



Sparsely Connected Neural Network for Massive MIMO Detection

Guili Gao, Chao Dong and Kai Niu

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

July 24, 2018

Sparsely Connected Neural Network for Massive MIMO Detection

Guili Gao^{1,2}, Chao Dong^{1,2}, Kai Niu¹

¹Key Laboratory of Universal Wireless Communications, Ministry of Education
Beijing University of Posts and Telecommunications

²Science and Technology on Communication Networks Laboratory
Email: {2016140061, dongchao, niukai}@bupt.edu.cn

Abstract—Deeping learning can achieve high parallelism and robustness, which is especially suitable for massive multiple-input multiple-output (MIMO) detection. There are already some well-developed deep learning models applied to MIMO detection, in which detection network is a typical representative model with excellent performance, but its complexity is high. This paper aims to simplify the detection network model, and the simplification runs through the entire data processing. This simplification includes three improvements. First, the number of inputs is reduced to simplify inputs; Second, the network connection structure is simplified by changing network from full connectivity to sparsely connectivity and reducing the number of network layers by half. Third, the loss function optimizes to avoid irreversible problems with the matrix. Base on the above improvements, the complexity of the network is reduced from $\mathcal{O}(64n^2)$ to $\mathcal{O}(3n)$. The simulation results indicates that the proposed structure has better performance than the existing detection network.

I. INTRODUCTION

Multiple-input multiple-output (MIMO) technology can improve spectrum efficiency and has been applied in many wireless communication standards, such as WiMax and LTE [1] [2]. Basically, the more antennas the transmitter/receiver is equipped with, the more possible signal paths and the better the performance in terms of data rate and link reliability. In future 5G development, massive MIMO is considered as a key technique with number of transmission and receiving antennas. It is getting great attention for need of high communication data rate, however, most of the MIMO used today is 4×4 or 8×8 . One of the reasons is that with the number of antennas increasing, the complexity becomes large, which is one of the key factors that restrict the number of antennas. So the key issue in using the massive MIMO is to reduce the detection complexity. The optimal detection scheme is the maximum likelihood (ML) detection, but it has the highest computational complexity. To reduce computational complexity, the linear detector is proposed, such as the minimum mean squared error (MMSE) and zero-forcing (ZF) [7] detectors. But the performance of linear detection is poor. There are other suboptimal algorithms, including approximate message passing (AMP) [8], semidefinite relaxation (SDR) [9], [10] and fixed-complexity sphere decoder (FSD) [4].

There are many simplifying algorithms, but as the number of antennas increases, their complexity becomes intolerable and performance deteriorates.

In the past few years, machine learning has achieved a great success in many fields. There are many models in the field of machine learning, such as Support Vector Machine (SVM), XGBoost [5], Decision Tree and Neural Networks. The rapidest development in recent years is the deep learning, especially in image processing and artificial intelligence. Deep learning is a multi-layer neural network constructed by complex connections, and the network structure is adjusted according to different application scenarios. The use of neural networks consists of two phases, training phase and application phase. During the training phase, pre-marked data are inputted into the network to adjust the network connection weights. The most commonly used network weight adjustment algorithm is the gradient descent method. But with the increase of network layer, the training time will increase, and the gradient will radiate or disappear [6]. Residual neural network (ResNet) [11] can increase the depth of the network, speed up convergence, improve the performance of the network, and avoid problems such as the radiation or the disappearing caused by too much network layers. ResNet is referenced in the network structure of this paper.

Detection network (DetNet [3]) is a multilayer deep neural network for massive MIMO detection. The performance of DetNet is much better than that of MMSE and ZF, especially when the number of antennas increases, it can approach the performance of AMP algorithm. The process of DetNet contains two phases, the training phase and the detection phase. During training, base on the number of antennas and receiving antennas, and the number of nodes in the network is determined; Each batch of training data run though different fast fading channels and different signal-to-noise ratio (SNR). The receiver will input the receiving signals into the network for training. The convergence of training parameters is guaranteed by backward propagation algorithm. In detection phase, cause the network has been trained, it can be applied to different fast fading channels with

different SNRs. The followings are the advantages and disadvantages of DetNet.

