Model-Aided Wireless Artificial Intelligence: Embedding Expert Knowledge in Deep Neural Networks Towards Wireless Systems Optimization

Alessio Zappone, *Senior Member, IEEE*, Marco Di Renzo, *Senior Member, IEEE*, Mérouane Debbah, *Fellow, IEEE*, Thanh Tu Lam, and Xuewen Qian

***Abstract*—Deep learning based on artificial neural networks is a powerful machine learning method that, in the last few years, has been successfully used to realize tasks, e.g., image classification, speech recognition, translation of languages, etc., that are usually simple to execute by human beings but extremely difficult to perform by machines. This is one of the reasons why deep learning is considered to be one of the main enablers to realize the notion of artificial intelligence. The current methodology in deep learning methods consists of employing a data-driven approach in order to identify the best architecture of an artificial neural network that allows one to fit input-output data pairs. Once the artificial neural network is trained, it is capable of responding to never-observed inputs by providing the optimum output based on past acquired knowledge. In this context, a recent trend in the deep learning community is to complement pure data-driven approaches with prior information based on expert knowledge. This work describes two methods that implement this strategy in the context of wireless communications, also providing specific case-studies to assess the performance compared to pure data-driven approaches.**

I. Introduction and motivation

Recently, deep learning has received attention as a technique to design and optimize wireless communication systems and networks, by employing fully data-driven approaches. We believe, however, that the application of deep learning to communication networks design and optimization offers more possibilities. As opposed to other fields of science, such as image classification and speech recognition, mathematical models for communication networks optimization are very often available, even though possibly simplified. We believe that this a priori expert knowledge, which has been acquired over decades of intense research, cannot be dismissed and ignored. In the present work, in particular, we put forth a new approach that capitalizes on the availability of (possibly simplified) theoretical models, in order to reduce the amount of empirical data to use and the complexity of training artificial neural networks (ANNs). Unlike other application fields, we concretely show, with the aid of some examples, that synergistically combining prior expert knowledge based on analytical models and data-driven methods is a suitable approach towards the design and optimization of communication systems and networks with the aid of deep learning based on ANNs.

The authors are with the Laboratory of Signals and Systems (CNRS - CentraleSupelec - Univ. Paris-Sud), Université Paris-Saclay, 3 rue Joliot-Curie, 91192 Gif-sur-Yvette, France, (alessio.zappone@l2s.centralesupelec.fr, marco.direnzo@l2s.centralesupelec.fr).

II. Artificial Intelligence by Deep Learning

Artificial intelligence (AI) broadly refers to the ability of machines to mimic the process of human intelligence. This is typically achieved through machine learning techniques, which enable machines to perform tasks by processing and learning from available data, instead of running a fixed computer program explicitly written for the problem at hand.

In the context of AI, deep learning is a specific machine learning method that implements the learning process by employing ANNs [1]. In principle, ANNs are able to operate in a fully data-driven fashion, thus dispensing system designers with the need of mathematical modeling and expert supervision of knowledge. When large datasets are available, moreover, deep learning is known to outperform other machine learning techniques. These features have made deep learning the most widely used machine learning technique in fields such as image classification, speech recognition, translation between languages, autonomous driving, etc., for which a mathematical description of the task to be executed is particularly challenging to be obtained. In these areas of research, however, an emerging opinion is that pure data-driven approaches may become unfeasible in the context of large-scale applications, due to the huge amount of required data, and to the related processing complexity. In [2], for example, image processing for object position detection in robotic applications is considered, and it is observed that augmenting a small training dataset of real images with a large dataset of synthetic images significantly improves the estimation accuracy with respect to processing only the small dataset of real images. Similar results have been obtained in [3] for application to speech recognition.

III. Deep Learning in Communications: Why Now?

Before discussing potential techniques to merge expert knowledge and deep learning, we discuss why deep learning is emerging now as a valuable tool for wireless networks design.

