



Article

Deep Learning for Joint MIMO Detection and Channel Decoding

Taotao Wang^{1,2,*} , Lihao Zhang² and Soung Chang Liew² 

¹ College of Information Engineering, Shenzhen University; ttwang@szu.edu.cn

² Department of Information Engineering, The Chinese University of Hong Kong; ttwang@ie.cuhk.edu.hk, zl018@ie.cuhk.edu.hk, soung@ie.cuhk.edu.hk

* Correspondence: ttwang@szu.edu.cn;

Abstract: We propose a deep-learning approach for the joint MIMO detection and channel decoding problem. Conventional MIMO receivers adopt a model-based approach for MIMO detection and channel decoding in linear or iterative manners. However, due to the complex MIMO signal model, the optimal solution to the joint MIMO detection and channel decoding problem (i.e., the maximum likelihood decoding of the transmitted codewords from the received MIMO signals) is computationally infeasible. As a practical measure, the current model-based MIMO receivers all use suboptimal MIMO decoding methods with affordable computational complexities. This work applies the latest advances in deep learning for the design of MIMO receivers. In particular, we leverage deep neural networks (DNN) with supervised training to solve the joint MIMO detection and channel decoding problem. We show that DNN can be trained to give much better decoding performance than conventional MIMO receivers do. Our simulations show that a DNN implementation consisting of seven hidden layers can outperform conventional model-based linear or iterative receivers. This performance improvement points to a new direction for future MIMO receiver design.

Keywords: deep learning; MIMO detection; channel decoding; deep neural network

1. Introduction

Multiple-antenna technology, also known as multiple-input multiple-output (MIMO), is one of the most important techniques for advanced wireless communications systems. It has already been incorporated into many wireless standards, e.g., 802.11n/ac [1] and LTE 4G [2]. It has also been shown theoretically that MIMO can increase spectrum efficiency linearly with the numbers of transmit and receive antennas [3]. Of much interest are low-complexity MIMO functional units that have good performance.

A MIMO transmitter transmits multiple data streams, one on each transmit antenna. A MIMO receiver receives a multiplexed copy of the multiple data streams plus noise on each receive antenna. A MIMO detector demultiplexes and decodes the multiplexed data on all the receive antennas into the originally transmitted multiple data streams plus noise and interference.

To achieve near-capacity performance, advanced channel coding schemes, such as LDPC and polar codes, have been suggested for 5G systems [4,5]. These channel codes protect the data streams against channel fading, interference, and noise. The output of a MIMO detector consists of a noisy version of the codeword transmitted by the transmitter. The function of channel decoding is to map the noisy codeword to the original information bits at the transmitter.

For optimal MIMO decoding, MIMO detection and channel decoding need to be performed in a joint manner. The conventional MIMO decoding schemes all use a model-based approach. However, due to the complex MIMO signal model, the optimal solution to the joint MIMO detection and channel decoding problem (i.e., the maximum likelihood decoding of the transmitted codewords from the received MIMO signals) is computationally infeasible.

As a practical measure, the current model-based MIMO receivers all use suboptimal MIMO decoding methods with affordable computational complexities. For example, instead of joint MIMO detection and channel decoding, [6–8] proposed to perform MIMO detection and channel decoding sequentially and separately, where MIMO detection is realized by linear equalizations with zero forcing (ZF) or minimum mean square error (MMSE) criteria. By contrast, [9–11] proposed to perform MIMO decoding and channel decoding iteratively with soft information exchanges between the two components. Thus, MIMO detection and channel decoding are performed in a joint manner. However, to contain complexity, the original MIMO signal model has been relaxed and replaced by an approximate model (i.e., it separately models the MIMO signal and the channel code). As a result, the solutions are still suboptimal. This leaves a gap for further performance improvement with better MIMO decoder designs.

To narrow the performance gap, this work applies the latest advances in deep learning for the design of MIMO receivers. In particular, we leverage deep neural networks (DNN) with supervised training to solve the joint MIMO detection and channel decoding problem. We show that DNN can be trained to give much better decoding performance than conventional MIMO receivers do. Our simulations show that a DNN implementation consisting of seven hidden layers can outperform conventional model-based linear or iterative receivers.

1.1. Related Work

Many MIMO detection schemes have been proposed [12]. Linear MIMO detection can first be used to cancel multiple-antenna interference with low complexities; after that channel decoding is performed [6–8]. In these schemes, linear MIMO detection and channel decoding operate in a sequential manner. Since linear MIMO detection introduces noise amplification and correlation, such sequential linear MIMO detection and channel decoding schemes typically result in large performance loss due to the mismatch between the noise models at the output of the MIMO detector and the input of the channel decoder.

To enhance the performance of MIMO detection, nonlinear MIMO detectors have also been proposed, e.g., MIMO detectors based on sphere decoding [13–15], semi-definite relaxation [16,17], and lattice reduction [15,18]. Unfortunately, these nonlinear MIMO detectors can only output hard estimates of channel symbols, making them incompatible with modern channel decoders that require soft input to achieve superior decoding performance.

Sphere decoding and list decoding algorithms were used for soft MIMO detection [9–11,19] that produces soft output. This soft information can then be fed to a channel decoding. Moreover, information exchange can be performed iteratively between soft MIMO detection and channel decoding to improve the overall performance of MIMO decoding. Although these iterative MIMO decoding schemes have better performance than the sequential schemes, their solutions are still approximate and suboptimal, due to the mismatch between the noise model of the soft output of the MIMO detector and the assumed noise model at the input of channel decoder. Furthermore, iterative information exchange introduces large decoding latencies.

