

Article

Update-Based Machine Learning Classification of Hierarchical Symbols in a Slowly Varying Two-Way Relay Channel

Jakub Kolář, Jan Sýkora *  and Petr Hron

Department of radio engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, 160 00 Prague, Czech Republic; Jakub.Kolar@seznam.cz (J.K.); hronpetr@fel.cvut.cz (P.H.)

* Correspondence: jan.sykora@fel.cvut.cz

Received: 20 October 2020; Accepted: 9 November 2020; Published: 11 November 2020



Abstract: This paper presents a stochastic inference problem suited to a classification approach in a time-varying observation model with continuous-valued unknown parameterization. The utilization of an artificial neural network (ANN)-based classifier is considered, and the concept of a training process via the backpropagation algorithm is used. The main objective is the minimization of resources required for the training of the classifier in the parametric observation model. To reach this, it is proposed that the weights of the ANN classifier vary continuously with the change of the observation model parameters. This behavior is then used in an update-based backpropagation algorithm. This proposed idea is demonstrated on several procedures, which re-use previously trained weights as prior information when updating the classifier after a channel phase change. This approach successfully saves resources needed for re-training the ANN. The new approach is verified via a simulation on an example communication system with the two-way relay slowly fading channel.

Keywords: wireless physical-layer network coding; machine learning; artificial neural networks; backpropagation; two-way relay channel; decision regions

1. Introduction

After the great success of machine learning (ML) methods in the classical application domains such as computer vision and speech recognition, they recently proved successful in other areas as well. The tasks suited to ML techniques can be divided into three main categories. First is the classification of objects based on rules inferred from a training set of examples (supervised learning). Second is the classification of objects without access to any examples, and the rules are found based only on a given distance metric. This technique (unsupervised learning) typically results in a cluster system minimizing and maximizing the intra- and inter-cluster distances, respectively. The third category (reinforcement learning) deals with learning how to interact with an unknown environment in order to reach a given goal. This method jointly addresses the problem of estimating the behavior of the environment and simultaneously learns a policy, which dictates actions to be taken in order to meet the goal. A good introduction to these ML topics can be found in [1,2].

In this paper, we focus on the first category, i.e., on supervised learning, implemented with an artificial neural network (ANN). The ANN mimics the structure of a brain in a sense, in that it consists of a number of individual neurons, each having multiple real inputs and one real output. Each neuron is parameterized by a set of weights and a bias. The operation performed by each neuron is a simple weighted sum of its inputs, the addition of the bias, and the application of a nonlinear activation function. Typically, the structure of an ANN consists of a number of layers of neurons. The output of a

neuron is connected to the inputs of all neurons in the subsequent layer. The total number of layers and number of neurons in each layer are the design parameters of the network. A special role is given to the input and output layer, which are always present (ANNs with hidden layers are termed deep neural networks). In the input layer, there is one neuron for each dimension of the input data. In the output layer, we have typically a neuron for each individual classification class, where we can apply a hard metric (for example, a max function) for a hard decision or a soft metric (soft-max function) to get a soft output, which can be interpreted as a probabilistic classification output.

Learning is performed by applying examples from the training set and optimizing the parameters of individual neurons w.r.t. a fidelity metric of the ANN output and the desired output given by the training set. This optimization can be efficiently implemented by the backpropagation algorithm, which performs a stochastic gradient descent on the fidelity metric. This is the basic principle behind ML and ANN in particular. One of the issues with such a simple approach is the large number of parameters that arise with high-dimensional input data. With a large number of parameters, the learning process takes longer and requires a larger training set. One way of eliminating this problem is to preprocess the data and reduce its dimensionality while maintaining the information needed for classification. This process is called feature extraction and is very dependent on the particular data source and application. Much work has been done to develop sophisticated methods for such feature extraction, for example convolutional layers for image recognition [3], cepstral coefficients for speech recognition [4] or modulation recognition [5] and the bag-of-words technique, developed for feature extraction of textual data [6], to name a few.

Another issue, which is also related to the number of parameters of the network, is the problem of over-fitting. The core of all ML methods is the ability to extract generalizing classification rules out of a set of particular examples: the training set. Over-fitting occurs when insignificant details of the training data are learned and degrade the performance of classifying data not present in the training set. To combat this problem, regularization techniques such as constraining neuron weights or random neuron dropout are used, to name a few.