- Advantage:
 - 1) The performance of DetNet is similar to the performance of suboptimal algorithm, and with the increase of the number of antennas, the performance is better.
 - 2) DetNet has good robustness, once trained, they can adapt to different SNRs and different channels.
 - 3) The structure of DetNet can be processed in parallel, especially when the current computing chip is providing better support for the parallel computing.
- Disadvantages:
 - 1) When the number of antennas is small, the performance is worse than linear detection.
 - 2) DetNet requires that the transmitter has fewer sending antennas than the receiving end, and if the number of sending antennas is close to or larger than the number of receiving antennas, the performance will be poor.

In this paper, we define the channel matrix as \mathbf{H} , the transmit vector as \mathbf{x} , and the receive vector as \mathbf{y} , Boldface uppercase letters denote matrices, Boldface lowercase letters denote vectors, the superscript $(\cdot)^T$ denotes the transpose.

II. SYSTEM MODEL

In this section, we first introduce the traditional MIMO detection algorithm, then explain the design idea of the DetNet, and finally introduce the parameters of the DetNet network in detail.

A. MIMO Detection

For a MIMO system, we consider an end-to-end communication system which contains n transmit antennas, m receiving antennas, where $n < m$. The communication model can be described as follows:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w} \quad (1)$$

Where \mathbf{y} is a real vector of $m \times 1$ dimensions, \mathbf{x} is a real vector of $n \times 1$ dimensions, \mathbf{w} is a real vector of $m \times 1$ dimensions, representing the additive white gaussian noise (AWGN) of with independent and identically distributed (i.i.d.), each with zero-mean and variance δ^2 , \mathbf{H} is $m \times n$ matrix, which represents the channel state information (CSI) that is supposed known perfectly on this model.

The goal of MIMO detection is to detect the transmission signals according to the signals received by the receiving antennas. The best algorithm is ML detection. According to ML, all possible transmission signals are sent over the known channel, and the detection result is

the estimate of the transmitted signals which is nearest to sending signals based on Euclidean distance. Since ML searches all the possible signals, the complexity of ML detection increases exponentially as the number of antennas and the modulation order increase. That is the reason why the ML detection is rarely used in practical MIMO detection. The formula (2) is the basic model of ML detection.

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \{\pm 1\}^K} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \quad (2)$$

Although ML detection is hard to be realized in the project, it has an enlightening effect on other detection algorithms. Many algorithms are derived from ML, DetNet is one of them.

B. DetNet

DetNet was proposed in "Deep MIMO detection" [3], it is a multi-layer neural network dedicated to MIMO detection, the overall structure of the network is as Fig.1.

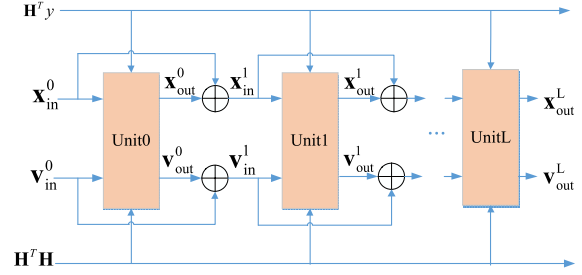


Fig. 1. DetNet Formwork

DetNet is cascaded through multiple units with same structure. There are four inputs in each unit: $\mathbf{H}^T \mathbf{y}$, $\mathbf{H}^T \mathbf{H}$, \mathbf{x}_{in}^l and \mathbf{v}_{in}^l , where $\mathbf{H}^T \mathbf{y}$ and $\mathbf{H}^T \mathbf{H}$ are the common inputs, \mathbf{x}_{in}^l and \mathbf{v}_{in}^l are changes with unit index, l represents the unit index. The residual structure is applied to increase the number of layer, the structure is as the formula (3). The input to the unit l is obtained by weighted averaging of the input of unit $l - 1$ and output of unit $l - 1$.

$$\begin{aligned} \mathbf{x}_{in}^l &= \mu \mathbf{x}_{out}^{l-1} + (1 - \mu) \mathbf{x}_{in}^{l-1} \\ \mathbf{v}_{in}^l &= \mu \mathbf{v}_{out}^{l-1} + (1 - \mu) \mathbf{v}_{in}^{l-1} \end{aligned} \quad (3)$$