General machine learning techniques are not new to wireless communications [4], [5], even though the use of deep learning has never been considered in the past. In our opinion, this is mainly due to the fact that, unlike other fields of science where theoretical modeling is particularly difficult to be performed and a data-driven approach is often the only solution available, wireless communications have always relied on strong theoretical models for their system design and optimization. This status quo, however, is rapidly changing, and very recently the use of deep learning has started being envisioned for wireless communication applications. Indeed, the increasing complexity of wireless networks makes it harder and harder to come up with theoretical models that are at the same time accurate and tractable. The rising complexity of 5G and beyond 5G networks is exceeding the modeling and optimization possibilities of standard mathematical tools. In addition, the use of deep learning for communications is further facilitated by:

- The recent exponential growth of connected devices, which provide communication network designers with an increasing amount of data to process.
- The recent technological improvements and enriched capabilities of specialized hardware for data processing (e.g., GPUs), which make deep learning algorithms applicable in practice in the context of wireless network optimization.
- The recent development of technologies (e.g., the Blockchain) that facilitate the secure and accurate processing of large databases that are distributed over multiple network nodes.

In addition to the above-mentioned enabling factors, it is our personal opinion that the use of deep learning for application to wireless communications provides communications theorists and engineers with another major opportunity. As discussed in Section II, an emerging research trend in the deep learning community is the development of techniques that exploit prior information that is available about the problem to solve. In the context of wireless communications, this represents a great opportunity, because theoretical models, despite their possible inaccuracy or cumbersomeness, are often available and provide much deeper prior information compared to other fields of science. This clear advantage of wireless communications should not be wasted. Accordingly, the aim of this work is to corroborate the intuition that available theoretical models and frameworks can indeed provide enough expert knowledge to facilitate the use of deep learning for application to wireless networks design. To this end, two main methods of embedding expert knowledge into deep learning techniques are discussed, and three specific case studies are analyzed.

IV. Learning to Optimize by Deep Neural Networks

A fundamental component of wireless networks management and operation is the allocation of the available resources to optimize desired performance functions, ensuring guaranteed performance to each user. Depending on the complexity of the system, the four scenarios in Table I can be identified.

Table I
SCENARIOS FOR RESOURCE MANAGEMENT IN WIRELESS NETWORKS.

C.1: An accurate and tractable theoretical model is available (e.g., point-to-point channel capacity, point-to-points bit error rates).
C.2: An accurate but intractable theoretical model is available (e.g., achievable sum-rate in interference-limited systems).
C.3: A tractable but inaccurate theoretical model is available (e.g., spectral / energy efficiency of ultra-dense networks, energy consumption models, hardware impairments).
C.4: Only inaccurate and intractable theoretical models are available (e.g., molecular communication networks, optical systems, end-to-end networks optimization).

As far as this work is concerned, the most interesting scenarios are represented by Cases C.2 and C.3. Cases C.1 and C.4, in fact, can be handled by traditional system design approaches and pure data-driven techniques, respectively. Cases C.2 and C.3, on the other hand, offer the opportunity of cross-fertilization between model-aided and data-driven approaches, as discussed in the next two sections. Moreover, the availability of theoretical models, although possibly inaccurate or intractable, represents the most common scenario in wireless communications.

A. *Learning to Optimize a Model*

Let us assume to be in Case C.2. Then, a mathematical formula for the performance metric to optimize is available, but it is too complex to be maximized by using traditional optimization theory methods, e.g., by directly tackling the optimization problem requires an exponential complexity in the number of variables to optimize. It is important to stress that in this case the issue is not solving the optimization problem, but rather the complexity and the time required to do so. This is a critical problem, for example, in mobile scenarios, wherein the network status changes rapidly (e.g., a user joins/leaves the network, the channel realizations/statistics change, new traffic requests occur, etc.) and thus the optimal resource allocation needs to be updated each time the network scenario changes (so, very often), thus making real-time implementation unfeasible.

In this context, the joint use of deep learning and traditional optimization theory can significantly speed up the computation of the optimal resources. The key idea is based on two observations:

- Resource allocation can be regarded as the problem of determining the map between the system parameters (e.g., the propagation channels, the number of active nodes, the users' positions, etc.) and the corresponding optimal resource allocation to use.
- ANNs are known to be universal function approximators, i.e., they can be trained to learn, virtually, any input-output map [6].