The problems of stochastic inference interpreted as ML tasks have attracted considerable attention in the field of physical layer communication [7–10]. The classical mathematical formulations of detection, estimation, and signal processing algorithms usually require a precise analytical knowledge of the system and observation model and usually lead to provable optimal closed-form results. Alternative ML-based approaches provide in such situations only numerically solved approximations to those solution. Although this has a limited applicability in classical scenarios, there are situations where the system and observation model are not fully known and/or the resulting closed-form solutions are too complex or unknown. In that situation, the ML approach provides an answer. As a demonstrative example, a particularly obvious area where this applies in the context of stochastic graphs is WPLNC (wireless physical layer network coding).

Recently, many interesting applications of ML were proposed for usage in wireless communications. To provide several motivating examples, in [11], the authors used ANN to perform joint OFDM channel estimation and symbol detection, where the learning phase was performed offline based on known channel statistics. In [8,9], a deep learning approach to jointly optimize the whole wireless communication chain was proposed, and this approach was successfully tested in an over-the-air transmission. Similar to this, in the context of WPLNC, the authors in [10] addressed parts of the network chain as individual deep neural networks, divided into the source modulator, a relay node, and the demodulator. Attention attracted also the approach of modulation classification in [12], where it was interpreted as an image classification problem. These papers largely motivated our work. So far, utilization of the ML in WPLNC is not deeply explored. The concept of WPLNC recently attracted much attention. A comprehensive tutorial of WPLNC techniques might be found in [13], where fundamental concepts, advantages, and challenges were covered.

This paper investigates the applicability of the ML concept in the parametric, slowly varying two-way relay channel (2WRC) WPLNC communication. This scenario is on the edge which still

can be handled by the traditional approaches, see [14]. This allows us to find the required reference scenarios to which we compare the ML solution and gives a hint about how the ML solution could be applicable in more complex scenarios, which are beyond the capability of classical analytical solutions. As was stated in [7], it is suitable to consider ML, when (1) the physical system model is not known, (2) the training dataset is available, (3) a clear metric of the task can be defined, and (4) a detailed explanation of the obtained result is not required. These general conditions are easily met in our context of wireless communication networks.

Our contribution in this paper is an exploration of a novel approach to compensate the effects of the variable phase of the wireless channel in a fundamental WPLNC scenario without the need for explicit channel state estimation (which would generally require orthogonal pilots). Instead, we use an ML approach that inherently tracks (learns) the varying hierarchical channel relative phase with tractable computational complexity. This is achieved with a hypothesis class of the ANN, and the task is interpreted as a classification problem. We explore methods to re-use previously trained coefficients of the ANN, thus saving computational resources. Validation of the results was performed by means of a numerical analysis.

The rest of the paper is organized as follows. Section 3 introduces the notation and describes the system model. Section 4 outlines a generic approach of ANN utilization for the classification of hierarchical symbols. Section 5 extends the previous approach for more practical scenarios and forms the main contribution of this paper. Section 6 concludes the paper and outlines future research directions.

2. Materials and Methods

Numerical analysis was performed using MATLAB software. The developed source code, utilized to produce the results, may be re-utilized without any restrictions and is available at: <https://www.dropbox.com/sh/vnc8j3ioksqhrmt/AAALaKeDHZyRBAJ-fnnFURfBa?dl=0>.

3. System Model

Let us assume a three-node network with two source nodes S_A , S_B and one intermediate relay node R . Further, let us assume a perfect time synchronization of symbols among the three nodes and the knowledge of the modulation pulse used. Such a network topology is referred to as a 2-way relay channel [13]. Let information bits $b_A, b_B \in \{0;1\}$ be at source nodes S_A, S_B mapped into the constellation space using binary phase shift keying (BPSK) as $s_A, s_B \in \{\pm 1\}$. For the topology of 2WRC, it is in the context of WPLNC efficient to utilize an XOR hierarchical network code map, such that only a hierarchical symbol $b = b_A \oplus b_B$ is required to be identified by the relay node (Briefly, the reason is that the relayed value of b is sufficient to recover information bit b_A at node S_B , as $b_A = b_B \oplus b$ and vice versa for b_B at S_A . See [13] for details.). A complex AWGN with variance σ_w^2 per dimension is denoted as $w \in \mathbb{C}$. A simplified channel model with a relative fading parameter $h \in \mathbb{C}$ is considered. For the considered analysis, the following constellation space system model is used:

$$x = s_A(b_A) + hs_B(b_B) + w, \quad (1)$$

where $x \in \mathbb{C}$ is the observation of node R . The sequence index is dropped for notational simplicity. The SNR will be related with respect to S_A and be denoted as $\text{SNR} = E[|s_A|^2] / \sigma_w^2$, where $E[\cdot]$ denotes the expectation operator.