Unlike the above model-based approaches, [20] proposed a deep learning approach for MIMO detection. Specifically, the method approximates MIMO detection using deep neural networks (DNN). The method progressively improves the approximation by adjusting the weights of a DNN based on a series of training MIMO signals. Compared with model-based MIMO detection, deep-learning MIMO detection achieves similar detection accuracies with faster detection speed. However, this deep-learning MIMO detection scheme can only perform hard MIMO detection and cannot be combined with a soft channel decoding scheme.

DNN is used to perform channel decoding for the first time in [21], followed by further work in [22,23]. It was shown that DNN channel decoding can approach the MAP performance with lower decoding latency than traditional channel decoding. Work [24] employed a neural network constructed by unfolding the factor graph of linear codes to improve the performance of belief

propagation decoding when the factor graph of the linear codes contains many small loops. Work [25] investigated DNN-based joint equalization and channel decoding problem for non-MIMO systems. A survey on the applications of deep learning to wireless systems can be found in [26].

The remainder of this paper is organized as follows. Section II presents the system model of MIMO systems. Section III reviews the existing model-based MIMO receivers. Section IV presents our deep learning MIMO receiver. Section V provides the simulation results. Finally, Section V concludes the paper.

2. System Model

This section presents the system model of MIMO systems and the format of the received MIMO signals.

Consider a MIMO system where the transmitter is equipped with M_T antennas and the receiver is equipped with M_R antennas. The channel between each transmit-receive antenna pair is assumed to incur frequency-flat fading and the channel state remains constant within one transmitted packet. We assume $M_T \leq M_R$ and M_T parallel data streams are transmitted, one on each transmit antenna.

Figure 1 shows the block diagram of the MIMO transmitter. At the transmitter side, a vector of K information bits, $\mathbf{b} = [b_1, b_2, \dots, b_K]^T$, is first channel-encoded into a codeword vector $\mathbf{c} = [c_1, c_2, \dots, c_N]^T$ of length $N = K/R$, where R is the code rate. The valid set of codewords is denoted by \mathbf{C} and thus $\mathbf{c} \in \mathbf{C}$. The coded bits in vector \mathbf{c} are modulated to a vector of complex data symbols, $\tilde{\mathbf{x}} = [x_1, x_2, \dots, x_{N/B}]^T$, where B is the number of code bits per complex data symbol. The modulation constellation is scaled so that the modulated symbols in $\tilde{\mathbf{x}}$ have unit average power. Through serial-to-parallel conversion, the vector $\tilde{\mathbf{x}}$ is partitioned into $L \triangleq N/(BM_T)$ consecutive data vectors of length M_T , $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_L\}$, i.e., we have $\tilde{\mathbf{x}} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \dots, \mathbf{x}_L^T]^T$. Then, L' pilot vectors of length M_T , $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{L'}\}$, are prepended to the data vectors $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_L\}$ to form an $M_T \times (L' + L)$ signal matrix $\mathbf{X} = [\mathbf{X}_p, \mathbf{X}_d]$, where $\mathbf{X}_d \triangleq [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_L]$ is the $M_T \times L$ data matrix that contains the data vectors, and $\mathbf{X}_p \triangleq [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{L'}]$ is the $M_T \times L'$ pilot matrix that contains the pilot vectors. We assume $L' \geq M_T$ to facilitate the channel estimation [27]. The signal matrix \mathbf{X} represents one transmitted packet. The M_T symbols of the t -th column vector in the signal matrix \mathbf{X} are simultaneous transmitted on the M_T transmit antennas in the t -th time slot.

At the receiver side, the received signals are written into an $M_R \times (L' + L)$ matrix, $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{L'+L}]$, where the t -th vector \mathbf{y}_t contains the received signals on the M_R receive antennas in the t -th time slot. The received signal matrix can be written as

$$\mathbf{Y} = \sqrt{\frac{1}{M_T}} \mathbf{H} \mathbf{X} + \mathbf{W} \quad (1)$$

where \mathbf{H} is an $M_R \times M_T$ complex channel matrix with zero-mean and σ^2 -variance independent complex Gaussian entries, and \mathbf{W} is the $M_R \times (L' + L)$ additive white Gaussian noise (AWGN) matrix that has zero-mean and unit-variance independent complex Gaussian entries. We also divide the received signal matrix and the AWGN matrix into two subparts: $\mathbf{Y} = [\mathbf{Y}_p, \mathbf{Y}_d]$, $\mathbf{W} = [\mathbf{W}_p, \mathbf{W}_d]$, where \mathbf{Y}_p is the $M_R \times L'$ matrix that contains the received signal vectors for the transmitted pilot vectors, \mathbf{Y}_d is the $M_R \times L$ matrix that contains the received signal vector for the transmitted data vectors, and \mathbf{W}_p , \mathbf{W}_d , are the matrices containing the noise components in \mathbf{Y}_p , \mathbf{Y}_d , respectively. The aim of the MIMO receiver is to decode the transmitted information bits in \mathbf{b} from the received signal matrix \mathbf{Y} .

For comparison with our proposed MIMO receiver, in Section 3 we review some conventional model-based MIMO receivers.

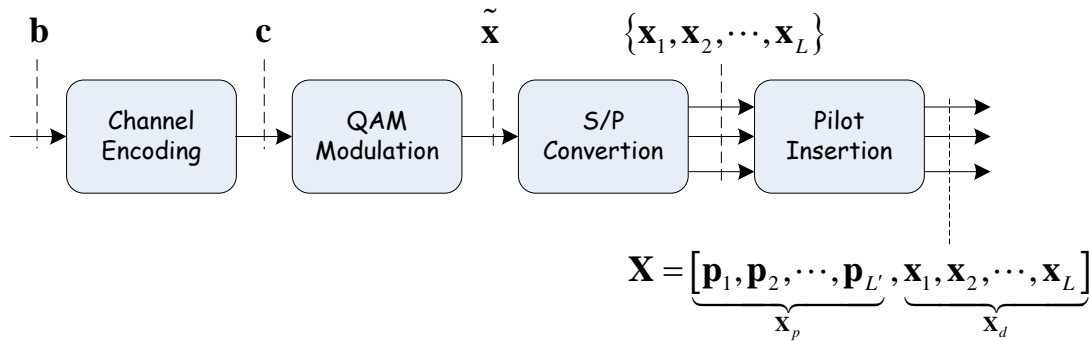


Figure 1. The block diagram for the MIMO transmitter.

