Online Energy-Efficient Power Control in Wireless Networks by Deep Neural Networks

Alessio Zappone, Merouane Debbah, Zwi Altman

ONLINE ENERGY-EFFICIENT POWER CONTROL IN WIRELESS NETWORKS BY DEEP NEURAL NETWORKS

Alessio Zappone[1], *Mérouane Debbah*[1,2], *Zwi Altman*[3]

1: Large Networks and Systems Group (LANEAS), Laboratoire des Signaux et Systmes
CentraleSupelec, CNRS, Univ. Paris Sud, Univ. Paris-Saclay, 91192 Gif-sur-Yvette, France
2: Mathematical and Algorithmic Sciences Laboratory,
France Research Center, Huawei Technologies, Paris, France
3: Orange Labs, 92794 Issy-les-Moulineaux, France.

ABSTRACT

The work describes how deep learning by artificial neural networks (ANNs) enables online power allocation for energy efficiency maximization in wireless interference networks. A deep ANN architecture is proposed and trained to take as input the network communication channels and to output suitable power allocations. It is shown that this approach requires a much lower computational complexity compared to traditional optimization-oriented approaches, dispensing with the need of solving the optimization problem anew in each channel coherence time. Despite the lower complexity, numerical results show that a properly trained ANN achieves similar performance as more traditional optimization-oriented methods.

1. INTRODUCTION

Future cellular networks will be required to serve more than 50 billions of devices by 2020, providing 1000x higher data rates compared to present systems, while at the same time halving the energy consumption. This means that the bit-per-Joule energy efficiency of future wireless networks will have to increase by a factor 2000x [1, 2]. This triggered a great deal of research aimed at allocating the available radio resources in order to maximize the network bit-per-Joule energy efficiency. In [3] a tutorial on optimization methods for energy efficiency is provided, describing the fundamental tool of fractional programming theory, and pointing out that energy efficiency maximization requires in general exponential complexity in realistic interference-limited network scenarios. To circumvent this issue, a simple approach is to resort to interference cancelation techniques or to orthogonal transmission schemes, thus falling back into the noise-limited regime [4–6]. However, this either leads to a poor resource reuse, or to noise enhancement effects and/or non-linear receive schemes. Moreover, unavoidable channel estimation errors also break the orthogonality in many cases. Another approach deals with interference by means of suboptimal optimization methods, mainly based on alternating optimization techniques [7, 8]. However, these approaches typically either do not guarantee convergence or are not supported by strong optimality properties. In order to overcome these issues, a recently proposed approach is the so-called sequential fractional programming (SFP) framework [9, 10], which merges fractional programming theory and sequential optimization tools, effectively decomposing complex energy efficiency maximization problems into a sequence of convex problems. This ensures strong optimality properties, while at the same time requiring affordable complexity. Moreover, it has been shown to achieve global optimality in several practical scenarios [10], by comparing it with global optimization methods. Nevertheless, all above approaches still require to perform the optimization anew anytime the propagation scenario changes. This is especially problematic when the resource allocation is based on the channel instantaneous realizations, which vary on a small scale in mobile environments. This forces to update the optimal resource allocation very frequently, thus causing a considerable complexity overhead, ultimately limiting the online implementation of available optimization frameworks, especially in complex networks.

The aim of this work is to show that these drawbacks can be overcome leveraging the emerging framework of deep learning [11]. Focusing on the problem of energy efficiency maximization, we show that it is possible to train a deep ANN to provide near-optimal energy-efficient power allocations. Overviews on deep learning applications to wireless communications have recently appeared in [12, 13]. In [14] and [15] deep learning is used for MMSE channel estimation and positioning, respectively, while data detection algorithms based on deep learning have been proposed in [16]. An overview of deep learning for resource allocation is provided in [17], while deep learning for power control is proposed in [18] and [19]. Specifically, [18] applies the recent tool of deep reinforcement learning to come up with online power

The research of A. Zappone and M. Debbah is supported by the H2020 MSCA IF BESMART, Grant 749336. The work of M. Debbah is also funded by the H2020-ERC PoC-CacheMire, Grant 727682.

control algorithms, whereas [19] proposes to train a deep neural network to learn the weighted MMSE algorithm for sum-rate maximization.

