DOI: https://doi.org/10.1109/CSNDSP49049.2020.9249528

A Survey about Deep Learning for Constellation Design in Communications

Manuel José López-Morales
Universidad Carlos III de Madrid
Madrid, Spain
manuelop@ing.uc3m.es

Kun Chen-Hu
Universidad Carlos III de Madrid
Madrid, Spain
kchen@ing.uc3m.es

Ana Garcia Armada
Universidad Carlos III de Madrid
Madrid, Spain
anagar@ing.uc3m.es

***Abstract*—The performance of communication systems depends on the choice of constellations, designed in an end-to-end manner. In case of a mathematical intractability, either because of complexity or even lack of channel model only sub-optimal solutions can be provided with an analytical approach. We present end-to-end learning, a recent technique in communications to learn optimal transmitter and receiver architectures based on deep neural networks (DNNs) architectures. We discuss cases in which this technique has been used to design constellations in which channel model intractability repressed from a mathematical analysis.**

***Index Terms*—Machine learning, end-to-end learning, channel imperfections, constellation design.**

I. Introduction

An important topic in digital communications is the design of constellation schemes for digital modulations. Over the years of digital communications [1], several constellation types have been defined. When designing constellations, we need to find the best location of the constellation points in transmission and the best decision regions in reception for those transmitted symbols. Therefore, this is an end-to-end problem which must be solved in both transmitter and receiver, to usually minimize the bit-error-rate (BER) or some other performance criteria.

From an information theory perspective, the aim is to maximize the mutual information $I(X_t; X_r)$ of the channel output X_r with the input X_t by finding an optimal constellation [2]. This can be solved through an analytical approach only when the channel distribution $f(X_t|X_r)$ is known. For simple scenarios, such as additive white Gaussian noise (AWGN) channels, a complete knowledge of $f(X_t|X_r)$ is available. Thus, optimal constellations can be designed mathematically [1].

There are other scenarios in which the channel distribution is extremely complex or even unknown, thus making it impossible to mathematically design constellations, unless several assumptions are made, that lead us to sub-optimal solutions. An example of a very complicated non-linear channel is the one present in [3]. In this case, high power amplifiers operate in compression to maximize SNR and power efficiency. This forces the use of lower-order or constant modulus constellations to cope with the severe non-linear distortion caused by the amplifier. Other non-linear channels are those present in fiber optics communications [4]. In these cases, assumptions must be made to make the problem tractable, thus leading to sub-optimal solutions.

Several approaches to the application of machine learning (ML) in telecommunications have been made in the last years [5]. With the emergence of deep learning (DL), researchers have started to work towards its application in communications networks [6]. Among the applications of deep learning in communications, which is an emerging field, there is a technique called end-to-end learning which aims at learning transmitter and receiver architectures to properly communicate under any imperfection or intractability of the channel [7]. Concretely, end-to-end learning can be applied to solve the problem of constellation design in the context of intractable channel models [8]–[10].

The aim of this paper is to serve as a survey in which the different approaches for designing constellations with ML are presented. Some examples of the use of this technique for different constellation design problems in communications are presented and explained. To the best knowledge of the authors, this is the first survey that specifically focuses on constellation design using ML tools.

The rest of this paper is organized as follows. Section II serves as a review of machine learning techniques. Section III introduces the classical approach for constellation design and explains the concept of end-to-end learning in communication systems. Section IV shows some of the results obtained from applying end-to-end learning for constellation design is different scenarios. Finally, Section V concludes the paper.

II. Machine Learning

Machine Learning is a subfield of artificial intelligence (AI) that provides a system with the ability to automatically learn and improve from experience without explicitly programming that system to do it. The learning process begins with the use of data, such as examples, direct experience or instruction, so as to look for patterns in data and perform better in the future based on the data provided. More information of any of the techniques explained can be found in [5].

This work has received funding from the European Union (EU) Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie ETN TeamUp5G, grant agreement No. 813391, and from the Spanish National Project TERESA-ADA (TEC2017-90093-C3-2-R) (MINECO/AEI/FEDER, UE).

