Deep Multi-Task Learning-Based Simultaneous Channel Tap and Coefficient Estimation

Mehmet Ali Aygül*†, Hakan Ali Çırpan*, Hüseyin Arslan‡

*Department of Electronics and Communications Engineering, Istanbul Technical University, Istanbul, 34467 Turkey

†Department of Research & Development, Vestel, Manisa, 45030 Turkey

‡Department of Electrical and Electronics Engineering, Istanbul Medipol University, Istanbul, 34810 Turkey

E-mails: {aygul21, cirpanh}@itu.edu.tr, huseyinarslan@medipol.edu.tr

2023 IEEE Future Networks World Forum (FNWF) | 979-8-3503-2458-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/FNWF58287.2023.10520577

***Abstract*—Wireless communication systems depend on accurate channel estimation to ensure efficient and reliable data transmission. The channel estimation process consists of two essential steps: channel tap and coefficient estimation. Physical layer features such as time arrival, and signal strengths are well used for the tap estimation. However, prior knowledge is required to use these methods. Recently, machine learning-based methods have been proposed. In particular, deep learning (DL)-based methods are promising because they can learn from raw data without much preprocessing, scale well with extensive and diverse datasets, and capture complex relationships. However, these methods overlook the relationship between the channel taps and coefficients. In this paper, we propose a DL-based multi-task learning method to estimate channel taps and coefficients simultaneously. Simulation results reveal that the performance of the proposed tap estimation method is superior to the traditional DL-based tap estimation. Furthermore, the proposed method removes the need to train two models to estimate channel taps and coefficients.**

***Index Terms*—Channel coefficients, channel tap estimation, deep learning, multi-task learning, wireless channel.**

I. Introduction

Wireless communication systems rely on appropriate channel estimation that accurately captures real-world conditions. Extraction of patterns and other characteristics from the channel estimation, enabling insights into how wireless signals are transmitted in complex environments. This is essential for wireless communication systems' reliable and efficient data transmission. Also, precise channel estimation supports network design, signal processing scheduling, and error correction methods and validates them, enabling collaboration and communication.

Channel estimation can be considered as two parts. These are the number of channel taps and coefficient estimations, respectively. Determining the optimal number of channel taps is particularly significant in wireless channel estimation since it is the first step when estimating the wireless channel [1]. Estimating channel taps from transmitted and received signals, without relying on pre-assumed scenarios, offers a promising approach. This helps infer channel characteristics and adapt communication strategies in unknown and dynamic environments.

There are several channel tap estimation methods in the literature. In [2], various channel characteristics are extracted based on signal strength and arrival time, such as delay spread, time delay, number of multipath, etc. [3] extracted the number of multipath, delay spread, and time delay from arrival time and signal strength information. However, these methods rely on prior assumptions. [4] proposes a channel coding type blind recognition method based on a cyclic neural network. In this work, the characteristics of the received related sequences are extracted by adequately utilizing the cyclic neural network. Then, a long string of sequences for segmented recognition is divided, and the final decision on the sequences is made using the principle of the minority obeying the majority. However, this method requires overheads, such as dividing long strings of sequences for segmented recognition and making the final decision on the sequences by using the principle of the minority obeying the majority. Also, identifying channel tap numbers cannot be achieved without recognizing coded sequences, and it cannot be considered an efficient maximally sparse representation.

To make blind estimation and eliminate the drawbacks of [4], a machine learning (ML)-based method is proposed [5]. In this paper, specifically, a deep neural network is used with the inputs as transmitted and received signals' samples and the outputs as the number of wireless channel taps. However, this paper overlooks the existing relationship between channel coefficients and taps. On the other hand, the relationship lies in coefficients defining how signal strength and phase change across taps, aiding tap estimation. Conversely, tap information helps extract coefficients by separating paths. Together, they enable accurate channel modeling, enhancing wireless system design through effective equalization, interference mitigation, and modulation strategies. Therefore, simultaneous estimation of the number of channel taps and its coefficient estimation is a promising approach. However, simultaneous estimation is a complex problem, so it is difficult to have an accurate estimation performance.

It is intuitively sound to think of using a deep learning (DL) method to solve complex problems [6]. Particularly, DL-based multi-task learning methods [7], [8] can address complex simultaneous estimation problems thanks to their ability to employ relationships between related tasks through their hidden layers. Different from the existing works, this paper proposes to estimate the number of taps, and channel coefficients simultaneously. A DL-based multi-task learning method is designed for this purpose. Simulation results reveal a performance improvement achieved by the proposed method compared to traditional DL-based channel tap estimation regarding classification accuracy. Moreover, instead of training a DL model for the number of channel tap estimation and a DL model for channel coefficients estimation, a single DL model is used in the proposed method.