μ is a residual coefficient. DetNet is an iterative network, the output of each unit can be used as the output of the whole network, and as the increasing number of network units, the output of each unit becomes closer to transmission signals based on Euclidean distance. For better performance, we should make the network as deep as possible. The design idea of the network comes from the formula (4).

$$\begin{aligned} \mathbf{x}_{in}^{l+1} &= \prod \left[\mathbf{x}_{in}^l - \lambda_l \frac{\partial \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{x}_{in}^l} \right] \\ &= \prod \left[\mathbf{x}_{in}^l - 2\lambda_l \mathbf{H}^T \mathbf{y} + 2\lambda_l \mathbf{H}^T \mathbf{H} \mathbf{x}_{in}^l \right] \end{aligned} \quad (4)$$

\mathbf{x}_{in}^l is the estimated signal of unit $l-1$ and λ_l is the stepping parameter. \prod is an iterative structure. \mathbf{x} init with a random vector, as the structure unit deepens, the vector becomes closer to the ideal vector. After one iteration, the structure performance will improve by the gradient descent algorithm and the backward propagation algorithm of the neural network. The output \mathbf{x}_{out}^l of each unit is gradually approaching the sending signal \mathbf{x} . It is shown in formula (4) that the performance of the network is just related to $\mathbf{H}\mathbf{y}$ and $\mathbf{H}^T \mathbf{H}\mathbf{x}$, so DetNet uses them as inputs of the network. DetNet also adds an input vector \mathbf{v} to each unit to expand the input dimension, which is similar to the offset of the input vector. The parameters of each unit are listed in Fig.2 [3].

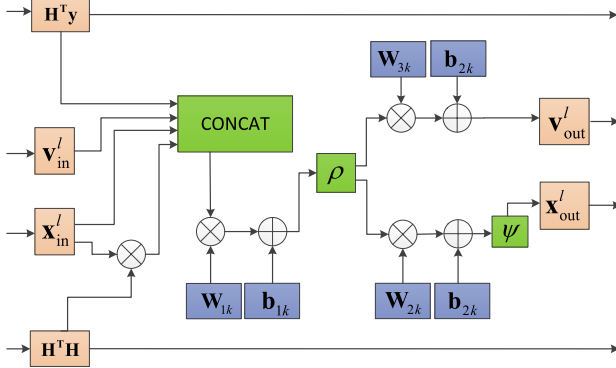


Fig. 2. Network Structure of Each Unit

The dimension of $\mathbf{H}^T \mathbf{y}$ is $n \times 1$, the dimension of \mathbf{v} is $2n \times 1$, the dimension of \mathbf{x} is $n \times 1$, the dimension of $\mathbf{H}^T \mathbf{H}$ is $n \times n$. CONCAT is used to connect all the input vectors and transform them into a one-dimensional vector, the dimension of the output vector from CONCAT is $5n \times 1$. Then the output is transmitted through a full-connected network with a large number of nodes to map the output to a higher dimension. ρ is sigmoid function as activation function. Because each unit needs to be iterated, the final output of each unit should have the same dimension as the input \mathbf{x} and \mathbf{v} . So the output of ρ is as the input to a layer of network for being compressed to \mathbf{v} and \mathbf{x} dimensions. Since the final $\mathbf{x} \in \{\pm 1\}^K$, the activation function used is similar to function \tanh whose range is $(-1, 1)$.

The loss function of the network is as follows:

$$Loss = \sum_{l=1}^L \log(l) \frac{\|\mathbf{x} - \hat{\mathbf{x}}_l\|^2}{\|\mathbf{x} - \tilde{\mathbf{x}}\|^2} \quad (5)$$

where:

$$\tilde{\mathbf{x}} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y}$$

L is the total number of units. $\hat{\mathbf{x}}_l$ is the unit l 's estimate of the transmitted vector.

III. IMPROVED DETNET

Although DetNet is a good-performance MIMO detection neural network model, we still find there is room for improvements. In this section, we simplify the network. The simplified network is a sparsely connected neural network called ScNet.

We take 2×2 MIMO structure as an example, and expand the network in Fig.2 into the form of nodes, as in Fig.3. Each node in Fig.3 represents one of the elements of vector.

