# End-to-end Learning for Physical Layer Communications

# End-to-end Learning for Physical Layer Communications

Sebastian Cammerer*, Sebastian Dörner*, Jakob Hoydis† and Stephan ten Brink*
* Institute of Telecommunications, Pfaffenwaldring 47, University of Stuttgart, 70659 Stuttgart, Germany
{cammerer,doerner,tenbrink}@inue.uni-stuttgart.de
†Nokia Bell Labs, Route de Villejust, 91620 Nozay, France, jakob.hoydis@nokia-bell-labs.com

## I. Introduction

Since Shannon's groundbreaking work on the fundamental limits of communications [1], engineers have been seeking to solve the task of "reproducing at one point either exactly or approximately a message selected at another point" [1] or, in other words, reliably transmitting a message from a source to a destination over a channel by the use of a transmitter and a receiver as illustrated in Fig. 1. *"Classical"* block-based signal processing has shown to be close to optimal while each sub-block can be optimized individually for a specific task such as equalization, modulation or channel coding.

At first glance, machine learning techniques do not appear to be a good match to communications on the physical layer, with 50 years of tremendous progress based on classic signal processing, communication and information theory, approaching close-to-optimal Shannon limit performance on many channels. However, several open problems remain, e.g., pertaining adaptivity and complexity of joint processing, where first results using machine learning-based approaches are promising (see [2], [3] and references therein).

Recently, the idea of deep learning (DL)-based communication was proposed in the literature [3], [4] based on the autoencoder concept ([5, Ch. 14]). In contrast to component-wise optimizations, the autoencoder approach now enables end-to-end training over any type of channel without the need for detailed prior mathematical abstraction of the channel model, breaking up restrictions commonplace in conventional block-based signal processing by moving away from handcrafted, carefully optimized sub-blocks towards adaptive and flexible (artificial) neural networks, leading to many attractive research questions. The benefits of machine learning approaches may include more flexible hardware, highly adaptive systems and less overall complexity. We thus pose the seemingly naive, yet, in fact, rather complicated and attractive research question: *Can we learn to communicate?*

We demonstrate the practical potential and viability of such a system by extending the idea of end-to-end learning of communications systems through deep neural network-based autoencoders to orthogonal frequency division multiplex (OFDM) with cyclic prefix (CP). This allows learning of transmitter and receiver implementations—without any prior

We would like to thank Maximilian Arnold for many helpful discussions.

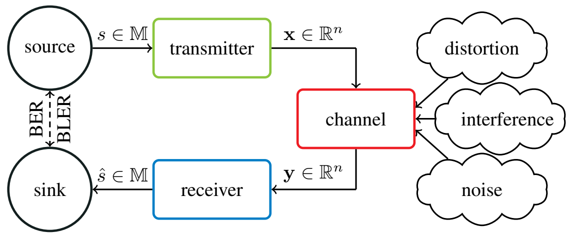


Fig. 1: Illustration of a simple communications system.

knowledge—that are optimized for an arbitrary differentiable end-to-end performance metric, e.g., block error rate (BLER). Our implementation shares the same benefits as a conventional OFDM system, namely single-tap equalization and robustness against sampling synchronization errors, which turned out to be one of the major challenges in prior single-carrier implementations [6]. We show that the proposed scheme can be realized with state-of-the-art deep learning software libraries, since transmitter and receiver solely consist of differentiable layers required for gradient-based training.

## II. Autoencoder-based Communication

As described in [3], a communications system can be interpreted as an autoencoder [5]. This is schematically shown in Fig. 2. An autoencoder describes a deep neural network consisting of various hidden layers that is trained to reconstruct the input (a so-called one-hot encoded vector representing one of the $m$ possible messages) at the output. As the information must pass each layer, the network needs to find a robust representation of the input message at every layer. In particular, the transmitter output (a real vector of dimension $n$) must be robust with respect to various channel impairments. Note that the channel is also represented by network layers (without trainable weights) that carry out stochastic transformations of the input data. It is crucial to have a good model that accurately reflects the real channel. The autoencoder is trained end-to-end using stochastic gradient descent (SGD). After training, the transmitter and receiver are fully described by their respective layer dimensions and weights and can operate in standalone mode to generate/process radio signals, e.g., on a software-defined radio (SDR) platform as shown in [6].

During training, the encoder part of the autoencoder has learned robust symbol sequence representations of all messages. Fig. 3 shows constellation diagrams of the IQ-symbols

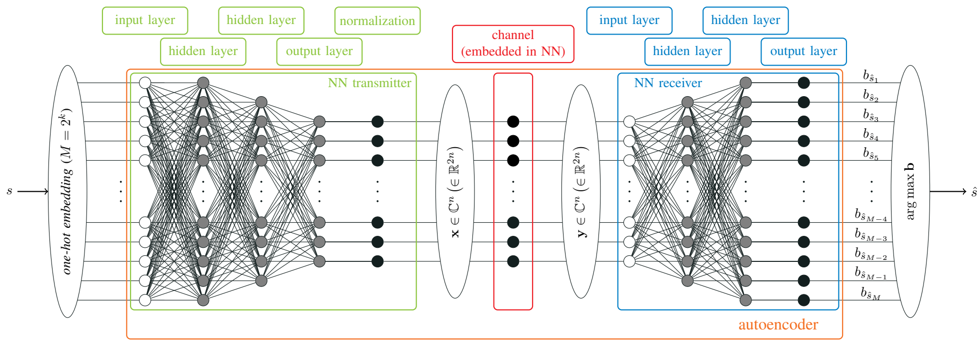


Fig. 2: Illustration of an end-to-end communications system as an autoencoder.

of all of the $m = 256$ possible messages of the single-carrier system, i.e., per subcarrier of the multi-carrier system. Each diagram shows all symbols at the same symbol position within a message, as each message consists of $\frac{1}{2}\log_2(m) = 4$ complex-valued IQ-symbols (we assume $n = 8$ and consider the first half of the transmitter output as the real and the second half as the imaginary part). Interestingly, we can observe that the autoencoder has learned some form of superimposed piloting since the center of the constellations is shifted away from the origin. For further details we refer to [6].

