Deep Learning-Based Narrowband Interference Cancellation Methods for OFDM Systems (Postprint)

Authors: Ma Tao, Li Yue, Yang Feng, Ding Lianghui

Date: 2019-04-01T00:00:00+00:00

Abstract

To address the performance degradation caused by narrowband interference (NBI) in orthogonal frequency division multiplexing (OFDM) systems, two deep learning (DL)-based narrowband interference cancellation structures are proposed. In both structures, the received signals are first preprocessed separately, and then a convolutional neural network (CNN) is utilized to extract features from the preprocessed data in the time domain and obtain interference estimates. Finally, the interference estimates are subtracted from the received signals. Simulation results demonstrate that both structures can effectively learn the narrowband interference in OFDM systems and improve system performance.

Full Text

Preamble

Ma Tao, et al.: Research on Narrowband Interference Cancellation Methods for OFDM Systems Based on Deep Learning, Vol. 37 No. 5, ChinaXiv Cooperative Journal

1 System Model

We assume ideal synchronization between the transmitter and receiver in the OFDM system. The OFDM signal has N subcarriers. After passing through an additive white Gaussian noise (AWGN) channel, the received signal can be expressed as

$$y = x + n$$

where x is the time-domain transmitted signal and n is time-domain noise. When narrowband interference (NBI) d is present, the equation becomes

$$y = x + n + d$$

where F is the discrete Fourier transform (DFT) matrix, D is the narrowband interference vector in the frequency domain, and β represents the offset ratio between the interference and transmitted signal subcarriers. It can be observed that D is a sparse vector whose non-zero elements represent interference components at corresponding positions. Assuming the number of interfered subcarriers is B, and letting I denote the set of indices for these subcarriers, then the i-th element of D is

$$D_i = \begin{cases} E_j e^{j2\pi f_c (i-f_c)\beta}, & i \in I \\ 0, & \text{otherwise} \end{cases}$$

where E_j represents the interference amplitude, f_c is the center frequency offset, and β is the offset parameter. Figure 1 shows the CNN-residual-iteration structure.

The received signal y contains noise and interference components. The interference vector d includes not only actual interference information but also erroneous information propagated from noise during transmission. Therefore, d can be written in the form of Equation (5), where n_d represents the erroneous component in the noise and NBI vector.

As mentioned earlier, in the architecture, the CNN acts as an interference estimator. It extracts time-domain features of interference from the input and outputs the corresponding interference estimate $\hat{d}$. We will discuss the CNN architecture in detail later. Subsequently, the estimated interference component $\hat{d}$ is subtracted from the received signal y to achieve interference cancellation:

$$\hat{x} = y - \hat{d}$$

where $\hat{x}$ is the interference-cancelled received signal. The interference vector d can be obtained from the signal-to-interference ratio (SIR).

2 Convolutional Neural Network

As a deep learning method, CNNs use local convolution operations to extract sample features from input data. A typical CNN includes convolutional layers, activation layers, pooling layers, and fully connected layers. During convolution, convolution kernels slide across the input vector or feature map, extracting local features and mapping them to new feature maps. Following convolutional layers are activation layers, which introduce non-linear features to the feature maps.

Common activation functions include tanh and rectified linear unit (ReLU). In pooling layers, feature maps from previous layers are partitioned and average or maximum values are output, reducing feature map dimensions while preserving main data characteristics. Finally, fully connected layers connect all previous feature maps and output corresponding decisions.

During training, CNNs use labeled data to minimize loss functions and obtain optimal network parameters. In summary, CNNs can fully extract local features from input data and train into a stable model. In this paper, considering that OFDM signals and interference have frequency-domain characteristics, we attempt to use CNNs to estimate and reconstruct narrowband interference signals, implementing narrowband interference cancellation in OFDM systems.

3 CNN-Based Interference Cancellation Architecture

In this paper, two deep learning-based OFDM communication architectures are adopted for narrowband interference cancellation: residual iteration and threshold filtering, as shown in Figures 1 and 2. In both architectures, the CNN serves as an interference estimator that estimates interference information from the time domain.

3.1 Two System Architectures

In the residual iteration architecture, the received signal y from Equation (2) is first fed into the receiver for demodulation and decoding. The receiver' s corresponding output $\hat{s}_0$ is re-encoded and modulated at the transmitter side, with the corresponding output result being $\hat{x}_0$. The difference between y and $\hat{x}_0$, denoted as $\hat{d}_0$, serves as input to the convolutional neural network, i.e.,

$$\hat{d}_0 = y - \hat{x}_0$$

In Equation (6), $\hat{x}_1$ is the received signal after interference removal, which will again be input to the receiver for demodulation, decoding, and re-encoding. The final output result is $\hat{s}_1$.