As a result, our proposed idea is to configure an ANN, whose inputs are the system parameters and whose outputs are the resources to allocate, so that its input-output relationship approximates the map between the system parameters and the optimal resource allocation. Once the ANN has been configured, it is possible to update the resource allocation without having to solve any optimization problems in real-time, i.e., every time that the system configuration changes. Indeed, the new system configuration needs only to be fed as the input of the already configured ANN, and the corresponding optimal resource allocation is obtained as the output of the ANN. This entails a negligible computational complexity, since an ANN performs only a composition of affine combinations and elementary function evaluations to compute its output.

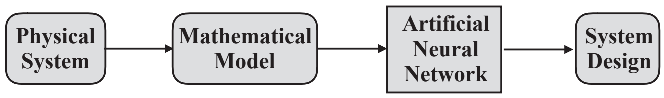

Figure 1. A training set is built from a mathematical model and used to train an ANN to perform resource allocation.

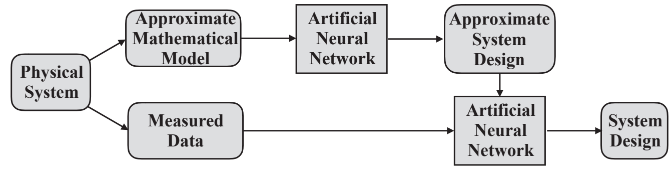

Figure 2. An approximate model is used to perform an initial training phase, which is later refined through a second training phase performed by employing a training set built from measured data.

Based on these considerations, the main issue to be discussed is the complexity required to configure the ANN. Unfortunately, the result from [6] is not constructive, since it only establishes that a suitable configuration exists. For this reason, the ANN needs to be trained in a supervised fashion to learn the desired map, which requires to process a *training set*, i.e., a dataset containing examples of system configurations with the corresponding optimal resource allocation. Thus, the complexity of the training process is due to the processing of the training set and to its generation. The former is accomplished by efficient, off-the-shelf, stochastic gradient descent algorithms that adjust the ANN parameters in order to reduce the error between the actual output and the desired training output. The latter, instead, is more computationally intensive, because it requires to solve the resource optimization problem by (traditional) optimization techniques. At a first sight, it might be argued that this defeats the complexity gain granted by the trained ANN. However, we stress that this is not the case for two main reasons:

- The training set can be generated *off-line* and then used to train the ANN. Thus, a much higher complexity can be afforded and real-time constraints does not constitute an issue for this phase.
- The training set can be updated at a much longer time-scale as opposed to the rate change of the network parameters.

Thus, the training process does not need to be executed each time a system parameter changes, and the solution does not need to be obtained before another system parameter changes. The use of traditional optimization theory to generate the training set together with the use of an available theoretical system model, on the other hand, constitute exactly the expert knowledge that is exploited to facilitate the use of deep learning techniques to perform real-time resource allocation strategies in wireless communication networks.

The approach described in this section is schematically depicted in Fig. 1. In a nutshell, the available accurate model is used to efficiently training the ANN, which can then be used for the efficient implementation of real-time resource allocation strategies.

B. Learning to Refine a Model

Let us assume to be in Case C.3. A tractable model is then available, but it is not sufficiently accurate. Nevertheless, even inaccurate models can provide useful information that should not be dismissed. In general, employing a fully data-driven approach to train an ANN requires the need of acquiring a huge amount of live data, which might not be practical due to the time, the complexity, or the economical reasons that this process entails. Instead, the availability of an approximate model can be exploited to perform a first rough training of the ANN, which can be subsequently refined only through a small set of real data.

More precisely, we propose the following approach:

- First, a training set based on a (possibly approximate) model is obtained, by using conventional optimization theory, as described in Section IV-A.
- Then, an ANN is trained by processing the generated training set, as discussed in Section IV-A. This provides an initial configuration for the ANN.
- Finally, the pre-trained ANN architecture is refined through a new training phase based on real/measured data (i.e., input-output pairs corresponding to the optimal network configuration).

Intuitively, the first training phase provides one with an estimate of the desired ANN configuration. This initialization may reduce the amount of measured data that is required in the second phase, when compared with the conventional approach of training an ANN by relying solely on measured data, without any initial guess of the ANN parameters. The method is schematically depicted in Fig. 2.

