Signal detection with co-channel interference using deep learning

Chenguang Liu\textsuperscript{a,b}, Yunfei Chen\textsuperscript{a}, Shuang-Hua Yang\textsuperscript{b,*}

\textsuperscript{a}School of Engineering, University of Warwick, Coventry, CV4 7AL, UK
\textsuperscript{b}Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China

Abstract

Signal detection using deep learning is a challenging and promising research topic. Several learning-based signal detectors have been proposed to produce significant results. However, most of them have ignored interference in their designs. In this paper, we evaluate the performance of learning-based signal detectors in the presence of co-channel interference under different channel conditions. Specifically, fully connected deep neural network (FCDNN) and convolutional neural network (CNN) are examined as the data-driven signal detector for blind signal detection without knowledge of the channel state information. Several important system parameters, including signal-to-interference ratio, number of interferences and type of interference, are considered. Numerical results show that FCDNN and CNN-based detectors have better performance and robustness to different SIRs conditions than traditional detectors in the presence of interference and FCDNN performs better than CNN when SIR is small and the order of interference modulation is high.

Keywords: Co-channel interference, deep learning, neural network, signal detection.

1. Introduction

The demand for concurrent data access has increased dramatically to satisfy the massive amount of mobile users. Multiple-input multiple-output (MIMO) technology is widely applied in communications systems to manage the limited spectrum resource and improve spectral efficiency. To maintain the quality of services and transmission efficiency, it is important to improve the ability of estimating channel state information and consequently detecting signals in communications system. Therefore, many works have been conducted on signal detection algorithms\cite{1,2}. Among them, the maximum likelihood detector can achieve the optimality by minimizing the probability of detection error. However, in many complicated scenarios, maximum likelihood detection is too complicated to implement. To cope with this complexity problem, researchers have considered suboptimal linear detectors, such as zero-forcing detector and minimum mean squared error detector, which have lower computational complexity at the cost of a worse detection performance.

\textsuperscript{*}This work was partly supported by the National Natural Science Foundation of China under Grant 61873119 and Grant 61911530247.

\textsuperscript{*}Corresponding author at: The Department of Computer Science and Engineering, Southern University of Science and Technology (SUSTech), 518055 Shenzhen, China

Email address: yangsh@sustech.edu.cn (Shuang-Hua Yang )

Preprint submitted to Journal of BLaX Templates

April 14, 2021
On the other hand, due to the powerful capability of recognizing patterns from high dimensional data, deep learning has been extensively utilized in many research areas, including computer image processing [3], visual tracking [4], speech recognition [5] and natural language processing [6]. Recently, deep learning and artificial neural networks have been extended to the physical layer of communications, such as channel extrapolation [7] [8], channel estimation and signal detection [9], [10], [11].

Previous works on the use of deep learning techniques in signal detection of communications systems include the following. In [12], a fully connected neural network was designed to estimate channel state information and then detect signals in an orthogonal frequency-division multiplexing communications system. It showed that the neural networks can achieve similar performance to traditional detection algorithms. A deep neural network detector, DetNet, was proposed in [13] and [14] for MIMO detection. DetNet achieves nearly optimal performances in challenging channels, but it assumes perfect channel state information in the detection, which is not practical in real-life communications. The work in [15] proposed a parallel detection network that improved detection performance by increasing the number of deep learning detection networks in parallel. In [16], a deep learning detector based on belief propagation algorithm was proposed for the detection of MIMO signals, which showed lower complexity and better performance than the tradition belief propagation algorithm. In [17], deep learning based massive MIMO detectors were proposed by unfolding belief propagation algorithms. The model reduces the number of trained parameters to improve the convergence efficiency. In [18], a model-driven DNN detector, OAMP-Net, was proposed by incorporating deep learning into orthogonal approximate message passing algorithm. The OAMP-Net considered imperfect channel state information and achieved better performance than orthogonal approximate message passing algorithm. Then, the work in [19] took channel estimation error into consideration and proposed an OAMP-Net based joint channel estimation and signal detection scheme, which showed strong robustness to the estimation error. In [20], CNN and RNN-based signal detectors were proposed to achieve similar performances to the maximum likelihood algorithm with perfect channel state information.

The work in [21] proposed a RNN-based sequence detector, CTBRNN, which directly learns from the training data and detects the desired signal sequence without estimating or knowing the channel model. It outperforms existing NN-based signal detectors in the experimental environment. These works have used deep learning algorithms to replace or improve the traditional signal detection algorithms. On the other hand, the communication system can also be treated as an end-to-end learning-based transceiver that signal detection can be performed based on historical data without any channel estimators or signal detectors. In [11], the possibility of using the deep learning based end-to-end system was discussed to reduce or remove the complexity involved in designing traditional transmitters and receivers. Furthermore, an unsupervised learning method, autoencoders, was proposed to decode the signal from fading channels by studying the features at both transmitter and receiver. The performance of the autoencoders-based system was evaluated in the presence of channel interference in Rayleigh fading [22]. Despite this, it is challenging for the architecture of unsupervised learning to be applied to complex communications systems. The work in [23] provided general insights into end-to-end learning-based communications systems, where CNN-based learning techniques were used to solve modulation recognition and wireless interference identification problems.