The approach taken in this work is inspired to that from [19], but with two main differences:

- while [19] focused on sum-rate maximization, this work is the first to consider the use of deep neural networks to learn fractional programming algorithms for energy efficiency maximization, based on the use of Dinkelbach's method and the SFP approach.
- while [19] uses a neural network to separately emulate each iteration of an iterative algorithm, this work takes an end-to-end perspective, proposing to use a deep network to learn the overall input-output map of a generic resource allocation problem.

The rest of this work is organized a follows. Section 2 formulates the problem of global energy efficiency (GEE) maximization in a generic interference-limited network. A deep ANN architecture is proposed in Section 3, showing how a suitable training set can be obtained and used to train the ANN for GEE optimization. Numerical results provided in Section 4 show that the proposed approach is able to provide satisfactory performance, while at the same time dispensing with the need to solve the problem for each new channel realization, thus facilitating online implementations.

2. SYSTEM MODEL AND PROBLEM STATEMENT

Let us consider the uplink of a multi-cell cellular network in which M base stations serve K mobile users. Each base station is equipped with N antennas whereas the mobile users have a single antenna, and let us denote by $\boldsymbol{h}_{k,m}$ the $N \times 1$ channel from mobile user k to base station m. Also, let p_k be the k-th user's transmit power, $\boldsymbol{c}_k$ the $N \times 1$ receive vector for the data from user k, and σ_m^2 the received noise power at base station m. Then, the signal-to-interference-plus-noise ratio (SINR) enjoyed by user k at its intended receiver m_k is expressed as:

$$\gamma_k = \frac{p_k |\boldsymbol{c}_k^H \boldsymbol{h}_{k,m_k}|^2}{\sigma^2 + \sum_{j \neq k} p_j |\boldsymbol{c}_k^H \boldsymbol{h}_{j,m_k}|^2} = \frac{p_k d_{k,k}}{\sigma^2 + \sum_{j \neq k} p_j d_{k,j}} \ , \tag{1}$$

with $d_{k,j} = |\boldsymbol{c}_k^H \boldsymbol{h}_{j,m_k}|^2$, for all k and j. Based on (1), the network GEE is given by

$$\text{GEE} = \frac{B \sum_{k=1}^K \log_2(1+\gamma_k)}{P_c + \sum_{k=1}^K \mu_k p_k} \quad \text{[bit/Joule]} \ , \tag{2}$$

wherein B is the communication bandwidth, P_c is the hardware static power consumed in the whole system[1], and μ_k the inverse of the efficiency of the power amplifier used by transmitter k.

[1]Clearly, P_c will depend on system parameters such as the number of antennas and efficiency of the hardware components deployed in the system. However, as far as power control is concerned, it will be a constant term with respect to the transmit power.

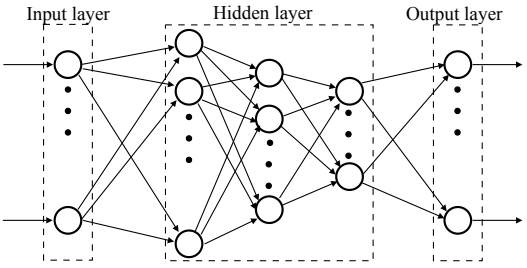

Fig. 1. Scheme of a deep ANN with I inputs, O outputs, L hidden layers, with N_ℓ units in layer ℓ, for all $\ell = 1, \ldots, L$.

Given this setup, the considered energy efficiency maximization problem is stated as the maximization of the GEE subject to power constraints, namely

$$\max_{\{p_k\}_{k=1}^K} \ \text{GEE}(p_1, \ldots, p_K) \tag{3a}$$

$$\text{s.t. } P_{min,k} \leq p_k \leq P_{max,k} \ , \forall \ k = 1, \ldots, K \tag{3b}$$

with $P_{max,k}$ and $P_{min,k}$ being the maximum feasible and minimum acceptable transmit powers for user k. Since the numerator of (3a) is not a concave function of $\boldsymbol{p} = \{p_k\}_{k=1}^K$, Problem (3) is a so-called non-concave fractional problem, for which no globally optimal, low-complexity optimization approach is available. Moreover, even by practical approaches, e.g. [3, 10], Problem (3) needs to be solved anew whenever the channel realizations $\{\boldsymbol{h}_{\ell,m_k}\}_{k,\ell}$ change. This represents a critical drawback, especially considering that the resource allocation process must be completed well before the end of the channel coherence time in order for the optimized power vector to be practically useful. The next section proposes the use of deep ANNs to solve these issues.