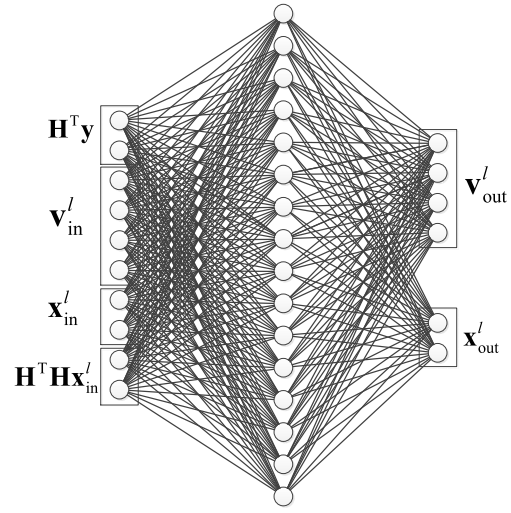


Fig. 3. DetNet Network Connection

Our improvement includes three aspects, next we will elaborate on these three aspects in detail.

A. Input Simplification

In Fig.3, although there are two outputs: \mathbf{x} and \mathbf{v} , only one output is used as the approximation of the transmitted signal \mathbf{x} . The other output \mathbf{v} does not carry any information, just as the input/output filling, its role is similar to the role of network bias. By adding \mathbf{v} , each unit increases a large number of connections and the complexity of the network. For the entire network, \mathbf{v} has no physical meaning in the field of communications, and the removal of \mathbf{v} simplifies the network structure remarkably. With \mathbf{v} , the number of edges per unit is $8n \times 8n$, and without \mathbf{v} , the number of edges is $4n \times 8n$. The total number of connections is reduced by half and the training parameters are reduced by half. After removing \mathbf{v} , the network is shown in Fig.4.

It tested the network after removing \mathbf{v} and find that the performance has small gain, and the training time is reduced.

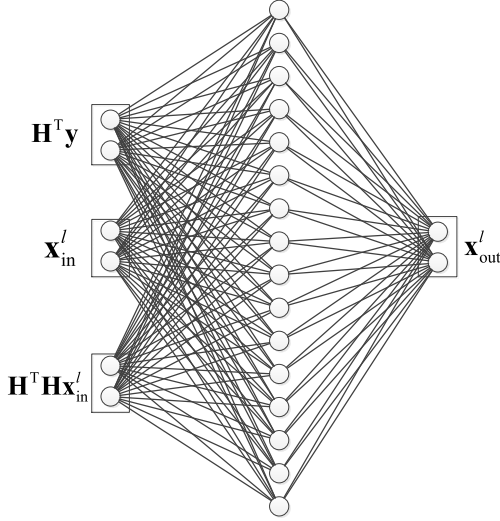


Fig. 4. DetNet Network Remove \mathbf{v}

B. The Simplification of the Network Connection

After removing \mathbf{v} , the network units are still fully connected structure. For each input node, it interacts with other nodes, but from formula (4), the iterative structure describes linear operation of vectors. As shown in Fig.5, only the same indexed elements are added or subtracted in linear operation. Inspired by this mathematical principle, we connect the same indexed vector in the network to the output. The connection relationship is as Fig.6.

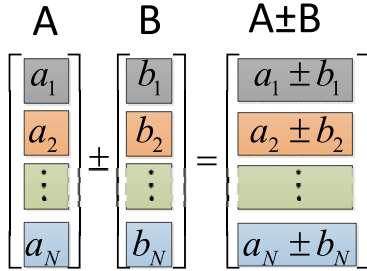


Fig. 5. Vector Plus Vector

It is a sparsely connected neural network(ScNet). In ScNet, the first node of each input is only connected with the first node of output, the second node only connected with the second node of output. Regard node $\mathbf{x}_{out}^l[i]$ of Fig.6 as a medium of information exchange between $\mathbf{H}^T \mathbf{y}[i]$, $\mathbf{x}_{in}^l[i]$ and $\mathbf{H}^T \mathbf{H} \mathbf{x}_{in}^l[i]$. i denotes the index.