All the aforementioned works have provided very useful guidance on the use of deep learning techniques to either improve existing signal detectors or replace existing signal detectors. However, most of them have ignored interference in their designs, except [22] that evaluated their design in the presence of interference. Co-channel interference is inevitable in many wireless
communications systems due to spectrum sharing and multiple access [24]. It is mainly caused by multiple radios transmitting on the same frequency band at the same time, which leads to degradation to the communications performance. In most previous works, the interference of the communication system is either not studied or simply treated as Gaussian noise. Although it was proven in [25] that weak interference can be treated as Gaussian noise to achieve the sum channel capacity, in general strong interference still requires sophisticated designs of transmitter and receiver to mitigate its influence [26]. The work in [22] considered the channel interference in an autoencoders-based SISO and MIMO system, but it did not delve into a complete study of the effects of different types of interference on signal detection.

In this paper, we will focus on learning based signal detection, inspired by the works in [12]-[21], to evaluate their detection performances in the presence of co-channel interference. The main contributions of this paper are summarized as follows:

1. Unlike most of the aforementioned works which either ignored interference or treated interference as Gaussian noise, we take co-channel interference into consideration and discuss explicitly the different effects of interference on the system performance, including different modulation types, different numbers of interferers and different power levels of interference. Fading and thermal noise in the channel are also considered.

2. Different from the work in [16]-[19] that incorporated deep learning methods into traditional signal detection algorithms, we directly apply deep neural networks as signal detector without the need for channel estimation, which simplifies the signal receiver by learning the channel characteristics in the training phase.

3. We implement full connected neural networks with adjusted parameters and convolutional neural networks as the learning-based signal detector and the system performance is studied. Based on the numerical results, CNN and FCDNN detectors outperform zero-forcing detectors with least-squares channel estimation in the presence of interference when the signal-to-interference ratio (SIR) is 0 dB or 20 dB for QPSK signals. Also, FCDNN outperforms CNN in low SIR conditions. In addition, the learning-based detectors show robustness to the different SIR conditions and it can be applied to detect higher-order modulated signals by conducting experiments on 8PSK modulated signals.

The rest of the paper is organized as follows. In Section 2, we will introduce the wireless communication system model with co-channel interference, the deep learning methods to be used and the learning based signal detection. In Section 3 we will discuss the simulation setup for deep learning based signal detection, data generation and model implementation. Simulation results will be presented in Section 4. Finally, Section 5 will conclude the work.

2. System model

Consider a communications system whose radio propagation channel suffers from co-channel interference, noise and fading. We first introduce the two deep learning algorithms used, convolutional neural network (CNN) and fully connected deep neural network (FCDNN). Then, the structure of learning-based detection will be described.

2.1. Problem formulation

We consider a wireless communications system having one transmitter equipped with $N_t$ antennas and one receiver equipped with $N_r$ antennas. The received signal vector $\mathbf{y} \in \mathbb{C}^{N_r}$ can be
expressed as
\[ y = Hx + I + n, \]  
\[ n \sim N_C(0, \sigma^2 \Lambda), \]
where \( n \sim N_C(0, \sigma^2 \Lambda) \) is the complex additive white Gaussian noise with mean zero and covariance matrix \( \sigma^2 \Lambda \), \( \Lambda \) is the identity matrix, \( H \in \mathbb{C}^{N_r \times N_t} \) is the channel matrix following Gaussian distributions, \( x \in \Theta^{N_t} \) is the transmitted signal from the discrete constellation set \( \Theta = \{s_1, s_2, s_3, \ldots, s_i\} \), \( I \in \mathbb{C}^{N_r} \) denotes co-channel interference caused by other signals transmitted in the same frequency at the same time, which can be expressed by
\[ I = \sum_{i=1}^{N_t-1} M_i p_i, \]
where \( M_i \) denotes the Gaussian channel matrix of the \( i \)-th interfering user and \( p_i \) is the signal vector of the \( i \)-th interfering user.

