Deep Learning for Real-Time Energy-Efficient Power Control in Mobile Networks

Bho Matthiesen*, Alessio Zappone†, Eduard A. Jorswieck*, and Merouane Debbah†‡
*Technische Universität Dresden, Institute of Communication Technology, Dresden, Germany
†Large Networks and Systems Group (LANEAS), Laboratoire des Signaux et Systèmes, CentraleSupelec, CNRS, Univ. Paris Sud, Univ. Paris-Saclay, 91192 Gif-sur-Yvette, France
‡Mathematical and Algorithmic Sciences Laboratory, France Research Center, Huawei Technologies, Paris, France
Email: {bho.matthiesen, eduard.jorswieck}@tu-dresden.de, {alessio.zappone, merouane.debbah}@l2s.centralesupelec.fr

***Abstract*—This work proposes an energy-efficient power control approach that can operate in an online fashion even in high-mobility scenarios. The proposed approach leverages an artificial neural networks to estimate the optimal relationship between the system propagation channels and the optimal power allocation rule. The approach also relies on an improved branch-and-bound algorithm that allows the offline generation of a large amount of training data with affordable complexity. Numerical results show the merits of the proposed approach, during both training and testing phase.**

***Index Terms*—Energy Efficiency, Deep Learning, Artificial Neural Networks, Power Control, Interference Networks.**

I. Introduction

Energy efficiency has emerged as a key performance indicator of future mobile networks, with a requirement of a 2000x bit-per-Joule energy efficiency (EE) increase [1]. In order to meet this requirement, energy-management must be tackled jointly by several tools [2], one of which is energy-efficient power control [3]. Nevertheless, maximizing the EE in interference-limited networks is known to be an NP-hard problem. Moreover, even available suboptimal optimization methods are based on iterative methods that solve concave or pseudo-concave relaxations of the energy-efficient problem. This is still not practical for online implementations in high-mobility systems where the channel coefficients vary rapidly, thus causing a significant complexity and feedback overhead.

This work shows how deep learning techniques based on artificial neural network (ANN) [4] can complement traditional optimization-oriented design methodologies, leading to an optimization technique that is able to provide a near-optimal power allocation rule in closed-form. This enables to update the optimal power allocation following the channel variations in real-time. Available research contributions that propose deep learning for the design of wireless networks have mainly focused on the development of improved decoding structures [5], [6]. Instead, the issue of power control by deep learning was discussed in [7], [8], where deep reinforcement learning was proposed, and in [9], where a deep neural network is used to emulate the weighted MMSE (WMMSE) algorithm. All of these works consider the maximization of the system sum-rate. Similarly, sum-rate maximization is also addressed in [10], where it is proposed to train a neural network employing the sum-rate as training cost function. A similar approach is employed in [11] where transfer learning is used to train a convolutional neural network. Specifically, the model is first pre-trained as in [9] and then the cost function is exchanged either for the weighted sum rate or the weighted sum energy efficiency (WSEE). A gain over the WMMSE solution, which maximizes the throughput, is observed but the results are not benchmarked against any algorithm that maximizes the EE.

Unlike most of these previous works, here we aim at developing power control algorithms to maximize the system EE, which includes the sum-rate maximization problem as a special case. Specifically, we focus on the maximization of the system WSEE, which is considered the most complex energy-efficient metric to maximize. Thus, developing an ANN-based effective online method for WSEE maximization appears as a very strong argument for the use of this tool to solve generic EE maximization problems. Besides the consideration of a different optimization metric, another major novelty of this work is that the proposed method does not train an ANN to emulate any given, suboptimal algorithm, as previous works do. Instead, we are able to globally solve the WSEE maximization problem with a novel branch-and-bound (BB) method with moderate complexity. This allows us to produce offline large training sets with limited complexity and then train a deep neural network to learn the optimal power allocation rule. The trained ANN is then able to predict with negligible complexity the optimal power allocation also for channel realizations that were not previously observed. In addition, our approach does not require to fix any initialization power vector, which is instead needed by previous implementations.

The work of B. Matthiesen and E. A. Jorswieck is supported in part by the German Research Foundation (DFG) in the Collaborative Research Center 912 "Highly Adaptive Energy-Efficient Computing." The work of A. Zappone and M. Debbah was funded by the European Commission through the H2020-MSCA IF-BESMART project under Grant Agreement 749336. We thank the Center for Information Services and High Performance Computing (ZIH) at TU Dresden for generous allocations of computer time.