The attractive feature of the proposed method lies in exploiting the available expert knowledge to select an efficient starting point for the second training phase. Selecting an efficient initialization point when training an ANN is recognized to be a relevant issue that significantly affects the performance of ANNs. To date, however, only heuristic methods or random initializations have been proposed in the deep learning literature [7], [8]. Instead, the initialization method that we propose provides one with a stronger theoretical justification, since it approximates the desired ANN configuration based on the expert knowledge at our disposal. Clearly, the performance of the proposed approach strongly depends on the accuracy of the model that is used. If the difference between the model and the real system is not too large, and the initial training phase is performed on enough data, then the second training phase will start from an ANN configuration that is already close to the desired configuration, and thus only a small dataset and a few gradient iterations might suffice to refine the ANN setup. On the other hand, if the model is a bad approximation of the true system, or a small training set is used, then the initial training might yield a misleading ANN configuration, thus requiring even more data in the second phase to obtain a good solution.

V. Applications

In this section, we consider three case-studies in order to substantiate the discussed model-aided ANN approaches. Our objective is two-fold. First, we show that, by using the approach introduced in Section IV-A, an ANN can be used to implement real-time resource allocation schemes that are too complex to be handled by available optimization-theoretic approaches. Then, based on the methodology introduced in Section IV-B, we show that an ANN can be first roughly trained by using (large) datasets based on analytical, but inaccurate, models, and subsequently fine-tuned by datasets of live data with limited size compared with what would be required if the initial training based on analytical models is not performed.

A. Real-Time Energy Efficiency Maximization in Multi-user Networks

Consider the uplink of a multi-user network with interfering mobile terminals. The objective is to allocate the transmit powers of the users in order to maximize the network bit-per-Joule energy efficiency, defined as the ratio between the system sum achievable rate over the total network power consumption. In this scenario, a model to formulate the energy efficiency optimization problem is available, but the presence of multi-user interference makes the problem too complex to be globally solved at an affordable computational complexity [9]. This is especially problematic when the optimization is performed based on instantaneous channel realizations, and thus needs to be performed anew every time the channel coefficients change. This causes a considerable complexity overhead preventing real-time implementations. The problem can be overcome by the approach from Section IV-A. We illustrate it by using a simple example.

A circular area with radius 500 m and 10 mutually interfering mobile users is considered. In order to build a training set, 10,000 independent scenarios are generated by randomly dropping the users in the service area, modeling the propagation losses according to [10] and the fading channels as standard complex Gaussian random variables. For each scenario, the optimal energy-efficient power allocation strategy is computed off-line by fractional programming [11]. Accordingly, 10,000 training samples are obtained.

A feedforward ANN with rectified linear unit activation functions and 10 layers is considered. Layers 1 and 2 have 18 neurons, and the number of neurons of the other layers decreases by 2 every two layers. Thus, the output layer has 10 neurons, providing the users' transmit powers.

The performance of the trained ANN is evaluated over a test set of 10,000 new channel scenarios, generated following the same procedure used for the training set, but with independent users' drops and channel realizations. Fig. 3 compares the average (over the test set) energy efficiency obtained using the trained ANN, and the global optimum obtained by fractional programming, versus the maximum feasible transmit power P_{max}. The energy efficiency obtained when all users transmit with power P_{max} is reported as well. It is observed that, despite the much lower complexity, the ANN-based scheme is optimal for low P_{max}, and near-optimal for larger P_{max}, where it achieves around 95% of the optimal value.

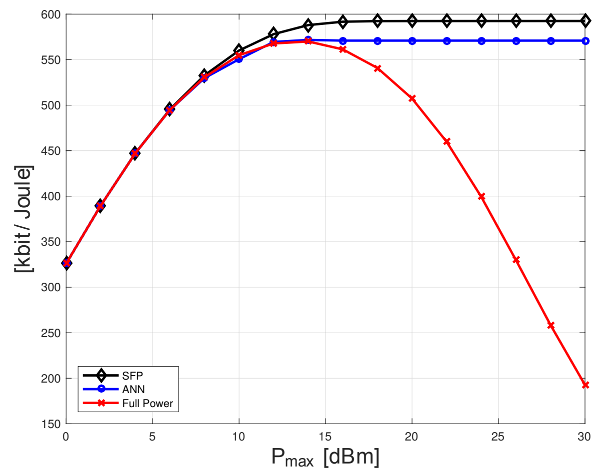

Figure 3. Energy efficiency versus P_{max} by: (Black line) Method from [11]; (Blue line) Deep learning by using ANN; (Red line) Full power allocation.