3. ENERGY EFFICIENCY MAXIMIZATION BY DEEP LEARNING

The idea of the proposed approach lies in observing that Problem (3) can be regarded as an unknown function mapping from the coefficients $\{d_{k,\ell}\}_{k,\ell}$ and maximum/minimum transmit powers P_{max} and P_{min} to the optimal power allocation vector $\boldsymbol{p}^*$, namely

$$\mathcal{F} : \boldsymbol{d} = \{d_{k,\ell}, P_{min,k}, P_{max,k}\}_{k,\ell} \in \mathbb{R}^{K(M+2)} \rightarrow \boldsymbol{p}^* \in \mathbb{R}^K \tag{4}$$

Then, leveraging the result that ANNs are universal function approximators [20], we propose to train an ANN to learn the unknown map (4). Specifically, we adopt a feedforward ANNs according to the general architecture shown in Fig. 1. The input layer feeds the input data in the form of the N_0-dimensional vector $\boldsymbol{x}_0 = \boldsymbol{d}$ to the rest of the network, with

$N_0 = K(M+2)$ for the case at hand. The input data is then processed by L so-called hidden layers, plus an output layer, each having N_ℓ processing units called neurons, $\ell = 1, \ldots, L+1$, to produce an N_{L+1}-dimensional output vector $\boldsymbol{x}_{L+1} = \boldsymbol{p}^*$, with $N_{L+1} = K$. Denoting by $\boldsymbol{x}_{\ell-1}$ the input to layer ℓ, each neuron n in layer ℓ, computes

$$\boldsymbol{x}_\ell(n) = f_{n,\ell}\left(\boldsymbol{w}_{n,\ell}^T \boldsymbol{x}_{\ell-1} + b_{n,\ell}\right) , \tag{5}$$

wherein $\boldsymbol{w}_{n,\ell} \in \mathbb{R}^{N_{\ell-1}}$ and $b_{n,\ell} \in \mathbb{R}$ are neuron-dependent weights and bias terms, while $f_{n,\ell}$ is a neuron-dependent nonlinear[2] map, called activation function. Typical choices for $f_{n,\ell}$ include sigmoidal, hyperbolic tangent, ReLU, and leaky ReLU functions [11]. Moreover, in order to enforce Constraint (3b) on the output vector, for all $n = 1, \ldots, N_{L+1}$, the output activation function is taken as

$$f_{n,L+1}(\cdot) = \max(P_{min,n}, \min(\cdot, P_{max,n})) , \tag{6}$$

The described ANN structure is called *feedforward* because the input data propagates only in the forward direction, from the input layer to the output layer, given that each neuron is only connected to the neurons in the following layer. Also, the considered network is referred to as a *deep* ANN, since multiple hidden layers are present. If instead only one hidden layer were used, it would be called a shallow ANN. Deep architectures are usually preferred since they require a lower number of neurons than shallow networks to learn a given input-output map [11].

3.1. ANN training

In order for the ANN to learn the desired input-output relation, it is necessary to tune the parameters $\boldsymbol{w}_{n,\ell} \in \mathbb{R}^{N_\ell - 1}$ and $b_{n,\ell} \in \mathbb{R}$ in a supervised learning fashion. To elaborate, the *training process* assumes to have a set of N_t pairs $\{(\boldsymbol{d}^{(nt)}, \boldsymbol{p}^{(nt)})\}_{nt=1}^{N_T}$, such that, for each nt, $\boldsymbol{p}^{(nt)}$ is the *desired output* when the input is $\boldsymbol{d}^{(nt)}$. Such a set is then typically split into a *training set*, which is used to train the network parameters, and a *validation set*, which is used to validate the trained model and obtain an estimate of the network generalization capabilities.