From the residual iteration architecture, it can be seen that the CNN, as an interference estimator, receives input $\hat{d}_0$ derived from the difference between the received signal and the retransmitted result.

The CNN-threshold filtering structure is shown in Figure 2 [Figure 2: see original paper]. Unlike the residual iteration architecture, in the threshold filtering structure, the received signal y first undergoes frequency-domain transformation. The frequency-domain signal is Y, the filter threshold is T, and the corresponding output after frequency-domain filtering is $\hat{D}$. Here, Y_i represents data on the i-th subcarrier. During filtering, we first calculate the amplitude $|Y_i|$ of all subcarriers in Y, while taking the threshold T as the average amplitude of all subcarriers in Y. Let $\hat{I}$ be the set of indices of subcarriers in Y whose amplitude exceeds the threshold. Then $\hat{D}$ can be viewed as the data set in Y at these indices. Finally, $\hat{D}$ is transformed back to the time domain, i.e., $\hat{d} = F^H \hat{D}$. Analyzing the components of $\hat{d}$, we find that $\hat{d}$ contains real interference, while the subcarriers in $Y \setminus \hat{D}$ contain signal and noise components. Based on this, $\hat{d}$ can be decomposed into

$$\hat{d} = d + n_d$$

where n_d represents the signal and noise components on the interfered subcarriers, and d represents the actual interference components present on these subcarriers and the interference components not in $\hat{I}$. Therefore, $\hat{d}$ contains interference information.

Connecting the CNN after the threshold filter attempts to extract interference features from $\hat{d}$ and obtain the corresponding estimate $\hat{d}_{CNN}$. Similar to the residual iteration architecture, we subtract $\hat{d}_{CNN}$ from $\hat{d}$ as shown in Equation (6). The result $\hat{x}$ will be input to the receiver for equalization, demodulation, and decoding, with the final estimated result being $\hat{s}$. From the description of both architectures, it can be seen that this paper generates CNN input data through two approaches: time-domain residual differences and frequency-domain filtering operations.

3.2 CNN-Based Interference Estimation

Before introducing the CNN architecture, we first explain the necessity of learning interference from the time domain. Directly estimating interference features from the frequency domain results in excessive redundancy in the input data, making it impossible to effectively learn interference frequency-domain information. Only the interfered subcarriers and their neighboring subcarriers contain useful information for interference learning and estimation.

Time-domain received signals include interference components after IFFT transformation. Noting that interference components are sparse in the frequency domain, the IFFT transformation is equivalent to a compressed sensing process that compresses the sparse interference vector. Therefore, in this work, estimating interference information from the time domain is similar to compressed sensing. However, the data components in the received signal pose challenges for interference estimation. Directly applying compressed sensing technology to map frequency-domain interference to the time domain requires prior knowledge of the interference sparsity pattern at the receiver, which reduces system spectral efficiency. This paper proposes using CNN as a blind estimator to identify the relationship between the interference vector and received signal without requiring prior information. Since CNNs can extract inherent relationships from network inputs, we select CNN as the interference estimator.

Table 1 shows the basic network configuration. The input and output sizes remain unchanged at $2 \times N$, where N is the number of OFDM signal subcarriers. Since the input and output data dimensions are not reduced, the CNN in this paper will only include convolutional layers and activation function layers.

Let L represent the number of convolutional layers, with an activation layer following each convolutional layer. Let k_l and n_l denote the kernel size and number of kernels in the l-th convolutional layer, respectively. The proposed CNN architecture is similar to that in reference [8]. Regarding the network design, several points are noteworthy: First, this network does not introduce pooling layers or fully connected layers. As previously described, during forward and backward propagation, the data dimension remains $2 \times N$ (separating real and imaginary components) without change. During convolution, zero-padding is used to ensure convolution operations do not reduce data dimensions. Second, after IFFT transformation, time-domain signals contain complex numbers. Therefore, this network uses the tanh function as the activation function, as shown in Equation (10), because ReLU cannot effectively map complex number features.

