Adaptive Channel Estimation based on Deep Learning

Abdul Karim Gizzini, Marwa Chafii, Ahmad Nimr, Gerhard Fettweis

Adaptive Channel Estimation based on Deep Learning

Abdul Karim Gizzini*, Marwa Chafii*, Ahmad Nimr†, Gerhard Fettweis†
*ETIS, UMR8051, CY Cergy Paris Université, ENSEA, CNRS, France
†Vodafone Chair Mobile Communication Systems, Technische Universitat Dresden, Germany
Email: {abdulkarim.gizzini, marwa.chafii}@ensea.fr , {ahmad.nimr, gerhard.fettweis}@ifn.et.tu-dresden.de

***Abstract*—Channel state information is very critical in various applications such as physical layer security, indoor localization, and channel equalization. In this paper, we propose an adaptive channel estimation based on deep learning that assumes the signal-to-noise power ratio (SNR) knowledge at the receiver, and we show that the proposed scheme highly outperforms linear minimum mean square error based channel estimation in terms of normalized minimum square error, with similar order of online computational complexity. The proposed channel estimation scheme is also evaluated for an imperfect estimation of the SNR and showed to be robust for a high SNR estimation error.**

***Index Terms*—Adaptive channel estimation; Deep learning; Machine learning; LMMSE**

I. Introduction

In wireless communications, the signal propagates between the transmitter and the receiver in a multi-path noisy environment. In such scenarios, link adaptation schemes such as adaptive modulation and coding [1], and transmit power control are truly essential to maximize the throughput performance for wireless communications, especially when the channel and signal-to-noise ratio (SNR) changes with time.

On the other hand, accurate estimation of the channel state information (CSI) is very relevant in several applications such as multiple-input multiple-output (MIMO) based 5G applications [2], and CSI-based localization schemes [3], [4], where CSI is used instead of the received signal strength indication (RSSI) for localization purposes, since RSSI in multi-path noisy transmissions cannot capture the information of each single path. CSI is also used in analyzing and designing physical layer security schemes [5], [6], due to its critical role in determining eavesdropping knowledge. Therefore, it is very important to accurately estimate the CSI, in order to improve the overall performance of wireless communication systems in such applications.

Recently, deep learning (DL) has gained great success in the fields of computer vision, natural language processing, among others, and has been considered for application in wireless communications. The potential applications of DL in physical layers are discussed in [7]. One promising application is channel estimation. In this context, the authors in [8] propose a DL based channel estimation scheme that uses conventional estimation methods as an initial coarse estimation, and then a fine estimation is achieved by means of deep neural networks (DNN). In [9], the authors present a DL-based channel estimation receiver called ComNet, that replaces the existing orthogonal frequency division multiplexing (OFDM) receiver in wireless communications. The proposed scheme consists of two blocks: (i) channel estimation, where the least squares (LS) channel estimate is enhanced by a fully connected neural network; (ii) signal detection, where bi-directional long short-term memory neural network is used to detect bits from the received signal. ComNet receiver achieves a performance approaching the linear minimum mean squared error (LMMSE) with less computational complexity. The authors in [10] propose a DL-based channel estimation scheme, by using DNN. The proposed scheme requires several inputs to the DNN, and suffers from considerable complexity. In [11], the authors investigate several adaptive filters used to estimate the CSI coefficients of different MIMO-OFDM indoor and outdoor channel models. The investigated adaptive filters are based on least mean squares algorithm, which does not require the exact signal statistics knowledge, and gives an acceptable performance.

In this paper, we propose an adaptive channel estimation scheme using DNN. Several DNNs trained offline for different SNR values are stored in the memory of the receiver and used depending on the estimated SNR value. The proposed scheme outperforms both accurate and adaptive LMMSE channel estimation in terms of normalized mean-squared error (NMSE). This later is also pre-calculated for different SNR values, and achieves the same order of complexity as the proposed DNN-based scheme.

The rest of this paper is organized as follows. The system model as well as LMMSE channel estimation schemes are presented in Section II. Proposed adaptive DL-based channel estimation scheme is defined and explained in Section III. Experimental results followed by complexity analysis for accurate and imperfect SNR estimation are provided in Section IV. Finally, Section V concludes the paper.