This paper is organized as follows. Section II revises the preliminaries and presents the system model. The proposed method for the number of channel taps and channel coefficients estimation is detailed in Section III. Section IV presents experiments conducted to evaluate the performance of the proposed method. Finally, the paper is concluded in Section V.

II. Preliminaries

A. System Model

Digital symbols at the transmitter are transmitted as [9]

$$x(t) = \sum_{k} p_k f(t - kT), \tag{1}$$

where T stands for the symbol period, $f(\tau)$ is the impulse response of the transmitter filter expressed as a delay function (τ), and p_k represents the symbol period. Discrete filter taps is used to represent the radio channel through which the broadcast signal travels. Thus, the received signal is expressed as follows when there is noise.

$$y(t) = \sum_{l=0}^{L-1} c(l) x(t - \tau(l)) + n(t), \tag{2}$$

where L stands for the number of channel taps, $c(l)$ stands for the lth complex channel coefficient, and $\tau(l)$ stands for the delay. The delays should be uniformly spaced according to the formula $\tau(l) = lT/W$, where W is an integer. For symbol-spaced channel modelling and fractionally-spaced channel modelling, W is often set to 1 and 2, respectively. The white complex Gaussian noise used to simulate the noise term, $n(t)$, is described.

At the receiver, the received signal is filtered by a filter that is matched to the pulse shape. Then, it sampled with the sampling period T_s, as follows [9].

$$r_k = \int f^*(\tau) y(\tau + kT_s) d\tau, \tag{3}$$

where superscript '$*$' denotes complex conjugate. Afterward, by replacing (2) into (3), the received signal samples can be denoted as

$$r_k = \sum_{j=0}^{J-1} h(j) b_{k-j} + z_k, \tag{4}$$

where $h(j)$ signifies jth composite channel coefficient, that follows a Rayleigh distribution.

B. Deep Learning

ML algorithms are successfully used in various applications, such as image processing and pattern recognition. This is inspired to apply them to wireless communication [10]. Then, these techniques became fundamental components of wireless communication systems for 5G and beyond [11]. Particularly, DL-based methods have become popular since using multiple hidden layers of DL methods allows for magnifying the intrinsic data features while suppressing the irrelevant information at each layer [12]. This is particularly true for complex problems, such as multiple problems that are tried to estimate simultaneously. In addition, raw data can be used without specific feature engineering in these methods, thanks to the hidden layers of the DL methods.

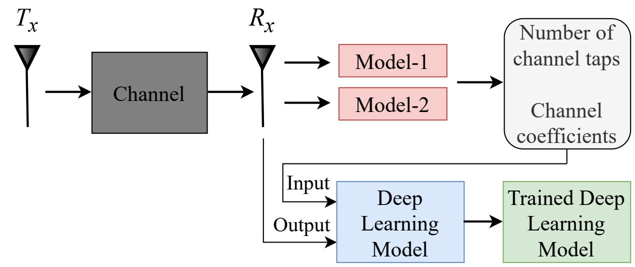
III. The Proposed Method

DL-based methods are widely used to solve complex problems [13]. However, in these methods, optimization is based on a specific metric, such as a score on a particular benchmark and a business key performance indicator. While focusing intensely on a single task obtains acceptable results, it may not be the optimum. On the other hand, simultaneously attempting to optimize multiple problems may result in an improvement since multiple problems can have correlations between them. Therefore, sharing representations between related tasks may improve the estimation performance of the original problem.

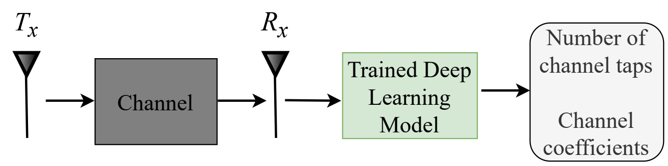
The number of channel taps and coefficients are strongly related to each other. The coefficients define how signal strength and phase change across taps, and a number of taps extract coefficients by separating paths. Therefore, they can be simultaneously estimated with the help of multi-task learning. Multi-task learning methods can gather helpful data from numerous related tasks to enhance individual estimates [14]. This is particularly true when using the DL-based multi-task learning method with multiple hidden layers. Along with this line, a DL-based multi-task learning method is proposed in this paper to estimate the number of channel taps and coefficients simultaneously.