4. Artificial Neural Network-Based Classification of Hierarchical Symbols

In this section, we briefly outline a generic approach, which is later improved and forms our main contribution. The main task is to infer the value of hierarchical symbol b , based on the observation x . Obviously, knowledge of the parameter h in (1) is crucial for this task. The expected channel phase drift over time opens a problem of ambiguity of deciding the hierarchical symbol b .

A basic background idea described in this paper is to utilize a classical ML approach for this problem, such that a training dataset $\mathcal{D} = \{(\mathbf{x}_d, \mathbf{t}_d)\}_{d=1}^D$ of size D is utilized to train an ANN classifier for this task. Vector $\mathbf{x}_d = [\Re\{x\}, \Im\{x\}]^T$ corresponds to the coordinates in the constellation space (an alternative would be to take $[|x|, \angle x]^T$ as the input vector, but it would introduce the problem of phase discontinuity). Vector \mathbf{t}_d with dimensionality 2 represents the desired output value of the ANN. It should be clear that even though we do not apply any explicit feature extraction, such preprocessing takes place in the form of a projection of the received signal onto a set of basis functions, thus giving the coordinates of \mathbf{x}_d in the constellation space. For the purpose of simulations, however, we work directly in the constellation space.

Let us now describe how to obtain the training dataset. Consider that initially, nodes S_A and S_B simultaneously transmit a predetermined, pseudorandom sequence of symbols $\{b_{A,d}\}_{d=1}^D, \{b_{B,d}\}_{d=1}^D$. Furthermore, let node R be aware of the resulting hierarchical target sequence $b_d = \{b_{A,d} \oplus b_{B,d}\}$. For the purpose of training the ANN using the backpropagation algorithm, this sequence is transformed to the expected ANN output vector \mathbf{t}_d as $\mathbf{t}_d = [b_d, 1 - b_d]$. Thus, the observation $\mathbf{x}_d(x)$ together with the corresponding value \mathbf{t}_d form the required training dataset for the supervised ML hypothesis class.

In the case of the considered BPSK modulation, the hierarchical symbol is binary. See Figure 1 for a demonstrative example. Therein, two colors distinguish the classes corresponding to the two values of the hierarchical target symbol b . The phase of h , denoted $\angle h$, dictates the rotation of the superimposed constellation.

It is natural to visualize the trained classifier in terms of its decision regions, as shown in [13]. An example of trained decision regions is shown in Figure 2. Therein, two reference solutions are provided. The trained decision regions are shown in the central subfigure. The true metric map is based on the analytically optimal metric; see [13] for details. The distance-based map is an approximation of the true metric map, based on the Euclidean distance between the neighboring points only, also derived in [13].

With the above description of the nature of the training dataset, it is straightforward to design an ANN and train it using the backpropagation algorithm and perform the classification of the hierarchical symbols; see [1]. The specific implementation is not the main objective of this paper. For more details; see [15].

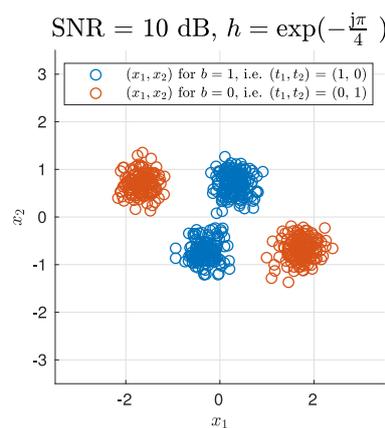


Figure 1. Demonstration of the training dataset.

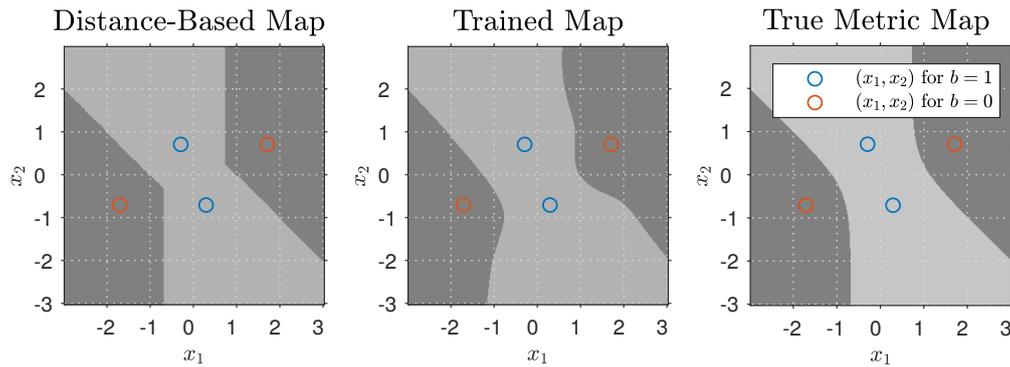


Figure 2. Illustration of trained decision regions and comparison with reference solutions.