II. LMMSE Channel estimation

This section sheds lights on the system model considered in our study, besides the mathematical representation of both accurate and adaptive LMMSE channel estimation schemes.

Authors acknowledge the CY Initiative for the support of the project through the ASIA Chair of Excellence Grant (PIA/ANR-16-IDEX-0008).

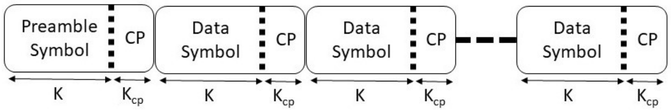

Fig. 1. OFDM Frame Structure.

A. System Description

We consider an OFDM transmitter that employs K subcarriers, and a cyclic prefix (CP) of length K_{cp} used to combat the inter symbol interference between two successive transmitted OFDM symbols. Each transmitted frame as shown in Fig. 1 consists of a preamble at the beginning, which is used for channel estimation at the receiver, followed by I OFDM symbols that contain the transmitted data. The input-output relation between the transmitted and the received OFDM frame at the receiver can be expressed as follows:

$$\boldsymbol{R}[k,i] = \tilde{\boldsymbol{H}}[k,i].\boldsymbol{S}[k,i] + \boldsymbol{N}[k,i], \tag{1}$$

where $\boldsymbol{S}[k,i]$, $\tilde{\boldsymbol{H}}[k,i]$, $\boldsymbol{R}[k,i]$, and $\boldsymbol{N}[k,i]$ denote the transmitted preamble symbol, the frequency domain channel gain, the received preamble symbol, and the noise symbol of the k-th sub-carrier of the i-th OFDM symbol, respectively. In general, OFDM transmitter uses $\mathcal{K}_{\text{on}}$ active sub-carriers out of K sub-carriers in each OFDM symbol. The other subcarriers are used as guard subcarriers. Assuming the channel is static during the transmission of the frame, the p-th preamble signal can be expressed as:

$$\tilde{\boldsymbol{r}}_p = \boldsymbol{S}_p.\tilde{\boldsymbol{h}}_p + \tilde{\boldsymbol{n}}_p \in \mathbb{C}^{|\mathcal{K}_{\text{on}}|\times 1}, \tag{2}$$

where $\tilde{\boldsymbol{r}}_p = \boldsymbol{R}[\mathcal{K}_{\text{on}},p]$, $\boldsymbol{S}_p = \text{diag}\left\{\boldsymbol{S}[\mathcal{K}_{\text{on}},p]\right\}$, and $\tilde{\boldsymbol{h}}_p = \tilde{\boldsymbol{H}}[\mathcal{K}_{\text{on}},p]$ is the frequency domain response of the channel at the preamble that needs to be estimated at the receiver.

B. Accurate LMMSE

In general, the LMMSE channel estimation is given by:

$$\hat{\tilde{\boldsymbol{h}}}_{p,\text{LMMSE}} = \boldsymbol{W}_{\text{LMMSE}}^{H}.\tilde{\boldsymbol{r}}_p, \tag{3}$$

where $\boldsymbol{W}_{\text{LMMSE}}^{H}$ denotes the LMMSE channel estimation matrix, which is expressed as:

$$\boldsymbol{W}_{\text{LMMSE}}^{H} = \boldsymbol{R}_{h_p}\left(\boldsymbol{R}_{h_p} + \boldsymbol{S}_p^{-1}\boldsymbol{R}_{\tilde{n}_p}\boldsymbol{S}_p^{-H}\right)^{-1}\boldsymbol{S}_p^{-1}. \tag{4}$$

