Deep Learning Decoder for MIMO Communications with Impulsive Noise

1st Oscar Delgado
Department of Electrical and Computer Engineering
McGill University
Montreal, Quebec, Canada
Email: oscar.delgadocollao@mail.mcgill.ca

2nd Fabrice Labeau
Department of Electrical and Computer Engineering
McGill University
Montreal, Quebec, Canada
Email: fabrice.labeau@mcgill.ca

***Abstract*—In this paper we consider signal detection in multiple-input-multiple-output (MIMO) systems with an impulsive noise channel. The existing, near optimal, sphere decoder (SD) achieves good performance, however, the computational complexity is directly related to the number of nodes visited during the tree search and the signal-to-noise ratio (SNR). Using neural network techniques, a Deep Learning Detector (DLD) is proposed. The DLD method detects signals transmitted in an impulsive noise channel, after an off-line training phase. The detection process of DLD has lower complexity than the average SD complexity, while exhibiting good performance. What is even more interesting is that the computational complexity of DLD is constant across SNR, in contrast to the SD detectors, which have an exponential complexity across the SNR. This constant complexity could be very helpful when implementing a detector in practice because it could allow for better optimization of resources. To evaluate the performance of our proposed method we have used a low level simulator that generates a fairly accurate model of a MIMO system with an impulsive noise channel. The complexity analysis and simulation results validate the arguments presented in this paper.**

***Index Terms*—Detector, Sphere decoder, MIMO, Deep learning, emerging technologies, wireless communications**

I. Introduction

In wireless communications, the variability of the transmission channel may significantly degrade data transmission. For example the presence of high voltage near electrical power stations creates fluctuations in the channel known as impulsive noise that reduces the overall transmission rate between a source and a destination [1]. In this paper we explore receiver design for channels impaired by such impulsive noise.

One of the major challenges in MIMO communication systems is to improve the bit error rate (BER) without increasing the complexity of the detector at the receiver [2]. The optimal receiver can be found by using the maximum a posteriori probability (MAP) algorithm. However, its complexity increases exponentially with respect to the modulation order, the number of transmit antennas and the SNR. Hence, a sub-optimal low-complexity detector is needed. In the high accuracy type of detectors the sphere decoder (SD), based on a tree search algorithm, was proposed. The SD offers a better computational complexity than MAP [3].

Although SD has lower complexity than MAP it also depends on the number of visited nodes in the tree search and the signal-to-noise ratio (SNR) [4], not to mention that SD cannot be parallelized due to its sequential nature, making it difficult for hardware implementations. Thus, a new detection method is needed.

One of the possible approaches to reduce complexity is to explore the tools of machine learning (ML). Machine learning (statistical learning, neural networks, etc.) is a set of techniques that enables systems with the ability to learn from experience without relying on an explicit algorithm. The learning process begins by observing the input data i.e., examples or direct experience; its objective is to find data patterns that will help to make better decisions in the future [5].

ML has been proven to work successfully for many different tasks and in different fields, such as data mining, computer vision, natural language processing etc.

ML techniques are becoming attractive because they can produce solutions that are easier to implement and can yield reasonably good performance [6]. In particular, deep learning fuelled by big data is able to capture complex correlations and minimize domain specific expertise. In essence this means that data pre-processing is reduced while still capturing abstract correlations. The term "deep" in "deep learning" refers to the number of layers the network has. In general most researchers agree that deep learning involves depth greater than two. Nevertheless networks with more than two layers are able to better capture feature information than shallow models [7].

In this paper, we address the problem of symbol detection on MIMO channels affected by impulsive noise via deep neural networks. The main contributions of this paper can be described as follows:

- We propose a new deep learning decoder (DLD) for MIMO communications in the presence of impulsive noise channels.
- We show that the proposed deep neural network can be trained on an impulsive noise channel and still have good performance on a Gaussian channel.
- Numerical results show that the proposed solution achieves lower computational complexity with similar detection performance to the sphere decoder.