The proposed method based on DL-based multi-task learning consists of two phases. First, the proposed method is performed in the training phase, where the dataset is generated, and a DL method is configured and trained. Then, it is performed in the testing phase, where the number of channel taps and coefficients are estimated simultaneously.

In training, transmitted signals (T_x) are transmitted via a wireless communication channel. Afterward, the receiver captures the received signals (R_x). The T_x and R_x signals are used to estimate channel taps and coefficients with the classical estimation models [9], [15] (a model for number of tap estimation and a model for coefficients estimation). Then, these estimated values are stored as output, and the input is stored as the received signals from which these values were obtained. Until a sufficient dataset is generated, these operations are repeated. The dataset size is chosen according to system requirements for the best performance, complexity,



(a)



(b)

Fig. 1: The proposed method for the number of channel taps and coefficients estimation; (a) training phase and (b) testing phase.

and memory usage. Afterward, the DL method[1] is trained with the created dataset. These processes are demonstrated in Fig. 1 (a). In addition, an example of the DL-based multi-task learning method is illustrated in Fig. 2[2]. Once the training and validation[3] loss convergence is done in the training phase, and the testing phase starts, which characterizes the run-time operation of the method.

In the testing phase, a signal passed over from the wireless channel is captured in the receiver. Afterward, this received signal is fed to the trained DL method. Then, the trained DL method simultaneously estimates the number of channel taps and coefficients. These processes are demonstrated in Fig. 1 (b). Also, Algorithm 1 provides the general operations of the proposed method.

A. Discussions on Computational Complexity

The suggested method's computing complexity is determined by the training and testing phases. The cost of the training phase depends on both model-based estimations and the DL approach, whereas the complexity of the testing phase depends on the DL technique.

A DL-based multi-task learning method is used in this work with an input layer, four hidden layers, and an output layer. This method has a units in the input layer, where a denotes the size of the input vector. Besides, it has b hidden units for simultaneous learning. Also, it has c, d, and e hidden units and f output units for the number of channel taps and f output units for channel coefficients. Thus, the

[1]All of the DL methods hyperparameters are empirically adjusted while considering the proposed methods' performance and generalizability.

[2]The hyperparameters used in this figure (such as number of hidden layers and units) are detailed in the next section.

[3]In the context of ML, the validation dataset often are used as a neutral assessment of a model's fit to the training dataset [16].

Algorithm 1 Estimating the number of channel taps and coefficients.

Input: S number of received signals for training (R_{xtrain}), N number of received signals for validation ($R_{xvalidation}$), initial hyperparameters, traditional methods to estimate number of channel taps and coefficients, and received signals for testing (R_{xtest}).

Output: Estimated number of channel taps and coefficients (E_{test}).

Training Phase:

for $s = 1$ **to** S **do**
  Receive R_{xtrain}.
  Traditional methods estimate number of channel taps and coefficients (E_{train}).
  A new data point R_{xtrain} and E_{train} is added to the training dataset (D).
end for
Train the DL method using the generated dataset D.
while Convergence of the training and validation loss graphs are achieved **do**
  Adjust the DL method's hyperparameters based on loss graphs.
  Train the DL method using the generated dataset D.
end while

Testing Phase:

Receive R_{xtest}.
Estimate number of channel taps and coefficients E_{test} using R_{xtest} and trained DL method.

overall training computational complexity of this method is $\mathcal{O}(ml \times (ab + 2(bc) + 2(cd) + 2(de) + 2(e + ef)))$, where m and l denotes the number of epochs and training examples, respectively. In addition, the computational complexity of the validation and the number of trials to select optimum hyperparameters of the DL method is added to the training complexity. Here note that the amount of tests and validation data relies on the application's complexity and reliability requirements. Since the testing phase does not require back-propagation, the computational complexity per sample is around half that of the training phase [17].

IV. Simulation Results

Illustrative simulations are conducted to demonstrate the performance of the number of channel taps estimation. Binary phase-shift keying is used as a modulation technique. All the simulation samples consider 1000 number of training pilot symbols. As a channel model, Rayleigh is used in which the number of channel taps is between $L = 1$ to $L = 10$ with a step size of 1 and $\boldsymbol{E}[|\boldsymbol{h}|^2] = 1$. The noises are modeled by $\mathcal{CN}(0, \sigma_N^2)$, i.e., zero-mean complex Gaussian samples with variance σ_N^2.

