Unsupervised Deep Learning for MU-SIMO Joint Transmitter and Noncoherent Receiver Design

Songyan Xue, Yi Ma, *Senior Member, IEEE*, Na Yi, *Member, IEEE*, and Rahim Tafazolli, *Senior Member, IEEE*

***Abstract*— This work aims to handle the joint transmitter and noncoherent receiver optimization for multiuser single-input multiple-output (MU-SIMO) communications through unsupervised deep learning. It is shown that MU-SIMO can be modeled as a deep neural network with three essential layers, which include a partially-connected linear layer for joint multiuser waveform design at the transmitter side, and two nonlinear layers for the noncoherent signal detection. The proposed approach demonstrates remarkable MU-SIMO noncoherent communication performance in Rayleigh fading channels.**

***Index Terms*—Unsupervised deep learning, joint transmitter and receiver design, noncoherent detection, multiuser single-input and multiple-output (MU-SIMO).**

I. INTRODUCTION

The use of neural networks for communicational signal processing can be traced back to 1990s, when the research was mainly focused on the autonomic modulation recognition, demodulation, and channel decoding (e.g. [1], [2]). With the recent success of deep learning in wide areas of applications (including natural language processing, image processing, autonomous driving, financial investment, computer games, and many others), neural networks have regained increasing interests in the domain of communicational signal processing (e.g. [3]–[5]). Their basic principle lies in the use of deep neural network (DNN) for more detailed waveform classification. Remarkably, an unsupervised deep learning approach has recently been proposed for joint transmitter and receiver (T-R) optimization, where the communication chain is modeled as an autoencoder combined with a parametric transform network (PTN) [6]. Through unsupervised offline learning given a specific fading channel model, the autoencoder-PTN approach is able to suggest a suitable transmitter and receiver structure for noncoherent communications.

This work is motivated by the fact that the current autoencoder-PTN structure involves considerable number of hidden layers, which could introduce unnecessary complexity and processing latency for both the learning and communication procedures. Moreover, a layer-reduced DNN might get the front layer better optimized due to its advantage of handling the *vanishing (or exploding) gradients problem* in the DNN training procedure [7], [8]. After a fundamental rethinking of the joint T-R design from the communication theoretic aspect, we show that the communication chain can be represented by a DNN with three essential layers. With specific to multiuser single-input multiple-output (MU-SIMO) noncoherent communications, the first layer is a partially-connected linear layer responsible for multiuser waveform joint optimization, and the others are nonlinear dense layers for noncoherent multiuser detection at the receiver side. Computer simulations show that the proposed deep-learning approach offers remarkable non-coherent communication performance in Rayleigh-fading channels due to the transmitter-receiver joint optimization gain. We also applied the proposed approach onto single-input single-output (SISO) system; as a special case of MU-SIMO. The proposed approach outperforms the autoencoder-PTN approach while offers much lower computational complexity.

Manuscript received February 27, 2018; revised July 12, 2018; accepted August 11, 2018. This work was supported in part by UK 5G Innovation Centre (5GIC). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. S. K. Mohammed.

The authors are from Institute for Communication Systems (ICS), University of Surrey, Guildford, England, GU2 7XH. Email: (songyan.xue, y.ma, n.yi, r.tafazolli)@surrey.ac.uk. Tel.: +44 1483 683609. Na Yi is also with DeepGo Ltd. (*Corresponding author:* Yi Ma)

II. SYSTEMS MODEL AND PROBLEM STATEMENT

Consider MU-SIMO communications, where a set of user terminals talk to an access point with N receive antennas. Each user terminal employs a single transmit-antenna to send a temporal sequence $\mathbf{c}_m \triangleq [c_{0,m}, ..., c_{L-1,m}]^T$, where m stands for the user index, L for the sequence length, and $[\cdot]^T$ for the matrix/vector transpose. Considering there are $M(\leq N)$ users involved in the communication, the received signal at the access point is described by the following matrix form

$$\mathbf{y}_l = \mathbf{H}\mathbf{x}_l + \mathbf{v}_l, \ {}_{0\leq l\leq L-1} \tag{1}$$

where $\mathbf{x}_l \triangleq [c_{l,0}, ..., c_{l,M-1}]^T$, $\mathbf{H}$ is the $(N) \times (M)$ random channel matrix, and $\mathbf{v}_l$ is the white Gaussian noise with zero mean and the variance σ_v^2.