Before simplification, the number of edges of each unit is $8n \times 8n$. After removing \mathbf{v} and simplifying the network connection, the number of connected edges of the network is only $3n$.

C. The Simplification of the Loss Function

The loss function for ScNet is as formula (6):

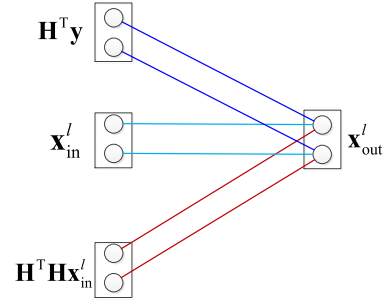


Fig. 6. neural network connection for ScNet

$$Loss = \sum_{l=1}^L \log(l) \|\mathbf{x} - \hat{\mathbf{x}}_l\|^2 \quad (6)$$

Our loss function removes $\|\mathbf{x} - (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y}\|^2$ compared to DetNet's loss function. Actually the removal formula is equivalent to $\|\mathbf{n}\|^2$. Our goal is to make Euclidean distance between estimations Euclidean distance, that isn't related to $\|\mathbf{x} - (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y}\|^2$, therefore, we remove the formula. Another reason that we remove this formula is: this formula contains matrix inversion operation, in many cases, the square matrix is not reversible and matrix inversion is a very complicated calculation. After our tests we find that after this formula is removed, the performance is improved slightly.

In our simulation, after the output of the last layer passes through the Ψ activation function, the range will be $\mathbf{y} \in (-1, 1)$, we judge the results by (7).

$$\mathbf{y}_{out} = \begin{cases} 1 & \mathbf{y}_{out}^N \geq 0 \\ 0 & \mathbf{y}_{out}^N < 0 \end{cases} \quad (7)$$

In this section, inspired by DetNet, we propose a simplified deep learning model for MIMO detection called ScNet. In the following sections, we will compare the performance of these two networks.

IV. SIMULATION RESULTS

In this section, we compared the performance of DetNet and ScNet, the simulation conditions are as follows: all simulation channels is given fast fading channels, and the SNR of each simulation is randomly chosen in the range of [7, 14]. At the beginning, the input \mathbf{x} and \mathbf{v} are into zero vectors. In DetNet, the dimension of \mathbf{v} is $2 \times n$, the extended dimension from inputs is $8 \times n$ and the learning rate is 0.0001. The layer number of the two networks is 90 and the residual coefficient of the residual network choose 0.9. During training, we send $5000 \times n$ bits per antenna, which is recorded as one iteration. Table.I is the comparison of DetNet and ScNet, that is a reference to the complexity of the network.

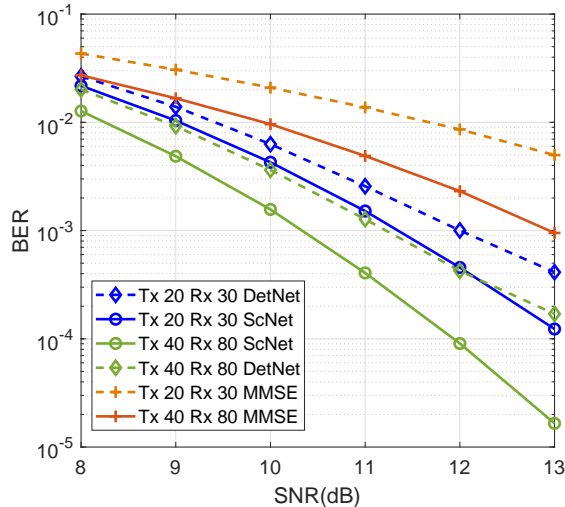


Fig. 7. Comparing the Performance of DetNet and ScNet

TABLE I
THE COMPARISON OF NETWORK UNIT EDGE NUMBER

antennas	network	unit edges
Tx 20 Rx 30	DetNet	25,600
	ScNet	60
Tx 40 Rx 80	DetNet	102,400
	ScNet	120

Fig.7 is a performance diagram comparing the two nets, it contains simulation results for two antenna configurations, Tx 20, Rx 30 and Tx 40, Rx 80. It can be seen that with Tx 40, Rx 80, the performance gain of ScNet over DetNet is about 1dB at 10^{-4} . Regardless of the antenna configuration, the performance of ScNet is slightly improved compared with DetNet, and far exceeds the performance of MMSE. Compared with the two antenna configurations of the same network, it can be seen that when the number of antennas increases, the performance gains. This shows that the ScNet with deep learning is more suitable for scenarios with large scale of antennas.