The remainder of this paper is organized as follows. Section II briefly describes the related work, Section III presents the problem statement and the system model. Based on this model, our proposed deep learning architecture is then described in

Section IV. In Section V we analyze the simulation results. Finally, Section VI presents our conclusions.

II. Related work

Deep learning is gaining attention in applications like MIMO. In [8]–[11] a comprehensive survey on the different aspects related to MIMO communications is presented, i.e., channel estimation, detection, end-to-end system design, resource management, power control, etc.. In [12] the authors proposed an auto-encoder. By taking into account the signal characteristics they used an auto-encoder as a feature extractor, then they describe an Extreme Learning Machine method to classify the input signals of an OFDM MIMO system. Their proposed solution achieves high detection accuracy with similar complexity as baseline solutions.

Similarly the authors in [13], [14] proposed the design of an end-to-end communication system and the use of auto-encoders to jointly learn transmitter/receiver implementations and signal encoding/decoding processes. Simulation results over an additive white Gaussian noise channel shows comparable performance than previous works. Nevertheless the scalability of their solution remains a challenge. Practical implementations of auto-encoders are also described in [15], [16].

Another interesting solution was proposed in [17]. There, a deep learning structure for symbol detection in molecular/optical networks was analysed. One important claim in [17] is that neural network detectors perform well even without knowledge of the channel model. Their simulation results using a Poisson channel model demonstrate better performance than the Viterbi detector.

In [18], the authors proposed deep learning to jointly estimate channel state information (CSI) and detect/recover the transmitted symbols by using the estimated CSI. Their simulation results show that their approach can detect the transmitted symbols with similar performance as the minimum mean-square error estimator but with reduced complexity.

In [19] the authors explored simple methodologies of deep learning to conventional MIMO systems. It is also exposed the practical challenges of applying deep learning, e.g., proper design of the network structure. Their simulations deal with simple network structures for signal detection of one-tap MIMO channels. and the use of convolutional neural networks and recurrent neural networks for multipath fading channels. It is also shown in their simulations that for low SNR the performance is close to the maximum likelihood detection method.

A deep neural network for detection of modulation symbols in a quasi-static channel was proposed in [20]. The proposed solution defines a multi-plateau sigmoid function in combination with a twin-network neural structure. Simulation results show that close to Maximum-Likelihood performance can be achieved with a network structure with relatively low number of parameters.

In [21] the authors suggest the use of deep neural networks for MIMO detection. Inspired by the projected gradient descent algorithm they design a low complexity method that works particularly well with high number of transmit and receive antennas in a white Gaussian channel. A previous work of the same authors [22] suggest than deep learning works well even without prior knowledge of the SNR.

III. Problem statement

We consider a MIMO system with n transmit antennas and m receive antennas that is mapped into complex symbols using quadrature amplitude modulation (QAM). The basic block diagram of a MIMO system is depicted in Fig. 1.

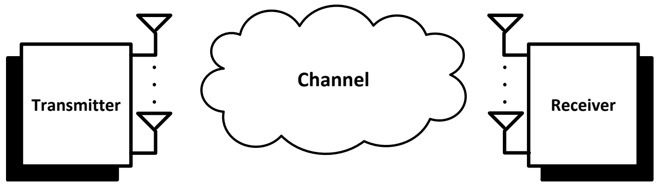

Fig. 1. Basic block diagram of a MIMO system.