Here, $\boldsymbol{R}_{h_p} = \text{E}\left[\tilde{\boldsymbol{h}}_p\tilde{\boldsymbol{h}}_p^{H}\right] \in \mathbb{C}^{|\mathcal{K}_{\text{on}}|\times|\mathcal{K}_{\text{on}}|}$, is the channel autocorrelation matrix. Assuming uncorrelated noise, $\boldsymbol{R}_{\tilde{n}_p} = \text{E}\left[\tilde{\boldsymbol{n}}_p\tilde{\boldsymbol{n}}_p^{H}\right] = \mathcal{N}_0 K\boldsymbol{I}_{|\mathcal{K}_{\text{on}}|}$. Knowing that, the transmitted preamble fulfils $\boldsymbol{S}_p^{H}\boldsymbol{S}_p = E_p\boldsymbol{I}$, where E_p denotes the preamble effective power, $\boldsymbol{W}_{\text{LMMSE}}^{H}$ can be expressed as:

$$\boldsymbol{W}_{\text{LMMSE}}^{H} = \boldsymbol{R}_{h_p}\left(\boldsymbol{R}_{h_p} + \frac{1}{\gamma}\boldsymbol{I}\right)^{-1}\boldsymbol{s}_p^{-1},\ \gamma = \frac{E_p}{K\mathcal{N}_0}. \tag{5}$$

In our study, we refer to $\boldsymbol{W}_{\text{LMMSE}}^{H}$ as the accurate LMMSE. In conventional approaches, $\boldsymbol{W}_{\text{LMMSE}}^{H}$ is computed online using the estimation of $\boldsymbol{R}_{h_p}$ and the SNR value γ.

C. Adaptive LMMSE

In adaptive LMMSE, $\boldsymbol{W}_{\text{LMMSE}}^{H}$ is computed offline from the statistical knowledge of the channel, i.e. $\boldsymbol{R}_{h_p}$, and for several SNR values defined by γ_u. The mapping $\eta \rightarrow \boldsymbol{W}_{\text{LMMSE}}^{H}[u]$, with:

$$\boldsymbol{W}_{\text{LMMSE}}^{H}[u] = \boldsymbol{R}_{h_p}\left(\boldsymbol{R}_{h_p} + \frac{1}{\gamma_u}\boldsymbol{I}\right)^{-1}\boldsymbol{S}_p^{-1}, \tag{6}$$

is used at the receiver to decide which matrix to be used for channel estimation. When $\eta = \eta_u$, the selected matrix provides accurate LMMSE estimation. Therefore, the estimated channel is given by:

$$\hat{\tilde{\boldsymbol{h}}}_{p,u} = \boldsymbol{W}_{\text{LMMSE}}^{H}[u]\tilde{\boldsymbol{r}}_p. \tag{7}$$

Adaptive LMMSE is computationally less complex than accurate LMMSE, as it does not need to calculate $\boldsymbol{W}_{\text{LMMSE}}^{H}$ each time the SNR changes.

III. Adaptive channel estimation based on deep learning

In this section, the DNN main principles are first presented. After that, the concept and architecture of the proposed adaptive DL-based channel estimation scheme are illustrated.

A. DNN Main Principles

DNNs are computational models consisting of many simple processing units called neurons, that work in parallel and are arranged in interconnected layers. Simple neural networks consist of an input layer and an output layer, when more layers are stacked, the networks are called deep [12]. A DNN learns to perform particular tasks through training, during which the strength of connections between units is learned. Subsequently, the trained DNN is used to perform the same task on novel inputs [13].

Let L be the number of hidden layers within the DNN, with J_l neurons for each layer l, where $1 \leq l \leq L$. $\boldsymbol{W}_l \in \mathbb{R}^{J_{l-1}\times J_l}$, and $\boldsymbol{b}_l \in \mathbb{R}^{J_l\times 1}$ are used to denote the weight and the bias matrices of the l-th hidden layer respectively. The DNN J_0 inputs are organized in a real-valued vector $\boldsymbol{x}_{(1)} \in \mathbb{R}^{J_0\times 1}$, fed to the DNN input layer. Similarly, the DNN J_{L+1} outputs are stacked in vector $\boldsymbol{y} \in \mathbb{R}^{J_{L+1}\times 1}$. The weight matrices for the input and output layers are denoted as $\boldsymbol{W}_1 \in \mathbb{R}^{J_0\times J_1}$ and $\boldsymbol{W}_{L+1} \in \mathbb{R}^{J_L\times J_{L+1}}$ respectively. Each neurons (l,j) performs a linear transformation represented by the activation function $f_{(l,j)}$ on the neuron's input $\boldsymbol{x}_{(l)} \in \mathbb{R}^{J_{l-1}\times 1}$ using its weight $\boldsymbol{\omega}_{(l,j)} \in \mathbb{R}^{J_{l-1}\times 1}$, and bias $b_{(l,j)}$ respectively, where $1 \leq j \leq J_l$. The neuron's output $y_{(l,j)}$ is

$$y_{(l,j)} = f_{(l,j)}\left(b_{(l,j)} + \boldsymbol{\omega}_{(l,j)}^{T}\boldsymbol{x}_{(l)}\right). \tag{8}$$

