

Channel Estimation with an Interpolation Trained Deep Neural Network

Yu Hu, Jianing Zhao, Bingyang Cheng

National Mobile Communications Research Laboratory, Southeast University, Nanjing, China

Email: 220190737@seu.edu.cn, jnzhao@seu.edu.cn, 220190736@seu.edu.cn

How to cite this paper: Hu, Y., Zhao, J.N. and Cheng, B.Y. (2021) Channel Estimation with an Interpolation Trained Deep Neural Network. *Journal of Computer and Communications*, 9, 123-131.
<https://doi.org/10.4236/jcc.2021.910008>

Received: October 10, 2021

Accepted: October 26, 2021

Published: October 29, 2021

Abstract

This paper proposes a deep learning-based channel estimation method for orthogonal frequency-division multiplexing (OFDM) systems. The existing OFDM receiver has low estimation accuracy when estimating channel state information (CSI) with fewer pilots. To tackle the problem, in this paper, a deep learning model is first trained by the interpolated channel frequency responses (CFRs) and then used to denoise the CFR estimated by least square (LS) estimation. The proposed deep neural network (DNN) can also be trained in a short time because it only learns the CFR and the network structure is simple. According to the simulation results, the performance of the DNN estimator can be compared with the minimum mean-square error (MMSE) estimator. Furthermore, the DNN approach is more robust than conventional methods when fewer pilots are used. In summary, deep learning is a promising tool for channel estimation in wireless communications.

Keywords

OFDM, Channel Estimation, Pilot, Deep Learning, Interpolation

1. Introduction

In an OFDM system, the CSI is important for precoding before transmission and decoding at the receiver. The accurate estimation of the CSI is used for the recovery of the transmitted symbols. It can effectively reduce the system error rate and improve system performance [1]. Usually, the CSI is estimated based on the pilot data sent by the transmitter and then the estimated CSI is used to recover the transmitted symbols.

The LS or MMSE estimation is normally used in traditional channel estimation methods [2] [3] [4]. The LS estimation is easy to implement since it needs no CSI. However, it ignores the influence of noise and leads to inadequate per-

formance. Compared to the LS estimation, the MMSE estimation utilizes the second-order statistics of the channel can achieve higher estimation accuracy. But, an MMSE estimator is undesirable. It needs to calculate the channel correlation matrix between the base station and user. Its complexity is also much higher than that of the LS estimation.

With the rapid development of machine learning technology and hardware performance in recent years, deep learning has been successfully applied to many fields including computer vision, natural language process, and so on [5]. Based on deep learning, the transmitter and receiver of the communication system can be regarded as two networks to complete end-to-end information transmission [6]. Deep learning can also be applied to the various modules of the communication system to optimize the performances of the modules [7] [8], such as channel estimation module and so on. In an OFDM system, deep learning is used to complete the channel estimation and signal detection and obtain a better signal recovery effect [9]. In addition, the deep neural network has excellent noise reduction capabilities. Before the LS estimation, the deep learning can be used to reduce noise to improve the estimation accuracy and research has found that such noise reduction network is robust to pilot pollution [4]. The DNN can directly learn the frequency domain information of the CSI. So it can perform noise reduction processing on the CSI estimated by the LS estimation and can be used for subsequent signal detection to avoid wasting resources by repeatedly sending pilots [10] [11] [12]. In the meantime, compressed sensing and deep learning can be combined to improve estimation accuracy [13] [14].

In this paper, we propose a deep learning-based channel estimation method in the OFDM system. We combine the DNN mode and LS estimation to make the two mutually benefited. To this end, we first train the DNN offline so that it learns the channel frequency characteristics in advance. In the online deployment stage, the trained DNN is capable of denoising the CSI estimated by the LS estimator. We found that directly using channel frequency characteristics as training data to train the DNN has poor robustness with fewer pilots. Therefore, we use the interpolated frequency characteristics to train the DNN and use the DNN recover the estimated CSI. The simulation results in this paper indicate that the proposed scheme has a good performance compared with traditional methods. In the future, deep learning can be potentially applied in channel estimation.