To elaborate further, denote by $\boldsymbol{x}_{L+1}^{(nt)}(\boldsymbol{W}, \boldsymbol{b})$ the *actual output* of the ANN corresponding to $\boldsymbol{d}^{(nt)}$ for given weights $\boldsymbol{W} = \{\boldsymbol{w}_{n,\ell}\}_{n,\ell}$ and bias terms $\boldsymbol{b} = \{b_{n,\ell}\}_{n,\ell}$, and define the *loss function* $\mathcal{L}(\boldsymbol{p}^{(nt)}, \boldsymbol{x}_{L+1}^{(nt)}(\boldsymbol{W}, \boldsymbol{b}))$ measuring the loss[3] between $\boldsymbol{p}^{(nt)}$ and $\boldsymbol{x}_{L+1}^{(nt)}(\boldsymbol{W}, \boldsymbol{b})$. The goal of the training process is to tune the network parameters $\boldsymbol{W}$ and $\boldsymbol{b}$ in order to minimize the average loss, defined as

$$L(\boldsymbol{W}, \boldsymbol{b}) = \frac{1}{N_t} \sum_{nt=1}^{N_t} \mathcal{L}(\boldsymbol{p}^{(nt)}, \boldsymbol{x}_{L+1}^{(nt)}(\boldsymbol{W}, \boldsymbol{b})) . \tag{7}$$

This minimization problem can be tackled by (stochastic) gradient search methods, i.e. iteratively updating the parameters according to the formulas:

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) - \alpha \nabla L(\boldsymbol{W}(t)) , \tag{8}$$

$$\boldsymbol{b}(t+1) = \boldsymbol{b}(t) - \alpha \nabla L(\boldsymbol{b}(t)) , \tag{9}$$

with α the learning rate, and where the gradients are conveniently estimated based on random subsets of the complete training set, called *mini-batches* [11, Ch. 8], and leveraging the *back-propagation* algorithm [11, Ch. 6.5].

Finally, it should be remarked that, in order to build a suitable training set, one should solve Problem (3) for several realizations of the entries of $\boldsymbol{d}$. This can be achieved by employing the monotonic fractional programming framework developed in [10], which however requires exponential complexity. For this reason, in this work we will apply the SFP framework, which requires polynomial complexity, while at the same time achieving near-optimal performance [10].

3.2. Online implementation and complexity

Once the parameters $\boldsymbol{W}$ and $\boldsymbol{b}$ to be used are determined as a result of the training process, the ANN is configured and able to compute energy-efficient power allocations corresponding to input vectors $\boldsymbol{d}$ that are not part of the training set. This means that every time the channel realizations change, the power control policy is updated by simply feeding the new $\boldsymbol{d}$ to the ANN, without any need to actually solve Problem (3) anew. This yields a significant complexity reduction, since in order to obtain the output $\boldsymbol{p}$ for a given input $\boldsymbol{d}$, an ANN must perform only $\sum_{\ell=1}^{L+1} N_{\ell-1} N_\ell$ real multiplications[4].

In addition, one should account for the complexity due to the training phase. However, network training can be performed at a much longer scale than the channel block duration. Indeed, once the network has been trained, the resulting training set can be used at least until the *statistics* of the channels significantly change. Instead, Problem (3) should be solved anew every time the *instantaneous realizations* of the channels change.

4. NUMERICAL RESULTS

We consider the uplink of a three-cell MIMO system with $K = 10$ users and $N = 10$ antennas at each base station. The path-loss model from [21] has been assumed, while fast fading terms have been modeled as realizations of zero-mean complex Gaussian random variables. Moreover, $\mu_k = 10$ for all k, whereas the circuit power P_c follows the model from [22]. Maximum ratio combining is adopted at all base stations. The considered ANN has $L = 10$ hidden layers,

[2] In principle linear maps could be used, but if all activation functions were linear, the ANN would be only capable of learning linear functions and there would be no advantage in using multiple layers.

[3] The loss function used in this work is specified in the numerical results.