Let H be the channel matrix, x the transmitted vector and y the received vector. The standard MIMO equation is:

$$y = Hx + \eta, \tag{1}$$

where η is the noise vector. Without loss of generality we assume that any complex vectors and matrices can be trivially transformed into their real representation and that the receiver has perfect knowledge of channel H. Then we have $H \in \Re^{2m\times 2n}$, $x \in \Re^{2n}$, $y \in \Re^{2m}$ and $\eta \in \Re^{2n}$. The aim is to recover the transmitted vector x from the received vector y with the smallest probability of error P_e.

$$P_e = P\{x \neq f(y)\}, \tag{2}$$

where $f(y)$ is the decoder estimate of x. The optimal decoder, also know as the Bayes decoder in the machine learning literature, can be found by using maximum a posteriori (MAP).

$$f(y)^{MAP} = \arg\max_x P(x|y), \tag{3}$$

assuming that the prior probability $P(x)$ is uniform we can find the maximum likelihood decoder $f(y)^{ML}$ as:

$$f(y)^{ML} = \arg\min_x \|y - Hx\|^2, \tag{4}$$

which in a sense means to find the x which minimizes the Euclidean distance. Solving equation (4) is NP-hard because a full search is needed (exponential complexity). To reduce complexity the SD decoder limits the search to those points inside a sphere of radius r centered around y. Introducing the radius constrain in the ML decoder we can define the SD decoder as:

$$f(y)^{SD} = \arg\min_x \{\|y - Hx\|^2 < r^2\}, \tag{5}$$

We should notice that the complexity of the SD decoder is much lower than the ML decoder but it is still exponential.

A. *Impulsive noise model*

Although the noise η is typically modelled as a Gaussian noise for classical wireless communication channels, in this paper, we use the noise model developed in [1] which is valid when there is a cohabitation of wireless networks and high power electricity networks (electric power equipment generates impulsive noise in bands covering the carrier frequencies of classic wireless communications). A representation of the Impulsive noise is depicted in Fig. 2.

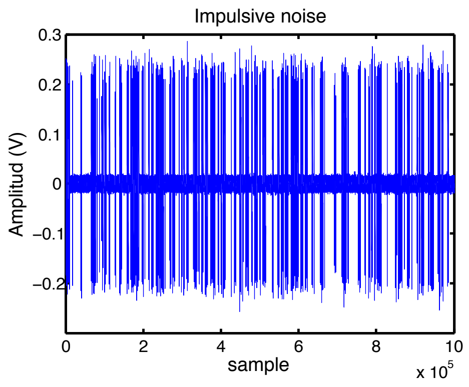

Fig. 2. Impulsive noise sample.

Impulsive noise is very different from Gaussian noise, as the impulsive noise is correlated and the samples follow a Gaussian mixture distribution. Specifically we have adopted the 6-state Partitioned Markov Chain model (PMC-6) that allow us to include the time-correlation of the noise observed in high voltage environments. PMC-6 offers sufficient degree of realism with lower implementation complexity.

IV. Deep Learning Decoder (DLD)

In this section, we describe our proposed deep learning decoder which is inspired by [21]. The estimating process is performed using a Deep Learning neural network through supervised learning. In essence we can think that the neural network will find the mapping of the function $f : y \rightarrow x$ which approximates the mapping of the system:

$$f(y;\theta) = \arg\max_{x} \hat{P}(x|y;\theta), \tag{6}$$

where θ denote the network parameters and $\hat{P}(x|y;\theta)$ is the estimated conditional probability density function. If the neural network is expressive enough it is possible to find a good estimator.

The deep learning process has two main stages, training and detection. In the first stage an off-line training, needed to find the network parameters, is performed. In the second stage, the neural network is deployed and used for detection.

A. *Training*

Based on the ML detector presented in equation (4), we can implement a projected gradient descent like solution, as described in [21]:

$$\begin{aligned}
\hat{x}_{k+1} &= \phi\Big[\hat{x}_k - \delta_k \frac{\partial \|y - Hx\|^2}{\partial x}\Big] \\
&= \phi[\hat{x}_k - \delta_k H^T y + \delta_k H^T H \hat{x}_k)] \\
&= \phi[(I + \delta_k H^T H)\hat{x}_k - \delta_k H^T y] \\
&= \phi[q_k],
\end{aligned} \tag{7}$$

where I is the identity matrix, $\phi[\bullet]$ is a projection operator, $\hat{x}_k$ is the estimate of x at layer $k = 1, .., L$, and δ_k is a step size.