The remainder of this paper is organized as follows. The system model is given in Section 2. The proposed DNN estimator is explained in Section 3. The simulation results are presented in Section 4. Finally, we conclude the paper in Section 5.

2. System Model

2.1. Deep Neural Network

Figure 1 shows the architecture of a DNN model. A DNN is composed of an input layer, an output layer, and several hidden layers. Each layer is composed of

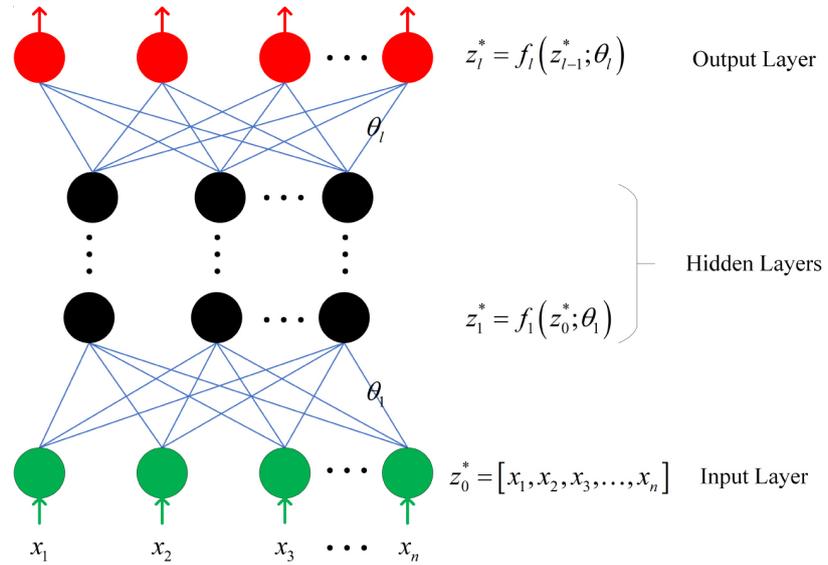


Figure 1. An example of deep neural network.

several neurons. The output of each neuron is a nonlinear function of the weighted sum of the output of previous neurons. The nonlinear function may be Rectified Linear Unit (RELU) function or Sigmoid function.

$$\sigma_{\text{ReLU}}(a) = \max(0, a) \tag{1}$$

$$\sigma_{\text{Sigmoid}}(a) = \frac{1}{1 + e^{(-a)}} \tag{2}$$

For an L layers DNN, the output at l -th layer with input $\mathbf{z}_{l-1}^* \in \mathbb{R}^{N_{l-1}}$ can be described as

$$\mathbf{z}_l^* = f_l(\mathbf{z}_{l-1}^*; \theta_l) = \sigma(\mathbf{W}_l \mathbf{z}_{l-1}^* + \mathbf{b}_l) \tag{3}$$

where $\mathbf{W}_l \in \mathbb{R}^{N_l \times N_{l-1}}$ and $\mathbf{b}_l \in \mathbb{R}^{N_l}$ represent the weight matrix and the bias vector, respectively. $\theta_l = \{\mathbf{W}_l, \mathbf{b}_l\}$ is the set of parameters for the l -th layer. Hence, the final output of the DNN is given by

$$\mathbf{z}_L^* = f_L(f_{L-1}(\dots(f_1(\mathbf{z}_0^*; \theta_1)\dots); \theta_{L-1}); \theta_L) \tag{4}$$

The DNN is trained with the training set which includes the training data and label data. The goal of the training is to minimize the loss function which is

$$L_{\text{MSE}}(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \frac{1}{N_L} \|\mathbf{z}_L - \mathbf{z}_L^*\|_2^2 \tag{5}$$

where $\mathbf{z}_L \in \mathbb{R}^{N_L}$ is the label data and $\boldsymbol{\theta} = \{\theta_1, \theta_2, \dots, \theta_L\}$ is the set of all training parameters [15].