The overall output the DNN l-th hidden layer is represented by the vector form

$$\boldsymbol{y}_{(l)} = \boldsymbol{f}_{(l)}\left(\boldsymbol{b}_{(l)} + \boldsymbol{W}_{(l)}.\boldsymbol{x}_{(l)}\right),\ \boldsymbol{x}_{(l+1)} = \boldsymbol{y}_{(l)}, \tag{9}$$

where $\boldsymbol{y}_{(l)} \in \mathbb{R}^{J_l\times 1}$ is the output of the l-th layer, $\boldsymbol{f}_{(l)}$ is a vector that results from the stacking of the J_l activation functions.

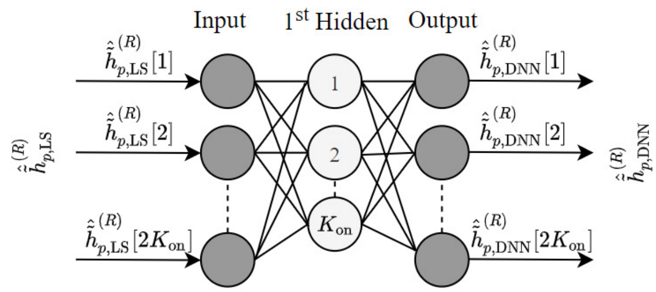

Fig. 2. DNN Architecture.

In general, DNN works in two phases: (i) training phase, where the DNN is given inputs-outputs data pairs, and it will try to learn the mapping relation between them; (ii) testing phase, where the DNN is fed by new unseen input data, and it will try to predict the relative output of this data. The DNN learning ability is mainly related to how DNN weights are adjusted during the training phase. Weight adjustment process can be done by applying two consecutive operations. First, the DNN applies what is known by forward propagation, by which the output $\boldsymbol{y}_{(l)}$ from each hidden layer moves as an input to the next hidden layer, and the same process is repeated until the DNN final output is obtained (9). Through forward propagation [14], the DNN approximates the DNN neurons weights in a way that DNN inputs and outputs map each others. The difference between the true output $\boldsymbol{y}^{(\mathrm{T})}$ and the predicted DNN output $\boldsymbol{y}^{(\mathrm{P})}$ can be represented by a cost function $J_{\boldsymbol{W},\boldsymbol{B}}\big(y^{(\mathrm{P})}_{(l,j)}, y^{(\mathrm{T})}_{(l,j)}\big)$ needed to be minimized. Backward propagation [15] is one of the methods used to minimize the cost function in DNN, through applying some optimizers.

These optimizers will minimize $J_{\boldsymbol{W},\boldsymbol{B}}\big(y^{(\mathrm{P})}_{(l,j)}, y^{(\mathrm{T})}_{(l,j)}\big)$ by iteratively updating $\boldsymbol{W}$ and $\boldsymbol{B}$ values during the DNN training phase. There are several optimizers that could be employed like stochastic gradient descent [13], root mean square prop [16], and adaptive moment estimation (ADAM) [17].

Updating $\boldsymbol{W}$ and $\boldsymbol{B}$ values is performed by calculating the derivative of $J_{\boldsymbol{W},\boldsymbol{B}}$ with respect to each neuron weight $\boldsymbol{\omega}_{(l,j)}$, and then each neuron's weight is updated according to the following updating rule

$$\boldsymbol{\omega}^{\text{new}}_{(l,j)} = \boldsymbol{\omega}_{(l,j)} - \rho . \frac{\partial J_{\boldsymbol{W},\boldsymbol{B}}}{\partial \boldsymbol{\omega}_{(l,j)}}, \tag{10}$$

where ρ represents the learning rate of the DNN, which controls how quickly the DNN model is adapted to the problem. Smaller learning rates require more training, given the smaller changes made to the weights in each update, whereas larger learning rates result in rapid changes and require fewer training.