[4] The complexity related to additions is negligible compared to that related to multiplications. Similarly, the complexity required to compute the activation functions is neglected, because the most widely used activation function, the ReLU, only requires determining the sign of the input.

while $n_\ell = n_{\ell-2} - 2$, if ℓ is odd, $n_\ell = n_{\ell-1}$ if ℓ is even, and $n_1 = 18$. $N_t = 10^4$ pairs $(\boldsymbol{d}, \boldsymbol{p})$ have been generated solving Problem (3) by the SFP framework, and further divided into a training set containing 9000 samples, and a validation set containing the remaining 1000 samples. In each scenario, mobile users' have been randomly and independently placed in a circular area with radius 500 m and assigned to the closest base station. The loss function (7) has been used, with $\mathcal{L}(\boldsymbol{p}^{(nt)}, \boldsymbol{x}_{L+1}^{(nt)}(\boldsymbol{W}, \boldsymbol{b})) = \|\boldsymbol{p}^{(nt)} - \boldsymbol{x}_{L+1}^{(nt)}(\boldsymbol{W}, \boldsymbol{b})\|^2$, and a mini-batch size of 256 has been used when implementing the stochastic gradient descent method. It should be stressed that typical deep learning applications consider much larger training sets, usually of the order of 10^6 samples. However, the choice of training the network by using only 10^4 samples is motivated by the consideration of reducing the complexity and storage requirements of the proposed approach.

After training and validation, the ANN performance has been tested over $N_r = 10^4$ previously unobserved network scenarios, that have been randomly and independently generated with respect to the training and validation samples. It should be noted that typical deep learning applications consider much smaller test sets compared to the training set. However, the choice $N_r = 10^4$ is motivated by practical considerations, since it allows evaluating the average performance of the proposed method after 10^4 unobserved channel realizations have occurred. The satisfactory results to be illustrated, show that indeed network training can be performed at a much longer scale than the channel block duration.

In Fig. 2, the average (over the test set) GEE versus P_{max} obtained by the SFP method from [10] and by the proposed ANN-based approach are compared. As a benchmark, the GEE obtained by full power allocation for all users is also reported. It is seen that, despite the much lower complexity, the ANN-based allocator performs like the sequential method from [10] for lower P_{max} values, while a small gap emerges for larger P_{max} values. However, also in this regime the ANN method achieves around 95% of the value obtained by SFP. A similar scenario is considered in Fig. 3, with the difference that the metric under analysis is the average per-user transmit power. Similar comments as for Fig. 2 apply. Finally, Tab. 4 shows the ratio of the average CPU times required to tackle (3) by SFP and by the proposed ANN-based approach. It is seen that the proposed method is always faster, especially for larger P_{max}. Indeed, as P_{max} increases, the space to search for the optimization algorithm grows wider, whereas the complexity of the ANN does not change. Moreover, we stress again that adopting any standard optimization algorithm would require to tackle (3) for each channel realization. Instead, this is not required by ANN-based methods.

5. CONCLUSIONS

It has been shown how deep ANNs are able to learn and compute energy-efficient power allocations in wireless interference networks, as a function of the system propagation channels. If properly trained, an ANN is able to approach the performance of sophisticated energy-efficient optimization frameworks, such as the near-optimal SFP method, while requiring a much lower complexity. In particular, ANN-based methods, do not require solving one optimization problem in each channel coherence time, thus lending themselves to effective online resource allocation also in complex and large interference networks. Numerical results have been provided to support the claimed advantages.

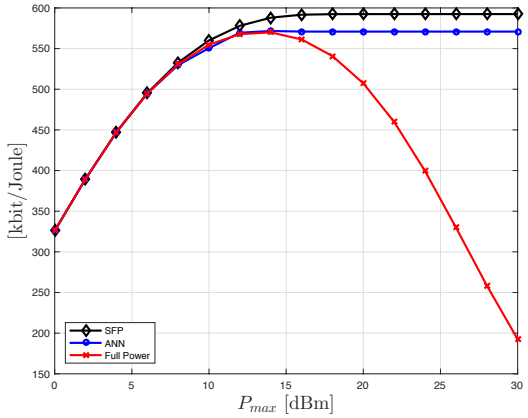

Fig. 2. GEE versus P_{max} by: (a) SFP from [10]; (b) Deep learning by ANN; (c) Full power allocation.