It is important to mention that our iterative solution differs from [21] by using a simplified version of the same paradigm. Notice that in (7) assuming that $I + \delta_k H^T H$ is invertible we can conveniently rewrite q_k as:

$$q_k = W_{1k}\hat{x}_k - W_{2k} H^T y, \tag{8}$$

which yields a more compact solution represented in Fig. 3. Notice that W_{1k} reduces complexity by avoiding the need of some matrix operations (multiplication and addition)

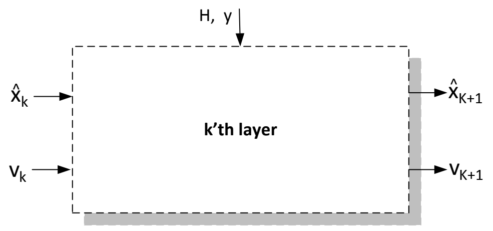

Fig. 3. Basic DLD diagram representing layer k

In Fig. 3 we have included v_k as an auxiliary input used to lift the inputs to a higher dimension; then the standard non-linearities of neural networks are applied. A detailed block diagram of our implementation is shown in Fig. 4.

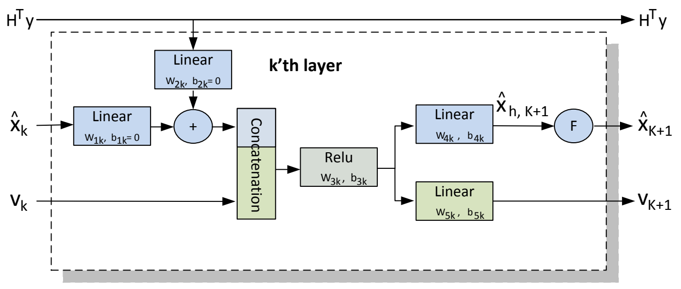

Fig. 4. Detailed DLD diagram representing layer k

Where $F(\hat{x}_{h,k}) = \hat{x}_k$ is the function used to transform the one-hot mapping $\hat{x}_{h,k}$, typically used by neural networks, to the real valued $\hat{x}_k$. One-hot representation implies that the allowed combinations of values are only those with a single (1) and all the others (0), e.g., in our simulations each real value is represented by a 4-bit one-hot vector. In summary the trainable parameters used in our implementation are:

$$\theta = \{W_{1k}, W_{2k}, W_{3k}, W_{4k}, W_{5k}, b_{3k}, b_{4k}, b_{5k}\}, \tag{9}$$

where W_{3k}, b_{3k} are the weight and bias terms used at the core of each layer in combination with a Relu activation function, and $W_{4k}, W_{5k}, b_{4k}, b_{5k}$ are the weights and bias terms used to shape the output of layer k. Now we define the loss function. it is important to mention that during training deep neural networks suffer from the vanishing gradients problem (back-propagation algorithm), in which the gradient becomes very small, thus preventing the weights from changing their value. To mitigate this problem GoogLeNet [23] proposed a loss function that is a combination of the intermediate loss and the final loss. We then define the loss function as:

$$l(\hat{x}_{h,k}(\theta)) = \sum_{k=1}^{L} \log(k) \|x_h - \hat{x}_{h,k}\|^2, \tag{10}$$

where x_h is the one-hot mapping representing the true x and $\hat{x}_{h,k}$ is the estimation of the one-hot mapping of x at layer k. If we fix the network structure and the network size then the parameters θ can be estimated as:

$$\theta^* = arg \min_{\theta} l(\hat{x}_{h,k}(\theta)). \tag{11}$$

Another well known technique, typically used and included in our implementation, to deal with the vanishing gradients problem is to use the residual feature from ResNet [24] in which the output of each layer is weighted with the output of the previous layer.

$$\begin{aligned} \hat{x}_{k+1} &= \alpha \hat{x}_k + (1-\alpha)\hat{x}_{k+1} \\ v_{k+1} &= \alpha v_k + (1-\alpha) v_{k+1}, \end{aligned} \tag{12}$$

where the ResNet parameter α is a parameter typically close to one ($0 < \alpha < 1$). Fig. 5 shows a block diagram of the full feed forward network implementation used by DLD.