In order to transmit the information efficiently, the desired signal \( x \) and interfering signal \( p \) are digitally modulated. For two-dimensional modulation, the modulated signal can be described by the in-phase and quadrature components as
\[ x(t) = A \cos(2\pi ft + \varphi) = \Re[(I + jQ)e^{2\pi ft}], \]
where \( A, f, \varphi \) are the amplitude, frequency and phase of the carrier signal, respectively, \( I \) is the in-phase component and \( Q \) is the quadrature component. \((I, Q)\) can also be considered as the coordinates of a point on the constellation diagram determined by varying both phase and amplitude, or one of them, to give the following modulation schemes:

1. Phase shift keying (PSK): \( M \) possible signal phases are utilized to map \( \log_2 M \) bits to \( M \) points on the constellation diagram. For example, when \( M = 4 \), the transmitter transmits sinusoids with a phase of \( \varphi = 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2} \) to represent bits 00, 01, 11 or 10 with a constant amplitude, resulting in points \((\frac{1}{\sqrt{2}}, 0), (0, \frac{1}{\sqrt{2}}), (-\frac{1}{\sqrt{2}}, 0), (0, -\frac{1}{\sqrt{2}})\) on the constellation, assuming \( A = 1 \). This gives QPSK.

2. Pulse amplitude modulation (PAM): The transmitter chooses one of the \( M \) possible signal amplitudes for sinusoids with a constant phase. For example, when \( M = 4 \), the sinusoids represent bits 00, 01, 11 or 10 with an amplitude of \( A = -3, -1, 1, 3 \) and a constant phase and frequency, resulting in points \((-3, 0), (-1, 0), (1, 0), (3, 0)\) on the constellation, assuming the minimum distance between each points is 2. This gives 4PAM.

3. Quadrature amplitude modulation (QAM): QAM combines PAM and PSK by changing both phase and amplitude and switching between \( M \) possible amplitudes and \( M \) possible phases to represent \( \log_2 (M^2) \) bits. For example, when \( M = 4 \), the sinusoids with a phase of \( \varphi = -3, -1, 1, 3 \) and an amplitude of \( A = -3, -1, 1, 3 \) to represent 4-digit bits 0000 to 1111, assuming the minimum distance between each point is 2. This gives 16QAM.

In this paper, we will use QPSK and 8PSK as modulation schemes for the desired transmitted signal, and BPSK, QPSK, 8PSK, 4PAM, 16QAM and 64QAM as modulation schemes for the interference signals. Our goal is to detect the transmitted signal \( x \) using the received signal \( y \) subject to interference \( I \), fading channel \( H \) and noise \( n \).
2.2. Deep learning methods

An example of a deep neural network structure is illustrated in Figure 1. Deep neural network is an upgrade of artificial neural network by introducing more hidden layers to enhance the convergence capacity for complex patterns. Each layer of the network is chained by employing the output of the current layer as the input of the next layer.

To implement a deep neural network, one method is fully connected neural network, which is known as the basic deep neural network architecture. It consists of a series of dense layers, and every neuron in one layer is connected with all neurons in the next layer. However, when the size of the network grows, this could lead to a large number of parameters causing overfitting. Therefore, a dropout layer can be inserted to the network to remove some random neurons to prevent overfitting[27]. On the other hand, convolutional neural network is built upon a fully connected neural network to extract features, which has been widely utilized in the computer vision area. It introduces a convolutional layer to convolve the inputs. After convolving, the size of the output decreases so that the number of parameters requiring training is reduced. Therefore, it provides a solution to reducing the complexity and improving training efficiency.

To obtain the best performance of the network, neural network is trained by inputting labelled historical datasets to seek the optimized values for the parameters of each layer, including weights and bias. Firstly, the loss between the predicted output and the actual result is calculated by forward propagation process. For multi-classification problem, categorical cross-entropy is commonly used to compute the loss. The loss function $L$ can be described by

$$L = -\frac{1}{m} \sum_{k=1}^{m} x_k \log \hat{x}_k + (1 - x_k) \log (1 - \hat{x}_k),$$

where $\hat{x}_k$ and $x_k$ denote the predicted label and the actual label, respectively. Then, the loss gradients of each layer are computed by calculating the partial derivatives of the loss function and back propagation is used to fit the network. After several iterations, we can find the optimum values of parameters to minimize the total loss, which makes the network more accurate and robust. The process of optimizing the parameters can be illustrated by the following equation

$$\theta_{m+1} = \theta_m - \alpha \frac{\partial L(x_k, f(\theta))}{\partial \theta},$$
where $\theta$ denotes the parameter in each layer, $\alpha$ denotes the learning rate of the network. The learning rate $\alpha$ determines the converging rate of gradients.

2.3. Learning based signal detection

A general structure of a learning-based signal detection system using deep neural networks can be described in Figure 2. It generally can be divided into two phases: model training and deployment. In the training phase, we collect the original transmitted signal corrupted by interference and noise, and mark the bits they represent. Then, we label these historical data as the training input for the neural work. The deep neural network is trained to detect the input signal array as the bit symbol it might represent. In the deployment phase, the simulated environment restores the actual process of signal transmission, and synthetic data is generated and transmitted in the channel to the receiver. We implement the learning-based detection at the receiver to map the sampled received modulated signals to the information bits. The learning-based detection cannot cancel or mitigate interference in the signal transmission, but a well-trained network can recognize the different patterns embedded in the signal and interference to make a highly reliable prediction based on the historical data.