A. Supervised Learning

Supervised machine learning algorithms starts from the analysis of a known training dataset and produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. Determining whether an image contains a certain object, would be a supervised learning problem, where the training data would include images with and without that object as input, and each image would have a label as output designating whether it contained the object. Defining $(x_i, y_i) \sim p(x, y)$, $i = 1, \ldots, N$ be samples taken from a training set $\mathfrak{D}$ belonging to a probability density function (PDF) $p(x, y)$, the aim is to find a mapping between all pairs of input-output. An inferred function $\boldsymbol{y} \approx \hat{f}(\boldsymbol{x}; \boldsymbol{\theta})$ would map input vector $\boldsymbol{x}$ to output $\boldsymbol{y}$, after it has learned the parameters $\boldsymbol{\theta}$. In an ideal scenario, unseen samples $\tilde{\boldsymbol{x}}$ will be mapped to the correct unknown target $\tilde{\boldsymbol{y}}$. Regression problems are those with continuous outputs, while problems with discrete output are classification problems. The learning process is approached through the definition of a cost function, which evaluates the quality of the predictions, and is defined as

$$C(\tilde{\boldsymbol{y}}) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathfrak{D}} \left\{\delta(\boldsymbol{y}, \tilde{\boldsymbol{y}})\right\} \tag{1}$$

where δ is a measure of distance between the wanted target $\boldsymbol{y}$ and the prediction $\tilde{\boldsymbol{y}}$, and E denotes expectation. Some examples of supervised learning techniques include: nearest neighbors, support vector machines and neural networks, among others.

B. Unsupervised Learning

Unsupervised learning is used when the information used to train is neither classified nor labeled. The aim is to infer a function to draw inferences from datasets to describe hidden structures from unlabeled data. Letting $\boldsymbol{x}_i \sim p(\boldsymbol{x})$, $i = 1, \cdots, N$ be sampled from a training set D belonging to the PDF $p(\boldsymbol{x})$, the aim is to extract certain features y which are not available in the training set D. Different tasks can be solved with unsupervised learning: clustering, which divides the data into clusters, feature extraction, which transforms data in a different latent space easier to handle and interpret, and synthesis of new samples, with the goal of learning the distribution $p(\boldsymbol{x})$ and to produce new samples from it. Unsupervised learning, contrary to supervised learning, does not have a unified accepted formulation. Many unsupervised learning tasks require the introduction of a hidden variable $\boldsymbol{z}_i$ for each sample, leading to the selection of different models under a probabilistic approach.

C. Reinforcement Learning

Reinforcement learning (RL) is a method that interacts with a dynamic environment by producing a series of actions and receives rewards according to the performance of such action with respect to the environment situation. The aim is to maximize the reward in the long-term. Basic RL can be modeled as a Markov Decision Process (MDP). Let S_t be the state environment provided to the agent at time t. The agent reacts by selecting an action A_t to obtain from the environment the updated reward Rt_{t+1}, and the next state S_{t+1}. In particular, the agent-environment interaction is formalized by a tuple $\langle S, A, T, r, \gamma\rangle$, where S is a finite set of states, A is a finite set of actions, $T(s, a, s') = P[S_{t+1} = s'|S_t = s, A = a]$ is the transition probability from state s to state s' under the action a, $r(s, a) = \mathbb{E}[R_{t+1}|S_t = s; A_t = a]$ is the reward function, and $\gamma \in [0, 1]$ is a discount factor. The agent builds a policy in order to find out which actions are good. $\pi : S\text{x}A \to [0, 1]$ defines the probability of taking an action a when state is s.

D. Deep Learning

Deep learning, a subset of ML, utilizes a hierarchical level of artificial neural networks to carry out the learning process, processing data with a nonlinear approach. Neural Networks (NNs) are among the most popular tools in the ML field, since they are known as universal function approximators [5] and can be trained by gradient backpropagation. A feedforward neural network with L layers maps a given input $\boldsymbol{x}_0 \in \mathcal{R}^{D_0}$ to an output $\boldsymbol{x}_L \in \mathcal{R}^{D_L}$ by implementing a function $\boldsymbol{f}(\boldsymbol{x}_0; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ represents the parameters of the NN. The input is processed through L successive steps

$$\boldsymbol{x}_l = \boldsymbol{f}_l(\boldsymbol{x}_{l-1}; \boldsymbol{\theta}_l), \quad l = 1, \cdots, L \tag{2}$$

where $\boldsymbol{f}_l(\boldsymbol{x}_{l-1}; \boldsymbol{\theta}_l)$ maps the input of the l-th layer to its output. The most used layer is the fully connected layer, defined as

$$\boldsymbol{f}_l(\boldsymbol{x}_{l-1}; \boldsymbol{\theta}_l) = \sigma(\boldsymbol{W}_l \boldsymbol{x}_{l-1} + \boldsymbol{b}_l) \tag{3}$$

where $\boldsymbol{W}_l$ and $\boldsymbol{b}_l$ are weights between layers and bias of each layer, respectively. Meanwhile, $\sigma(\cdot)$ is the activation function, which has to be non-linear in order to benefit from the multi-layer structure. Depending on the application, several different types of layers and activation functions can be defined.

