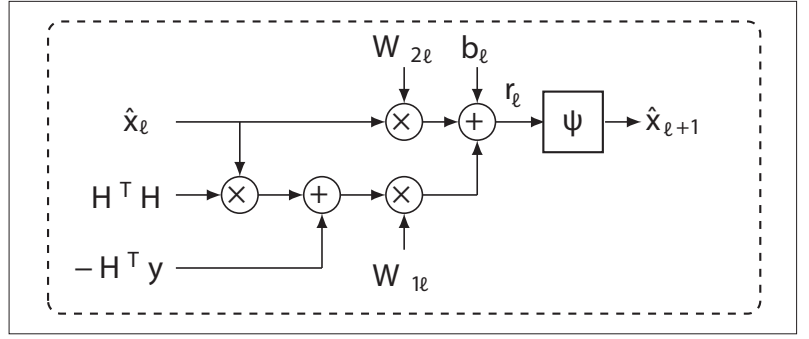

FIGURE 2. The ℓth layer of FS-Net with trainable parameters $\theta_\ell = \{\mathbf{W}_{1\ell}, \mathbf{W}_{2\ell}, \mathbf{b}_\ell\}_{\ell=1}^L$ with diagonal $\mathbf{W}_{1\ell}$ and $\mathbf{W}_{2l}$ and a soft quantizer ☐.
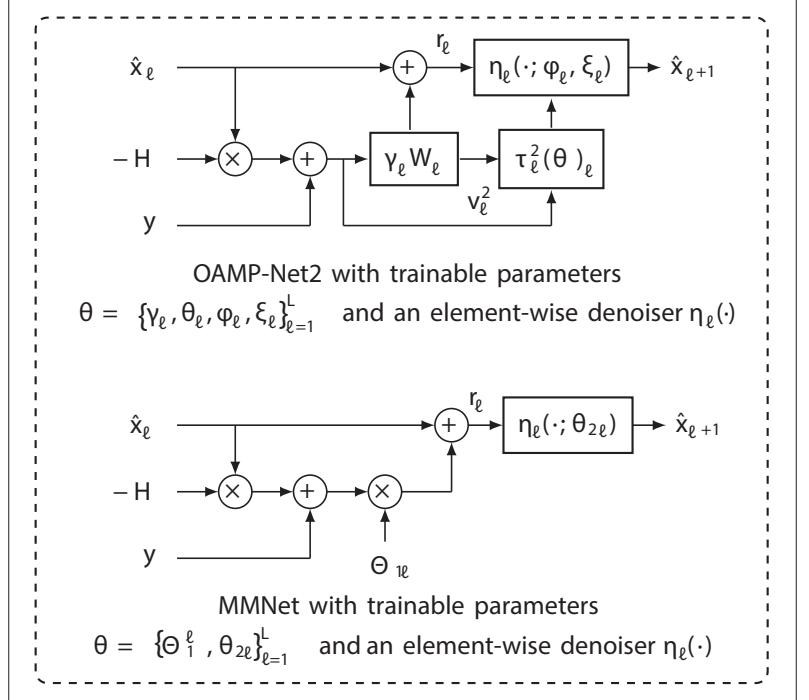

FIGURE 3. The ℓth layer of OAMP-Net2 and MMNet with trainable parameters θ.

OAMP-Net2: OAMP-Net2, as illustrated in Fig. 3, and its predecessor OAMP-Net strictly follows the OAMP framework. Specifically, He *et al.*, [8] proposed the training of four variables $\{\gamma_\ell, \theta_\ell, \phi_\ell, \xi_\ell\}$ at each layer to form the linear estimate $\hat{\mathbf{x}}_\ell + \gamma_\ell\mathbf{W}_\ell(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}_\ell)$ and the denoiser $\eta(\cdot; \phi_\ell, \xi_\ell, \tau_\ell) = \phi_\ell\tilde{\eta}(\mathbf{x}\,|\,\mathbf{r}_\ell, \tau_\ell) - \xi_\ell\mathbf{r}_\ell(\gamma_\ell)$. The trained parameters can significantly improve the accuracy and convergence of the nonlinear estimator. Specifically, $\{\gamma_\ell, \theta_\ell\}$ can improve the accuracy in estimating the prior mean $\mathbf{r}_\ell$ and variance $t_\ell^2(\theta_\ell)$ in the nonlinear estimator. At the same time, ϕ_ℓ and ξ_ℓ are trained to achieve a better divergence-free nonlinear estimator $\hat{\mathbf{x}}_{\ell+1}$ than the analytical solution in [7].