Specifically, to detect the transmitted signal $x_t$, the learning based FCDNN detector can be described by

$$\hat{x} = f^L(f^{L-1}(\cdots f^1(y, W_1, b_1))))$$

(7)

$$f^m(\theta) = \beta(W_{m-1}z_{m-1} + b_{m-1})$$

(8)

$$z_1 = y$$

(9)

$$\theta = [(W_m, b_m)]_{m=1}^L$$

(10)

where $\hat{x}$ denotes the predicted output or the data decision of the transmitted signal $x$, $y$ denotes the received signal in (1) and is also the input to the detector, $f^m(\theta)$ represents the fully connected layer of the detector, $\theta$ represents the parameters of the learning based detector consisting of weights $W$ and bias $b$ to be optimized by training with the signal datasets, $\beta$ denotes the activation function, such as ReLU [28] and SoftMax [29], to introduce nonlinearity into the network for better performance.

Figure 2: The structure of learning based detection system.
For CNN detector, the convolutional layer can be described by

\[ f^m_{(i,j)} = (z_{m-1} \ast K_m)_{(i,j)} + h_m, \]  

(11)

where \( z_{m-1} \) denotes the input matrix, \( K_m \) denotes the convolutional kernel with the same dimension, \( \ast \) denotes the convolution operation with kernel \( K \). The convolution operation reduces the complexity of the input \( y \) in (9) but preserves the features for further predicting the signal \( \hat{x} \) as in (7).

The parameter \( \theta \) of each layer is trained by reducing the loss between the predicted transmitted signal \( \hat{x} \) and the actual signal \( x \). Once we find the optimized \( \theta \), the network can be considered as a function that takes the received signal \( y \) as input and the data decision of the transmitted signal \( \hat{x} \) as output. The learning-based detector does not require knowledge of the interference \( I \), the channel matrices \( H \) or the noise level \( n \), which simplifies the receiver greatly.

3. Deep learning based signal detector

In this section, we will introduce the wireless transmission system with co-channel interference, noise and varying channel status to simulate the realistic communications conditions. After describing the communications system used, we will introduce the method of generating synthetic datasets for the purpose of training and testing. Subsequently, we will discuss the structure of two learning-based signal detectors, CNN and FCDNN.

3.1. Simulated environment

The simulated wireless communications environment is implemented by using GNU radio which is an open-source toolkit for signal simulation[30]. It contains a lot of essential components for radio communications, including channel models, modulators and noise generators. To simulate a real-life wireless communications system, the implemented components are shown below.

1. Input bitstream: A text file is converted to a binary bitstream as the input of the signal transmitter.
2. Digital modulator: Binary information is modulated to I/Q sampled signal and various digital modulators are considered in the system, including BPSK, QPSK, 8PSK, 4PAM, 16QAM, 64QAM.
3. Pulse shaping: Root raised-cosine filter is used to filter the transmitted waveform of I/Q signal for transmission over the channel and to resample the waveform. We choose 16 samples per symbol as the sampling rate at the pulse shaping resampler.
4. Fading channel: Rician fading channel is applied in the system to describe a line-of-sight radio propagation with interference. The desired power delay and sample rate offset are considered to simulate the realistic fading scenario.
5. AWGN: The thermal noise at the receiver is described by additive white Gaussian noise and the variance is applied to model the noise power.
6. Transmitter: The target signal is transmitted to the receiver and QPSK and 8PSK are considered to modulate the desired signal.
7. Interferer: Multiple interferers with similar radio propagation conditions are added to the system to describe the scenario of co-channel interference. Different numbers, modulations, SIRs and delays of the interference are considered to have a comprehensive study for the comparison of the performance.
8. Receiver: The learning-based signal detector is deployed as the receiver to detect the signal.

To define the interference level of the radio system, the SIR is defined by

\[
SIR = 10 \log \frac{A_s}{A_i},
\]

(12)

where \(A_s\) denotes the power of transmitted signal, which is normalized as unit in the simulation; \(A_i\) denotes the peak power of the interference signal. The noise level of the system is determined by the signal-to-noise ratio (SNR)

\[
SNR = 10 \log \frac{A_s}{A_n},
\]

(13)

where \(A_n\) denotes the noise power which specifies the variance of the AWGN process. We consider the SNR in the range of -20 dB to 20 dB, to evaluate the performance under different noise and interference conditions.

