Low-complexity Neural Network-based MIMO Detector using Permuted Diagonal Matrix

Siyu Liao[1], Chunhua Deng[1], Lingjia Liu[2], Bo Yuan[1]
[1]Department of Electrical & Computer Engineering, Rutgers University
[2]Department of Electrical & Computer Engineering, Virginia Tech
siyu.liao@rutgers.edu, chunhua.deng@rutgers.edu
ljliu@vt.edu, bo.yuan@soe.rutgers.edu

1 Abstract

DNN has achieved state-of-the-art performance in MIMO detection problem. However, the deep and large model is hard to deploy to resource constrained platforms. In this work, we propose to provide a sparse DNN model using permuted diagonal matrices. As a result, our model is with low complexity and doesn't have indexing overhead like other sparsity method. Experiment shows that our method can achieve high sparsity while maintaining the model performance.

2 Introduction

The demand of transferring big data has been increasing for many years. As one of the solutions, the multiple-input multiple-output (MIMO) technology is to increases the spatial bandwidth by adding more antennas [1]. For this technology, a difficult challenge is the signal detection problem.

On the other hand, deep neural network (DNN) has gradually become a powerful model which learns to address the problem where underlying data characteristics are unknown. It is found that DNN is also applicable to MIMO detection and has achieved state-of-the-art performance [2].

Although DNN has extraordinary performance in many applications, their deep architecture comes with millions of parameters. It is a big challenge to deploy the model into resource constrained platforms. A popular solution is to generate sparse DNN models with low complexity. But such sparsity is heuristic and irregular, incurring indexing overhead for nonzero model parameters [3].

In this work, we propose to use permuted diagonal matrices to generate regularly sparse DNN model for MIMO detection problem. This method results in regular sparsity so that indexing overhead can be avoided and the sparsity can be controlled by configuring the model hyper-parameters.

3 MIMO Detection

Let $\mathbf{H} \in \mathbb{C}^{N \times K}$ be the channel matrix and $\mathbf{y} \in \mathbb{C}^N$ be the received vector. We have $\mathbf{X}$ as input symbols and $\mathbf{b}$ as noise vector in Gaussian distribution with zero mean and variance σ^2. A linear MIMO model is formulated as below:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{b}. \tag{1}$$

The goal is to detect $\mathbf{x}$ given $\mathbf{H}$ and $\mathbf{y}$. The DNN based solution is to build a network that learns model parameters $\mathbf{W}$ to give correct detection. The learning process is through minimizing the distance between detection result and ground truth $\mathbf{x}$ over given training data.

More specifically, DNN learns a function $f(\mathbf{W})$ by stacking multiple non-linear layers which is formulated as:

$$\mathbf{a} = g(\mathbf{W}\mathbf{x}), \tag{2}$$

where $g(\cdot)$ is an non-linear activation function such as hyperbolic tangent and $\mathbf{W}$ is the weight parameter for this layer. Here bias parameters are included in $\mathbf{W}$. Stacking means that output of a layer can be the input to the next layer. Theoretically, stacking multiple layers allows DNN to approximate any continuous function [4].

4 Proposed Method

In this section, we introduce the permuted diagonal matrix for DNN based MIMO detection. Let $\mathbf{W} \in \mathbb{R}^{m \times n}$ be the weight matrix which can be divided into small blocks in shape $k \times k$. In total, there will be $m/k \times n/k$ blocks. Let each block $\mathbf{I}(l)$ be a matrix generated by permuting a diagonal matrix and the permutation is shifting from the main diagonal to the l-th diagonal. More specifically, the representation of $\mathbf{W}$ is formulated as following:

$$\mathbf{W} = \begin{bmatrix} \mathbf{I}(l_{1,1}) & \dots & \mathbf{I}(l_{1,n/k}) \\ \vdots & \ddots & \vdots \\ \mathbf{I}(l_{m/k,1}) & \dots & \mathbf{I}(l_{m/k,n/k}) \end{bmatrix}, \tag{3}$$

where $i = 1, \dots, m/k$ and $j = 1, \dots, n/k$. Each $\mathbf{I}(l_{i,j})$ is a matrix defined as:

$$\mathbf{I}(l_{i,j}) = \begin{pmatrix} 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \dots & 0 & 1 & 0 \end{pmatrix}^{l_{i,j}} \times diag([w_1^{l_{i,j}}, \dots, w_k^{l_{i,j}}]), \tag{4}$$

and $diag(\cdot)$ function outputs a diagonal matrix for given vector. Those values of $w_p^{l_{i,j}}$ are weight parameters inside the sub-matrix for $p = 1, \dots, k$. Therefore, the density of the block permuted diagonal matrix $\mathbf{W}$ is $\frac{1}{k}$. The sparsity $(1 - \frac{1}{k})$ is controlled via the setting of value k. With larger k, the matrix is more sparse.

5 Experiment

Experiment is performed to explore the trade off between sparsity and bit error rate (BER). In this work, we take the state-of-the-art DNN model DetNet [2] and use the same configurations. Instead of using general dense matrices, we apply our permuted diagonal matrices to the DNN model. Experiment is conducted with Tensorflow [5] and Tesla V100 GPU.

We set different block size to achieve different sparsity and measure the corresponding BER. More specifically, the block size setting is applied to all weight matrices inside the model. For example, when $k = 4$, all weight matrices are with 75% sparsity.

As shown in figure 1, we set block size as $k = 2, 4, 8, 16$ and measure corresponding bit error rate for BPSK. The baseline model is the original DetNet with dense matrices. It can be seen that with large block size, although the model gets more sparse, the bit error error will increase for all different SNR. For $k = 2, 4, 8$, BER remains almost the same but decreases a lot when k gets to 16.

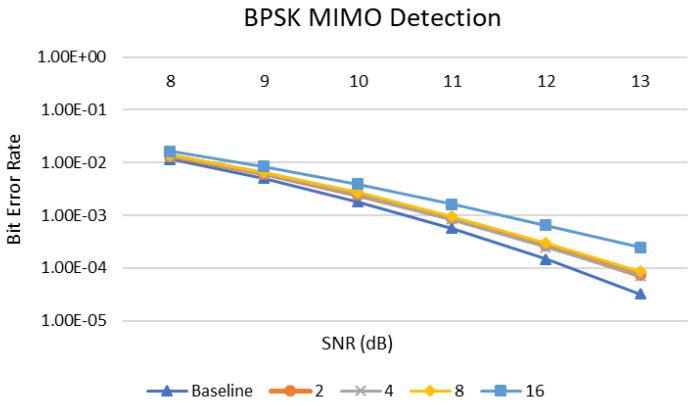

Figure 1: Different Block Size Settings for BPSK MIMO Detection.

6 Conclusion

In this work, we propose to address the problem of deployment of DNN models for MIMO detection by using permuted diagonal matrices. This method generates sparse DNN model but without indexing overhead. Experimental results show that this method can output DNN model with low complexity while having negligible BER increase.

References

[1] Mirsad Čirkić. *Efficient MIMO Detection Methods*. PhD thesis, Linköping University Electronic Press, 2014.

[2] Neev Samuel, Ami Wiesel, and Tzvi Diskin. Learning to detect. *IEEE Transactions on Signal Processing*, 2019.

[3] Chunhua Deng, Siyu Liao, Yi Xie, Keshab K Parhi, Xuehai Qian, and Bo Yuan. Permdnn: Efficient compressed dnn architecture with permuted diagonal matrices. In *2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 189–202. IEEE, 2018.

[4] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[5] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016.