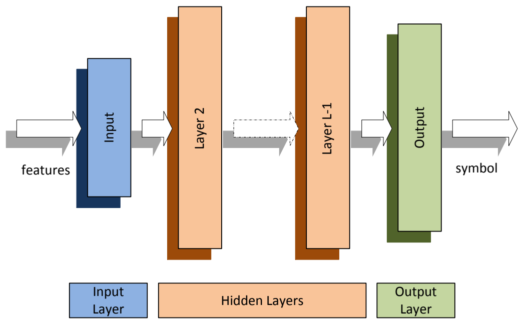

Fig. 5. Feed forward neural network used for DLD

B. Complexity analysis

There are different ways to analyze the complexity of a decoder. For the sphere decoder some authors focus on the number of points visited during the tree search given a chosen radius over particular values of SNR. Others prefer to use the run time needed to perform the detection. In this paper we analyse the number of multiplications and additions needed to perform the detection because it offers a fair perspective for comparison between the sphere decoder and our proposed deep learning detector. From our design we find that the total number of multiplications performed by DLD is:

$$[4n^2(2 + 2\omega\nu + \omega q + \omega)]L, \tag{13}$$

and the total number of additions is:

$$[4n^2(2 + 2\omega\nu + \omega q + \omega) - 2n(\omega + \nu + q - 1)]L, \tag{14}$$

where ω is the neural network width (maximal number of nodes of the ReLu block), ν is the size of vector v_k ($\omega \approx \nu$) and q is the modulation order. The complexity of the proposed detector DLD is in the order of $\mathcal{O}(n^2\omega^2 L)$, i.e. the algorithm has linear complexity in the number of layers L and a quadratic complexity in the number of antennas n and the neural network width ω. n is a given parameter, L and ω are network parameters to be tuned when designing the neural network; in practice we have found that the network is sensitive to big changes of L but not so sensitive to big changes of ω.

Due to space constraint we do not include the training complexity analysis, nevertheless to give an idea, using a desktop PC Intel Core i7 with 8MB of RAM it takes around 60 hours to train the network.

V. Numerical Results

In this section we evaluate the performance of DLD based on the Impulsive noise model and the Gaussian noise model.

A. Simulation settings

In our case, the dataset is composed of samples generated using the link level Vienna simulator [25]. We consider an LTE wireless communication network, using a 4×4 MIMO system. The input bits are modulated using a 16QAM modulation scheme.

1) Data collection: We assume that we know channel H for each transmitted symbol as well as the received signal y. We compare our results with the soft sphere decoder [26]. Table I summarizes the simulation parameters used for data collection.

TABLE I
Wireless Network Parameters

Parameter	value
Communication system	LTE
Carrier frequency	2.1 GHz
Modulation scheme	16 QAM
Transmission type	MIMO
Antenna configuration	4 transmit, 4 receive (4 x 4)
Channel model	3GPP TU
Decoder (used for comparison)	Soft Sphere decoder
Impulsive noise model	PMC-6

2) Training: We design and evaluate the performance of our deep learning algorithm using Tensorflow in Python. During training one of the questions that arises besides the classical question of how to tune the hyper-parameters is what is the SNR level at which we can train the network so that we get the best performance. From our experience, we have observed that when one sample out of hundred is misclassified the neural network manages to explore the whole symbol space and captures the relationships needed to have good estimations, i.e., training at the SNR when the probability of error P_e is at around 10^{-2}. In our case this happens at $26dB$. Table II summarizes the neural network parameters used for training.