3.2. Data generation and processing

We generate multiple sets of synthetic datasets, based on different values of several variables in the system, including SNR, SIR, number of interferences, type of interference modulation etc. The output data received is a series of sampled I/Q complex data. However, it is difficult for neural networks to learn from the complex data as the parameter weights and bias are all real values. Therefore, we transform the received complex signal to a 2-dimensional vector by separating the real and imaginary numbers as

\[
y = \left( \Re(y), \Im(y) \right)
\]

(14)

Subsequently, we normalize the data to the interval between 0 and 1 to avoid the influence of singular samples. Then, these datasets are labelled by the original bits they represent. To better aggregate the output of network and the original bit labels, we apply one-hot function, introduced in [14], to describe the possible symbol distribution by mapping the discrete signal symbols to a unit vector array to match the output of the deep learning detector.

3.3. Structure of FCDNN and CNN

The FCDNN structure is described in Table 1 consisting of 7 dense layers. The input signal data is processed by fully connected layers and ReLU activation function. In addition to dense layers, dropout layers are also inserted into the middle of hidden layers to prevent overfitting. To avoid removing the essential neurons and affect the performance, 50% dropout rate is chosen according to the number of parameters. The input data size of 2×16 is determined by 16 samples-per-symbol separating the real and imaginary number. We use SoftMax as the activation at the output layer to obtain a probability distribution over the predicted symbol classes.

The structure of CNN shown in Table 2 is similar to what was used in the work by Visual Geometry Group [31], which combined multiple convolutional layers with fully connected layers. The work in [32] modified this structure and showed good performance of classifying modulation type. In our work, the CNN structure is redesigned to fit smaller input (2 x 16) for signals. Therefore, the pooling function is not adopted as it can blur the input data due to its small size and affect the authenticity of the data. We select 1 x 3 for the kernel size of the convolution.
Table 1: FCDNN structure

<table>
<thead>
<tr>
<th>Layer</th>
<th>Parameters</th>
<th>Activation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input size</td>
<td>$2 \times 16$</td>
<td></td>
</tr>
<tr>
<td>Dense layer</td>
<td>64 neurons</td>
<td>ReLU</td>
</tr>
<tr>
<td>Dense layer</td>
<td>128 neurons</td>
<td>ReLU</td>
</tr>
<tr>
<td>Dropout</td>
<td>50%</td>
<td></td>
</tr>
<tr>
<td>Dense layer</td>
<td>512 neurons</td>
<td>ReLU</td>
</tr>
<tr>
<td>Dropout</td>
<td>50%</td>
<td></td>
</tr>
<tr>
<td>Dense layer</td>
<td>512 neurons</td>
<td>ReLU</td>
</tr>
<tr>
<td>Dropout</td>
<td>50%</td>
<td></td>
</tr>
<tr>
<td>Dense layer</td>
<td>128 neurons</td>
<td>ReLU</td>
</tr>
<tr>
<td>Dense layer</td>
<td>64 neurons</td>
<td>ReLU</td>
</tr>
<tr>
<td>Output size</td>
<td>$4 \times 1$(QPSK)</td>
<td>SoftMax</td>
</tr>
<tr>
<td></td>
<td>$8 \times 1$(8PSK)</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: CNN structure

<table>
<thead>
<tr>
<th>Layer</th>
<th>Parameters</th>
<th>Activation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input size</td>
<td>$2 \times 16$</td>
<td></td>
</tr>
<tr>
<td>Conv layer</td>
<td>$1 \times 3$ filter</td>
<td>ReLU</td>
</tr>
<tr>
<td>Conv layer</td>
<td>$1 \times 3$ filter</td>
<td>ReLU</td>
</tr>
<tr>
<td>Dense layer</td>
<td>512 neurons</td>
<td>ReLU</td>
</tr>
<tr>
<td>Dropout</td>
<td>50%</td>
<td></td>
</tr>
<tr>
<td>Dense layer</td>
<td>128 neurons</td>
<td>ReLU</td>
</tr>
<tr>
<td>Dropout</td>
<td>50%</td>
<td></td>
</tr>
<tr>
<td>Dense layer</td>
<td>128 neurons</td>
<td>ReLU</td>
</tr>
<tr>
<td>Dropout</td>
<td>50%</td>
<td></td>
</tr>
<tr>
<td>Output size</td>
<td>$4 \times 1$(QPSK)</td>
<td>SoftMax</td>
</tr>
<tr>
<td></td>
<td>$8 \times 1$(8PSK)</td>
<td></td>
</tr>
</tbody>
</table>

window as it is suitable for the input $2 \times 16$. After inserting the convolutional layer, dense layers and 50% dropout are applied.

The FCDNN has 413,316 trainable parameters, while the CNN has 107,802 to train the network. We use the Adam optimizer [33], which can adaptively adjust parameters during training to reduce the time cost for convergence. The learning rate is set at 0.001 and early stopping techniques are applied at the training phase to prevent overfitting. Both FCDNN and CNN detectors are implemented using open-source library Keras in Python. We use TensorFlow as the
backend, and implement the deep learning program on the platform of Google Colab using its GPU resources on the cloud.