B. Energy Efficiency Maximization in non-Poisson Cellular Networks

Consider the problem of optimizing the deployment density of a cellular network, given the transmit power of the base stations, and aiming at energy efficiency optimization. By assuming that the base stations are distributed according to a Poisson point process, an accurate and realistic analytical model energy efficiency optimization has been recently proposed in [12], and the optimal deployment density of the cellular base stations has been formulated in a tractable analytical form. Leveraging [12], large datasets can be generated with low computational effort, which provides system designers with the optimal base station density as a function of the base stations transmit power. It is known, on the other hand, that similar tractable analytical frameworks cannot be easily obtained if the cellular base stations are distributed according to non-Poisson spatial models, which makes the energy efficiency optimization of such network cellular deployments very difficult [13]. In addition, the generation of large datasets based on non-Poisson point processes is a time and memory consuming task, which makes it difficult to obtain large datasets containing the optimal data pairs (optimal deployment density, transmit power) for training purposes.

The two-step approach introduced and described in Section IV-B is thus motivated. We consider an ANN whose objective is to provide the system designer with the optimal deployment density of the base stations that optimizes the energy efficiency for a given transmit power of the base stations. Thus, the transmit power of the base stations is the input of the ANN and the optimal deployment density of the base stations is the output of the ANN. It is assumed, in particular, that the base stations are distributed according to a non-Poisson point process, whose exact distribution is not known, and that just (some) empirical datasets are available about the locations of the base stations.

We aim at understanding whether by performing an initial training of the ANN based on a large, Poisson-based dataset, followed by a second training based on a smaller dataset of empirical (or synthetic from simulations) data, we can obtain similar performance as using only a large training set of real data. To answer this question, the following approach is used:

1) A large dataset is generated, by assuming that the base stations are distributed according to a Poisson point process, and by computing the optimal base station density by using the analytical framework recently proposed in [12]. The ANN is first trained by using the obtained dataset.
2) A smaller dataset is generated, by assuming the actual (non-Poisson) spatial distribution for the locations of the base stations and by computing the optimal transmit power through an exhaustive search. The pre-trained ANN is refined by employing this second training set.

The results are illustrated in Fig. 4, where the horizontal axis reports the amount of non-Poisson empirical data used to train the ANN and the vertical axis shows the mean relative square error. Both the training error and the validation errors are reported. It is assumed that the total amount of data used for training the ANN is 10,000 samples. In the horizontal axis, x% (of 10,000 samples) denotes the amount of non-Poisson data used during the refinement phase. This implies that (100-x)% (of 10,000 samples) is the amount of data used for the initial training of the ANN based on the Poisson data obtained from the analytical framework in [12]. A square grid model is considered, as far as the non-Poisson model is concerned [13]. For the ANN network architecture summarized in the caption of the figure, we observe that there exist a few values of x% for which the mean relative square error of training and validation phases are smaller than the corresponding values obtained by training the ANN by using only non-Poisson datasets. This is because the gradient search used for training is better initialized and the refinement phase can converge to a better solution. On the other hand, if the amount of Poisson data used for the first training phase is too small, the first phase may not converge appropriately, which may result in a bigger relative mean square error. This result highlights the potential of using (even inaccurate) analytical models to better train ANNs, but, at the same time, the need of judiciously choosing the amount of data to use from the assumed model and from the actual empirical data.

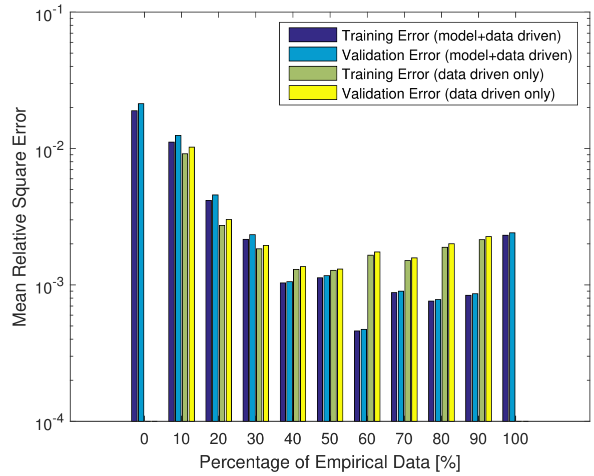

Figure 4. Feedforward ANN with 5 layers, 4 neurons in each hidden layer, and sigmoidal activation functions. The ANN is trained by using the Bayesian regularization back propagation algorithm that uses, at most, 100 iterations for each training phase. The bars corresponding to 0% and 100% empirical data are obtained by training the ANN with only Poisson and non-Poisson data, respectively. Training and validation errors are computed from 1,000 samples of non-Poisson data.