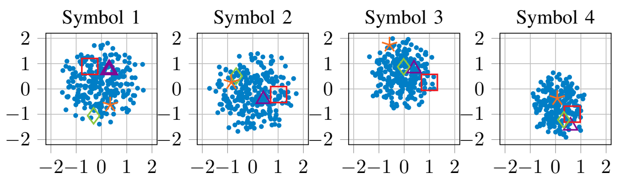


Fig. 3: Scatter plot of the learned constellations for all $M = 256$ messages using average power normalization $\|\mathbf{x}\|^2 \leq n$. The symbols of four individual messages are highlighted by different color markers.

## III. OFDM Extensions

We extend our work of [6] from single-carrier to multi-carrier, i.e., OFDM with CP as shown in Fig. 4. Note that a single autoencoder message $\mathbf{x}$ is represented by $\frac{n}{2}$ complex-valued IQ-symbols. Instead of directly transmitting the encoder's output $\mathbf{x}$, an inverse discrete Fourier-transform (DFT) of width $w_{\text{FFT}}$ is applied on a set of $w_{\text{FFT}}$ independent autoencoder messages, i.e., $w_{\text{FFT}}$ equivalent independent sub-channels are created, where independent autoencoder messages are assigned to each subcarrier.[1] As each autoencoder still requires $\frac{n}{2}$ channel uses, we generate $\frac{n}{2}$ complex-valued OFDM symbols $\mathbf{x}_{\text{OFDM}}$, each of length $w_{\text{FFT}}$. For additional robustness against sampling synchronization errors and to avoid inter-symbol interference (ISI) , we further add a CP of length $\ell_{\text{CP}}$, i.e., $w_{\text{FFT}}$ independent autoencoder symbols form one single OFDM symbol $\mathbf{x}_{\text{OFDM,CP}}$ of total length $w_{\text{FFT}} + \ell_{\text{CP}}$. Thus, a sequence of $\frac{n}{2}(w_{\text{FFT}} + \ell_{\text{CP}})$ complex-valued symbols is subsequently transmitted over the (mutlipath) channel.

[1]Remark: as no additional piloting is assumed, we cannot simply distribute the $\frac{n}{2}$ symbols of a message within the same OFDM symbol. Otherwise the unknown phase rotation per subcarrier would destroy the message.

At the receiver side, the CP can be used for frame synchronization through autocorrelation with peak detection; synchronization turned out to be a challenging step in singe-carrier autoencoder-based communication [6]. Finally, a DFT recovers the inputs for the $w_{\text{FFT}}$ independent autoencoder receivers.

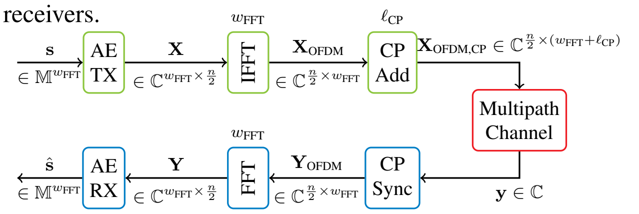


Fig. 4: OFDM extension to the autoencoder system.

At first glance it may appear counterintuitive that the autoencoder system benefits from such an explicit structure as it could also *learn* to compensate for these effects with a single (large) neural network. However, we observe for a *single* neural network that training complexity tremendously increases and practically limits the system performance (see [7]). Thus, the benefits of the proposed system are:

1) robustness against sampling synchronization errors
2) single-tap equalization[2]
3) moderate training complexity due to independent and short length sub-carrier messages (i.e., small $n$)

This enables reliable communication over multipath channels and makes the communication scheme suitable for commodity hardware with imprecise oscillators.

## References

[1] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. Journal*, vol. 27, pp. 379–423, 623–656, 1948.

[2] M. Ibnkahla, "Applications of neural networks to digital communications: A survey," *Signal Process.*, vol. 80, no. 7, pp. 1185–1215, 2000.

[3] T. J. O'Shea and J. Hoydis, "An introduction to machine learning communications systems," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, Dec 2017.

[4] T. J. O'Shea, K. Karra, and T. C. Clancy, "Learning to communicate: Channel auto-encoders, domain specific regularizers, and attention," *IEEE Int. Symp. Signal Process. Inform. Tech. (ISSPIT)*, pp. 223–228, 2016.

[5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[6] S. Dörner, S. Cammerer, J. Hoydis, and S. ten Brink, "Deep learning-based communication over the air," *IEEE Journal of Selected Topics in Signal Processing*, 2018.

[7] T. Gruber, S. Cammerer, J. Hoydis, and S. ten Brink, "On deep learning-based channel decoding," in *Proc. of CISS*, 2017.

[2]The autoencoder inherently has to *learn* how to synchronize.