III. End to End Learning

A. Classical Design Approach

Classically, constellation design main goal is to maximize the mutual information $I(X_t; X_r)$ of the channel output X_r with the input X_t, which is equal to maximizing channel capacity $C = \max_{p(x_t), X_t} I(X_t; X_r)$, where $p(x_t)$ is the marginal distribution of X_t. Both $p(x_t)$ and $f(X_t|X_r)$ must be known to solve the problem of capacity maximization. For simple scenarios, such as additive white Gaussian noise (AWGN), a complete knowledge of $f(X_t|X_r)$ is available. If $p(x_t)$ is uniform, the symbols in the constellation are equally probable and the maximum likelihood is used as a decision criterion

$$x_r^{ML} = \arg\max_{X_r, X_t} \{f(x_r|x_t), x_r \in X_r, x_t \in X_t\}. \tag{4}$$

When $f(x_r|x_t)$ is unknown or too complicated to be mathematically tractable, we cannot rely on the maximum likelihood approach to solve the problem of constellation design.

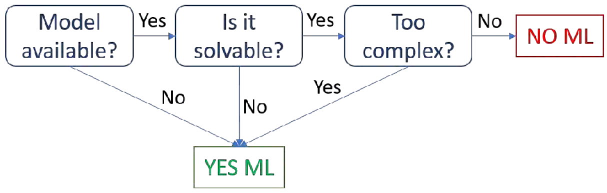

Fig. 1: Should you investigate ML in your problem?

B. End-to-end Learning

The application of machine learning in any discipline must be assesed following the scheme in Fig. 1.

End-to-end learning aims to learn transmitter and receiver implementations optimized for a specific performance metric and channel model. It was first presented in [7]. The initial idea is to interpret the whole communication chain as an autoencoder, an unsupervised learning technique. In communications, the autoencoder wants to learn representations x_t of a set of possible messages s that are transmitted through a channel with impairments and then recover the message s in reception as $\hat{s}$, with a small error probability. The main limitation relates to the training of the whole autoencoder, since the channel must be represented as a neural network, i.e. it must be differentiable. For any real system, the channel is a black-box with an unknown transfer function $f(x_r|x_t)$, so gradient backpropagation is impossible. To overcome this problem, 3 main approaches are proposed:

- Analytical channel model and receiver fine-tuning [11].
- Learn a generative channel model [12].
- Reinforcement learning in transmission [9].

The first approach starts with end-to-end training on an analytical channel differentiable model $f(x_r|x_t)$. It does not cover all hardware and channel inefficiencies, so the performance is limited by the accuracy of the model. The second step is to fine-tune the receiver by using a physical realistic channel. The TX weights are fixed and we only train the receiver. This approach benefits from an easy fine-tuning of the receiver but lacks of not being able to fine-tune the transmitter. Another limitation is that an analytical channel model is required, which is never available for completely unknown channels. It is easy to implement, since it is a simple supervised training, but maximal performance can never be reached due to transmitter mismatch.

The second approach first learns a generative channel model from data and then train the full autoencoder over the learned channel model. Generative adversarial networks (GANs) [13], whose aim is to learn to mimic the distribution of any type of data, can be used to learn such generative channel model. GANs are composed of the generator $G(z)$ and the discriminator $D(x)$. The former generates new data samples using a latent variable z (typically Gaussian noise) and tries to deceive the discriminator, which attempts to distinguish fake from real samples. For communications, conditional GANs are proposed, since the generator is varied to $G(z, x_t)$, which accounts for the presence of a signal x_t to be transmitted over the channel. This is a very elegant method to learn, purely from observations, a NN implementation of a channel model. Unfortunately, it is unclear how well GANs are capable of modeling complex channels with a lot of randomness.