C. Energy Efficiency Maximization with Unknown Power Consumption Models

In this case study, we consider again the problem of optimizing the deployment density of a cellular network, given the transmit power of the base stations and by considering the energy efficiency as the key performance indicator of interest. As opposed to the first case study, we assume that the cellular base stations are distributed according to a Poisson point process, which is considered to be accurate for the application of interest. We assume, however, that only a simplified statistical model for the static and idle hardware power consumptions of the cellular base stations is available. Details about the definitions of static and idle power consumptions can be found in [12]. Specifically, we consider that the static and idle hardware power consumptions are distributed according to two uniform random variables with some given mean and variance. Although tractable, this model is clearly a rough approximation of the actual hardware power consumption, which will in practice deviate from the considered model. Nevertheless, the numerical results that are shown in this section confirm that even such a simple model can provide us with enough prior information.

Based on the adopted uniform model, the optimal deployment density of the cellular base stations is computed, as a function of the transmit power and of the static and idle power consumptions, by employing the optimization framework proposed in [12], which allows us to easily produce large datasets for training ANNs. Subsequently, we generate a smaller dataset based on the actual realizations of the static and idle hardware power consumptions, which, for ease of reproducibility, are assume to follow a Gaussian model with fixed mean and variance. We note that, as opposed to the case study in the previous section, the considered ANN has three inputs (the transmit power of the base stations, the static power consumption, the idle power consumption) and one output (the optimal deployment density of the base stations), and so it is more difficult to train as opposed to the one-input and one-output ANN considered in the previous case study.

With these two datasets available, we use a similar procedure as the one discussed in Section V-B to train the considered ANN:

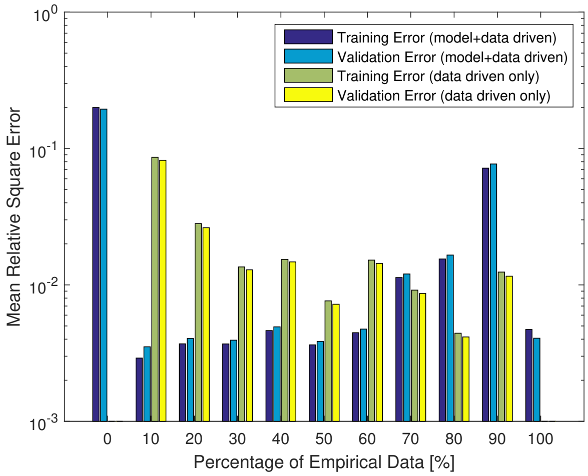

Figure 5. Feedforward ANN with 6 layers, 4 neurons in each hidden layer, and sigmoidal activation functions. The ANN is trained by using the Bayesian regularization back propagation algorithm that uses, at most, 250 iterations for each training phase. The bars corresponding to 0% and 100% empirical data are obtained by training the ANN with only Poisson and non-Poisson data, respectively. Training and validation errors are computed from 1,000 samples of non-Poisson data.

1) First, the initial training of the ANN is performed based on the training set obtained from the approximate power consumption model (uniform distribution).
2) Then, the dataset obtained from the true values of the static and idle hardware power consumptions are used to refine the initial training.

The results are illustrated in Fig. 5. The figure shows similar trends as those reported in Fig. 4. This clearly highlights that using data from (possibly inaccurate) models may be beneficial to reduce the amount of required empirical data. It is apparent, in addition, that the amount of data to be used for the two phases is a critical hyper-parameter to be optimized, which highly depends on the ANN architecture being used.

VI. Conclusion and Open Research Issues

Based on theoretical arguments and numerical evidence illustrated in the present work, it is possible to conclude that mathematical models and optimization techniques provide unique insights to complement and improve ANN data-driven approaches. Unlike other application fields where deep learning may be employed, the solid theoretical understanding of wireless communication systems and networks that communication theorists have developed during the last decades provide us with unique opportunities to be exploited.