Dataset[4] is generated by MATLAB simulation environment. The dataset includes three phases: training, validation, and

[4]Dataset will be publicly available online after the acceptance.

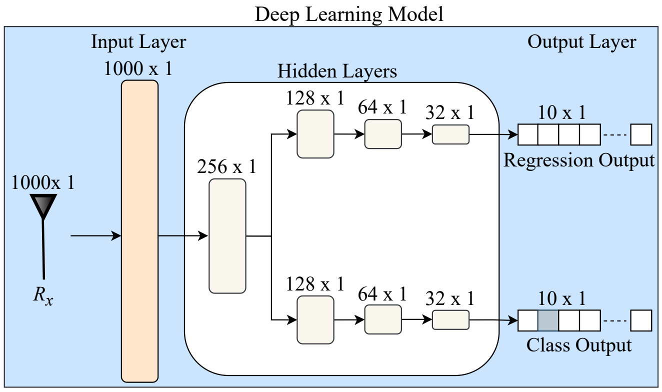


Fig. 2: Illustration of the DL model.

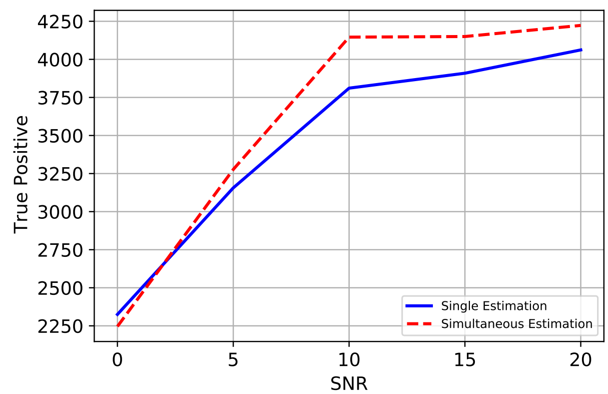


Fig. 3: True positive results for single and simultaneous estimation.

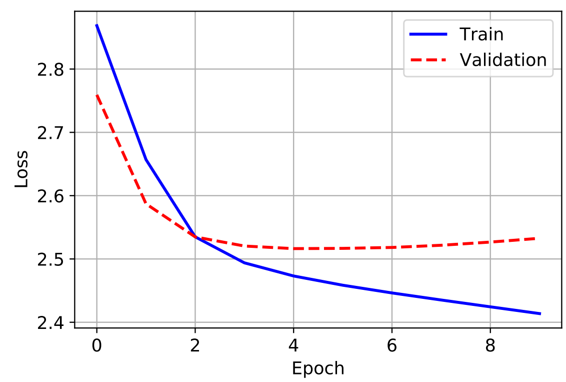


Fig. 4: The loss graph for the proposed method.

testing. In all phases, SNR varies from 0 to 20 with a step size of 5. In addition, 60000, 20000, and 20000 samples are generated for each SNR value in the training, validation, and testing phases. Therefore, in total, 300000, 100000, and 100000 samples are generated for the training, validation, and testing phases.

The open-source ML library Keras [18], which operates in Python, implements the proposed DL method. All of the simulations are made on an MSI computer with a Windows 10 operating system, an Intel® Core™ i7-7700HQ central processing unit (CPU), a GeForce GTX 1050 Ti graphics processing unit (GPU), and 16 GB RAM.

An input layer, four hidden layers, and an output layer are used in the proposed method. Specifically, 1000 units are used in the input layer. Afterward, simultaneous learning of the number of channel taps and coefficients is made in the first hidden layer with 256 units to learn the relationship between them. Then, three hidden layers are used to learn the number of channel taps and coefficients. In these layers, 128, 64, and 32 units are used, respectively. Afterward, in the output layer, 10 units are used for the number of channel taps estimation, and 10 units (unit per maximum number of taps) are used for channel coefficients. Note that these hyperparameters' usages are illustrated in Fig. 2. In all of the layers, the rectified linear unit is used as an activation function. However, since the number of channel tap estimation is a classification problem, the softmax activation function is used in the output layer to estimate the number of taps. The DL method is trained with a batch size of 16 and 10 epochs. ADAM [19] is used for adaptive learning rate optimization and the optimum learning rate in this method was found at 0.00001.

For the traditional method, an input layer, four hidden layers, and an output layer are used. In the input layer, 1000 units are used, while 256, 128, 64, and 32 hidden units are used in the hidden layers. Also, 10 units are used in the output layer. Other hyperparameters are the same as the proposed method.