3. Model-based MIMO Receivers

Traditional MIMO receivers have been extensively studied in the literature and implemented in real systems. This section gives a brief overview of these MIMO receivers.

A symbol-wise optimal MIMO receiver decodes each information bit, b_k , from the received signal matrix \mathbf{Y}_d by minimizing the symbol error probability or equivalently maximizing the *a posteriori probability* (APP):

$$\hat{b}_k = \arg \max_{b_k} p(b_k | \mathbf{Y}_d, \mathbf{H}, C) \quad (2)$$

where \hat{b}_k denotes the estimate of the information bit b_k , and $k \in \{1, 2, \dots, K\}$. The problem as expressed in (2) is in fact a *joint MIMO detection and channel decoding* problem, since data symbol detection and the channel decoding are implicitly performed in (2). We point out that joint MIMO detection and channel decoding as in (2) require the knowledge of the channel matrix \mathbf{H} . In practice, the channel matrix is typically estimated from the received pilot signals \mathbf{Y}_p , e.g., the least square (LS) estimate of the channel matrix is given by: $\hat{\mathbf{H}} = \sqrt{M_T} \mathbf{Y}_p \mathbf{X}_p^H (\mathbf{X}_p \mathbf{X}_p^H)^{-1}$ [27]; then, the channel matrix estimate $\hat{\mathbf{H}}$ is substituted back to (2) to replace the real channel matrix \mathbf{H} .

Even with the above approximation which replaces \mathbf{H} by $\hat{\mathbf{H}}$, the exact computation of APP, $p(b_k | \mathbf{Y}_d, \hat{\mathbf{H}}, C)$, is difficult and highly complex. The computation difficulty is due to: i) the correlation among the data symbols introduced by channel encoding; ii) the parallel signal interference caused by the MIMO channel. Therefore, suboptimal MIMO detection and channel decoding schemes with manageable implementation complexities are typically used in practice. We overview two suboptimal schemes in the following.

3.1. Linear MIMO Receivers

One suboptimal MIMO detection and channel decoding approach is to cancel the parallel signal interference with a linear MIMO detection first and then perform channel decoding next. We refer to this approach as linear MIMO receivers. For example, the zero-forcing (ZF) detection [6] removes the interference by

$$\tilde{\mathbf{Y}}_d = \sqrt{M_T} (\hat{\mathbf{H}} \hat{\mathbf{H}}^H)^{-1} \hat{\mathbf{H}}^H \mathbf{Y}_d = \mathbf{x}_d + \tilde{\mathbf{W}}_d \quad (3)$$

where $\tilde{\mathbf{Y}}_d$ is the post-cancellation signals and $\tilde{\mathbf{W}}_d = \sqrt{M_T} (\mathbf{H}_d \mathbf{H}_d^H)^{-1} \mathbf{W}_d$ is the post-cancellation noise. Since parallel signal interference is already removed in (3), the post-cancellation signals, $\tilde{\mathbf{Y}}_d$, can be fed to a traditional channel decoder to recover data symbols. Figure 2 shows the block diagram for this linear MIMO receiver.

There is no loss of information in (3) since one can get back \mathbf{Y}_d from $\tilde{\mathbf{Y}}_d$. The suboptimality in the linear MIMO decoding arises from the fact that the traditional channel decoder assumes the transformed noise $\tilde{\mathbf{W}}_d$ is white, but it is actually not after the transformation in (3). Although the complexity of this linear MIMO receiver is low, its performance is far from optimal.

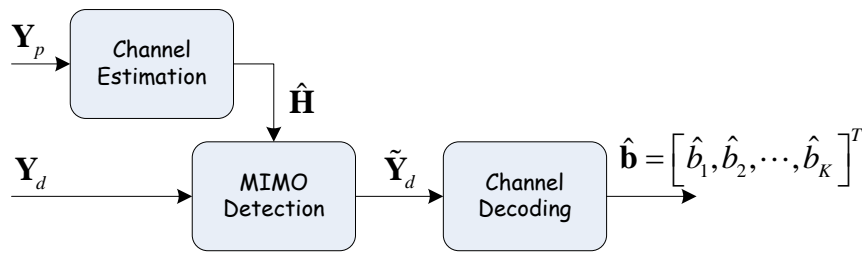


Figure 2. The block diagram for the linear MIMO receiver

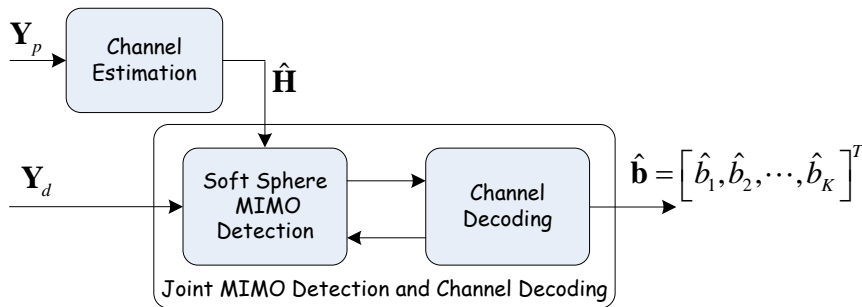


Figure 3. The block diagram for the iterative MIMO receiver.