OAMP-Net2 achieves an impressive performance improvement compared to the conventional linear/nonlinear detectors. Specifically, a numerical example for an 8×8 MIMO system with i.i.d. Rayleigh fading channels shows that it can perform 5-dB and 10-dB better than the classical OAMP and LMMSE schemes [8]. However, like OAMP, OAMP-Net2 is strictly based on the assumption of unitarily-invariant channels. Therefore, it has a significant performance loss for real-
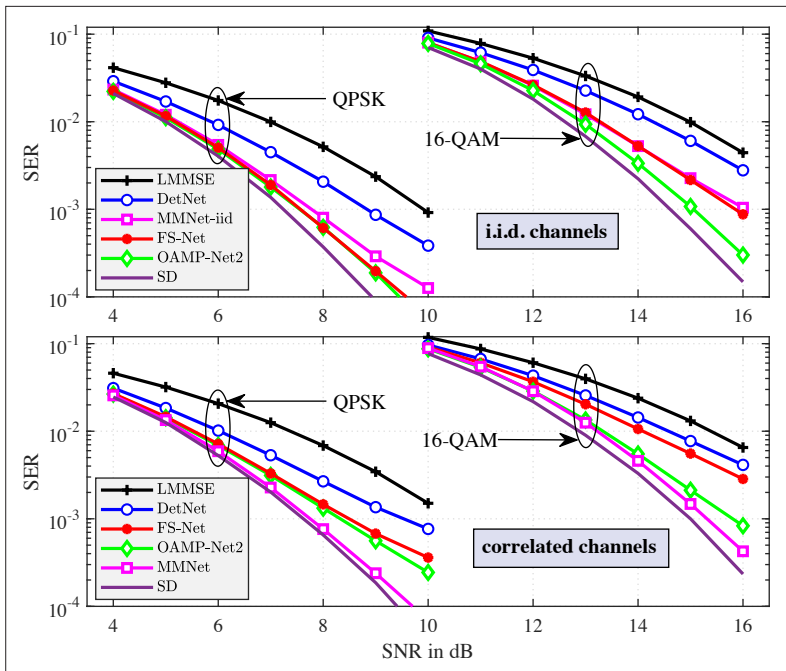
FIGURE 4. Performance comparison between LMMSE, SD, and different detection networks with i.i.d. and correlated channels.
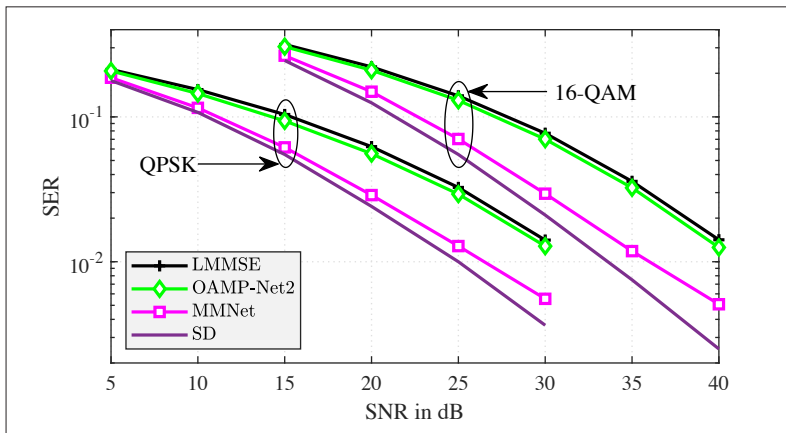

FIGURE 5. Performance comparison between LMMSE, SD, and different detection networks with realistic 3GPP channels.

istic correlated channel models [9]. Furthermore, OAMP-Net2 imposes high complexity (even higher than the classical OAMP) due to the additional trainable parameters.

MMNet: Khani *et al.*, proposed MMNet [9] to overcome the limitations of OAMP-Net. Similar to OAMP-Net and OAMP-Net2, MMNet unfolds the iterative update of the linear and nonlinear estimators. However, a significant improvement is made to overcome the poor performance of the OAMP algorithm for correlated channels. As illustrated in Fig. 3, MMNet can be summarized as follows:

- Matrix $\mathbf{W}_\ell$, which represents the linear transformation in the linear estimator of OAMP and OAMP-Net2, is cast as a trainable matrix variable $\theta_{1\ell}$ in MMNet. This allows the linear estimator $\mathbf{r}_\ell$ to include more trainable parameters that can be optimized for each channel realization. Moreover, it also avoids the matrix inversion in $\mathbf{W}_\ell$, required by OAMP and OAMP-Net2.
- To handle the cases in which different transmitted symbols have different noise levels,

the error variance τ_ℓ^2 is parameterized by a length-*K* vector $\theta_{2\ell}$, corresponding to the estimated error variances for the *K* users at the denoiser input.