II. System Model and Problem Statement

We consider the uplink of a multi-cell interference network in which L single antenna user equipments (UEs) are served by

M base station (BS), equipped with n_R antennas at each BS. After maximum ratio combining at the receiver, the achievable rate enjoyed by user i at its BS $a(i)$ is:

$$R_i = B\log\left(1+\frac{\alpha_i p_i}{1+\sum_{j\neq i}\beta_{i,j}p_j}\right) \tag{1}$$

with $\alpha_i = \frac{\|\boldsymbol{h}_{a(i),i}\|^2}{\sigma_i^2}$ and $\beta_{i,j} = \frac{|\boldsymbol{h}_{a(i),i}^H \boldsymbol{h}_{a(i),j}|^2}{\sigma_i^2\|\boldsymbol{h}_{a(i),i}\|^2}$, $\boldsymbol{h}_{a(i),j} \in \mathbb{C}^{n_R}$ is the channel from UE j to BS $a(i)$, $x_j \in \mathbb{C}$ the symbol transmitted by UE j, and $\boldsymbol{z}_{a(i)}$ zero-mean circularly symmetrical complex Gaussian noise with power σ_i^2, and B is the communication bandwidth. Each UE is subject to an average transmit power constraint, i.e., $p_j \leq P_j$ where p_j is the average power of x_j, and the BSs use matched filtering.

In this context, the EE of the link between UE i and its intended BS is defined as the benefit-cost ratio in terms of the link's achievable rate and power consumption necessary to operate the link, i.e.,

$$\text{EE}_i = \frac{B\log\left(1+\frac{\alpha_i p_i}{1+\sum_{j\neq i}\beta_{i,j}p_j}\right)}{\mu_i p_i + P_{c,i}} \tag{2}$$

where μ_i is the inefficiency of UE i's power amplifier and $P_{c,i}$ is the total static power consumption of UE i and its associated BS.

The aim of this work is to develop a near-optimal approach to tackle the maximization of the WSEE, that is also fast enough to be easily implemented online. The WSEE optimization problem is stated as:

$$\max_{\boldsymbol{p}} \sum_{i=1}^{L} w_i \frac{\log\left(1+\frac{\alpha_i p_i}{1+\sum_{j\neq i}\beta_{i,j}p_j}\right)}{\mu_i p_i + P_{c,i}} \tag{3}$$

$$\text{s.t.}\, 0 \leq p_i \leq P_i, \quad \text{for all } i = 1,2,\ldots,L, \tag{4}$$

which falls within the category of sum-of-ratios problems, that is NP-hard and considered the hardest class of fractional problems to solve.

III. ANN-based power control

The proposed approach relies on the universal approximation property of ANNs [12], which states that the input-output relationship of a feedforward neural network with with fully-connected layers can emulate any continuous function. Equipped with this result, it is possible to regard Problem (4) as the map:

$$\mathcal{F} : \boldsymbol{a} = (\alpha_i, \beta_{i,j}, P_i)_{i,j} \in \mathbb{R}^{L(L+1)} \mapsto \boldsymbol{p}^* \in \mathbb{R}^L, \tag{5}$$

with $\boldsymbol{p}^*$ the optimal power allocation corresponding to $\mathbf{a}$. Then, we can employ an ANN with fully-connected layers to approximate (5). In the following we briefly describe the ANN that will be employed.[1]

We consider a feedforward ANN which takes as input a realization of the parameter vector $\mathbf{a}$, and outputs an estimate $\boldsymbol{p}$ of the corresponding optimal power allocation vector $\mathbf{p}^*$. The input and output layers are separated by K, fully-connected hidden layers, wherein the k-th layer has N_k neurons. For all $k = 1,\ldots,K+1$, the neuron n of layer k computes

$$\zeta_k(n) = f_{n,k}\left(\boldsymbol{\gamma}_{n,k}^T \boldsymbol{\zeta}_{k-1} + \delta_{n,k}\right) \tag{6}$$

wherein $\boldsymbol{\zeta}_k = (\zeta_k(1),\ldots,\zeta_k(N_{k+1}))$ is the $N_{k+1}\times 1$ output vector of layer k, $\boldsymbol{\gamma}_{n,k} \in \mathbb{R}^{N_{k-1}}$ and $\delta_{n,k} \in \mathbb{R}$ are neuron-dependent weights and bias terms, respectively, while $f_{n,k}$ is the activation function of neuron n in layer k.

Based on the universal function approximation property, provided a sufficient number of neurons are deployed, the weights and bias of the ANN can be adjusted so that the input-output relationship of the ANN approximates arbitrarily well the optimal power allocation rule. However, the universal function approximation property does not specify how the weights and bias terms should be set in order to achieve a satisfactory approximation performance. To this end, a supervised training procedure needs to be used, which requires a training set containing examples of desired power allocation, i.e. $\{(\boldsymbol{a}_n, \boldsymbol{p}_n^*)\,|\, n = 1,\ldots,N_T\}$, wherein $\boldsymbol{p}_n^*$ is the optimal power allocation corresponding to $\boldsymbol{a}_n$. Then, the training process adjusts the weights and biases of the ANN in order to minimize the loss between the actual and desired output, i.e. tackling the problem:

$$\min_{\boldsymbol{\Gamma},\boldsymbol{\delta}} \frac{1}{N_T}\sum_{n=1}^{N_T} \mathcal{L}(\hat{\boldsymbol{p}}_n(\boldsymbol{\Gamma},\boldsymbol{\delta}), \boldsymbol{p}_n^*) \tag{7}$$