3.2. Iterative MIMO Receivers

The second MIMO detection and channel decoding approach performs iterative soft-in soft-out MIMO detection and channel decoding. Using $\hat{\mathbf{H}}$, \mathbf{Y}_d and the prior information about the data symbols, a soft MIMO detector computes the extrinsic information about the data symbols [9] and delivers the soft information to a soft channel decoder. The soft channel decoder then computes the new extrinsic information about the data symbols and send the computed new extrinsic information back to the soft MIMO detector for further iteration.

In the next round of iteration, the soft MIMO detector replaces the prior information about the data symbols with the information sent from the soft channel decoder and re-computes its extrinsic information about these data symbols again. Several rounds of such iterations are performed to ensure the convergence of the overall MIMO detection and channel decoding process. We refer to such iterative MIMO detection and channel decoding schemes as iterative MIMO receivers. It yields an approximate solution to the joint MIMO detection and channel decoding problem expressed in (2).

Figure 3 shows the block diagram for the iterative MIMO receiver. The soft MIMO detection often used is the sphere algorithm [11] and the soft channel decoding often used is the belief propagation algorithm. The complexity of the iterative MIMO receiver is much higher than that of the linear MIMO receiver. Although the iterative MIMO receiver has better performance than the linear MIMO receiver does, there is still a large performance gap with respect to the optimal MIMO receiver. Moreover, the iterative information exchange introduces large decoding latency.

4. Deep-Learning MIMO Receivers

We propose to employ deep neural networks (DNN) to solve the joint MIMO detection and channel decoding problem stated in (2) with the goal of improving performance. The DNNs are trained under the framework of supervised learning.

We consider the training of DNN at the MIMO receiver after the channel matrix estimate $\hat{\mathbf{H}}$ has already computed from the received pilot signals. Using this channel matrix estimate $\hat{\mathbf{H}}$ at the MIMO receiver, we generate a set of training signals to train a DNN to solve the joint MIMO detection and channel decoding problem (2) under the framework of supervised learning. The training and deployment framework of DNN for MIMO is illustrated in Figure 4. We describe the associated procedures in the following.

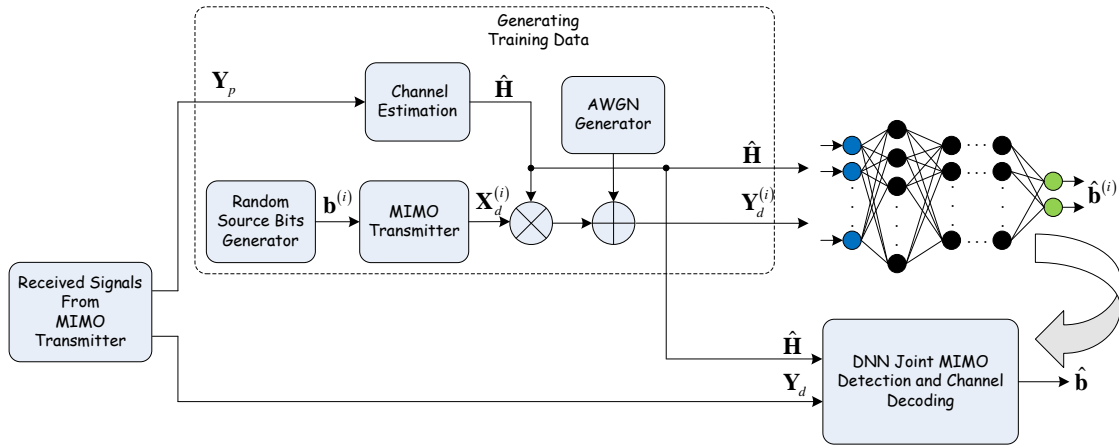


Figure 4. The block diagram for the iterative MIMO receiver.

The receiver generates the training data by calling a functional block that mimics the operation at the MIMO transmitter. Specifically, for training purposes, the receiver randomly generates many length- K binary vectors, $\mathbf{b}^{(i)}$, $i = 1, 2, \dots, Z$. Each binary vector $\mathbf{b}^{(i)}$ is transformed into a data matrix $\mathbf{X}_d^{(i)}$ using the functional block of the MIMO transmitter as described in Section II. Then, with the channel matrix estimate $\hat{\mathbf{H}}$ given by the channel estimator, the receiver generates a training signal by multiplying $\hat{\mathbf{H}}$ with $\mathbf{X}_d^{(i)}$ followed by adding AWGN:

$$\mathbf{Y}_d^{(i)} = \sqrt{1/M_T} \hat{\mathbf{H}} \mathbf{X}_d^{(i)} + \mathbf{W}_d^{(i)}$$

where $\mathbf{Y}_d^{(i)}$ is the i -th training signal and $\mathbf{W}_d^{(i)}$ is the corresponding generated AWGN. The training set is given by $D(\hat{\mathbf{H}}) = \left\{ \mathbf{Y}_d^{(i)}, \mathbf{b}^{(i)} \mid \hat{\mathbf{H}} \right\}_{i=1}^Z$, where $\mathbf{Y}_d^{(i)}$ is the i -th training signal and $\mathbf{b}^{(i)}$ is the corresponding label for $\mathbf{Y}_d^{(i)}$. We emphasize that the training set is dependent on the channel matrix estimate $\hat{\mathbf{H}}$.