MMNet offers more flexibility in designing the linear estimator and the denoiser, compared to the OAMP algorithm. Simulation results in [9] showed that MMNet outperforms OAMP-Net by 3-dB and reduces the computational complexity by a factor of 10–15 for practical 3GPP channels. It is, however, noted that MMNet requires retraining for each channel realization. A simplified version of MMNet, called MMNet-iid, was also proposed in [9] for detection with i.i.d. Gaussian channels. In MMNet-iid, the trainable matrix/vector $\theta_{1\ell}$ and $\theta_{2\ell}$ are replaced by $\theta_{1\ell}\mathbf{H}^T$ and $\theta_{2\ell}$, respectively, where $\theta_{1\ell}$ and $\theta_{2\ell}$ are trainable scalars.

NUMERICAL EXAMPLES AND DISCUSSION

NUMERICAL EXAMPLES

Figures 4 and 5 provide performance comparisons between the discussed detection networks (i.e., DetNet, MMNet, FSNet, and OAMPNet2) and the conventional LMMSE and SD detectors. We consider $(K, N) = (16, 32)$ and set $L = 10$ for QPSK and $L = 15$ for 16-QAM. In the training phase, we set the learning rate to 10^{-3} and the batch training size to 1000. Simulations were implemented on a standard Intel Xeon CPU E3-1270 v5, 3.60 GHz with 16-GB RAM, using the Tensorflow library. It should be noted that except for MMNet, all the other detection networks are trained offline. MMNet was designed to be trained online (i.e., it has to be retrained whenever the channel matrix H changes).

The performance comparison in the upper part of Fig. 4 is, for the case of i.i.d. Rayleigh, fading channels. It shows that the DNN-based detectors outperform the LMMSE scheme. Among the considered DNN-based detectors, DetNet provides the worst performance. Compared to LMMSE, the gain of DetNet is only about 1-dB and 0.5-dB for QPSK and 16-QAM, respectively. The OAMP-Net2 detector provides the best performance (quite close to that of the SD method) with a 2-dB gain compared to LMMSE in both the cases of QPSK and 16-QAM. FS-Net performs as well as OAMP-Net2 for the case of QPSK, but worse for 16-QAM. The performance of MMNet-iid is between DetNet and OAMP-Net2.

The lower part of Fig. 4 presents a performance comparison for the case of spatially correlated channels. We assume that the channels from different users to the BS are uncorrelated but the channels from a given user to the receive antennas are spatially correlated and follow a typical urban channel model as described in [12]. It is also observed that the DNN-based detectors outperform the LMMSE scheme. Among the considered DNN-based detectors, DetNet also provides the least performance gain at about 1-dB and 0.5-dB over LMMSE for the case of QPSK and 16-QAM, respectively. MMNet achieves the lowest SER (also quite close to that of the SD method) in this correlated channel scenario thanks to its online training strategy, but with the cost of excessively high computational complexity. In contrast, the other DNN-based detectors are trained offline before the online detection (re-training is not required)

	LMMSE	FS-Net	MMNet-iid	OAMP-Net2	DetNet	SD	MMNet (include training time)		
							10 epochs	|100 epochs	500 epochs
QPSK	0.4×10^{-6}	2×10^{-6}	11×10^{-6}	13×10^{-6}	15×10^{-6}	$> 5 \times 10^{-4}$	1.4	12	60
16-QAM	0.4×10^{-6}	3×10^{-6}	23×10^{-6}	25×10^{-6}	50×10^{-6}	$> 6 \times 10^{-4}$	2.2	19	94

TABLE 1. Computational complexity comparison in terms of average run time (seconds).

and, thus, they have lower computational complexities compared to MMNet. Note that the complexity of offline training is generally ignored in the literature [3, 6]. The gain of MMNet compared to LMMSE is significant (more than 2-dB). While FS-Net and OAMP-Net2 give similar performance for QPSK, FS-Net performs worse than OAMP-Net2 for 16-QAM, similar to what was observed in i.i.d. channels.