True positive values[5] are given to show the effectiveness of the proposed method. Figure 3 plots SNR versus true positive for single and simultaneous estimation. This figure shows that the proposed method is superior to the single estimation method.

Since the proposed method is ML-based, ensuring that the proposed method is well-generalized is essential, which means that the inputs should not be memorized during the training phase. The proposed method's training and validation losses versus epochs are shown in Fig. 4. The figure shows that the training sets converge to the validation set. This demonstrates no overfitting during training, demonstrating the suitability of the proposed method for use with unknown data.

V. Conclusion

This paper proposed the simultaneous estimation of the number of channel taps and coefficients to exploit the relationship between them. A DL-based multi-task learning method was designed for this purpose. This method estimated the number of channel taps and coefficients simultaneously. Therefore, there was no need to train two models to estimate them. Simulation results showed that the proposed method can identify the number of channel taps with higher performance. Furthermore, the simulations proved that the proposed DL method did not exhibit overfitting or underfitting. Last but

[5]A true positive value is an outcome where the model correctly predicts the positive class and it is widely used to compare classification methods performance [20].

important, the proposed method works automatically to identify the number of channel taps. In future work, the proposed method will be investigated in the real environment.

ACKNOWLEDGEMENT

This work was supported in part by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under Grant No. 119E433. Also, this work was supported in part by Vestel Elektronik Sanayi ve Ticaret A.Ş.

REFERENCES

[1] X. Wautelet, C. Herzet, A. Dejonghe, and L. Vandendorpe, "Mmse-based and em iterative channel estimation methods," in *Proc. IEEE 10th Symp. Commun. Veh. Technol. (SCVT)*, 2003.

[2] Z. Xiao, H. Wen, A. Markham, N. Trigoni, P. Blunsom, and J. Frolik, "Non-line-of-sight identification and mitigation using received signal strength," *IEEE Trans. Wireless Commun.*, vol. 14, no. 3, pp. 1689–1702, 2014.

[3] D. Li, G. Wu, J. Zhao, W. Niu, and Q. Liu, "Wireless channel identification algorithm based on feature extraction and BP neural network," *J. Inf. Process. Systems*, vol. 13, no. 1, 2017.

[4] C. Huang, B. Shen, and H. Wu, "A kind of channel coding type blind-identification method based on recognition with recurrent neural network," Nov. 2018, University of Electronic Science and Technology of China, CN109525369A.

[5] A. M. Jaradat, K. W. Elgammal, M. K. Özdemir, and H. Arslan, "Identification of the number of wireless channel taps using deep neural networks," in *Proc. 19th IEEE Int. New Circuits Syst. Conf. (NEWCAS)*. IEEE, 2021, pp. 1–4.

[6] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *J. Big Data*, vol. 2, no. 1, pp. 1–21, 2015.

[7] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, pp. 41–75, 1997.

[8] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[9] H. Arslan and G. E. Bottomley, "Channel estimation in narrowband wireless communication systems," *Wireless Commun. Mob. Comput.*, vol. 1, no. 2, pp. 201–219, 2001.

[10] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, 2016.

[11] J. Kaur, M. A. Khan, M. Iftikhar, M. Imran, and Q. E. U. Haq, "Machine learning techniques for 5G and beyond," *IEEE Access*, vol. 9, pp. 23 472–23 488, 2021.

[12] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation network," *Advances Neural Inf. Proces. Syst.*, vol. 2, 1989.

[13] L. Deng, D. Yu *et al.*, "Deep learning: Methods and applications," *Foundations Trends Signal Process.*, vol. 7, no. 3–4, pp. 197–387, 2014.

[14] Y. Zhang and Q. Yang, "An overview of multi-task learning," *National Sci. Review*, vol. 5, no. 1, pp. 30–43, 2018.

[15] T. Yucek and H. Arslan, "Time dispersion and delay spread estimation for adaptive OFDM systems," *IEEE Trans. Veh. Technol.*, vol. 57, no. 3, pp. 1715–1722, 2008.

[16] E. Alpaydin, *Introduction to Machine Learning*. London: The MIT Press, 2009.

[17] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2015, pp. 5353–5360.

[18] F. Chollet, "Keras," https://keras.io, accessed: 2023-09-23.

[19] D. Kinga, J. B. Adam *et al.*, "A method for stochastic optimization," in *Proc. Int. Conf. Learning Representations (ICLR)*, vol. 5. San Diego, California;, 2015, p. 6.

[20] S. Raschka, "An overview of general performance metrics of binary classifier systems," *arXiv preprint arXiv:1410.5330*, 2014.