TABLE II
NEURAL NETWORK PARAMETERS

Parameter	value
Number of samples used for training	40 Million
Number of layers	50
x size	8
v size	$16 \times x$ size
Relu layer width	$18 \times x$ size
x_h size	$4 \times x$ size
Batch size	5000
SNR for training	26dB
Gradient descent method	Adam optimizer
Learning rate	0.01
Decay factor	0.97
Decay step	300
Resnet parameter α	0.93
$\hat{x}_0$, v_0	0
weights, bias terms	start at random

B. *Simulation results*

In order to quantify the performance of our deep learning decoder we analyse the BER and the symbol error rate (SER) response with respect to variations of the SNR.

1) Gaussian noise: First we analyse DLD with the Gaussian noise model. Fig. 6 and Fig. 7 show the BER vs. SNR and SER vs. SNR respectively. There we see that the DLD performance is very close to the soft sphere decoder, when the BER and SER are low ($< 10^{-1}$) the performance is pretty much the same as the SSD decoder, and when the BER and SER are high ($> 10^{-1}$) the SNR difference is smaller than 0.5dB.

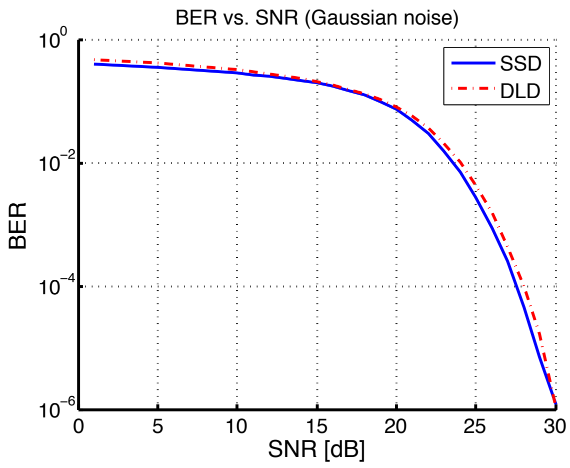

Fig. 6. BER performance of a 4×4 MIMO system, 16 QAM, Gaussian noise

2) Impulsive noise: For the DLD with the Impulsive noise model. Fig. 8 shows the BER vs. SNR curve. In this case the DLD performance is very close to the SSD decoder for all the SNR values (difference < 0.1 dB). Notice that it is harder to detect the received signal in the case of impulsive noise. Similar

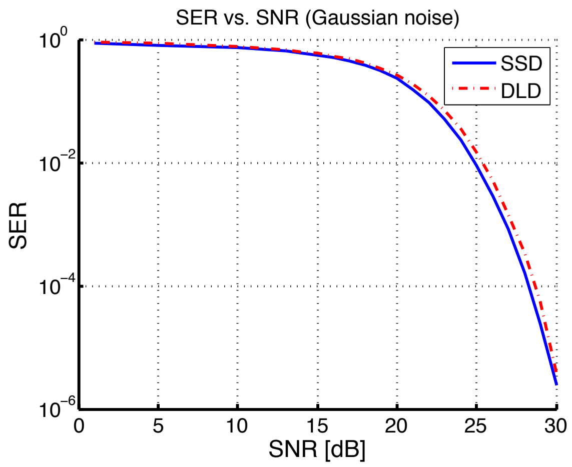

Fig. 7. SER performance of a 4×4 MIMO system, 16 QAM, Gaussian noise.