Suppose: A1) the receiver does not know the channel matrix $\mathbf{H}$, and A2) elements in $\mathbf{c}_m$ can be mutually correlated with respect to l. The receiver aims to reconstruct the sequences $\mathbf{c}_m, \ {}_{\forall m}$, from $\mathbf{y}_l, \ {}_{\forall l}$, through noncoherent sequence detection. To facilitate our discussion, we represent the linear model (1) into a more compact form as below

$$\mathbf{Y} = \mathbf{X}\mathbf{H}^T + \mathbf{V} \tag{2}$$

where $\mathbf{X}$ is a $(L) \times (M)$ matrix with the m^{th} column formed by $\mathbf{c}_m$, $\mathbf{Y}$ is a $(L) \times (N)$ matrix with the l^{th} row formed by $\mathbf{y}_l^T$, and $\mathbf{V}$ the noise matrix corresponding to $\mathbf{v}$. Then, the noncoherent receiver forms the following mathematical relationship

$$\hat{\mathbf{X}} = G(\mathbf{Y}), \tag{3}$$

where $G(\cdot)$ denotes the function for signal detection.

Due to the channel randomness and the assumption A1), $\mathbf{X}$ might not be uniquely determined by $\mathbf{Y}$ even in the noiseless case. Such is called channel ambiguity which is the dominating factor that limits the noncoherent detection performance. This is the problem we aim to handle through the deep-learning assisted joint T-R design.

III. Deep Learning for Joint T-R Optimization

A. Minimization of Channel Ambiguity

We start from the maximum a posteriori (MAP) receiver of (3) which maximizes the probability of $\mathbf{X}$ conditioned on $\mathbf{Y}$

$$\hat{\mathbf{X}} = \arg\max_{\mathbf{X}\in\mathbf{A}} p(\mathbf{X}|\mathbf{Y}) \tag{4}$$

subject to $\mathbf{X}$ drawn from a finite-alphabet set $\mathbf{A} = \{\mathbf{\Theta}_0, ..., \mathbf{\Theta}_{J-1}\}$, where $\mathbf{\Theta}_j$ is a $(L) \times (M)$ matrix, and J the size of $\mathbf{A}$. The m^{th} column of $\mathbf{\Theta}_j$ is the m^{th} user's codeword $\mathbf{c}_m$, which is independently drawn from their user-specific codebooks $\mathbf{C}_m$. Assuming each codebook having K codewords, we have: $J = K^M$. In (4), $p(\cdot)$ denotes the probability.

The goal of joint T-R design is to find a set $\mathbf{A}^\star$ (or equivalently $\mathbf{C}^\star_{m,\forall m}$) that minimizes the error probability $p(\hat{\mathbf{X}} \neq \mathbf{X})$. From the information-theoretic point of view, $\mathbf{A}^\star$ is a set of joint typical sequences. As long as $\mathbf{A}^\star$ is determined, every user terminal will utilize their user-specific codebook $\mathbf{C}^\star_m$ for communications, and there is no need of cooperation between users.

Definition 1: The objective function in (4) can be viewed as a waveform classification problem as illustrated in Fig. 1 (a). Given J bins labeled with $\mathbf{\Theta}_0, ..., \mathbf{\Theta}_{J-1}$, the role of receiver (classifier) is to throw $\mathbf{Y}$ into their corresponding bins according to the a posteriori probability. Consider a received waveform $\mathbf{Y} = \mathbf{\Theta}_j\mathbf{H}^T + \mathbf{V},\ _{0\leq j\leq J-1}$, the waveform is called miss-classified if we have $\hat{\mathbf{X}} \neq \mathbf{\Theta}_j$. The probability of missed classification (PMC) can be directly translated into the communication error probability.