We use the generated training set to train a DNN, $f_\theta(\cdot)$, that approximates the solution to problem (2), where θ is the set containing all the weights of the edges in the DNN. When we feed the training signals $\left\{ \mathbf{Y}_d^{(i)} \right\}_{i=1}^Z$ to the inputs of the DNN, we also feed the channel matrix estimate $\hat{\mathbf{H}}$ to the DNN (as illustrated in Figure 4). We optimize the DNN weights by miming the cross entropy loss function [28]:

$$L(\theta, D(\hat{\mathbf{H}})) = \frac{1}{ZK} \sum_{i=1}^Z \sum_{k=1}^K \left[b_k^{(i)} \ln(\hat{b}_k^{(i)}) + (1 - b_k^{(i)}) \ln(1 - \hat{b}_k^{(i)}) \right] \quad (4)$$

where $b_k^{(i)} \in \{0, 1\}$ is the k -th target information bit of the i -th label vector $\mathbf{b}^{(i)}$, $\hat{b}_k^{(i)}$ is the soft estimate of $b_k^{(i)} \in \{0, 1\}$ given by the DNN. The training algorithm used to minimize (4) for DNN is the so called stochastic gradient descent (SGD) algorithm [28]. After the training is finished, the weights of the DNN are fixed to $\hat{\theta}$ and we can use the trained DNN $f_{\hat{\theta}}(\cdot)$ to decode the received signals as $\hat{\mathbf{b}} = f_{\hat{\theta}}(\mathbf{Y}_d)$. We have the following remarks on this DNN for MIMO:

- The variables of interests to the DNN are the data symbols in $\mathbf{X}_d^{(i)}$. The size of the variable space is thus 2^K , where K is the length of $\mathbf{b}^{(i)}$ (Note that we have the one-to-one mapping: $\mathbf{b}^{(i)} \rightarrow \mathbf{X}_d^{(i)}$). According to the results shown in [21], if the DNN can see all possible codewords, the decoding performance of the DNN is the best. Like the investigation in [21], we also adopt short codes and train the DNN with all different codewords.
- The training of the DNN is quite time-consuming. Therefore, the training procedure will introduce a large decoding latency and it cannot be deployed for applications with stringent latency

requirements, such as voice transmissions; it is, however, suitable for data transmissions with relaxed latency requirements.

5. Simulation Results

In this section, we present simulation results for the evaluation of the proposed DNN MIMO receiver. The modulations used are BPSK and QPSK. The channel code used is the polar code [5] with code rate 1/2. We assume that each packet consists of $K = 16$ bits in the simulations. The adoption of the short packet length is due to the exponential training complexity when DNN is used to perform channel decoding [21].¹ Packets of short length are of interest in some practical systems such as the internet of things (IoT). After channel encoding and modulation, $K = 16$ information bits are transformed to 32 BPSK symbols or 16 QPSK symbols. Our simulations assume MIMO matrices of dimensions $M_R \times M_T = 2 \times 2, 4 \times 4$ and 8×8 .

We implement a DNN consisting of one input layer, six hidden layers and one output layer using the deep-learning software toolkit of Keras. The nonlinear activation function at the neurons of the input layer and the hidden layers is the Rectified linear unit (ReLU) function [28]. The input layer is a densely-connected layer. Each hidden layer is a densely-connected layer with batch normalization (BN) operations before the operations by ReLU. The output layer is a densely-connected layer with the sigmoid activation functions. The architecture of the DNN is illustrated in Figure 5. We train our NN over several “epochs”. In each epoch, the gradient of the loss function is computed over the entire training set using Adam, a method for stochastic gradient descent optimization [29]. Our training set contains all different 2^K codewords, K is the length of information bits. Setting the number of learning epochs to 10^5 , we train the DNN with datasets of different training SNRs (from 0 dB to 6 dB). After the training is finished, the trained DNN is used to decode the received MIMO signals.

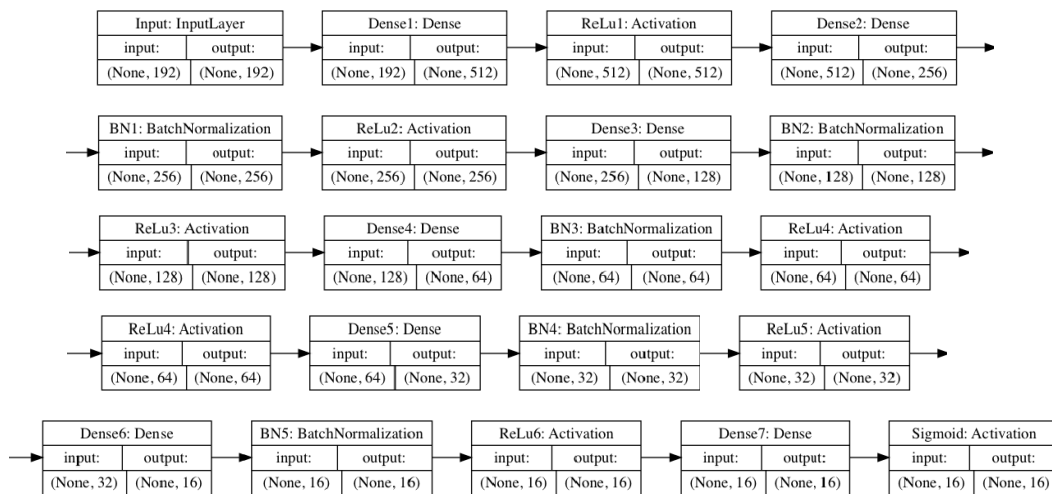


Figure 5. The architecture of the adopted DNN consisting of six hidden layers with 512, 356, 128, 64, 32, and 16 neurons respectively.