2.2. System Architecture

Figure 2 shows the architecture of a baseband OFDM system model. At the transmitter, the transmitted signal are converted into a parallel data stream, then the signal are transformed from the frequency domain to the time domain by inverse discrete Fourier transform (IDFT). After that, the inter-symbol interference

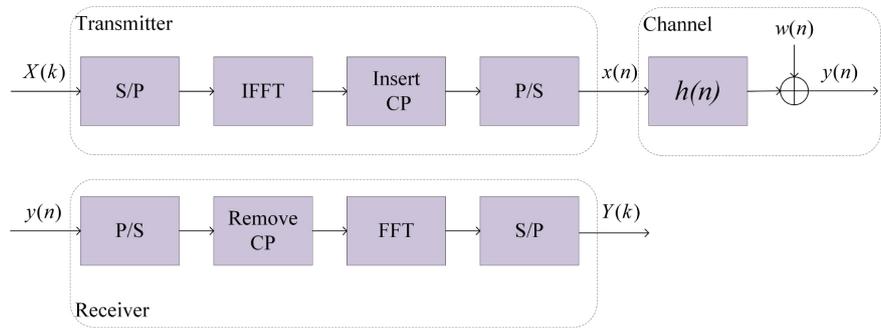


Figure 2. OFDM system.

(ISI) can be mitigated by inserting cyclic prefix (CP). Then the signal is converted into a serial data stream and sent to the channel. At the receiver, the received serial data stream is converted to a parallel data stream. Then the CP is removed and the signal are transformed to the time domain by discrete Fourier transform (FFT). Finally, the signal is converted to a serial data stream for output. The received signal $Y_i(k)$ of the k -th subcarrier in the i -th OFDM symbol can be expressed as

$$Y_i(k) = H_i(k)X_i(k) + W_i(k) \tag{6}$$

where $H_i(k)$, $X_i(k)$, and $W_i(k)$ denote the CFR, transmitted signal, and complex Gaussian noise with zero mean and variance σ_w^2 , respectively. We assume that the first OFDM symbol is a pilot symbol and the subsequent OFDM symbols are composed of the transmitted data. These symbols from a frame. Each OFDM symbol contains 64 subcarriers including data subcarriers index set \mathbb{S}_D and pilot subcarriers index set \mathbb{S}_P .

It is straightforward to express LS estimation as

$$H_p(k) = \frac{Y_p(k)}{X_p(k)}, \quad k \in \mathbb{S}_P \tag{7}$$

where $X_p(k)$ and $Y_p(k)$ are the pilot symbol and corresponding received signal. Then we get the estimated CFR $\hat{H}(k)$ by interpolating $H_p(k)$.

3. Deep Channel Estimator

The LS estimation has low complexity. However, its estimation accuracy is poor. Thus, we design a DNN to improve estimation accuracy. Our estimation method consists of two steps as shown in **Figure 3**.

- Estimating the CFR by the LS estimation.
- Using the DNN to denoise the estimated CFR. $\tilde{H}(k)$ is the CFR with denoise processing.

We use the FFT operation to convert the channel impulse responses to the CFRs as training data. We build a five layers DNN including the input layer and the output layer. The neurons in each layer are 128, 80, 40, 80, 128, respectively. Separate the real and imaginary parts of the CFR, so the number of neurons in the input and output layer are both 128, *i.e.*, the input vector

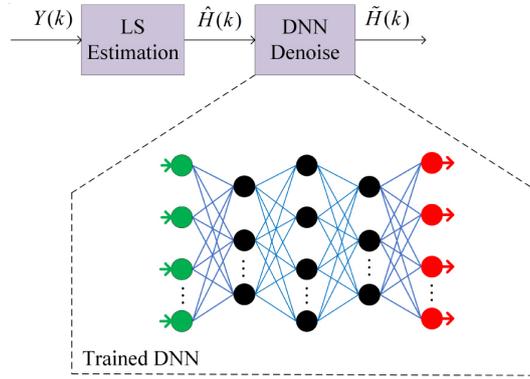


Figure 3. Schematic diagram of proposed channel estimation based on deep learning.

$$\mathbf{h}_0 = [\Re\{\mathbf{H}\}, \Im\{\mathbf{H}\}]^T \tag{8}$$

where $(\cdot)^T$ represents the transpose operation, $\Re\{\cdot\}$ and $\Im\{\cdot\}$ denote the real and imaginary parts of a complex CFR $\mathbf{H} = \{H(k)\}_{k \in \mathbb{S}_p \cup \mathbb{S}_D}$.