Realistic 3GPP channels are considered in Fig. 5, where the QuaDRiGa 3GPP model [13] is adopted. We observed that the training process of DetNet and FS-Net did not converge with this channel model (a similar observation was reported in [9]). Therefore, we compare the two detection networks OAMP-Net2 and MMNet with LMMSE and SD. MMNet performs closest to SD and much better than OAMP-Net2 and LMMSE. As explained earlier, this is due to the online training strategy of MMNet.

Table 1 compares the computational complexity of the detection methods in terms of average run time. It is obvious that LMMSE has the lowest complexity since it is a linear detector. The complexity of FS-Net is the lowest among the network detectors. The run times of MMNet-iid and OAMP-Net2 are longer than that of FS-Net, because they use more complex denoisers and OAMP-Net2 requires a matrix inversion in each layer. Among the DNN detectors that use offline training, Det-Net has the longest run time because its layered structure is more sophisticated with many parameters and the input of each layer is also lifted to a much higher dimension. All the offline-training DNN detectors run faster than the SD detector. The computational complexity of MMNet is much higher than that of the other detectors because it must be trained online.

Discussion

DNN-based detection requires an offline training phase. The resulting trained DNN model is saved at the base station for online application. Once the DNN is deployed, the base-station does not require further training data, except for MMNet which uses online retraining. The following is a summary of the advantages and disadvantages of the presented detection networks:

- *DetNet* is better than LMMSE but computationally expensive due to its sophisticated structure.
- *FS-Net* has lower complexity than DetNet, but its performance is degraded with large constellations. Both FS-Net and DetNet may not converge for certain practical channels.
- *OAMP-Net2* performs well with i.i.d. and correlated channels, but not as well with realistic channels and is computationally expensive due to the use of matrix inversions.
- *MMNet*, while working well with any channel model, is very computationally expensive due to the need for online retraining. The simpli-

fied version MMNet-iid with offline training performs well for its targeted i.i.d. channels.

Given the above discussion, it is clear that the development of more efficient, low-complexity, and universally-applicable DNN-based detectors is of significant interest.

Open Research Problems

Learning to Learn the MIMO Detector

The aforementioned DNN detectors tune their inference rules based on the training data. If there is a change in the data distribution (e.g., spatially correlated channel or sparse channel, a new mapping for the transmitted symbols, or a spatially correlated noise model), the trained DNN detector may become obsolete. Retraining the DNN detector from scratch for each new data distribution may not be feasible. This issue prompts the consideration of meta-learning in the DNN detector design.

Meta-learning, also known as "learning to learn," aims to design a model that learns from the output of other learning models using previously observed tasks. A notable meta-learning approach is to train the meta-learner's initial parameters such that the model has maximal performance on new tasks with just a few gradient update steps [14]. In the context of DNN-based detection, it would be interesting to investigate how to apply meta-learning to pre-train the weights of the DNN detector to a good initialization point that generalizes well to new underlying data distributions.

Channel Estimation and Channel Decoding

A DNN detector requires knowledge of the channel, which must be estimated before the data detection phase. It would be interesting to investigate the performance of the DNN detectors with potential channel estimation mismatch. In addition, the novel model-based DNN architectures can be designed to carry out both channel estimation and data detection tasks.

Channel encoding/decoding is another integral part of communications systems. A well performing code typically requires soft inputs from the demodulator. Thus, it is important for a DNN detector to provide soft detection outputs to the channel decoder. In addition, the DNN detector should be able to accept soft outputs from the channel decoder as prior information for the data symbol vector. This implementation would allow turbo-like joint MIMO detection and channel decoding with DNNs.

DNN-Based Detection for Nonlinear MIMO Channels

The majority of the proposed DNN-based detectors in the literature tackle the detection problem in linear MIMO channels. However, a cost-efficient and energy-efficient massive MIMO system may use non-ideal hardware that is prone to

For massive MIMO systems that exhibit nonlinear power amplifiers and phase noise, developing novel DNN detectors is an open research direction.

impairments and nonlinear distortions. A DNN detector for massive MIMO systems with one-bit ADCs, proposed in a recent work [15], has shown significant performance gain over algorithm-based approaches. For massive MIMO systems that exhibit nonlinear power amplifiers and phase noise, developing novel DNN detectors is an open research direction.