For comparison, we treat the following two traditional MIMO receivers as our benchmarks: i) the linear MIMO receiver that employs ZF MIMO detection followed by the MAP polar decoding of [5], ii) the iterative MIMO receiver that iterates between the sphere MIMO detection of [11] and the MAP polar decoding of [5]. We investigate the performance of MIMO receivers with perfect knowledge as well as with imperfect knowledge of the channel matrix. For the latter, we assume LS estimation [27]

¹ The extension of extend DNN channel decoding to long packet length can follow the solution of [23]. We will consider how to incorporate the solution of [23] into our DNN joint MIMO detection and channel decoding scheme in future work.

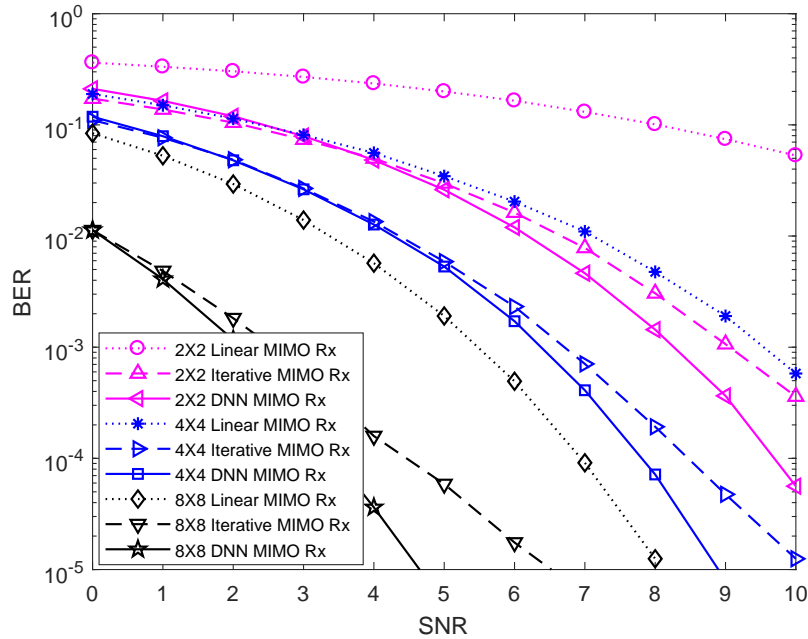


Figure 6. BER of the MIMO receivers with perfect knowledge of the MIMO channel matrix for BPSK.

is used to estimate the channel matrix. For a fixed SNR, we evaluate the average BER results of the MIMO receivers over 100 different MIMO channel realizations.

Figure 6 and Figure 7 show the BER of the MIMO receivers with perfect knowledge of the MIMO channel matrix for BPSK and QPSK, respectively. We can observe that our DNN MIMO receiver can indeed outperform the linear and iterative MIMO receivers in terms of BER. For example, the DNN MIMO receiver has around 1 dB and 3.5 dB SNR gain over the linear and iterative MIMO receivers, respectively, at the BER of 10^{-4} for BPSK and 8×8 MIMO channels.

Figure 8 and Figure 9 show the BER of the MIMO receivers with imperfect knowledge of the MIMO channel matrix for BPSK and QPSK, respectively. For the channel matrix estimation, we place a Hadamard matrix at the beginning of the packets as pilots and use the LS estimation based on the received pilots to estimate the channel matrix at the receivers. In general, the performance trend for the cases of perfect and imperfect channel estimates are the same. The only difference between them is that for the cases of imperfect channel estimates, the gain obtained by our DNN MIMO receiver is even larger. For example, the DNN MIMO receiver now has around 2 dB and 10 dB SNR gain over the linear and iterative MIMO receivers at the BER of 10^{-4} for BPSK and 8×8 MIMO channels.

6. Conclusions

This work used a deep-learning tool, deep neural network, to develop a new solution to the problem of joint MIMO detection and channel decoding. Conventional MIMO receivers perform MIMO detection and channel decoding in a sequential or an iterative manner. The algorithms of these conventional MIMO receivers relax the signal model of coded MIMO. As a result, they are suboptimal solutions to the joint MIMO detection and channel decoding problem, leaving the possibility for further improvement. Our deep learning solution uses a DNN for joint MIMO detection and channel decoding under the framework of supervised learning. The deep-learning MIMO receiver does not separate the MIMO detection and channel decoding into two parts and does not perform sequential or iterative operations on them. It treats the MIMO detection and channel decoding as a joint decoding process and employs a single DNN to approximate the joint decoding process. This joint process improves the overall decoding performance. In our simulations, we trained a DNN consisting of six hidden

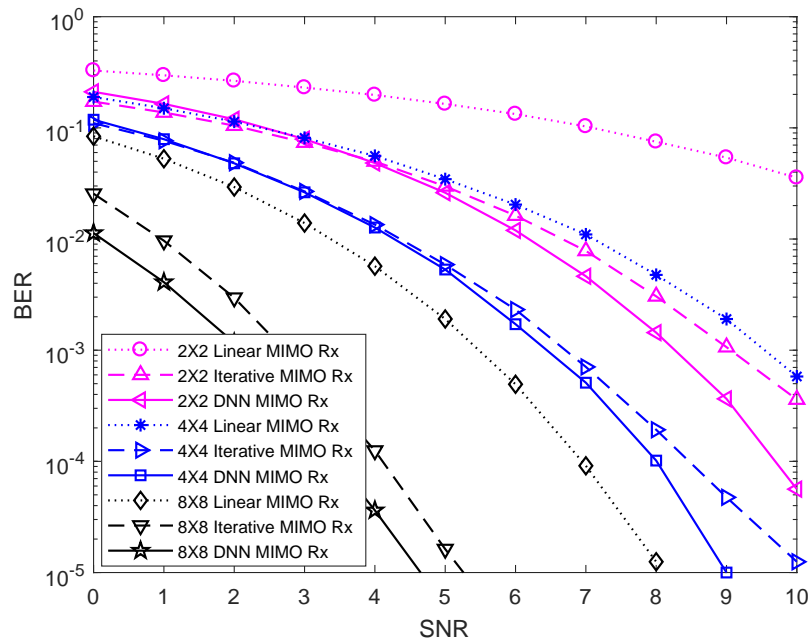


Figure 7. BER of MIMO receivers with the perfect knowledge of the MIMO channel matrix for QPSK.

layers to decode MIMO signals. The simulation results demonstrate notable gains obtained by our deep-learning MIMO receiver over the conventional linear and iterative MIMO receivers.