4. Simulation results and discussion

In this section, numerical results are presented to describe the performance of the learning-based detectors for QPSK and 8PSK signals under various interference conditions. Firstly, the performance of signal detection for varying signal-to-noise ratios is analyzed. Afterwards, we investigate the performance for varying SIRs. Finally, we consider different numbers of interferences.

4.1. Performance under AWGN

![Signal detection accuracy of CNN, FCDNN and zero-forcing detectors with varying SNR for QPSK signals.](image)

We consider the detector for various SNRs from -20 to 20 dB with a step size of 2 dB. FCDNN and CNN are applied to detect QPSK and 8PSK signals. An interferer with 0 dB and 20 dB SIR is also considered to compare the performances with interference. Zero-forcing detector with least-squares channel estimator and the ideal performance without interference are also given to benchmark the performance gap for the learning-based detectors.

Figure 3 describes the accuracy of signal detection for QPSK modulated signals. One sees that CNN and FCDNN have comparable performances generally when there is no interference. Also, FCDNN and CNN achieve the best performance at nearly 99.5% accuracy when SNR increases to 11 dB. Beyond 11 dB, the accuracy has an upper limit and improves little when the SNR increases further. There is a large performance gap between the ideal performance without interference and FCDNN and CNN based detectors from -20 dB to 12 dB, which narrows down from about 40% at -20 dB to 0% at 12 dB. For detection with a SIR of 20 dB, the learning-based detectors show similar performances to the ones with no interference and FCDNN and
CNN have better accuracy than the zero-forcing detector when SNR is beyond 6 dB. When the SIR is 0 dB, FCDNN generally outperforms CNN by approximately 3% accuracy when SNR is larger than 6 dB and both FCDNN and CNN outperform the zero-forcing detector when SNR is beyond 4 dB. The performance difference widens as the SNR grows. Specifically, the limiting accuracy of the learning-based detectors for 0 dB SIR is about 91% and 99.5% for 20 dB SIR or no interference, while the limiting performance of the zero-forcing detector decreases from about 98% to 58% when SIR changes from 20 dB to 0 dB. This is because the zero-forcing detector cannot correctly obtain the channel state information. This shows that learning-based detectors have better robustness than zero-forcing detectors with least-squares channel estimator in the presence of co-channel interference and noise.

![Figure 4: Signal detection accuracy of CNN, FCDNN and zero-forcing detectors with varying SNR for 8PSK signals.](image)

We extend the test to detecting 8PSK modulated signal, which is shown in Figure 4. In this case, for detection with no interference and 20 dB SIR, FCDNN and CNN have similar trends in the performance and reach the best detection performance at around 93% when the SNR is large. However, the detection with 20 dB SIR requires 3 dB more SNR to achieve the same accuracy as the detection without interference. The ideal performance without interference and the performance of the zero-forcing detector with 20 dB SIR are better than the FCDNN and CNN detectors for all SNR conditions, while the difference decreases as SNR grows. For the detection with 0 dB SIR, FCDNN and CNN can achieve up to 77% and 69% accuracy, respectively, while zero-forcing detector can only achieve about 40% accuracy. This shows significant deterioration in the performance compared with the detection with no interference or 20 dB SIR. Also, the zero-forcing detector has better performance than the learning-based detectors when SNR is less than -2 dB while the learning-based detectors outperform the zero-forcing detector when SNR is larger than 0 dB. Furthermore, FCDNN outperforms CNN from SNR=-4 dB and the gap between FCDNN and CNN widens until reaching about 8% when SNR=10 dB. These results have shown that the FCDNN and CNN deal with interference better than the zero-forcing detector. Addition-
ally, FCDNN outperforms CNN when SIR is low because more trainable parameters in FCDNN makes the detector more capable of classifying signals in a poor environment. Compared with QPSK signals, the performance of the learning-based detectors can be compromised due to reduced Euclidean distance of 8PSK signals when SNR and SIR are high. For both QPSK and 8PSK signals, the detection with and without interference has no difference until SNR is higher than 6 dB, which shows the robustness of the detectors for interference and low SNR conditions compared with the zero-forcing detector.

Figure 5: QPSK signal detection accuracy of FCDNN at 4 dB SNR.

To show more details on what symbol affects the classification performance of the learning-based detectors, Figure 5 and 6 show the confusion matrix of FCDNN for QPSK and 8PSK when SNR is 4 dB and 8 dB respectively. In Figure 5, there is about 20% possibility that symbol 00 is predicted as 01 and 10, whereas it is less likely to be confused by 11, because it requires the change of two bits to become 11 and only one bit to become 01 or 10. In Figure 6, symbol 111 can be mistaken as its neighbors on the constellation diagram, 101 and 110. Thus, these results demonstrate that it is more likely for the detector to be confused by symbols with one-bit difference.