There are two factors that can cause the waveform missed classification. One is the white Gaussian noise $\mathbf{V}$, and the other is the channel ambiguity as already briefly discussed in Section II. While the noise issue has been well addressed in the literature (e.g. [6]), our focus is on the channel ambiguity which should be minimized using deep learning for the joint T-R design.

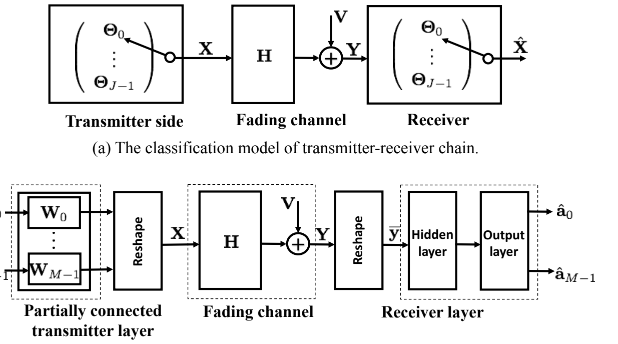
Proposition 1 (channel ambiguity): Given the waveform set $\mathbf{A}$ with $\mathbf{\Theta}_j \in \mathbf{A}$ to be equally probable, the probability of waveform missed classification in the noise-free (NF) case is

$$\text{PMC}_{\text{NF}} = \mathop{\mathbb{E}}_{\mathbf{H}_j}\left(\frac{\sum_{k\neq j} p(\mathbf{\Theta}_k^{-1}\mathbf{\Theta}_j\mathbf{H}_j^T)}{\sum_{k=0}^{J-1} p(\mathbf{\Theta}_k^{-1}\mathbf{\Theta}_j\mathbf{H}_j^T)}\right) \tag{5}$$

where $\mathbb{E}(\cdot)$ denotes the expectation, and $\mathbf{H}_j \triangleq \mathbf{\Theta}_j^{-1}\mathbf{Y}$ in the noise-free case.

The term PMC_{NF} in (5) is due to the channel ambiguity and thus noise independent. It can cause communication error floor at higher signal-to-noise ratios (SNRs) if we have $\mathbf{\Theta}_k, \mathbf{\Theta}_j \in \mathbf{A}$ inappropriately designed. Hence, it must be minimized with the following objective function

$$\min_{\mathbf{A}} \text{PMC}_{\text{NF}} = \min_{\mathbf{A}}(1 - \Gamma(\mathbf{H}_j, \mathbf{\Theta}_k, \mathbf{\Theta}_j)) \tag{6}$$



(a) The classification model of transmitter-receiver chain.

(b) The proposed deep neural network architecture.

Fig. 1. The proposed DNN structure for joint transmitter and noncoherent receiver design.

with

$$\Gamma(\mathbf{H}_j, \mathbf{\Theta}_k, \mathbf{\Theta}_j) \triangleq \mathbb{E}\left(\frac{p(\mathbf{H}_j)}{\sum_{k=0}^{J-1} p(\mathbf{\Theta}_k^{-1}\mathbf{\Theta}_j\mathbf{H}_j^T)}\right),\ \mathbf{\Theta}_k, \mathbf{\Theta}_j \in \mathbf{A} \tag{7}$$

Since $0 < \Gamma(\mathbf{H}_j, \mathbf{\Theta}_k, \mathbf{\Theta}_j) \leq 1$, the optimization problem can be stated by

Proposition 2: The channel-ambiguity minimization problem (6) is equivalent to

$$\max_{\mathbf{A}} \Gamma(\mathbf{H}_j, \mathbf{\Theta}_k, \mathbf{\Theta}_j),\ \ \mathbf{\Theta}_k, \mathbf{\Theta}_j \in \mathbf{A} \tag{8}$$

where $\mathbf{H}_j$ is a random channel matrix following a certain probability distribution.