A drawback of the current proposed deep-learning MIMO receiver is that the DNN needs to be trained for each different channel matrix, introducing a large decoding latency. In general, to train the same DNN for MIMO decoding with different channel matrices is challenging, since the space of all possible channel matrices is huge. It is impossible to let the DNN see all the channel realizations. In [20], a scheme to construct one DNN for MIMO detection with different channel matrices is given. However, it is not clear how to extend the associated DNN to solve the problem of joint MIMO detection and channel decoding. A DNN for joint MIMO detection and channel decoding that can handle different channel matrices with one training (i.e., no need to train and readjust the weights in the DNN for each different channel matrix) is an interesting direction for further investigations.

Author Contributions: conceptualization, T.W. and S.L.; methodology, T.W. and S.L.; simulation, L.Z.; validation, L.Z., T.W. and S.L.; formal analysis, T.W., S.L. and L.Z.; writing—original draft preparation, T.W. and S. L.; writing—review and editing, T.W. and S.L.

Funding: The work was supported in part by the National Natural Science Foundation of China under Grant 61701311, and in part by the Natural Science Foundation of Guangdong Province, China, under Grant 2017A030310334, and in part by the Shenzhen NSF project under grant JCYJ20170818095107583.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

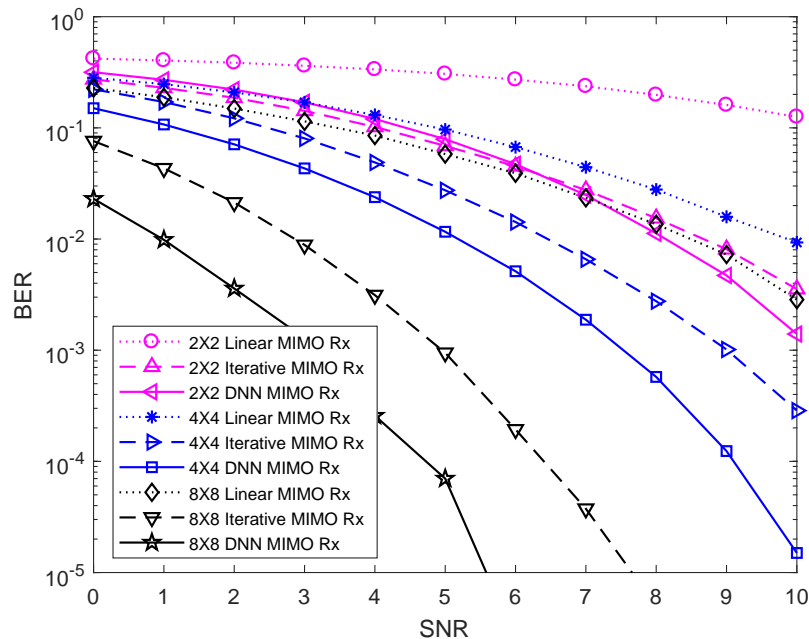


Figure 8. BER of MIMO receivers with imperfect knowledge of the MIMO channel matrix for BPSK.

MIMO	Multiple-input multiple-output
LDPC	Low density parity check code
DNN	Deep neural networks
LTE	Long term evolution
ZF	Zero forcing
MMSE	Minimum mean square error
LS	Least square
APP	A posteriori probability
BPSK	Binary phase shift keying
QPSK	Quadrature phase shift keying

References

1. Bejarano, O.; Knightly, E.; Park, M. IEEE 802.11ac: from channelization to multi-user MIMO. *IEEE Communications Magazine*, **2013**, *51*(10), 84–90.
2. Ghosh, A.; Ratasuk, R. Essentials of LTE and LTE-A. **2011**.
3. Goldsmith, A.; Jafar, S. A.; Jindal, N. et al. Capacity limits of MIMO channels. *IEEE Journal on Selected Areas in Communications*, **2003**, *21*(5), 684–702.
4. Richardson, T. ; Kudekar, S. Design of Low-Density Parity Check Codes for 5G New Radio. *IEEE Communications Magazine*, **2018**, *56*(3), 28–34.
5. Arikan, E. Channel polarization: A method for constructing capacity achieving codes for symmetric binary-input memoryless channels. *IEEE Trans. Inform. Theory*, **2009**, *55*(7), 3051–3073.
6. Haustein, T.; Helmolt, C. V.; Jorswieck, E. et al. Performance of MIMO systems with channel inversion. *IEEE Vehicular Technology Conference*, **2002**.
7. Kumar, K. R. ; Caire, G.; Moustakas, A. L. Asymptotic Performance of Linear Receivers in MIMO Fading Channels. *IEEE Transactions on Information Theory*, **2009**, *55*(10), 4398–4418.
8. Hedayat, A. Nosratinia, A. Outage and Diversity of Linear Receivers in Flat-Fading MIMO Channels. *IEEE Transactions on Signal Processing*, **2007**, *55*(12), 5868–5873.
9. Hochwald, B. M.; Brink, T. et al. Achieving near-capacity on a multiple-antenna channel. *IEEE Transactions on Communications*, **2003**, *51*(3), 389–399.