5. Slowly Varying Fading Parameter in 2WRC

Above, we briefly addressed how an ANN might be simply utilized for the classification of hierarchical symbols in 2WRC. Therein, we considered the training dataset \mathcal{D} to be obtained for a single fixed value of $\angle h$. In this section, we address a more practical situation, when the value of $\angle h$ is slowly varying. Practically, effects such as slow user mobility, changes in the propagation environment, or drift of the internal oscillators might be considered as the causes of it.

Considering the ANN, its functionality is fully determined by its structure (number of layers and number of neurons). The training dataset is obtained as a pilot signal known to both Tx and Rx, as described previously, and we shall focus on the minimization of resources required for training. The reason is two-fold. Firstly, the number of training samples used for training directly affects signaling overhead. Thus, it is desirable to minimize the number of training samples D . Secondly, the training process of ANN using backpropagation itself is time consuming. Besides the number of training samples D , its time complexity is typically dependent on the number of training epochs E and learning rate η . From a matrix implementation point of view, it can be easily seen that the time complexity of backpropagation scales quadratically with the number of nodes. The time complexity dependence is linear w.r.t. the size of the training set. Therefore, we relaxed the employment of a relatively small ANN with 4 layers. The core of our contribution in this paper is exploring the ideas of how to minimize the resources required for training the ANN in the case of hierarchical symbol classification in the 2WRC.

Note that in the simulations, we consider a 2WRC with BPSK modulation, and the employed ANN consists of 2 hidden layers, each having 40 neurons. The network has 2 inputs, corresponding to the coordinates in the constellation space. Its two outputs are designed to indicate the binary value of the hierarchical symbols, which might be easily scaled using the soft-max approach to interpret the outputs as classification likelihoods.

5.1. Numerical Analysis of the Effect of Training Dataset Size

Unfortunately, no general rules are available to determine what size of the training dataset is required to achieve the desired accuracy or error rate. This issue is traditionally addressed heuristically. Luckily, for our problem, it is straightforward to perform an exhaustive number of Monte Carlo simulations to determine, how the quality of the trained classifier depends on the ANN parameters.

The size of the training dataset D is hereby considered to be a key parameter, since it dictates the transmission overhead. In Figure 3, we present an exemplary result of such a numerical analysis. Therein, we observe that for more than $D = 2000$ training samples, the accuracy of classification no longer increases, and therefore, it is not desirable to waste valuable radio resources to obtain them.

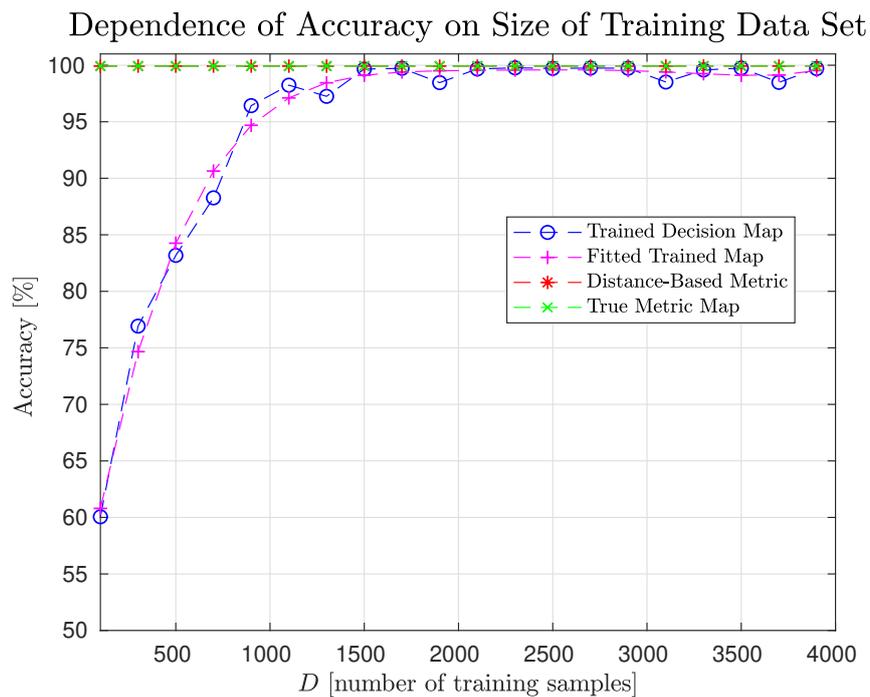


Figure 3. Numerical analysis of how the training dataset size affects the resulting accuracy of classification.