In general cases, the optimization problem (8) is mathematically intractable. We might employ the Cauchy-Schwarz inequality to obtain

$$\Gamma(\mathbf{H}_j, \mathbf{\Theta}_k, \mathbf{\Theta}_j) \leq \sqrt{\mathbb{E}\left(\frac{p(\mathbf{H}_j)}{(\sum_{k=0}^{J-1} p(\mathbf{\Theta}_k^{-1}\mathbf{\Theta}_j\mathbf{H}_j^T))^2}\right)\mathbb{E}(p(\mathbf{H}_j))} \tag{9}$$

where the upper bound is achievable at

$$p(\mathbf{H}_j) + \sum_{k\neq j,\forall k} p(\mathbf{\Theta}_k^{-1}\mathbf{\Theta}_j\mathbf{H}_j^T) = \lambda \text{ (constant)}. \tag{10}$$

Such yields a necessary condition for the optimization problem (8). Note that it is mathematically challenging to find sufficient conditions for (8). Nevertheless, optimization based on the necessary condition (10) is an integer linear programming problem which is already NP hard. This renders DNN a considerable approach helping us to find a suitable waveform set $\mathbf{A}$.

B. The Proposed Deep Learning Algorithm

Fig. 1 (b) illustrates the block diagram of the proposed DNN architecture for the joint transmitter and noncoherent receiver optimization. The transmitter side is modeled as a partially-connected linear layer [1], where the weighting matrix is partitioned into M sub-matrices $\mathbf{W}_m,\ _{0\leq m\leq M-1}$, with each

[1] Here, we emphasize that $\mathbf{W}_m,\ _{0\leq m\leq M-1}$, are not connected with each other. Although they are jointly optimized in the learning procedure, there is no indication of user cooperation in the communication procedure.

corresponding to the user-specific codebook $\mathbf{C}_m,\ _{0\leq m\leq M-1}$, for the m^{th} user terminal, and so $\mathbf{W}_m$ has the size of $(L)\times(K)$. The input to the transmitter layer is a set of one-hot vectors denoted by $\mathbf{a}_m,\ _{0\leq m\leq M-1}$, with the size of $K\times 1$, and $\mathbf{a}_m$ follows the uniform distribution. The output of the linear layer is: $\mathbf{z}_m = \mathbf{W}_m\mathbf{a}_m,\ _{0\leq m\leq M-1}$, which is a column of $\mathbf{W}_m$ selected by $\mathbf{a}_m$. The 'reshape' component at the transmitter side relates $\mathbf{z}_m$ to $\mathbf{X}$ as: $\mathbf{X} = [\mathbf{z}_0, ..., \mathbf{z}_{M-1}]$. After the offline training, the linear layer will form the finite-alphabet set $\mathbf{A}$ (see (4)) with an appropriate optimization. The receiver layers play a central role for waveform classification. The input to the receiver layers is a column vector $\overline{\mathbf{y}}$ by reshaping the matrix $\mathbf{Y}$, and the output is a set of estimated one-hot vectors $\hat{\mathbf{a}}_m,\ _{0\leq m\leq M-1}$. The entire DNN is trained with the objective of minimizing the difference $(\hat{\mathbf{a}}_m - \mathbf{a}_m),\ _{0\leq m\leq M-1}$ (equivalent to the categorical cross-entropy minimization). According to the deep learning principles [9], [10], two nonlinear layers (1 hidden layer and 1 output layer) are sufficient for the classification task.