4.2. Performance under various interference

Figure 7 compares the performances of the FCDNN and CNN detectors when the interference is modulated by BPSK, QPSK, 8PSK, 4PAM, 16QAM and 64QAM under varying SIRs. The SNR at receiver is set at 10 dB when the detector has excellent performance from Figure 3.

When SIR is less than 0, FCDNN shows better performance for detection of low-order modulated interference (BPSK, QPSK, 4PAM), while CNN and FCDNN can have almost identical
Figure 6: 8PSK signal detection accuracy of FCDNN at 8 dB SNR.

Figure 7: Signal detection accuracy of CNN and FCDNN with different types of interference at SNR=10 dB for QPSK signals.
performance for high-order interference (8PSK, 16QAM, 64QAM in all cases). As SIR grows up to 24 dB, the accuracy of CNN and FCDNN is between 97% and 99%, regardless of the modulation type of the interfering signal. To sum up, at low SIR conditions, the modulation type of the interference can significantly affect the performance of the detector. Specifically, BPSK modulated interference tends to have the minimal impact on the performance, followed by 4PAM, QPSK, 8PSK. Also, 16QAM and 64QAM modulated interferences have similar influence on the performance, but the accuracy of 16QAM and 64QAM still has over 40% less than the accuracy of BPSK modulated interference.

![Figure 8: QPSK signal detection accuracy of FCDNN at -8 dB SIR with BPSK interference.](image)

Figure 8 and 9 describe the confusion matrix when SIR is -8 dB for the BPSK and 64QAM modulated interference, respectively. The BPSK interference causes difficulty in distinguishing the adjacent symbols for the desired signals, while the confusion on the classification spreads to further symbols in 64QAM and the performance with the 64QAM interference degrades significantly. Therefore, it is shown that low-order modulation of the interference can have less negative impact on the signal detection than high-order modulation. Specifically, symbols of interfering signal can overlap with points of the desired signal on the constellation diagram. Thus, the points with the same phase shift and amplitude can represent different binary digits of information. This creates the barrier for learning-based detectors due to the identical features representing different symbols. Interference with high-order modulation exacerbates this dilemma by increasing the number of overlapping points and reducing the distance between symbols on the constellation diagram. As a result, the boundaries between different symbols are seriously blurred in detection. However, this problem can be partly fixed by studying the historical signal distribution. Different numbers and frequencies of signal occurrence can improve the learning-based detector.

Figure 10 compares the performance of detecting 8PSK signals with different modulated inter-
Figure 9: QPSK signal detection accuracy of FCDNN at -8 dB SIR with 64QAM interference.

Figure 10: Signal detection accuracy of CNN and FCDNN with different types of interference at SNR=10 dB for 8PSK signals.
interferences. At low SIR, similar observation can be made from Figure 10 where low-order modulation of interference has less negative influence on the detection performance than high-order modulation. However, at high SIR, the accuracy achieves the best performance at $\text{SIR}=20 \text{ dB}$ and stays around 80%. Compared with the detection of QPSK signals, the gap of improvement by using FCDNN at low SIR decreases to about 2-5%.

4.3. Performance under various numbers of interference

![Figure 11: Signal detection accuracy of CNN and FCDNN with varying numbers and types of interference for QPSK signals.](image)

In this case, we consider the impact of the number of interferers. We simulate 0 to 90 interferences with a step size of 10 for interference modulated by BPSK, QPSK, 8PSK, 16QAM and 64 QAM. We also randomly modulate interferences so that part of the interferers has been modulated by BPSK, QPSK while other parts have been modulated by 16QAM and 64QAM etc. The SIR and SNR are all set at 10 dB.

Figure 11 shows the performance for detecting QPSK signals when increasing the number of interferences. In this figure, the performances of FCDNN and CNN with the same type of interference have almost identical curves. However, FCDNN still has slightly higher accuracy than CNN in most cases. In general, the accuracy of the detection linearly decreases from about 98% to roughly 50%-79%, when the number of interferences increases from 0 to 90. The curve for BPSK modulated interference has a higher accuracy than other interferences when the number of interferences is greater than 30. The performance with random interference is better when the number of interferences is lower than 30. The 64QAM interference has relatively more negative impact on the detection performance than other types of interference, whose accuracy is up to 20% less than that for BPSK interference. The accuracy of the detection with random interference fluctuates between 20 and 90 interferences, which outperforms signal detection with 16QAM, 64QAM and 8PSK interferences.
In Figure 12, a similar figure is shown for 8PSK. The accuracy decreases from roughly 80% to about 30%-50%, which is approximately 20% less than the detection for QPSK signals. Besides, FCDNN and CNN have very similar performances. In this figure, only the detection with BPSK and 4PAM interference can achieve accuracy above 50% when the number of interferers increases to 90, while others drop below 40%.