5.2. Updates of Previously Trained Weights

As stated previously, training of the ANN-based classifier is further determined by parameters E and η . A basic approach for training an ANN classifier is to randomly initialize its weights and then perform backpropagation. Assume this is done for the training dataset $\mathcal{D}(\angle h_0)$, parameterized by the channel phase. The trained weights are denoted by α_0 . Consider further that after the training process is performed, the ANN can be used to classify the received symbols for a certain time, while $\angle h \approx \angle h_0$. Due to the variability of the wireless channel, consequently $\angle h = \angle h_1 \neq \angle h_0$. Inevitably, the classifier needs to be updated according to $\mathcal{D}(\angle h_1)$, resulting in new weights α_1 . In general, we ask whether we can re-use previously trained results α_{i-1} for obtaining new weights α_i . To provide an answer, we propose three procedures, which differ in the strategy of re-initialization and compare their classification performance.

For minimizing the number of operations for training, we propose the following general scheme with diverse epoch numbers and learning rates. This procedure repeats with period I . First, the network is trained on $\mathcal{D}(\angle h_0)$ over E_1 epochs with rate η_1 . For $I - 1$ subsequent sets $\mathcal{D}(\angle h_i)$, the network is trained over $E_2 < E_1$ epochs with rate $\eta_2 > \eta_1$. The motivation is such that a higher number of training epochs E_1 is used to train weights via a smaller learning rate η_1 , leading to more precise initial training.

5.2.1. Procedure P1: Always Re-Initialize

The first procedure, P1, is visualized in Figure 4 with green color. The weights of the ANN are initialized. A long training process is performed, lasting E_1 epochs with learning rate η_1 , and subsequently for $I - 1$ realizations of $\angle h_i$, the training is performed within E_2 epochs with η_2 . Weights α_i are re-initialized before each training.

5.2.2. Procedure P2: Regular Re-Initialization

The second procedure, P2, is visualized in Figure 4 with blue color. Compared to the previous routine, the weights are re-initialized after every I th training.

5.2.3. Procedure P3: Never Re-Initialize

The third procedure, P3, is visualized right-most in Figure 4 with red color. Weights α_i are randomly initialized only before the first training process. For the subsequent values of $\angle h_i$, the weights are updated and utilized as prior information.

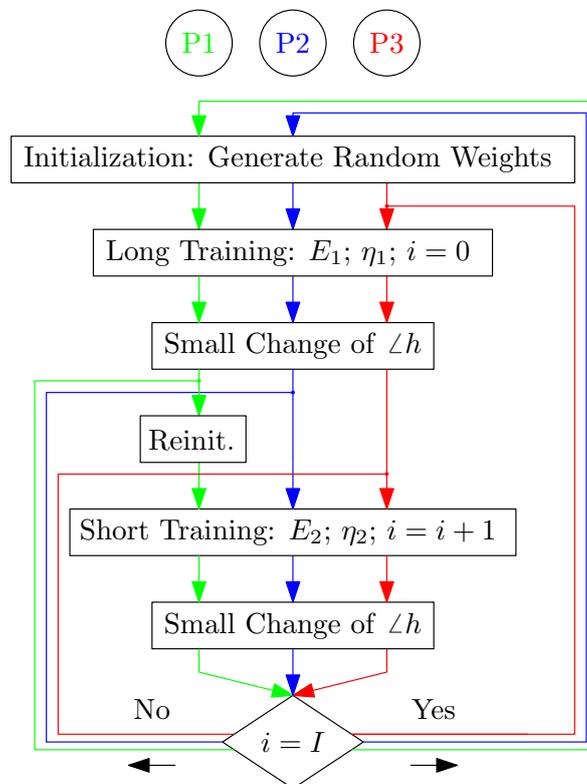


Figure 4. An illustrative block diagram representing procedures P1 (green), P2 (blue), and P3 (red). The difference to be emphasized is the strategy of re-initialization of the ANN weights.