The third approach consists of implementing the transmitter as an agent of reinforcement learning. In this case, no backpropagation is needed from the receiver to the transmitter, but some ”reward” signal should be seen by the transmitter to perform training. For instance, this reward signal can be defined as $R = -\log_{10}(BER)$, since for smaller values of BER, turns into bigger values. Another definition could be $R = 1/BER$. The training process would be performed alternatively between receiver and transmitter. Both start with a random initialization, receiver is trained for fixed transmitter for a certain receiver training time T_{tr}^{RX}, then transmitter is trained for fixed receiver for a certain transmitter training time T_{tr}^{TX}. This is repeated several times until a certain stop criteria is met. This approach benefits from the fact that it can train over any type of channel without any modelling needed and can be applied to real systems without modification, but its convergence is slower and it is very difficult to do hyperparameter tuning.

IV. Examples of Constellations Design using End-to-end Learning

In the literature, constellations have been designed substituting the whole transmitter and receiver architectures by neural networks [2], [7], [8], [10], [14]–[17]. In these papers, the transmitter and receiver only purpose is to modulate and demodulate, that is why whole transmitter and receivers where substituted by neural networks and trained.

The concept of end-to-end learning was proposed and applied to the two-user interference channel in [7]. In [14], O'Shea showed the result of the application of end-to-end learning to the problem in [3]. The autoencoder is capable of creating an encoder network that learns a novel transmit waveform optimized for the non-linear compression caused by the amplifiers. The decoder network in the receiver is capable of yielding a separable 32-symbol QAM-like constellation. This is shown in Fig. 2.

The authors in [8] applied the first approach of end-to-end learning explained in Section III to intensity modulation/direct detection (IM/DD) fiber optics communications. Both the constellations and pulse shapes are learned with the use of neural networks and it is shown they can achieve bit error rates below 6.7% hard-decision forward error correction (HDFEC) threshold.

In [2], both geometric and probabilistic constellation shaping is learned to get constellations that get closer than state-of-the-art constellations to Shannon capacity limit. The transmitter structure is adapted to account for the probability of a symbol and the geometry of the constellation. The shape learned by the system is similar to a two-dimensional Gaussian distribution as shown in Fig. 3. The points with a greater occurrence are placed in the center, and this occurrence is greater for lower SNRs.

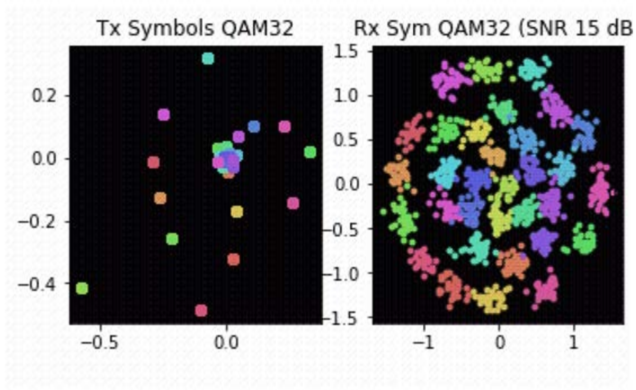

Fig. 2: Fig. 5 in [14]. Learned transmit constellation (left), received constellation (right), for an SNR of 15 dB.

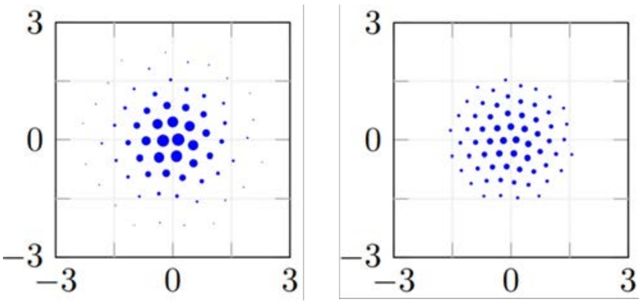

Fig. 3: Fig. 5 of [2] for an AWGN channel with constellation size 64 and SNR 5dB (left) and 18dB (right). Points size proportional to probabilities of occurrence.

In [15], Matsumine et. al. applied end-to-end learning for the optimization of constellations in two-way relaying with physical-layer network coding. DNN-based modulation and demodulation are employed at each terminal and relay node. Jones et. al. [16] did the same for geometric constellation shaping including fiber nonlinearities. The channel model included modulation dependent nonlinear effects. The algorithm yields a constellation mitigating them, with gains up to 0.13 bit/4D. In [17], Albergue used end-to-end learning in non-orthogonal multiple access to define the constellations of users in the downlink. Cammerer et. al. showed in [10] the constellations learned when end-to-end learning is performed in the context of bit-metric decoding receivers. Joint optimization of constellation shaping and labeling is performed for iterative demapping and decoding receivers. Fig. 4 shows that, when only constellation shaping is done, the results are different from those with both shaping and labeling.