B. Proposed DL-based Adaptive Channel Estimation Scheme

The proposed DNN channel estimation scheme depends mainly on LS channel estimates of $\tilde{\boldsymbol{h}}_p$ from the received preamble, where DNN is employed as an additional module besides the conventional LS channel estimation scheme. Combining both conventional channel estimation schemes with DNN will significantly improve the overall performance with a considerable decrease in the computational complexity as we will discuss in section IV.

TABLE I
PROPOSED DNN PARAMETERS.

Parameter	**Values**
Hidden layers	1
Number of neurons	52
Activation function	ReLU
Number of epochs	500
Batch size	32
Training Samples	8000
Testing Samples	2000
Optimizer	ADAM
Loss function	MSE
Learning rate	0.001
Training SNR	30 dB

Considering that $\boldsymbol{S}_p$ is invertible, the LS channel gain estimation at the k-th sub-carrier of $\tilde{\boldsymbol{h}}_p$ can be expressed as

$$\hat{\tilde{\boldsymbol{h}}}_{p,\text{LS}}[k] = \frac{\boldsymbol{r}_p[k]}{\boldsymbol{S}_p[k]}. \tag{11}$$

First of all, $\hat{\tilde{\boldsymbol{h}}}_{p,\text{LS}}$ is converted from complex to real-valued domain by arranging the real and imaginary values of $\hat{\tilde{\boldsymbol{h}}}_{p,\text{LS}}$ in one vector $\hat{\tilde{\boldsymbol{h}}}^{(R)}_{p,\text{LS}} \in \mathbb{R}^{2\mathcal{K}_{\text{on}}\times 1}$. After that, these inputs are normalized to have a zero mean and unit variance, since differences in the scales across input variables may increase the difficulty of the training in the problem being modeled. Finally, the normalized LS estimated channel is fed as an input to the DNN as shown in Fig 2.

The proposed DNN is trained using the parameters illustrated in Table I, and aims to learn the error of the LS channel estimation in order to correct $\hat{\tilde{\boldsymbol{h}}}^{(R)}_{p,\text{LS}}$ by minimizing a cost function $J_{\boldsymbol{W},\boldsymbol{B}}\big(\tilde{\boldsymbol{h}}^{(R)}_p, \hat{\tilde{\boldsymbol{h}}}^{(R)}_{p,\text{DNN}}\big)$, where $\hat{\tilde{\boldsymbol{h}}}^{(R)}_{p,\text{DNN}}$ is the output of the DNN when the input is the LS estimated channel $\hat{\tilde{\boldsymbol{h}}}^{(R)}_{p,\text{LS}}$, and $\tilde{\boldsymbol{h}}^{(R)}_p \in \mathbb{R}^{2\mathcal{K}_{\text{on}}\times 1}$ represents the real and imaginary values of the ideal channel.

At the end of the training, the corrected LS channel estimate $\hat{\tilde{\boldsymbol{h}}}^{(R)}_{p,\text{DNN}}$ is processed again to get back $|\mathcal{K}_{\text{on}}|$ complex valued records. The complex valued DNN output is denoted by $\hat{\tilde{\boldsymbol{h}}}_{p,\text{DNN}}$ such that $\tilde{\boldsymbol{h}}_{p,\text{DNN}} \in \mathbb{C}^{\mathcal{K}_{\text{on}}\times 1}$. It is worth mentioning that at the end of the training phase, we only save the weights of the epoch having the highest training and validation accuracy, instead of averaging the weights over all the epochs. Moreover, the DNN training is performed using SNR = 30dB to achieve the best performance as illustrated in [18], due to the fact that when the training is performed for a high SNR value, the DNN is able to better learn the channel statistics, and due to its good generalization DNN ability, it can still perform well

TABLE II
CHANNEL MODEL.