5.3. Numerical Analysis

The above-described procedures P1 – P3 were implemented, and we provide numeric simulation results. The angle of relative fading $\angle h$ of consequent pilot sequences used for the training process changes uniformly as $\angle h_{i+1} - \angle h_i = 4.5^\circ$. As justified by results presented in Figure 3, a pilot sequence of length $D = 2000$ was considered. We experimentally determined $E_1 = 30$ and $E_2 = 20$ to be the suitable numbers of training epochs with the corresponding learning rates $\eta_1 = 0.05$ and $\eta_2 = 0.07$. With respect to the block diagram in Figure 4, consider $I = 3$. To provide an in-depth look, an analysis of the ANN weights’ evolution was performed. To evaluate the rates of change of all weights α in each layer of the ANN, the following expression was used:

$$a^{(j)}(e) = \frac{1}{A^{(j)}} \sum_i \left[\alpha_i^{(j)}(e) - \alpha_i^{(j)}(e - 1) \right]^2, \tag{2}$$

where the upper index (j) specifies the layer, $A^{(j)}$ is the number of weights in layer (j) , e is a counter of training epochs, and i identifies the specific weights of the layer. For comparison, this expression is evaluated and graphically represented in Figure 5 for procedure P3, without re-initialization, and in Figure 6 for procedure P1, where the weights are always re-initialized, and thus no prior information is utilized (graphically, the inspection of procedure P2 resulted in similar results as for P3, and the figure is therefore omitted). Note that in these figures, it is clearly observable how the training process

converges. The peaks of the curves correspond to the changes of the parameter $\angle h_i$, as marked by arrows. These results are useful to optimize parameters E_1, η_1, E_2, η_2 .

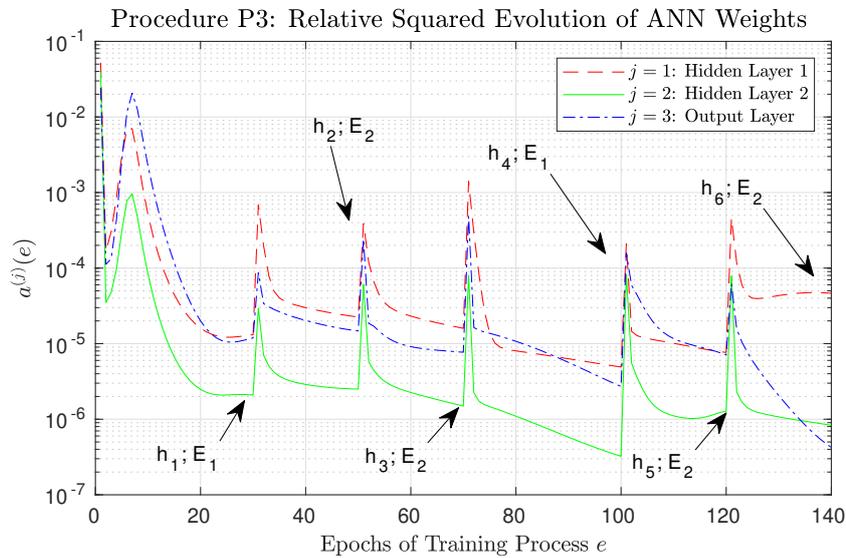


Figure 5. Relative squared evolution of the weights for procedure P3, where the weights are initialized only once, and for every subsequent value of $\angle h_i$, the weights are updated via backpropagation without subsequent re-initializations.

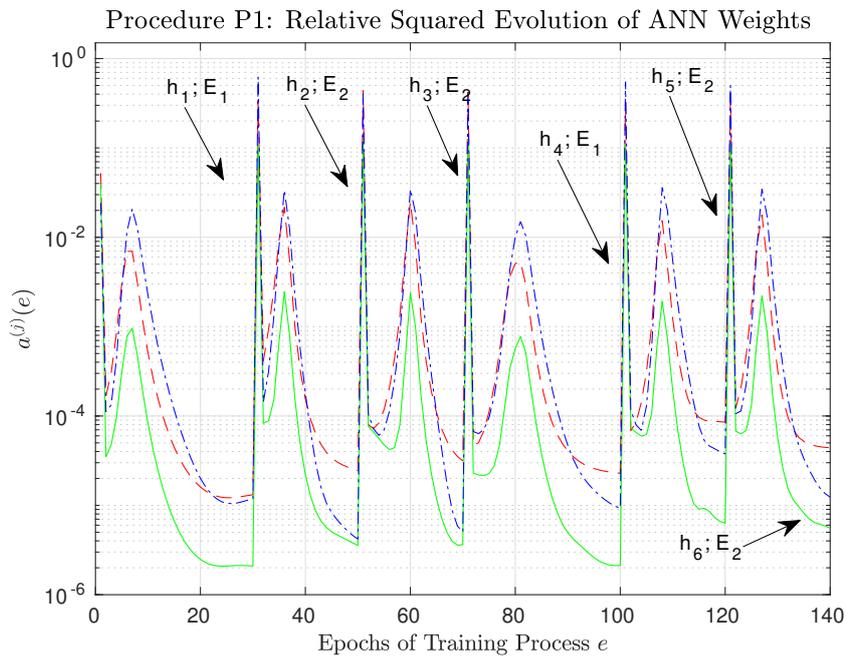


Figure 6. Relative squared evolution of weights for procedure P1, where the weights are initialized, and for every subsequent value of $\angle h_i$, the weights are re-initialized prior to the further training process. The legend from Figure 5 is used.