3.3 Network Training Process

To train the CNN into a stable, converged model for practical application, this section discusses the training and optimization process for the proposed network. We generate OFDM signals and corresponding narrowband interference according to Equations (2) and (3). When generating narrowband interference, B represents the interference bandwidth, f_c is the center frequency offset, and β is the offset parameter. The noise and interference components can be determined by the signal-to-noise ratio (SNR) and signal-to-interference ratio (SIR). We assume B is fixed during training, meaning the number of interfered subcarriers remains constant. The center subcarrier f_c is selected from several predetermined positions, and β follows a uniform distribution $\beta \sim U[-\frac{1}{2}, \frac{1}{2}]$. SNR and SIR represent signal transmission conditions and determine communication performance. In this paper, R_{SIR} and R_{SNR} denote the SIR and SNR values used during training data generation.

We employ mini-batch gradient descent for network training. Since the CNN model is a regression network, we define the loss function as L2 loss:

$$\text{Loss} = \frac{1}{N}\|\hat{d} - d\|_2^2$$

During training, we use the Adam optimizer to minimize the loss and update network parameters. A portion of the training data is selected as the validation set. Network parameters are updated on the training set while monitoring performance on the validation set. Training stops early if the validation loss ceases to decrease. Additionally, the learning rate is appropriately reduced during training to decrease parameter update magnitude and ensure convergence. The learning rate at step t is defined as

$$\alpha_t = \alpha_0 \cdot \frac{c_{\text{step}} - t}{c_{\text{step}}}$$

where α_0 is the initial learning rate, c_{step} is the total number of training steps, and t is the current step. Although training data volume is substantial, making training cost and complexity significant, the proposed architecture remains valuable for practical applications. We will train the network offline, and the generated training data will also be produced and stored offline. To ensure the trained model is stable and applicable to more scenarios, we include as many diverse situations as possible during training data generation. In practical application, the pre-trained CNN model can be used directly.

4 Simulation Results

Unless otherwise specified, all simulations use the parameters in Table 2 . At the transmitter, binary phase shift keying (BPSK) modulation and '1/3' rate Turbo coding are used. For each (SIR, SNR) scenario, we generate 500 frames (selecting 100 as the validation set), with each frame containing 10 Turbo blocks of length 194, i.e., 101 OFDM symbols per frame. To evaluate system performance, we select the "frequency-domain threshold filtering" method from reference [3] as the baseline. We first calculate the amplitude of each OFDM subcarrier and the corresponding threshold, then null data on subcarriers whose amplitude exceeds the threshold. Since multiple network structures are evaluated, we denote 'CNNk_i' as a CNN with k convolutional layers in configuration i, e.g., 'CNN5_0' . Network training and testing are completed on TensorFlow, while other system components are built and simulated in MATLAB.

4.1 Network Structure Selection

As previously mentioned, the CNN extracts interference features from the time domain and outputs the estimated interference vector. Therefore, the network structure significantly impacts system performance. We trained several CNN structures shown in Table 3 and compared their loss functions and corresponding system performance. Ultimately, we determined that CNN5_5 provides the most stable training effect and good performance on the test set for both architectures.

When comparing various models, we observed that increasing network depth beyond $L = 5$ yields minimal improvement in learning performance. Analysis suggests this is because the input data contains many low-level features but few high-level features, making deeper networks unnecessary. Increasing kernel size actually degrades performance, likely because convolution kernels process local information in the input signal. Given the input data dimensions, larger kernels cannot effectively extract local features. However, increasing network width improves learning performance, indicating that input data still contains numerous low-level features for CNN extraction. We also attempted removing the frequency-domain filter in Figure 2 and estimating interference directly from the time-domain received signal. While the CNN could still learn interference features, performance improvement was limited compared to including the filter. We attribute this to the presence of many useful signals not affected by interference in the time-domain received signal, which impacts interference estimation accuracy. Although threshold filtering may remove some interfered information, making CNN input incomplete, the overall system performance is still better than without filtering.

4.2 Network Performance Verification

In previous simulations, training and test sets were generated under identical conditions (SNR, SIR, B, f_c, β). However, in practical scenarios, these conditions may differ between training and application. Therefore, the proposed system needs robustness—effective interference learning and performance improvement under conditions different from training.

We test the trained CNN model under various scenarios to verify robustness. First, we evaluate system performance under different SIRs, fixing the SIR set to $[-6 : 1 : -3]$ dB and using the CNN5_4 model. The resulting performance curves are shown in Figure 3 [Figure 3: see original paper].

As shown in Figure 3, both proposed architectures reduce the bit error rate of interfered OFDM systems and improve performance. Compared to frequency-domain threshold filtering, the deep learning-based frequency-domain filtering architecture can more effectively estimate interference features and improve system performance. While the residual iteration architecture only slightly improves communication performance, the threshold filtering architecture can more accurately learn interference information and provide greater performance gains. Analysis suggests that due to excessive erroneous information, the CNN input in the residual iteration architecture contains too much redundant information, affecting learning effectiveness.