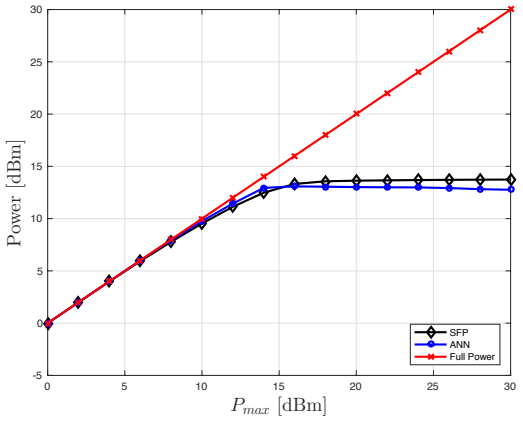

Fig. 3. Per-user transmit power versus P_{max} by: (a) SFP from [10]; (b) Deep learning by ANN; (c) Full power allocation.

$P_{\max}$ [dBm]	0	10	20	30
Time Ratio	1.04	21.75	56.27	103.84

Table 1. Ratio of average CPU times required by SFP and proposed ANN-based approach to tackle (3).

6. REFERENCES

[1] "NGMN alliance 5G white paper," *https://www.ngmn.org/5g-white-paper/5g-white-paper.html*, 2015.

[2] S. Buzzi, C.-L. I, T. E. Klein, H. V. Poor, C. Yang, and A. Zappone, "A survey of energy-efficient techniques for 5g networks and challenges ahead," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, 2016.

[3] 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.

[4] D. W. K. Ng, E. S. Lo, and R. Schober, "Energy-efficient resource allocation in multi-cell OFDMA systems with limited backhaul capacity," *IEEE Transactions on Wireless Communications*, vol. 11, no. 10, pp. 3618–3631, October 2012.

[5] Q. Xu, X. Li, H. Ji, and X. Du, "Energy-efficient resource allocation for heterogeneous services in OFDMA downlink networks: Systematic perspective," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 5, pp. 2071–2082, June 2014.

[6] J. Xu and L. Qiu, "Energy efficiency optimization for MIMO broadcast channels," *IEEE Transactions on Wireless Communications*, vol. 12, no. 2, pp. 690–701, February 2013.

[7] B. Du, C. Pan, W. Zhang, and M. Chen, "Distributed energy-efficient power optimization for CoMP systems with max-min fairness," *IEEE Communications Letters*, vol. 18, no. 6, pp. 999–1002, 2014.

[8] S. He, Y. Huang, L. Yang, and B. Ottersten, "Coordinated multicell multiuser precoding for maximizing weighted sum energy efficiency," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 741–751, February 2014.

[9] A. Zappone, L. Sanguinetti, G. Bacci, E. A. Jorswieck, and M. Debbah, "Energy-efficient power control: A look at 5G wireless technologies," *IEEE Transactions on Signal Processing*, vol. 64, no. 7, pp. 1668–1683, April 2016.

[10] 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.

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

[12] 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, December 2017.

[13] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks," *https://arxiv.org/pdf/1710.02913.pdf*, 2017.

[14] D. Neumann, T. Wiese, and W. Utschick, "Learning the MMSE channel estimator," *https://arxiv.org/abs/1707.05674v2*, 2017.

[15] J. Vieira, E. Leitinger, M. Sarajlic, X. Li, and F. Tufvesson, "Deep convolutional neural networks for massive MIMO fingerprint-based positioning," *https://arxiv.org/pdf/1708.06235.pdf*, 2017.

[16] N. Farsad and A. Goldsmith, "Detection algorithms for communication systems using deep learning," *https://arxiv.org/abs/1705.08044*, 2017.

[17] F. D. Calabrese, L. Wang, E. Ghadimi, G. Peters, and P. Soldati, "Learning radio resource management in 5G networks: Framework, opportunities and challenges," *https://arxiv.org/abs/1611.10253*, 2017.

[18] J. Fang, X. Li, W. Cheng, Z. Chen, and H. Li, "Intelligent power control for spectrum sharing: A deep reinforcement learning approach," *https://arxiv.org/pdf/1712.07365.pdf*, 2017.

[19] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for wireless resource management," *https://arxiv.org/pdf/1705.09412.pdf*, 2017.

[20] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[21] 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.

[22] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks spectral, energy, and hardware efficiency," *Foundations and Trends in Signal Processing*, 2017.