It is our hope that the present article will spur the interest of our research community, towards developing efficient ways of combining emerging deep learning wireless applications with theoretical prior knowledge originating from traditional modeling and optimization theory. The present work has introduced and substantiated an approach towards this direction, which, however, represents only the tip of the iceberg. Many important questions and research issues need to be addressed to synergistically combine conventional model-based and innovative data-driven approaches. Some of them as listed below:

- How does the dimension of the training set scale with the number of parameters to learn? It is anticipated that larger problems will require more training data, but different scaling laws might be observed for different classes of optimization problems.
- Is it possible to infer the resource allocation for a large system from the resource allocation of a smaller system? This would enable to scale up known ANN configurations to higher dimensions without explicitly performing a new training process.
- When using a simplified model for pre-training, how is the accuracy of the simplified model related to the amount of training data to use in both phases? The presented simulation have shed some light in this direction, but more investigation is needed for a solid understanding of this issue.

Finally, while the proposed approach represents a possible method to combine deep learning with mathematical modeling, it is to be mentioned that other approaches exist. In particular, we mention the framework of *deep reinforcement learning* [14], [15], which merges deep learning and reinforcement learning. In principle, deep reinforcement learning is able to learn optimal action policies in a fully data-driven fashion, by simply learning from direct interaction with the environment. Also in this case, however, the availability of prior information can significantly speed up the convergence of the algorithm and reduce the amount of required data. A dedicated comparison between the approach discussed in the present work and the use of model-aided deep reinforcement learning represents an interesting future research direction.

References

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

[2] T. Inoue, S. Chaudhury, G. D. Magistris, and S. Dasgupta, "Transfer learning from synthetic to real images using variational autoencoders for robotic applications," *https://arxiv.org/pdf/1709.06762.pdf*, 2017.

[3] C. Kim, E. Variani, A. Narayanan, and M. Bacchiani, "Efficient implementation of the room simulator for training deep neural network acoustic models," *https://arxiv.org/pdf/1712.03439.pdf*, 2017.

[4] O. Simeone, "A brief introduction to machine learning for engineers," *Foundations and Trends® in Communications and Information Theory*, pp. 1—191, 2017.

[5] S. Shalev-Shwartz, "Online learning and online convex optimization," *Foundations and Trends® in Communications and Information Theory*, vol. 4, no. 2, pp. 107–194, 2012.

[6] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[7] A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," in *International Conference on Learning Representation*, 2013.

[8] D. Sussillo and L. F. Abbott, "Random walks: Training very deep nonlinear feed-forward networks with smart initialization," *https://arxiv.org/abs/1412.6558v3*, 2015.

[9] A. Zappone and E. Jorswieck, "Energy efficiency in wireless networks via fractional programming theory," *Foundations and Trends® in Communications and Information Theory*, vol. 11, no. 3-4, pp. 185–396, 2015.

[10] G. Calcev, D. Chizhik, B. Goransson, S. Howard, H. Huanga, A. Kogiantis, A. Molisch, A. Moustakas, D. Reed, and H. Xu, "A wideband spatial channel model for system-wide simulations," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 2, March 2007.

[11] A. Zappone, E. Björnson, L. Sanguinetti, and E. Jorswieck, "Globally optimal energy-efficient power control and receiver design in wireless networks," *IEEE Transactions on Signal Processing*, vol. 65, no. 11, pp. 2844–2859, June 2017.

[12] M. D. Renzo, A. Zappone, T. T. Lam, and M. Debbah, "System-level modeling and optimization of the energy efficiency in cellular networks—a stochastic geometry framework," *IEEE Transactions on Wireless Communications*, 2018.

[13] M. D. Renzo, S. Wang, and X. Xi, "Inhomogeneous double thinning – modeling and analysis of cellular networks by using inhomogeneous poisson point processes," *IEEE Transactions on Wireless Communications*, 2018.

[14] Y. Li, "Deep reinforcement learning: An overview," *https://arxiv.org/abs/1701.07274*, 2017.

[15] K. Arulkumaran, M. P. Deisenroth, and M. B. A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, November 2017.