In Figure 7, we provide a comparison of the approaches in terms of the bit-error rate (BER) over different values of the SNR. Therein, two reference solutions are provided, as addressed in Section 4. Recall that these reference solutions are based on perfect knowledge of $\angle h_i$ and therefore are naturally superior to the trained solutions. Procedures P1–P3, represented in Figure 4, were tested for two values of the parameter I , namely $I = 3$ and $I = 5$. In this result, we observe that the most efficient strategy

is procedure P3. Clearly, it is also better to keep longer training sequences more often, i.e., $I = 3$ is preferable to $I = 5$.

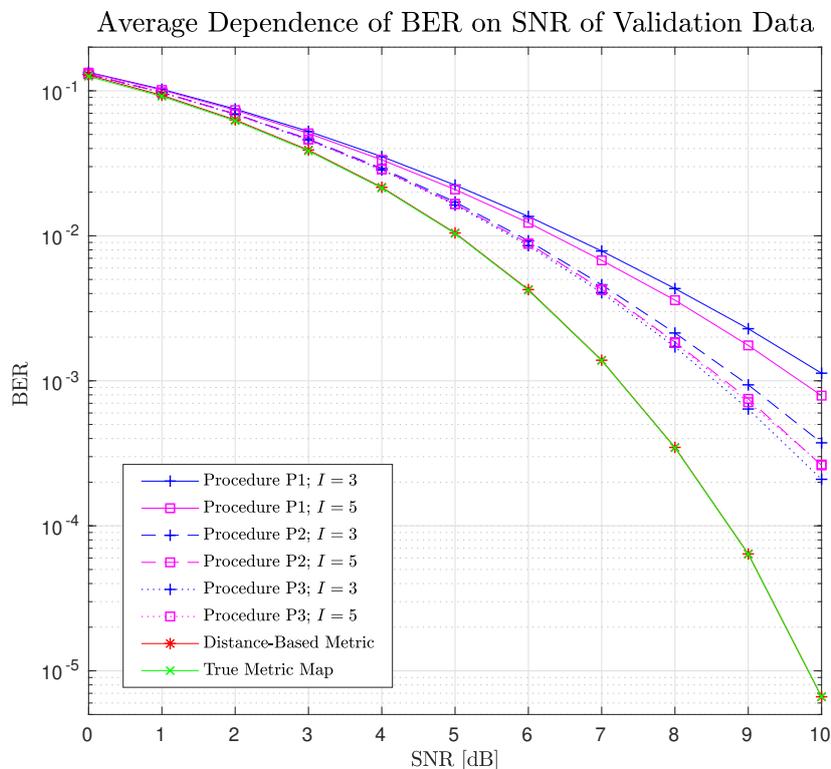


Figure 7. Comparison of different proposed approaches together with reference solutions. Procedure P3 with $I = 3$ performed the best. The validation dataset of 100,000 samples was used.

6. Discussion

In this paper, we considered the problem of the classification of received symbols in 2WRC according to a many-to-one hierarchical map given from a WPLNC scenario with the employment of an ANN as a classifier. It was shown that for a slowly varying phase of the wireless channel, it is possible to utilize previously trained weights of the ANN as prior information. An effort was made to minimize the resources required for the training of the system, and the number of resources required for training was optimized. The simulation results showed that the utilization of the prior information was beneficial and improved the overall classification results. Three different procedures of training were proposed, implemented, and evaluated. The procedure denoted P3 in the text, where the weights of the ANN were not re-initialized with consecutive changes of the relative fading parameter, was identified to be the best one.

Future research will focus on (1) the optimization of the parameters of the proposed methods, where the challenge is recognized in the number of parameters that the ANN model contains and a more systematic approach would be desired, (2) the implementation of the approach for over-the-air transmission, which seems straightforward using the concept of software-defined radio (however on-line processing of data opens timing issues in practical networks), and (3) exploiting of the principle for more complex networks, where new challenges will emerge. As noted previously, the presented scenario is still on the edge, which can be handled by traditional, analytically derived approaches. However, these are bearable only for simple network topologies. Hopefully, we shall be able to extend the approach described in this paper to more complex scenarios. These extensions might focus on

more complex modulation schemes, an increased number of network nodes, and relaxed knowledge of the network topology.