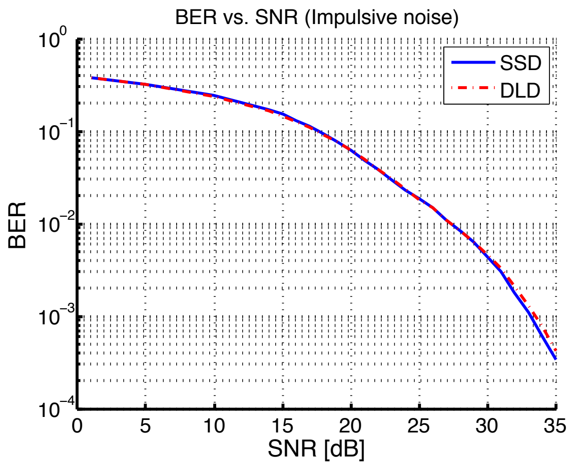

Fig. 8. BER performance of a 4×4 MIMO system, 16 QAM, Impulsive noise

results can be observed in Fig. 9 for the case of SER vs. SNR. Since the performance in terms of BER is very similar to the soft sphere decoder in the presence of Gaussian and impulsive noise, we focus our attention on another important performance metric which is the computational complexity.

C. *Complexity cost*

We should first note that the complexity of the sphere decoder depends on the SNR, so we use the expected number of operations in order to compare it with DLD. According to [3] the total number of multiplications and additions of the sphere decoder is 5.5×10^6. Considering the network parameters used in our implementation and summarized in Table II the total number of operations of DLD is 24% lower than the sphere decoder (see Table III). It is worth noticing than further gains can be obtained due to the fact that DLD has constant complexity for all SNR, allowing the possibility to have more efficient implementations.

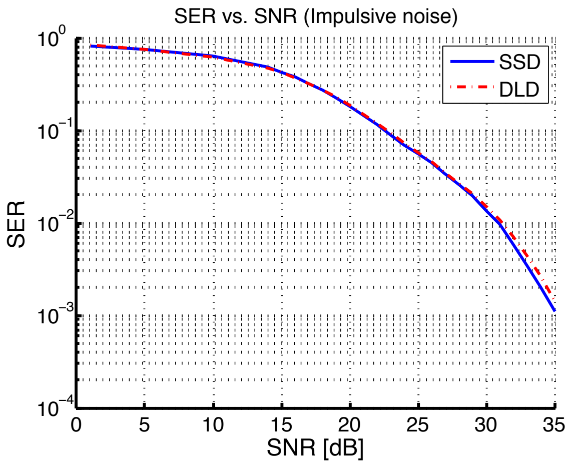

Fig. 9. SER performance of a 4×4 MIMO system, 16 QAM, Impulsive noise.

TABLE III
COMPLEXITY BREAKDOWN

Block	Multiplications	Summations
$H^T y$	64	56
$\hat{x}_k$	64	56
ReLu	19584	19584
$\hat{x}_{k+1}$	4608	4320
$\hat{v}_{k+1}$	18432	17280
Operations per layer	42752	41296
Total	2137600	2064800

VI. Conclusion

In this paper we have investigated symbol detectors in MIMO systems with impulsive noise channels. Using deep learning a new detector (DLD) is proposed. DLD detects signals after an off-line training phase. DLD exhibits good performance, and lower computational complexity than the sphere decoder. One of the important results is that the DLD computational complexity is constant across the SNR; in contrast to the SD detectors which have exponential complexity. We argue that this constant complexity could be very helpful in practical implementations because better resource optimizations are possible. To evaluate the performance of DLD, we have used a low level simulator that generates a very accurate model of the wireless communication network. The simulation results show that we can design very accurate detectors that demonstrate < 0.1dB difference with respect to the SD decoder; moreover the complexity analysis reveals that our proposed DLD detector requires 24% less arithmetic operations than the average number of arithmetic operations of the SD decoder.

References

[1] F. Sacuto, "Modeling of the impulsive noise in the power substation environment and its application to receiver design," Ph.D. dissertation, McGill University, Montreal, QC, Canada, 2015.

[2] L. Azzam and E. Ayanoglu, "Reduced Complexity Sphere Decoding for Square QAM via a New Lattice Representation," *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 4242–4246, November 2007.