In our experiment, the DNN is trained to minimize the mean square error (MSE) between the input CFR and the output. The MSE is expressed as

$$L_{\text{MSE}}(\theta) = \arg \min_{\theta} \frac{1}{N_L} \|\mathbf{h}_L - \mathbf{h}_L^*\|_2^2 \tag{9}$$

where $h_L = h_0^*$ and h_L^* denote the label and output data. According to the simulation results in Section 3, the trained DNN performs poorly with a few pilots. Therefore, for a few pilots, e.g. 8 pilots subcarriers with index set

$\mathbb{S}_p = \{0, 8, 16, 24, 32, 40, 48, 56\}$. We train DNN through the following two steps.

- Extracting $H_1(k)$ from $H(k)$ according to the known pilot index $k \in \mathbb{S}_p$. $\bar{H}(k)$ can be obtained by interpolating $H_1(k)$ where $k \in \mathbb{S}_p \cup \mathbb{S}_D$. We use third-order spline interpolation to interpolate $H_1(k)$.
- Inputting the $\bar{H}(k)$ as training data into the DNN and the label data is $H(k)$. From the simulation results in Section III, the DNN trained by this method has a significant noise reduction with fewer pilots.

4. Simulation Results

In this section, we show the simulation results of the proposed method. In our experiment, an OFDM system with 64 subcarriers and the QPSK modulation method is considered. The wireless channel follows the WINNER II channel model, where the carrier frequency is 2.6 GHz and the number of paths is 24. The typical urban channels with maximum delay 16 sampling period are used. We train the DNN in different pilots with Adam optimizer. We initialize the weights and biases by Glorot Uniform to achieve the best learning performance [16]. The normalized mean square error (NMSE) is used as a performance matrix which is defined as

$$\text{NMSE} = \mathbb{E} \left[\frac{\|\tilde{\mathbf{H}} - \mathbf{H}\|_2^2}{\|\mathbf{H}\|_2^2} \right] \tag{10}$$

where $\tilde{\mathbf{H}}$ and \mathbf{H} are estimated CFR and actual CFR. The proposed method is compared with the LS, MMSE, deep learning-based [9], and AE estimation method [11]. [9] and [11] use the ideal CSI to train the neural network. In [9], The neural network performs not only channel estimation but also signal detection. The transmitted signal through the channel is used as the input signal of the neural network and the trained neural network will complete the channel estimation and demodulation and it restores the received signal to the transmitted binary bit stream. In [11], the author utilizes the trained neural network in the channel estimation procedure as shown in **Figure 3**. The trained neural network uses the channel matrix estimated by LS estimation as the input signal and it outputs the denoised CSI which can be used for data demodulation later.

Figure 4 compares the NMSE of the different channel estimation methods. The MMSE estimation method performs best with 64 pilots because the MMSE estimation utilizes the second-order statistical characteristics of the channel. For 64 pilots, the proposed DNN estimation is equivalent to the AE estimation, because the training data at this time is a 64-pilot channel matrix which is equivalent to the training data of the AE estimation. However, there are often no 64 pilots in actual situations. When 16 pilots are used, the NMSE curves of the MMSE, LS, and AE estimation decrease slightly when the SNR is above 15 dB. But the DNN estimation still has the ability to reduce NMSE when the SNR is greater than 15 dB. In the case of 8 pilots, the MMSE, LS, and AE estimation reach the performance limit at the SNR of 10 dB, while the DNN estimation method performs better than the above three methods from 0 dB and can also estimate well when the SNR is above 10 dB. **Figure 5** shows the bit-error rate