CONCLUSION

We have reviewed several recent developments in DNN-based massive MIMO detection. By imitating the iterations in established MIMO detection algorithms with a predetermined number of layers, a DNN detector with learned and fine-tuned parameters can offer fewer detection errors with lower computational complexity at run time. We believe that DNN-based detection can contribute to the development of low-complexity technologies for modern and emerging wireless networks.

ACKNOWLEDGMENT

REFERENCES

[1] L. Lu *et al.*, "An Overview of Massive MIMO: Benefits and Challenges," *IEEE JSAC*, vol. 8, no. 5, Oct. 2014, pp. 742–58.

[2] F. Rusek et al., "Scaling up MIMO: Opportunities and Challenges With Very Large Arrays," IEEE Signal Process. Mag., vol. 30, no. 1, Jan. 2013, pp. 40–60.

[3] N. T. Nguyen, K. Lee, and H. Dai, "Application of Deep Learning to Sphere Decoding for Large MIMO Systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 10, May 2021, pp. 6787–6803.

[4] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless Networks Design in the Era of Deep Learning: Model-Based, AI-Based, or Both?" *IEEE Trans. Commun.*, vol. 67, no. 10, June 2019, pp. 7331–76.

[5] N. Samuel, T. Diskin, and A. Wiesel, "Learning to Detect," *IEEE Trans. Signal Process.*, vol. 67, no. 10, May 2019, pp. 2554–64.

[6] N. T. Nguyen and K. Lee, "Deep Learning-Aided Tabu Search Detection for Large MIMO Systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, June 2020, pp. 4262–75.

[7] J. Ma and L. Ping, "Orthogonal AMP," *IEEE Access*, vol. 5, Jan. 2017, pp. 2020–33.

[8] H. He *et al.*, "Model-Driven Deep Learning for MIMO Detection," *IEEE Trans. Signal Process.*, vol. 68, Feb. 2020, pp. 1702–15.

[9] M. Khani *et al.*, "Adaptive Neural Signal Detection for Massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, Aug. 2020, pp. 5635–48.

[10] J. R. Hershey, J. L. Roux, and F. Weninger, "Deep Unfolding: Modelbased Inspiration of Novel Deep Architectures," arXiv:1409.2574, 2014.

[11] M. A. Albreem, M. Juntti, and S. Shahabuddin, "Massive MIMO Detection Techniques: A Survey," *IEEE Commun. Surveys and Tutorials*, vol. 21, no. 4, 4th Qtr. 2019, pp. 3109–32.

[12] K. Pedersen, P. Mogensen, and B. Fleury, "A Stochastic Model of the Temporal and Azimuthal Dispersion Seen at the Base Station in Outdoor Propagation Environments," *IEEE Trans. Veh. Technol.*, vol. 49, no. 2, Mar. 2000, pp. 437–47.

[13] S. Jaeckel *et al.*, "QuaDRiGa: A 3-D Multi-Cell Channel Model With Time Evolution for Enabling Virtual Field Trials," *IEEE Trans. Antennas and Propag.*, vol. 62, no. 6, June 2014, pp. 3242–56.

[14] C. Finn, P. Abbeel, and S. Levine, "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks," *Proc. Int'l. Conf. Machine Learning (ICML)*, Sydney, NSW, Australia, July 2017.

[15] L. V. Nguyen, A. L. Swindlehurst, and D. H. N. Nguyen, "Linear and Deep Neural Network-Based Receivers for Massive MIMO Systems With One-Bit ADCs," *IEEE Trans. Wireless Commun.*, vol. 20, no. 11, Nov. 2021, pp. 7333–45.

BIOGRAPHIES

LY V. NGUYEN is currently pursuing a Ph.D. degree in a joint doctoral program with San Diego State University and the University of California, Irvine, USA.

NHAN THANH NGUYEN received a Ph.D. degree from Seoul National University of Science and Technology. He is currently with University of Oulu (UOULU), Finland.

NGHI H. TRAN [SM] received a Ph.D. degree from University of Saskatchewan, Canada. He is currently an Associate Professor at The University of Akron, USA.

MARKKU JUNTTI [F'20] received a Dr.Sc. degree from UOULU, Finland, where he has been a Professor since 2000.

A. LEE SWINDLEHURST [F] received a Ph.D. degree from Stanford University, USA. He is currently a Professor at University of California, Irvine, USA.

DUY H. N. NGUYEN [SM'19] received a Ph.D. degree from McGill University, Canada. He is currently an Associate Professor at San Diego State University, USA.