Discrete delay (ns)	0	1	2	100	101	102	200	201	300	301	400	401
Average path gain (dB)	0	0	0	-9.3	-9.3	-9.3	-20.3	-20.3	-21.3	-21.3	-28.8	-28.8

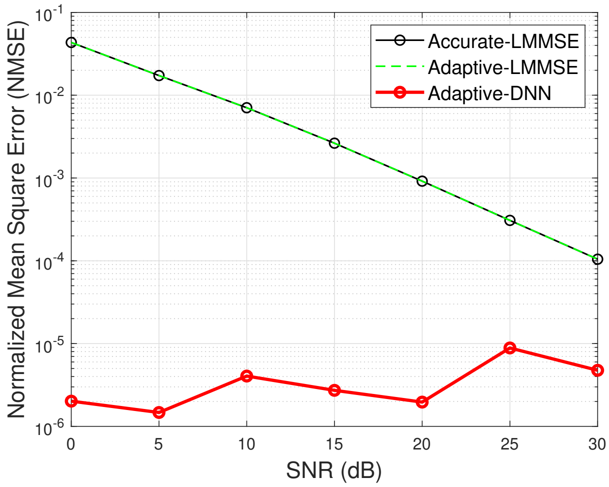

Fig. 3. NMSE Performance for Perfect SNR Estimation.

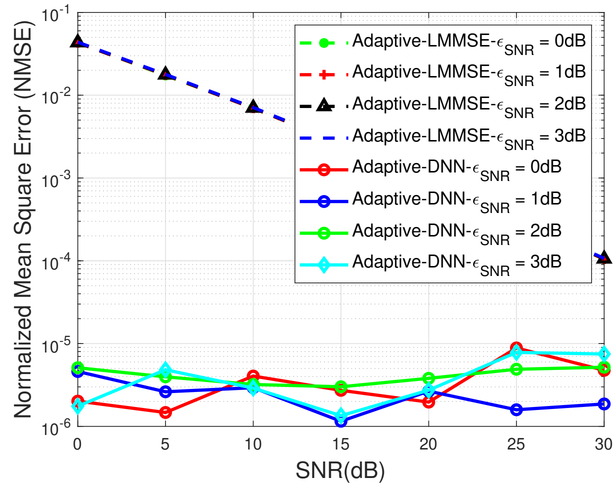

Fig. 4. NMSE Performance for Imperfect SNR Estimation.

in low SNR regions, where the noise is dominant. It is worth mentioning that the training is performed offline which does not increase the online computational complexity. Moreover, intensive experiments are performed using the grid search algorithm [19] in order to select the best suitable DNN hyper parameters in terms of both performance and complexity.

The proposed adaptive DNN channel estimation scheme assumes that the SNR estimation is available at the receiver. Based on this assumption, several DNNs are trained offline, each for different SNR value. According to the estimated SNR value, DNN selection will be performed to process the received signals. Moreover, using the SNR estimation error term

$$\epsilon_{\text{SNR}} = \mid \text{SNR}_{\text{accurate}} - \text{SNR}_{\text{estimated}} \mid, \quad (12)$$

the proposed DNN works in two different scenarios: (i) Perfect SNR knowledge, where $\epsilon_{\text{SNR}} = 0$. This means that the SNR used in the training is equal to the exact received SNR; (ii) Imperfect SNR knowledge, where $\epsilon_{\text{SNR}} > 0$. In this case, the estimated SNR deviates from the accurate SNR value. Thus, in order to take into consideration the estimation errors of the SNR value, the DNN will be trained on different SNR values within the interval $[\text{SNR} - \epsilon_{\text{SNR}}, \text{SNR} + \epsilon_{\text{SNR}}]$.

IV. SIMULATION RESULTS

In this section, NMSE is used to evaluate the performance of the proposed adaptive DL-based channel estimation scheme versus accurate and adaptive LMMSE channel estimation schemes in two different scenarios as defined in the previous section. After that, a computational complexity between the studied channel estimation schemes is carried on. The channel delay and power profile of the channel used in our simulations are defined in Table II. The total number of sub-carriers for an OFDM symbol is equal to $K = 64$ where only $|\mathcal{K}_{\text{on}}| = 52$ sub-carriers are activated.