4.4. Performance over different fully connected layers

In general, increasing the number of fully connected layers in neural networks can enhance the learning ability, whereas it was proven in [13] and [14] that the system performance cannot be improved by deepening the network beyond a certain number of layers. Also, more hidden layers can result in overfitting and consequently compromise the system accuracy. Therefore, we conducted experiments to compare the performances of the FCDNN detectors with different numbers of fully connected layers for detecting QPSK signals when SIR changes from -20 dB to 20 dB with a step size of 2 dB, to discover how the number of layers affects the system performance.

In Figure 13, FCDNN with 5 layers and 10 layers has better performance than other detectors in most cases. When SIR is less than 0 dB, FCDNN with 1 layer has the lowest accuracy as a result of lacking feature extraction capacity, whereas the performance gap between FCDNN with 1 layer and other detectors decreases from about 30% to about 5% as the SIR increases to 2 dB. When SIR is beyond 4 dB, FCDNN can have comparable performance with other detectors, while FCDNN with 20 layers has the worst performance and FCDNN with 5 layers and 10 layers always outperforms FCDNN with 20 layers by about 5%. Therefore, it can be concluded that increasing layers does not always improve the system performance. Figure 14 shows the process of reducing the loss in training and validation datasets for FCDNN with different layers. To
Figure 13: Signal detection accuracy of FCDNN with varying numbers of fully connected layers for QPSK signals.

Figure 14: Training Performance of FCDNN with varying numbers of fully connected layers for interfered QPSK signals.
prevent overfitting, early stopping technique automatically finishes the training process when the difference between training loss and validation loss increases over a certain number of epochs. From this figure, about 140 epochs and 85 epochs are required for the FCDNN with 5 layers and 10 layers to complete fitting, while 15 layers and 20 layers only need 50 and 25 epochs. Also, the loss curve of FCDNN with 15 and 20 layers fluctuates more and has a larger gap between its training loss and validation loss than FCDNN with 5 layers and 10 layers, which explains why over 10 layers can cause overfitting and affect the detection performance.

5. Conclusion

In this paper, we have proposed a deep learning-based signal detection scheme for wireless communications. FCDNN and CNN-based signal detectors have been implemented. Different types of co-channel interferences in the communications system have been studied and compared under several scenarios, including signal types, varying SNR, SIR, numbers of interferences and layers of neural networks. Simulation results have demonstrated that CNN and FCDNN detectors can outperform zero-forcing detector in the presence of interference and FCDNN outperforms CNN in low SIR conditions. At 0 dB SIR condition, the learning-based detector has an upper limit of accuracy at about 91% for QPSK signals and around 77% for 8PSK signals. Furthermore, lower-order modulated signals and interference can have less negative effect on the detection accuracy for the learning-based detectors. Although the learning-based signal detectors still have performance gaps at high SIR conditions compared with the ideal detection and zero-forcing detector, the learning-based detectors show better robustness to the different SNRs and SIRs conditions and simplify the design of the signal receiver by directly learning the characteristics of the interfering channels without estimating and equalizing the channel. In addition, the simulation results have proven that deepening the network cannot help improving the detection accuracy and even degrade the system when the number of layers is beyond a certain amount.

References


URL https://doi.org/10.1145/3065386


Chenguang Liu  
Chenguang Liu received his B.E. degree in software engineering from Dalian University of Technology, Dalian, P.R.China, in 2016 and M.S. degree in advanced computer science from The University of Manchester, UK, in 2017. He is currently studying as an PhD student at the University of Warwick, UK. His research interests include wireless communications, signal detection, deep learning.

Yunfei Chen  
Yunfei Chen (S’02-M’06-SM’10) received his B.E. and M.E. degrees in electronics engineering from Shanghai Jiaotong University, Shanghai, P.R.China, in 1998 and 2001, respectively. He received his Ph.D. degree from the University of Alberta in 2006. He is currently working as an Associate Professor at the University of Warwick, UK. His research interests include wireless communications, cognitive radios, wireless relaying and energy harvesting.

Shuang-Hua Yang  
Shuang-Hua Yang received the B.S. degree in instrument and automation and the M.S. degree in process control from the China University of Petroleum (Huadong), Beijing, China, in 1983 and 1986, respectively, and the Ph.D. degree in intelligent systems from Zhejiang University, Hangzhou, China, in 1991. He was awarded DSc from Loughborough University in 2014 to recognize his academic contribution to wireless monitoring research. He is currently a chair professor of the department of computer science and engineering and vice dean of graduate school with Southern University of Science and Technology (SUSTech), Shenzhen, China. Before joined SUSTech in 2016 he had spent over two decades in Loughborough University, as a professor in computer science and head of department. His current research interests include cyber-physical system safety and security, Internet of Things, wireless network-based monitoring and control. He is a Fellow of IET and a Fellow of InstMC, U.K. He is an Associate Editor of the IET Journal Cyber-Physical Systems Theory and applications, the InstMC Journal Measurement and Control, and the International Journal of Computing and Automation.