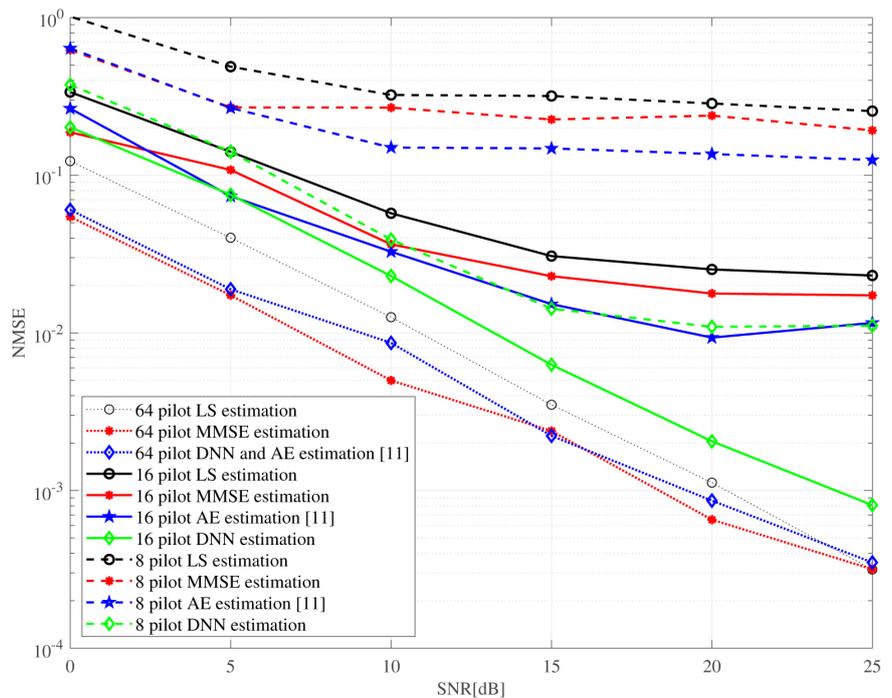


Figure 4. NMSE performance comparison in different pilots with respect to SNR.

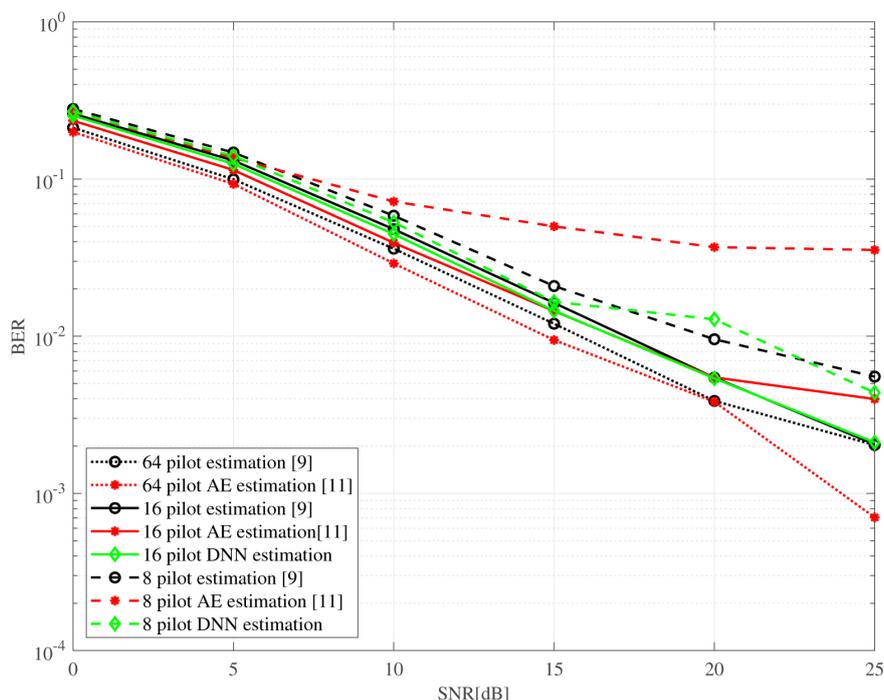


Figure 5. BER performance comparison in different pilots with respect to SNR.