A. NMSE for Perfect SNR knowledge

In the first scenario, we assume a perfect SNR knowledge at the receiver. We have trained seven neural networks each on a training dataset generated using a fixed SNR value from the set $\{0, 5, \ldots, 30\}$dB. It is clearly shown from Fig. 3 that adaptive DNN highly outperforms both accurate and adaptive LMMSE, which reveals that the adaptive DNN channel estimation scheme was able to overcome the presence of the noise.

B. NMSE for Imperfect SNR Knowledge

In the second scenario, channel estimation is performed assuming that the SNR estimation at the receiver includes some errors. In order to consider this issue in the training phase of the DNN, each DNN is now trained on a data set generated by choosing uniformly an SNR value within the interval $I_{\text{SNR}} = [\text{SNR} - \epsilon_{\text{SNR}}, \text{SNR} + \epsilon_{\text{SNR}}]$ with a step of 0.1. In these simulations, we consider: $\epsilon_{\text{SNR}} \in \{1\text{dB}, 2\text{dB}, 3\text{dB}\}$. For example, if the estimated SNR is equal to 5dB, but the actual value of the SNR belongs to the interval $I_5 = [5 - \epsilon_{\text{SNR}}, 5 + \epsilon_{\text{SNR}}]$, the DNN trained on this SNR interval will be used for channel estimation. The accuracy shown for SNR=5dB corresponds to averaging all the samples where the estimated SNR is 5dB and the actual SNR belongs to I_5.

The generated dataset is divided into 80% training data and 20% testing data.

We can observe from Fig. 4, that adaptive DNN performance varies with different SNR error intervals. It is difficult to get a predictable behaviour of the DNN depending on the value of ϵ_{SNR}. In fact, when the DNN is trained with different SNR values within an interval I_{SNR}, it is supposed to learn better in high SNR values within this interval (where there is less noise) compared to low SNR values within the same interval (where the noise is more dominant).

It is worth mentioning that the fluctuations in the DNN NMSE curves in Fig. 3 and 4 are related to the ability of the DNN in learning the channel statistics. There is no accurate prediction of the exact DNN behaviour, since its performance depends on the SNR value and the hyper parameters used in the training process. However, integrating the proposed DNN architecture in the channel estimation process leads to significant performance improvement, even though adaptive DNN performance varies for different ϵ_{SNR}, it still outperforms both accurate and adaptive LMMSE channel estimation schemes, even if LMMSE is more robust to SNR estimation errors.

C. Computational complexity analysis

In DNN, a matrix multiplication is computed for each transition between the l-th and $(l-1)$-th layers. Thus, the online DNN computational complexity is represented by the total number of real-valued multiplications required in the DNN network:

$$N_{\text{mul}} = \sum_{l=1}^{L+1} J_{l-1} J_l, \;\; J_0 = 2|\mathcal{K}_{\text{on}}|, \;\; J_{L+1} = 2|\mathcal{K}_{\text{on}}|. \tag{13}$$

As mentioned in Section III, the proposed adaptive DNN based channel estimation scheme consists of one hidden layer ($L = 1$) having $|\mathcal{K}_{\text{on}}|$ neurons, therefore the number of real-valued multiplications for adaptive DNN is $(4|\mathcal{K}_{\text{on}}|^2)$. The computational complexity of accurate LMMSE channel estimation scheme is of order $|\mathcal{K}_{\text{on}}|^3$ complex multiplications [20], which is equivalent to $4|\mathcal{K}_{\text{on}}|^3$ real-valued multiplications. On the other hand, for adaptive LMMSE assuming the SNR knowledge at the receiver, thus pre-calculating LMMSE estimation matrix, makes the number of real-valued multiplications for adaptive LMMSE equal to those needed for the adaptive DNN. However, the accuracy of the adaptive DNN is still appealing, and outperforming the performance of both accurate and adaptive LMMSE, even in imperfect SNR estimation scenarios.