Fig.8 shows the convergence speeds of the two network trainings. The random SNR is used to test when these two networks are trained, so the BER after convergence does not reach the minimum value, and the network fluctuation is also slightly larger. It can be seen that, however, the simplification of the network has not affected the convergence performance of the network.

V. CONCLUSIONS AND FUTURE WORK

In this paper, an improved deep learning model is proposed for detection in large-scale MIMOs, based on the analysis of DetNet model. Numerical analysis indicates that ScNet not only simplifies the complexity, but also improves the performance, especially when

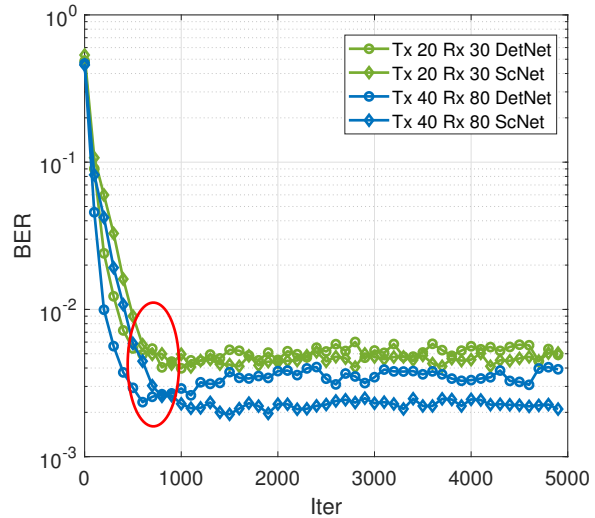


Fig. 8. Network Convergence Speed

sending and receiving end equipped with large scale antenna.

VI. ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (No. 61601047, 61671080, 61771066), and Science and Technology on Communication Networks Laboratory Open Project (KX172600028).

REFERENCES

- [1] E. Dahlman, S. Parkvall, J. Skold, P. Beming, "3G Evolution HSPA and LTE for Mobile Broadband". Academic, 2008.
- [2] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, "Scaling up mimo: Opportunities and challenges with very large arrays", *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 40 C60, 2013.
- [3] N. Samuel, T. Diskin and A. Wiesel, "Deep MIMO detection" *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Sapporo, 2017, pp. 1-5. doi: 10.1109/SPAWC.2017.8227772
- [4] C. Xiong, X. Zhang, K. Wu and D. Yang, "A simplified fixed-complexity sphere decoder for V-BLAST systems", in *IEEE Communications Letters*, vol. 13, no. 8, pp. 582-584, August 2009.
- [5] T. Chen and C. Guestrin. "Xgboost: A scalable tree boosting system". CoRR, abs/1603.02754, 2016.
- [6] S. Ioffe and C. Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". arXiv:1502.03167, 2015
- [7] S. Yang and L. Hanzo, "Fifty years of mimo detection: The road to large-scale mimos", *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 1941 1988, 2015.
- [8] S. Wu, L. Kuang, Z. Ni, J. Lu, D. Huang and Q. Guo, "Low-Complexity Iterative Detection for Large-Scale Multiuser MIMO-OFDM Systems Using Approximate Message Passing", in *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 902-915, Oct. 2014.
- [9] Z. Ma, M. Zhao, Q. Chen and P. Fan, "MIMO detection for high-order QAM constellations based on successive decision feedback semidefinite relaxation," *Proceedings of the Fifth International Workshop on Signal Design and Its Applications in Communications*, Guilin, 2011, pp. 173-176.

- [10] J. Jalden and B. Ottersten, "The diversity order of the semidefinite relaxation detector", *IEEE Transactions on Information Theory*, vol. 54, no. 4, pp. 1406-1422, 2008.
- [11] K. He, X. Zhang, S. Ren, J. Sun "Deep Residual Learning for Image Recognition", in *Computer Vision and Pattern Recognition IEEE* pp. 770-778 2016.