(BER) of the three deep learning-based channel estimation methods. The BER of the three estimation methods has little difference with 16 pilots. But in the case of 8 pilots, the BER estimated by DNN estimation is much smaller than AE estimation at high SNR. Compared with the method in [9], although the BER of our DNN estimation is not significantly reduced, the network structure is much simpler than that in [9], which means that we can train the network in shorter time. A simpler network structure can be adjusted more quickly when using samples from the real wireless channels to fine-tune the DNN model. Besides, our network is more flexible. For example, the denoised channel matrix can be used for precoding at the transmitting end while [9] has no estimated channel matrix because it directly completes channel estimation and detection. It is obvious that the proposed DNN estimation can learn channel characteristics well with fewer pilots and has higher estimation accuracy compared with other methods according to the simulation results above.

The comparison of different network structures is shown in **Figure 6**. When we use the 40-20-40 network, the NMSE is the highest. As the number of hidden neurons increases, the NMSE will decrease slightly. However, the training time increases exponentially with increase of the complexity of the network structure. Therefore, there is a trade-off between network complexity and estimation accuracy. If the network is too simple, its learning channel characteristics effect and noise reduction effect are not significant. When the network is too complex may require a lot of training time. Thus, we should choose a suitable network on the different situations in order to achieve the higher estimation accuracy with less resource consumption.

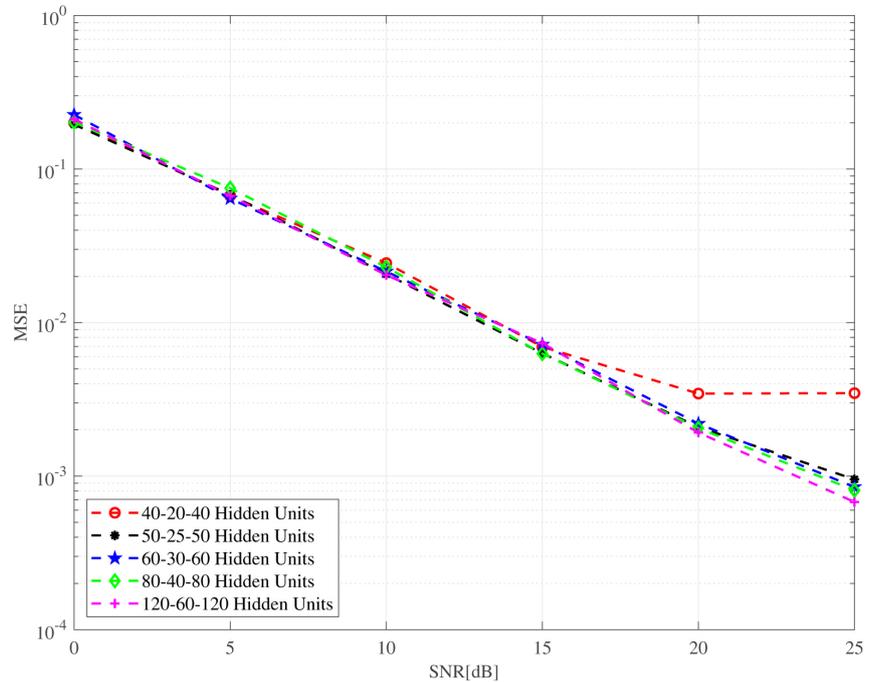


Figure 6. The MSE performance of the proposed estimator with respect to hidden units.

5. Conclusion

In this paper, we have proposed a novel deep learning-based channel estimator composed of DNN and LS estimation to improve the performance of the conventional LS estimation. To overcome the low estimation accuracy with few pilots, we developed a combined strategy of the DNN and the LS estimation. The DNN channel estimator exploits channel characteristics in the frequency domain very effectively and then denoises the LS estimation results. It was shown from the simulation results that the proposed scheme obtains an obvious performance gain over the conventional LS estimation and other deep learning estimation methods, which is attributed to that the training data provided in this paper are closer to the real channel estimation. In this work, the training data are generated by the WINNER II channel model. For practice use, we can collect samples generated from real wireless channels to retrain or fine-tune the model.