V. Conclusion

This paper presents a novel adaptive DNN channel estimation scheme, based on the estimation of the SNR value at the receiver. First, classical channel estimation schemes illustrated by accurate and adaptive LMMSE have been surveyed. After that, a brief DNN introduction, followed by presentation of the proposed adaptive DNN channel estimation scheme is discussed. Unlike accurate and adaptive LMMSE channel estimation schemes, the proposed scheme does not require specific channel statistics knowledge, making it more suitable to real case scenarios. Simulation results reveal that the proposed adaptive DNN channel estimation scheme outperforms both accurate and adaptive LMMSE channel estimation in both perfect and imperfect SNR estimation scenarios, with similar computational complexity.

References

[1] M. A. Azza, M. El Yahyaoui, and A. El Moussati, “Throughput Performance of Adaptive Modulation and Coding Schemes for WPAN Transceiver,” in *2018 International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, Nov 2018, pp. 1–4.

[2] S. Ehsanfar, M. Matthé, M. Chafii, and G. P. Fettweis, “Pilot- and cp-aided channel estimation in mimo non-orthogonal multi-carriers,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 650–664, 2019.

[3] W. Kui, S. Mao, X. Hei, and F. Li, “Towards Accurate Indoor Localization Using Channel State Information,” in *2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, May 2018, pp. 1–2.

[4] S. Abdul Samadh, Q. Liu, X. Liu, N. Ghourchian, and M. Allegue, “Indoor Localization Based on Channel State Information,” in *2019 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet)*, Jan 2019, pp. 1–4.

[5] J. M. Hamamreh, H. M. Furqan, Z. Ali, and G. A. S. Sidhu, “An Efficient Security Method Based on Exploiting Channel State Information (CSI),” in *2017 International Conference on Frontiers of Information Technology (FIT)*, Dec 2017, pp. 288–293.

[6] S. Chetry and A. Singh, “Physical Layer Security of Outdated CSI Based CRN,” in *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, July 2018, pp. 1–5.

[7] T. O’Shea and J. Hoydis, “An introduction to deep learning for the physical layer,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, Dec 2017.

[8] A. K. Gizzini, M. Chafii, A. Nimr, and G. Fettweis, “Deep learning based channel estimation schemes for ieee 802.11p standard,” *IEEE Access*, vol. 8, pp. 113 751–113 765, 2020.

[9] X. Gao, S. Jin, C. Wen, and G. Y. Li, “ComNet: Combination of Deep Learning and Expert Knowledge in OFDM Receivers,” *IEEE Communications Letters*, vol. 22, no. 12, pp. 2627–2630, Dec 2018.

[10] Y. Yang, F. Gao, X. Ma, and S. Zhang, “Deep Learning-Based Channel Estimation for Doubly Selective Fading Channels,” *IEEE Access*, vol. 7, pp. 36 579–36 589, 2019.

[11] M. Ghosh, C. Srinivasarao, and H. K. Sahoo, “Adaptive Channel Estimation in MIMO-OFDM for Indoor and Outdoor Environments,” in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, March 2017, pp. 2743–2747.

[12] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. [Online]. Available: https://doi.org/10.1038/nature14539

[13] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Networks*, vol. 61, p. 85–117, Jan 2015.

[14] G. Xie and J.-H. Lai, “An interpretation of forward-propagation and back-propagation of dnn,” in *PRCV*, 2018.

[15] H. Li, R. Zhao, and X. Wang, “Highly efficient forward and backward propagation of convolutional neural networks for pixelwise classification,” 2014.

[16] S. ichi Amari, “Backpropagation and stochastic gradient descent method,” *Neurocomputing*, vol. 5, no. 4, pp. 185 – 196, 1993.

[17] S. De, A. Mukherjee, and E. Ullah, “Convergence guarantees for rmsprop and adam in non-convex optimization and an empirical comparison to nesterov acceleration,” 2018.

[18] A. K. Gizzini, M. Chafii, A. Nimr, and G. Fettweis, “Enhancing least square channel estimation using deep learning,” in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5.

[19] F. Pontes, G. Amorim, P. Balestrassi, A. Paiva, and J. Ferreira, “Design of experiments and focused grid search for neural network parameter optimization,” *Neurocomputing*, vol. 186, pp. 22 – 34, 2016.

[20] V. Savaux and Y. Louët, “LMMSE Channel Estimation in OFDM Context: A Review,” *IET Signal Processing*, vol. 11, no. 2, pp. 123–134, 2017.