V. Conclusions

In this paper, we make a brief survey about machine learning techniques and end-to-end learning. End-to-end learning consists of a technique in which full transmitter and receiver architectures based on DNNs can be learned. We present the different ways to perform the training of end-to-end learning

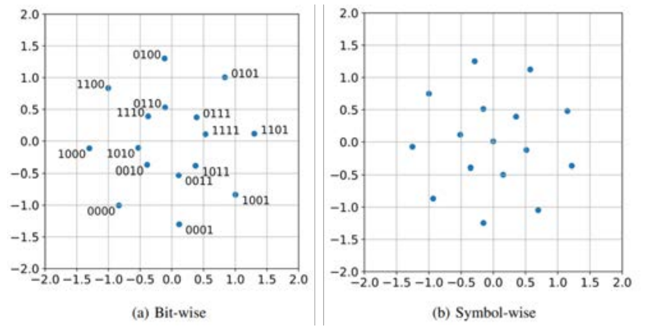

Fig. 4: Constellation and labeling for the bit-wise autoencoder (left) and without bit labeling (right). Obtained for 4 bits/symbol and SNR=4dB. Results come from Fig. 4 of [10].

and some cases in which end-to-end learning has been used for constellation design.

References

[1] Proakis, *Digital Communications 5th Edition*. McGraw Hill, 2007.

[2] M. Stark, F. A. Aoudia, and J. Hoydis, "Joint Learning of Geometric and Probabilistic Constellation Shaping," 2019.

[3] "Tracking and Data Relay Satellite, National Aeronautics and Space Administration (NASA)," https://www.nasa.gov/directorates/heo/scan/services/networks/tdrs_main, accessed: 2020-02-14.

[4] E. Agrell, A. Alvarado, G. Durisi, and M. Karlsson, "Capacity of a Nonlinear Optical Channel With Finite Memory," *Journal of Lightwave Technology*, vol. 32, no. 16, p. 2862–2876, Aug 2014.

[5] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. Chen, and L. Hanzo, "Machine Learning Paradigms for Next-Generation Wireless Networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, April 2017.

[6] C. Zhang, P. Patras, and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2224–2287, thirdquarter 2019.

[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] B. Karanov, M. Chagnon, F. Thouin, T. A. Eriksson, H. Bulow, D. Lavery, P. Bayvel, and L. Schmalen, "End-to-End Deep Learning of Optical Fiber Communications," *Journal of Lightwave Technology*, vol. 36, no. 20, p. 4843–4855, Oct 2018.

[9] F. A. Aoudia and J. Hoydis, "End-to-End Learning of Communications Systems Without a Channel Model," 2018.

[10] S. Cammerer, F. A. Aoudia, S. Dörner, M. Stark, J. Hoydis, and S. ten Brink, "Trainable Communication Systems: Concepts and Prototype," 2019.

[11] S. Dörner, S. Cammerer, J. Hoydis, and S. t. Brink, "Deep Learning Based Communication Over the Air," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 132–143, Feb 2018.

[12] H. Ye, G. Y. Li, B. F. Juang, and K. Sivanesan, "Channel agnostic end-to-end learning based communication systems with conditional gan," in *2018 IEEE Globecom Workshops (GC Wkshps)*, Dec 2018, pp. 1–5.

[13] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, "Recent progress on generative adversarial networks (gans): A survey," *IEEE Access*, vol. 7, pp. 36 322–36 333, 2019.

[14] T. R. Ben Hilburn, Tim O'Shea and N. West, "DeepSig: Deep Learning for Wireless Communications," Nvidia Blog, 2018.

[15] T. Matsumine, T. Koike-Akino, and Y. Wang, "Deep learning-based constellation optimization for physical network coding in two-way relay networks," 2019.

[16] R. T. Jones, T. A. Eriksson, M. P. Yankov, and D. Zibar, "Deep learning of geometric constellation shaping including fiber nonlinearities," in *2018 European Conference on Optical Communication (ECOC)*, Sep. 2018, pp. 1–3.

[17] F. Alberge, "Constellation design with deep learning for downlink non-orthogonal multiple access," in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sep. 2018, pp. 1–5.