Discussion on Key Novelties: The proposed DNN approach shares the same principle as the autoencoder approach originally proposed in [6] for single-input single-output (SISO) communications in the sense of utilizing unsupervised deep learning for joint transmitter and receiver design. One of major differences between the two approaches lie in:

1) The proposed DNN minimizes the use of hidden layers at both the transmitter side and the receiver side. Such largely reduces the DNN complexity, and mitigates the vanishing (or exploding) gradients problem in the training procedure. Our simulation results (see Section IV) show that the simplified DNN structure does not introduce any performance penalty in communications.

In addition, the proposed DNN approach is extended to the MU-SIMO system that

2) employs a partially-connected linear layer to model the behavior of multi-transmitter concurrent transmissions. Such allows multiple transmitters' waveform (i.e., $\mathbf{W}_m,\ _{0\leq m\leq M-1}$) to be jointly optimized through the deep learning. Thanks to the partially-connected structure, $\mathbf{W}_m$ can be immediately downloaded onto each individual transmitter after the offline training, and there is no need for user cooperation in communications.
3) employs the asymmetric DNN structure to model the communication chain instead of using the symmetric structure in the auto-encoder approach. Despite the use of partially connected layer at the transmitter side, fully connected layers are employed at the receiver side to facilitate the multiuser joint detection. The asymmetric DNN structure is more appropriate for modeling the MU-SIMO system.

IV. Computer Simulations and Discussion

Our computer simulations are structured into two experiments with respect to the SISO and MU-SIMO communication cases. The former is mainly focused on the comparison with the state-of-the-art PTN-DNN approach, and the latter introduces a novel neural network architecture which aims to implement the MU-SIMO communication system. The performance of the proposed DNN approach is evaluated using the block error rate (BLER) averaging over sufficient Monte-Carlo channel trials. The SNR is defined by the average received bit-energy to noise ratio (E_b/N_0).

Experiment 1 (DNN for SISO): Table I provides the setup of the proposed DNN for the SISO case. Basically, an information bit-stream is divided into a number of blocks with each having 4 bits. Each block is represented by an one-hot vector $\mathbf{a}_0$ which is fed into the linear layer. The output of the linear layer is a $(8)\times(1)$ normalized vector; such introduces the code rate of $1/2$. The loss function is the categorical cross-entropy. The communication channel is a 3-tap time dispersive channel with Rayleigh distribution. The DNN is trained using the stochastic gradient descent (SGD) with Adam optimizer at the learning rate $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ (see [11] for the detailed description of parameters). The batch size of each epoch is 1000, and the DNN is trained at $E_b/N_0 = 15$ dB. This configuration is in line with the reference configuration in [6].

TABLE I
DNN SETUP FOR THE SISO CASE.

No. Transmitters (M)	1
One-hot vector ($\mathbf{a}_0$)	$(16)\times(1)$ column vector
Transmitter layer ($\mathbf{W}_0$)	Activation function: linear; the output of the linear layer is normalized
Rx Hidden layer	Nonlinear layer; Activation function: ReLU
Output layer	Nonlinear layer; Activation function: softmax

Fig. 2 shows the BLER performance of the proposed DNN approach. The baseline for performance comparison includes: the DNN-PTN approach in [6] as well as the DBPSK-MLE with Hamming $(7,4)$ code. It is shown that both deep-learning approaches outperform the conventional hand-engineered non-coherent communication throughout the whole SNR range. The performance gain mainly comes from the joint transmitter and receiver optimization through deep learning. Moreover, it is interesting to find that the proposed DNN offers slightly better performance (around 0.8 dB) in comparison with the DNN-PTN approach. This gain attributes to the mitigation of vanishing gradient problem, and thus the proposed DNN gets well trained particularly for a better transmitter design.

Experiment 2 (DNN for MU-SIMO): In this experiment, a partially-connected linear layer is employed at the transmitter side for the multi-transmitter joint optimization. At the receiver, we divide the decoded information into several blocks corresponding to their transmitted blocks, respectively. The activation function for the output layer is sigmoid. The rest of the network setup is the same as that for SISO communications.