[3] R. E. Chall, F. Nouvel, M. Hélard, and M. Liu, "Performance and Complexity Evaluation of Iterative Receiver for Coded MIMO-OFDM Systems," *Mobile Information Systems*, vol. 2016, p. 22, December 2015.

[4] M. El-Khamy, M. Medra, and H. M. ElKamchouchi, "Reduced Complexity List Sphere Decoding for MIMO Systems," *Digital Signal Processing*, vol. 25, no. C, pp. 84–92, February 2014.

[5] J. L. Berral-García, "When and How to Apply Statistics, Machine Learning and Deep Learning Techniques," *20th International Conference on Transparent Optical Networks (ICTON)*, pp. 1–4, July 2018.

[6] D. Cote, "Using machine learning in communication networks [Invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 10, pp. D100–D109, October 2018.

[7] S. Liang and R. Srikant, "Why Deep Neural Networks for Function Approximation?" *5th International Conference on Learning Representations (ICLR)*, March 2017.

[8] C. Zhang, P. Patras, and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey," *IEEE Communications Surveys Tutorials*, March 2019.

[9] D. Gündüz, P. de Kerret, N. D. Sidiropoulos, D. Gesbert, C. Murthy, and M. van der Schaar, "Machine Learning in the Air," *Computing Research Repository (CoRR) Cornell University*, April 2019.

[10] T. Diamandis, "Survey on Deep Learning Techniques for Wireless Communications," *EE 359 Final Project Stanford university*, 2017.

[11] T. Wang, C. Wen, H. Wang, F. Gao, T. Jiang, and S. Jin, "Deep learning for wireless physical layer: Opportunities and challenges," *China Communications*, vol. 14, no. 11, pp. 92–111, November 2017.

[12] X. Yan, F. Long, J. Wang, N. Fu, W. Ou, and B. Liu, "Signal detection of MIMO-OFDM system based on auto encoder and extreme learning machine," *International Joint Conference on Neural Networks (IJCNN)*, pp. 1602–1606, May 2017.

[13] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Physical layer deep learning of encodings for the MIMO fading channel," *55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 76–80, October 2017.

[14] T. J. O'Shea and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, December 2017.

[15] T. J. O'Shea, T. Roy, N. West, and B. C. Hilburn, "Demonstrating Deep Learning Based Communications Systems Over the Air In Practice," *IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–2, October 2018.

[16] S. Dörner, S. Cammerer, J. Hoydis, and S. t. Brink, "Deep Learning Based Communication Over the Air," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 132–143, February 2018.

[17] N. Farsad and A. Goldsmith, "Neural Network Detection of Data Sequences in Communication Systems," *IEEE Transactions on Signal Processing*, vol. 66, no. 21, pp. 5663–5678, November 2018.

[18] H. Ye, G. Y. Li, and B. Juang, "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, February 2018.

[19] M. Baek, S. Kwak, J. Jung, H. M. Kim, and D. Choi, "Implementation Methodologies of Deep Learning-Based Signal Detection for Conventional MIMO Transmitters," *IEEE Transactions on Broadcasting*, pp. 1–7, January 2019.

[20] V. Corlay, J. J. Boutros, P. Ciblat, and L. Brunel, "Multilevel MIMO Detection with Deep Learning," *52nd Asilomar Conference on Signals, Systems, and Computers*, pp. 1805–1809, October 2018.

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

[22] ——, "Deep MIMO detection," *IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, July 2017.

[23] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, June 2015.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, June 2016.

[25] M. Rupp, S. Schwarz, and M. Taranetz, *The Vienna LTE-Advanced Simulators: Up and Downlink, Link and System Level Simulation*, 1st ed. Springer Publishing Company, Incorporated, 2016.

[26] C. Studer, M. Wenk, A. Burg, and H. Bolcskei, "Soft-Output Sphere Decoding: Performance and Implementation Aspects," *2006 Fortieth Asilomar Conference on Signals, Systems and Computers*, pp. 2071–2076, Oct 2006.