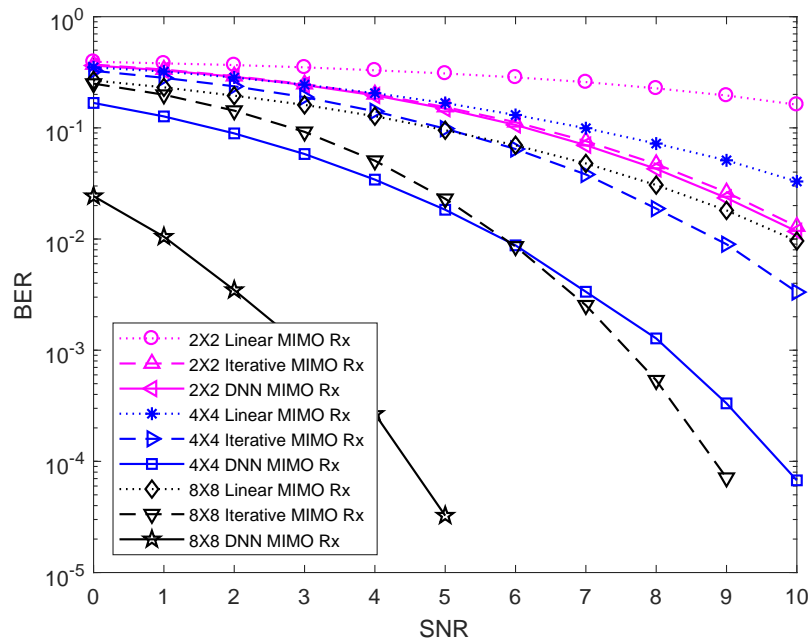


Figure 9. BER of MIMO receivers with imperfect knowledge of the MIMO channel matrix for QPSK.

10. Witte, E. M.; Borlenghi, F.; Ascheid, G. et al. A Scalable VLSI Architecture for Soft-Input Soft-Output Depth-First Sphere Decoding. *IEEE Transactions on Circuits & Systems II Express Briefs*, **2009**, 57(9), 706–710.
11. Studer, C. Helmut, B. Soft-Input Soft-Output Single Tree-Search Sphere Decoding. *IEEE Transactions on Information Theory*, **2010**, 56(10), 4827–4842.
12. Larsson, E. G. MIMO Detection Methods: How They Work (Lecture Notes). *IEEE Signal Processing Magazine*, **2009**, 26(3), 91–95.
13. Viterbo, E.; Boutros, J. A universal lattice code decoder for fading channels. *IEEE Transactions on Information Theory*, **1999**, 45, 1639–1642.
14. Barbero, L. G.; Thompson, J. S.; Fixing the complexity of the sphere decoder for MIMO detection. *IEEE Trans. Wireless Commun.*, **2008**, 7, 2131–2142.
15. Agrell, E.; Eriksson, T.; Vardy, A.; Zeger, K. Closest point search in lattices. *IEEE Transactions on Information Theory*, **2002**, 48, 2201–2214.
16. Tan, P.; Rasmussen, L. The application of semidefinite programming for detection in CDMA. *IEEE J. Select. Areas Commun.*, **2001**, 19, 1442–1449.
17. Steingrimsson, B.; Luo, Z.-Q.; Wong, K. Soft quasi-maximum-likelihood detection for multiple antenna wireless channels. *IEEE Trans. Signal Processing*, **2003**, 51(11), 2710–2719.
18. Windpassinger, C.; Fischer, R. Low-complexity near-maximum-likelihood detection and precoding for MIMO systems using lattice reduction. *IEEE Information Theory Workshop*, **2003**, 345–348.
19. Larsson, E. G.; Jaldén, J. Soft MIMO detection at fixed complexity via partial marginalization. *IEEE Trans. Signal Processing*, **2008**, 56, 3397–3407.
20. Samuel, N.; Diskin, T.; Wiesel, A. Deep MIMO detection. Deep MIMO Detection. *IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, **2017**.
21. Gruber, T.; Cammerer S.; Hoydis J.; Brink, S. On deep learning-based channel decoding. *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, **2017**.
22. Seo, J.; Lee, J.; Kim K. Decoding of Polar Code by Using Deep Feed-Forward Neural Networks. *2018 International Conference on Computing, Networking and Communications (ICNC)*, **2017**.
23. Cammerer, S.; Gruber, T.; Hoydis, J.; Brink, S. Scaling Deep Learning-based Decoding of Polar Codes via Partitioning. *arXiv:1702.06901*, **2017**.
24. Nachmani, E.; Marciano, E.; Lugosch, L.; Gross, W. J; Burshtein, D. Deep Learning Methods for Improved Decoding of Linear Codes. *IEEE Journal of Selected Topics in Signal Processing*, **2018**, 12(1), 119–131.

25. Ye, H. ; Li. G. Y. Initial Results on Deep Learning for Joint Channel Equalization and Decoding. *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, **2017**.
26. Wang, T.; Wen C.-K. ; Wang, H.; Gao, F. ; Jiang, T.; Jin S. Deep learning for wireless physical layer: Opportunities and challenges. *China Communications* ,**2017**, *14(11)*, 92–111.
27. Biguesh, M.; Gershman, A. Training-based MIMO channel estimation: A study of estimator tradeoffs and optimal training signals. *IEEE Trans. Signal Process.*,**2006**, *54(3)*, 884—893.
28. Goodfellow, Y. B.; Courville, A. Deep Learning. *MIT Press*,**2016**.
29. Kingma, D. P. ; Ba, J. Adam: A method for stochastic optimization. *CoRR*,**2017**, **2014**,