Fig. 3 illustrates the BLER for MU-SIMO systems with various configurations of the MIMO size and code rate. The MU-SIMO channel matrix is i.i.d. Rayleigh. The baseline for performance comparison is the coherent linear minimum mean-square error (LMMSE) MIMO receiver with BPSK modulations at the transmitter side. For the symmetric 2-by-2 MU-MIMO full-rate (i.e., (4,4) code rate) communications,

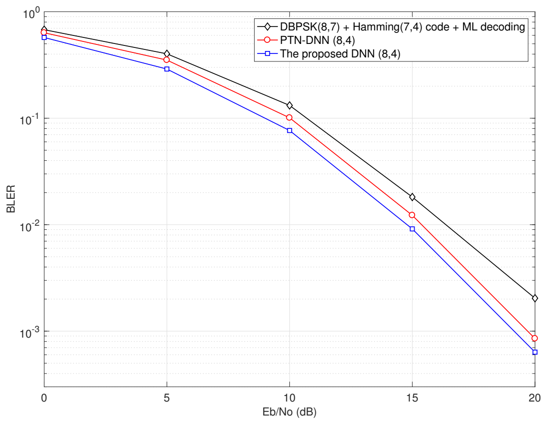


Fig. 2. BLER as a function of Eb/No for the SISO case.

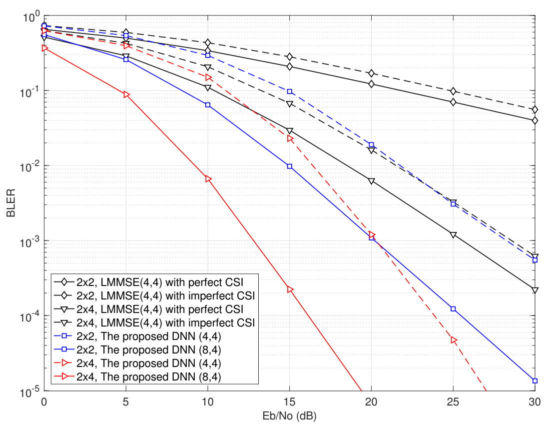


Fig. 3. BLER as a function of Eb/No for the MU-SIMO case.

it is found that the DNN approach outperforms the LMMSE receiver only at the high-SNR range (e.g. $E_{\rm b}/N_0 > 7$ dB). This is reasonable since the LMMSE receiver has the perfect channel knowledge for coherent detection. When the perfect channel knowledge is replaced by the least-square channel estimate using 4 pilots/transmitter, the DNN approach shows better performance almost throughout the whole SNR range. The most interesting phenomenon is that the DNN approach is able to exploit the channel spatial diversity gain in the symmetric MU-MIMO case. The BLER performance improves significantly when employing 2 more receive antennas. This coincides with the information-theoretic result for diversity gain [12]. In addition, it is also shown that half-rate (i.e., (8,4) code rate) MU-SIMO significantly outperforms the full-rate case thanks to the coding gain.

V. Conclusion

This letter presented a novel unsupervised deep learning approach for joint transmitter and noncoherent receiver design in MU-SIMO systems. The principle of the proposed DNN was mathematically studied. It was shown that the proposed DNN offered remarkable BLER performances in Rayleigh fading channels. When the proposed approach was applied to SISO, it slightly outperformed the state-of-the-art in performance whilst largely reduced the complexity due to the use of less nonlinear layers.