To fully evaluate network performance, we also simulate system performance under different SNRs. The results are shown in Figure 4 [Figure 4: see original paper]. The curves demonstrate that the CNN-threshold filtering structure achieves excellent interference cancellation across various SNRs. The "no interference cancellation" curve shows that narrowband interference is the primary factor affecting OFDM communication performance.

During training data generation, the number of interfered subcarriers was fixed at $B = 7$. Next, we test system performance with different interference bandwidths. Figure 5 [Figure 5: see original paper] shows simulation results at SIR = -4 dB. As the number of interfered subcarriers increases, the system bit error rate rises, but the proposed CNN-threshold filtering structure can still effectively learn interference features and reduce error rates. Compared to threshold filtering, the performance improvement of the CNN-based approach becomes more significant as more subcarriers are interfered.

Furthermore, during training, the interference center frequency f_c was limited to a few positions. In practice, the interference center frequency may not align with training scenarios. Therefore, we test system stability under frequency offset conditions. As shown in Table 2, we define a center frequency every 16 subcarriers. In this simulation, we assume the center frequency in the test set is offset by 4-8 subcarriers relative to the training set, with other conditions unchanged. The results are shown in Figure 6 [Figure 6: see original paper]. The "no cancellation" and "frequency-domain filtering" methods show little variation with subcarrier offset. However, the performance of our proposed structure degrades as the offset increases. Nevertheless, even when the offset reaches 7-8 subcarriers (making interfered subcarriers non-overlapping between training and test sets), the system can still estimate and cancel interference accurately. When the offset is no more than 5 subcarriers, the proposed model outperforms frequency-domain filtering.

References

[1] Coulson A J. Bit error rate performance of OFDM in narrowband interference with excision filtering [J]. IEEE Trans on Wireless Communications, 2006, 5(9): 2484-2492.

[2] Darsena D. Successive narrowband interference cancellation for OFDM systems [J]. IEEE Communications Letters, 2007, 11(1).

[3] Shi Kai, Zhou Yi, Kelleci B, et al. Impacts of narrowband interference on OFDM-UWB receivers: analysis and mitigation [J]. IEEE Trans on Signal Processing, 2007, 55(3): 1118-1128.

[4] Gomaa A, Al-Dhahir N. A compressive sensing approach to NBI cancellation in mobile OFDM systems [C]//Proc of Global Telecommunications Conference. Piscataway, NJ: IEEE Press, 2010: 1-5.

[5] Goodfellow I, Bengio Y, Courville A, et al. Deep learning [M]. Cambridge: MIT Press, 2016.

[6] Wang Tianqi, Wen Chao-Kai, Wang Hanqing, et al. Deep learning for wireless physical layer: Opportunities and challenges [J]. China Communications, 2017, 14(11): 92-111.

[7] Ye Hao, Li G Y, Juang B H F, et al. Channel agnostic end-to-end learning based communication systems with conditional GAN [J]. arXiv preprint arXiv: 1807.00447, 2018.

[8] Liang Fei, Shen Cong, Wu Feng. An iterative BP-CNN architecture for channel decoding [J]. IEEE Journal of Selected Topics in Signal Processing, 2018, 12(1): 144-159.

[9] O' Shea T J, Roy T, Clancy T C. Over-the-air deep learning based radio signal classification [J]. IEEE Journal of Selected Topics in Signal Processing, 2018, 12(1): 168-179.

5 Conclusion

This paper addresses OFDM systems and deep learning technology, investigating narrowband interference cancellation in OFDM systems. We first analyze and evaluate traditional narrowband interference cancellation methods and their limitations. We then introduce the application of deep learning in physical layer communications. Two deep learning-based architectures are proposed: residual iteration and threshold filtering. By learning the mechanism of OFDM narrowband interference cancellation, we utilize convolutional neural networks to learn interference information from the time domain and perform cancellation.

Combined with the actual interference model, we design a CNN regression network for interference learning and estimation. The learned interference is subtracted from the received signal in the time domain, and the interference-cancelled signal is processed by the receiver to obtain the final result. Simulation results demonstrate that both proposed architectures can effectively learn interference features and achieve accurate estimation. The CNN-threshold filtering structure provides better interference estimation than the CNN-residual iteration structure. This work represents the first introduction of deep learning into OFDM narrowband interference cancellation. The deep learning-based interference cancellation architecture can accurately estimate and cancel narrowband interference without increasing system complexity, making it applicable to various scenarios.

Note: Figure translations are in progress. See original paper for figures.