Author Contributions: Development of the algorithms and writing, J.K.; supervision and editing, J.S.; validation and editing, P.H. All authors read and agreed to the published version of the manuscript.

Funding: This work was supported by the Grant Agency of the Czech Technical University in Prague, Grant No. SGS20/068/OHK3/1T/13, B2.3 (6.11.2020).

Acknowledgments: The authors would like to thank the two anonymous reviewers for their helpful comments, which improved the text.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

2WRC	2-way relay channel
WPLNC	wireless physical layer network coding
ANN	artificial neural network
ML	machine learning
SNR	signal-to-noise ratio
BPSK	binary phase shift keying
BER	bit-error rate

References

1. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: New York, NY, USA, 2006.
2. Sutton, R.; Barto, A.G. *Reinforcement Learningan Introduction*, 2nd ed.; The MIT Press: London, UK, 2018.
3. O'Shea, K.; Nash, R. An introduction to convolutional neural networks. *arXiv* **2015**, arXiv:1511.08458.
4. Souissi, N.; Cherif, A. Speech recognition system based on short-term cepstral parameters, feature reduction method and artificial neural networks. In Proceedings of the 2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Monastir, Tunisia, 21–23 March 2016; pp. 667–671.
5. Keshk, M.E.H.M.; Abd El-Naby, M.; Al-Makhlaway, R.M.; El-Khobby, H.A.; Hamouda, W.; Abd Elnaby, M.M.; El-Rabaie, E.S.M.; Dessouky, M.I.; Alshebeili, S.A.; Abd El-Samie, F.E. Automatic modulation recognition in wireless multi-carrier wireless systems with cepstral features. *Wirel. Pers. Commun.* **2015**, *81*, 1243–1288. [[CrossRef](#)]
6. Sheikh, I.; Illina, I.; Fohr, D.; Linares, G. Learning word importance with the neural bag-of-words model. In Proceedings of the 1st Workshop on Representation Learning for NLP, Berlin, Germany, 11 August 2016; pp. 222–229.
7. Simeone, O. A Very Brief Introduction to Machine Learning With Applications to Communication Systems. *IEEE Trans. Cogn. Commun. Netw.* **2018**, *4*, 648–664. [[CrossRef](#)]
8. O'Shea, T.; Hoydis, J. An Introduction to Deep Learning for the Physical Layer. *IEEE Trans. Cogn. Commun. Netw.* **2017**, *3*, 563–575. [[CrossRef](#)]
9. O'Shea, T.J.; Karra, K.; Clancy, T.C. Learning to communicate: Channel auto-encoders, domain specific regularizers, and attention. In Proceedings of the 2016 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Limassol, Cyprus, 12–14 December 2016; pp. 223–228.
10. Matsumine, T.; Koike-Akino, T.; Wang, Y. Deep Learning-Based Constellation Optimization for Physical Network Coding in Two-Way Relay Networks. In Proceedings of the ICC 2019—2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–6.
11. Ye, H.; Li, G.Y.; Juang, B.H. Power of deep learning for channel estimation and signal detection in OFDM systems. *IEEE Wirel. Commun. Lett.* **2017**, *7*, 114–117. [[CrossRef](#)]
12. Peng, S.; Jiang, H.; Wang, H.; Alwageed, H.; Zhou, Y.; Sebdani, M.M.; Yao, Y. Modulation Classification Based on Signal Constellation Diagrams and Deep Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 718–727. [[CrossRef](#)] [[PubMed](#)]
13. Sykora, J.; Burr, A. *Wireless Physical Layer Network Coding*; Cambridge University Press: Cambridge, UK, 2018.

14. Hron, P.; Sykora, J. Performance Analysis of Hierarchical Decision Aided 2-Source BPSK H-MAC CSE with Feed-Back Gradient Solver for WPNC Networks. In Proceedings of the 2019 IEEE Microwave Theory and Techniques in Wireless Communications (MTTW), Riga, Latvia, 1–2 October 2019.
15. Kolar, J. Machine Learning Algorithms in Wireless Physical Layer Network Coding. Master's Thesis, Czech Technical University in Prague, Prague, Czech Public, 2020. Available online: <http://hdl.handle.net/10467/86135> (accessed on 29 January 2020).

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).