Appendix A
Proof of Proposition 1

In the noise-free case, the probability of missed classification is dominated by the random channel matrix $\mathbf{H}$. Suppose that DNN has been trained to categorize $\mathbf{Y}$ into the group $\mathbf{\Theta}_j$. The combinations $\mathbf{Y} = \mathbf{\Theta}_k\mathbf{H}_k^T$, $\forall k{\neq}j$, will be miss-classified into the group of $\mathbf{\Theta}_j$. Then, the probability of missed classification is

$$\mathrm{PMC}_{\mathrm{NF}} = \mathop{\mathbb{E}}_{\mathbf{Y}}\Big(\sum_{k\neq j} p(\mathbf{\Theta}_k\mathbf{H}_k^T|\mathbf{Y})\Big) \tag{11}$$

$$= \mathop{\mathbb{E}}_{\mathbf{Y}}\left(\frac{\sum_{k\neq j} p(\mathbf{\Theta}_k\mathbf{H}_k^T)p(\mathbf{Y}|\mathbf{\Theta}_k\mathbf{H}_k^T)}{p(\mathbf{Y})}\right) \tag{12}$$

$$= \mathop{\mathbb{E}}_{\mathbf{Y}}\left(\frac{\sum_{k\neq j} p(\mathbf{\Theta}_k\mathbf{H}_k^T)}{\sum_{k=0}^{J-1} p(\mathbf{\Theta}_k\mathbf{H}_k^T)}\right) \tag{13}$$

where Bayes' rule is employed to yield (12), and we applied $p(\mathbf{Y}|\mathbf{\Theta}_k\mathbf{H}_k^T) = 1$ and $p(\mathbf{Y}) = \sum_{k=0}^{J-1} p(\mathbf{\Theta}_k\mathbf{H}_k^T)$ to reach (13). Considering $\mathbf{H}_k$ and $\mathbf{\Theta}_k$ to be independent as well as the condition c2), (13) reads as

$$\mathrm{PMC} = \mathop{\mathbb{E}}_{\mathbf{Y}}\left(\frac{\sum_{k\neq j} p(\mathbf{H}_k)}{\sum_{k=0}^{J-1} p(\mathbf{H}_k)}\right) \tag{14}$$

Here, we utilize the equation $p(\mathbf{\Theta}_k\mathbf{H}_k^T) = p(\mathbf{\Theta}_k)p(\mathbf{H}_k)$, which is valid as $\mathbf{\Theta}_k$, $\forall k$, forms a finite-alphabet set. Given $\mathbf{H}_k^T = \mathbf{\Theta}_k^{-1}\mathbf{\Theta}_j\mathbf{H}_j^T$, (14) is equivalent to (5).

References

[1] A. K. Nandi and E. E. Azzouz, "Algorithms for automatic modulation recognition of communication signals," *IEEE Trans. Commun.*, vol. 46, no. 4, pp. 431–436, Apr 1998.
[2] A. Hamalainen and J. Henriksson, "A recurrent neural decoder for convolutional codes," in *IEEE Int. Conf. Commun.*, vol. 2, 1999, pp. 1305–1309 vol.2.
[3] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," *2017 IEEE 18th International Workshop on SPAWC*, pp. 1–5, 2017.
[4] N. Farsad and A. J. Goldsmith, "Detection algorithms for communication systems using deep learning," *CoRR*, vol. abs/1705.08044, 2017.
[5] A. Li, Y. Ma, S. Xue, N. Yi, and R. Tafazolli, "A carrier-frequency-offset resilient OFDMA receiver designed through machine deep learning," in *IEEE PIMRC*, Sept. 2018, pp. 1–5.
[6] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec 2017.
[7] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen Netzen. Diploma thesis, Institut für Informatik, Lehrstuhl Prof. Brauer, Technische Universität München," 1991.
[8] S. Hochreiter, Y. Bengio, and P. Frasconi, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," in *Field Guide to Dynamical Recurrent Networks*, J. Kolen and S. Kremer, Eds. IEEE Press, 2001.
[9] G.-B. Huang and H. A. Babri, "Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions," *IEEE Trans. Neural Netw.*, vol. 9, no. 1, pp. 224–229, Jan 1998.
[10] G.-B. Huang, "Learning capability and storage capacity of two-hidden-layer feedforward networks," *IEEE Trans. Neural Netw.*, vol. 14, no. 2, pp. 274–281, Mar 2003.
[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.
[12] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.