Acknowledgements

This work has been supported by the Research Fund of National Mobile Communications Research Laboratory, Southeast University (No. 2021C01).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Shannon, C.E. (1958) Channels with Side Information at the Transmitter. *IBM Jour-*

- nal of Research and Development*, **2**, 289-293. <https://doi.org/10.1147/rd.24.0289>
- [2] Sorenson, H.W. (1970) Least-Squares Estimation: From Gauss to Kalman. *IEEE Spectrum*, **7**, 63-68. <https://doi.org/10.1109/MSPEC.1970.5213471>
- [3] Cho, Y.S., Kim, J., Yang, W.Y. and Kang, C.G. (2010) Channel Estimation. *IEEE*, 187-207. <https://doi.org/10.1002/9780470825631.ch6>
- [4] Balevi, E., Doshi, A. and Andrews, J.G. (2020) Massive MIMO Channel Estimation with an Untrained Deep Neural Network. *IEEE Transactions on Wireless Communications*, **19**, 2079-2090. <https://doi.org/10.1109/TWC.2019.2962474>
- [5] Zhang, K., Zuo, W., Chen, Y., Meng, D. and Zhang, L. (2017) Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Transactions on Image Processing*, **26**, 3142-3155. <https://doi.org/10.1109/TIP.2017.2662206>
- [6] Dörner, S., Cammerer, S., Hoydis, J. and Brink, S.T. (2018) Deep Learning Based Communication over the Air. *IEEE Journal of Selected Topics in Signal Processing*, **12**, 132-143. <https://doi.org/10.1109/JSTSP.2017.2784180>
- [7] O'Shea, T. and Hoydis, J. (2017) An Introduction to Deep Learning for the Physical Layer. *IEEE Transactions on Cognitive Communications and Networking*, **3**, 563-575. <https://doi.org/10.1109/TCCN.2017.2758370>
- [8] Bkassiny, M., Li, Y. and Jayaweera, S.K. (2013) A Survey on Machinelearning Techniques in Cognitive Radios. *IEEE Communications Surveys Tutorials*, **15**, 1136-1159. <https://doi.org/10.1109/SURV.2012.100412.00017>
- [9] Ye, H., Li, G.Y. and Juang, B. (2018) Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems. *IEEE Wireless Communications Letters*, **7**, 114-117. <https://doi.org/10.1109/LWC.2017.2757490>
- [10] Soltani, M., Pourahmadi, V., Mirzaei, A. and Sheikhzadeh, H. (2019) Deep Learning-Based Channel Estimation. *IEEE Communications Letters*, **23**, 652-655. <https://doi.org/10.1109/LCOMM.2019.2898944>
- [11] Han, S., Oh, Y. and Song, C. (2019) A Deep Learning Based Channel Estimation Scheme for IEEE 802.11p Systems. *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, 1-6. <https://doi.org/10.1109/ICC.2019.8761354>
- [12] Gizzini, A.K., Cha_i, M., Nimr, A. and Fettweis, G. (2020) Deep Learning Based Channel Estimation Schemes for IEEE 802.11p Standard. *IEEE Access*, **8**, 113751-113765. <https://doi.org/10.1109/ACCESS.2020.3003286>
- [13] Ma, W., Qi, C., Zhang, Z. and Cheng, J. (2020) Sparse Channel Estimation and Hybrid Precoding Using Deep Learning for Millimeter Wave Massive MIMO. *IEEE Transactions on Communications*, **68**, 2838-2849. <https://doi.org/10.1109/TCOMM.2020.2974457>
- [14] Ma, W., Qi, C., Zhang, Z. and Cheng, J. (2019) Deep Learning for Compressed Sensing Based Channel Estimation in Millimeter Wave Massive MIMO. *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, 1-6. <https://doi.org/10.1109/WCSP.2019.8928030>
- [15] Goodfellow, I., Bengio, Y., Courville, A. and Bengio, Y. (2016) Deep Learning. MIT Press. <http://www.deeplearningbook.org>
- [16] Glorot, X. and Bengio, Y. (2010) Understanding the Difficulty of Training Deep Feedforward Neural Networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 249-256.