with $\mathcal{L}(\cdot,\cdot)$ any suitable error measure between the actual ANN output $\boldsymbol{p}_n$, and the desired output $\boldsymbol{p}_n^*$. Problem (7) is tackled by state-of-the-art, off-the-shelf stochastic gradient descent methods specifically designed for ANNs training [4]. Instead, it is important to observe that the training process can be significantly simplified by reformulating the WSEE maximization problem constraining all transmit powers to lie in the interval $[0,1]$. To elaborate, upon applying the variable change $p_i \to \tilde{p}_i P_i$, for all $i = 1,\ldots,L$, Problem (4) can be equivalently stated as

$$\max_{\tilde{\boldsymbol{p}}} \sum_{i=1}^{L} w_i \frac{\log\left(1+\frac{\tilde{\alpha}_i \tilde{p}_i}{1+\sum_{j\neq i}\tilde{\beta}_{i,j}\tilde{p}_j}\right)}{\mu_i P_i \tilde{p}_i + P_{c,i}} \tag{8}$$

$$\text{s.t.}\, 0 \leq \tilde{p}_i \leq 1, \quad \text{for all } i = 1,2,\ldots,L, \tag{9}$$

wherein $\tilde{\alpha}_i = \alpha_i P_i$, $\tilde{\beta}_i = \beta_i P_i$, $\tilde{\mu}_i = \mu_i P_i$, for all $i = 1,\ldots,L$. This implies that the normalized training set to be used for training purposes is $\mathcal{S}_T = \{(\tilde{\boldsymbol{a}}_n, \tilde{\boldsymbol{p}}_n^*)\,|\, n = 1,\ldots,N_T\}$ with parameter vector $\tilde{\boldsymbol{a}} = (\tilde{\alpha}_i, \tilde{\beta}_{i,j}, P_i)_{i,j}$. The motivation to consider such a reformulation is that for any value of the maximum power vectors $P_1,\ldots,P_L$, the optimal solution of (9) will always have transmit powers lying in the set [0,1]. Thus, in Problem (9) the dependence between the optimal solution and the maximum power constraints is much weaker than in Problem (4). This means that Problem (9) will exhibit a simpler structure of the optimal solution as a function of the input parameters $\boldsymbol{a}$, among which we find the maximum power constraints $P_1,\ldots,P_L$, which simplifies the task of learning the optimal power allocation rule.

After the training procedure is complete, all weights and biases of the ANN are fixed, which allows writing the ANN input-output relationship in closed-form as the composition of the affine combinations and activation functions of all neurons.

[1]For an extensive explanation on the different neural network architectures, we refer to [4].

Thus, the trained ANN provides a closed-form approximation of the map (5), whose accuracy can be adjusted by employing a properly sized ANN that is accurately trained. Therefore, the optimal power allocation can be tuned in real-time as $\boldsymbol{a}$ varies, by simply using the derived closed-form map, without having to solve Problem (4) anew.

A. Computational complexity

The complexity of the proposed ANN-based power allocation method depends on two main tasks: (a) computing the power allocation vector using the trained ANN; (b) building the training set and processing it to train the ANN.

As for the first task, it requires performing $\sum_{k=1}^{K+1} N_{k-1}N_k$ real multiplications,[2] and evaluating $\sum_{k=1}^{K+1} N_k$ scalar activation functions $f_{n,k}$, which poses a very limited computational burden, especially considering that the activation functions are all elementary functions that do not involve any numerical integration or fixed point iteration.

Instead, the second task appears to be more complex. However, we argue in the rest of this section that its complexity does not significantly impact on the overall complexity of the power allocation method. First of all, let us observe that most of the complexity lies in the generation of the training set, rather than in its processing during the training procedure.[3] Instead, generating the training set requires globally solving the NP-hard Problem (9) for N_T different realizations of the system parameter vector $\boldsymbol{a}$. Nevertheless, this is still quite affordable taking into account the following three main points:

- The training set can be generated *offline*. Thus, real-time constraints do not apply and a much higher complexity can be afforded.
- The training set can be updated at a *much longer time-scale* compared with the channels coherence time.
- Despite the first two points, due to the fact that Problem (9) is NP-hard, even its offline solution could be problematic considering that reasonably-sized training set should have hundreds of thousands of samples. For this reason, the next section proposes a novel and improved BB method that makes the offline solution of (9) fully affordable in a reasonable amount of time.

Finally, we should also observe that the proposed method does not require to set any initial power vector, as it would be necessary if the ANN were used to learn a suboptimal power allocation algorithm. As a result, our method reduces the size of the map to estimate, lowering the input size from $L(L+2)$ to $L(L+1)$, which allows for a simpler training procedure compared to other available approaches.

IV. Globally optimal power control

Due to space constraints, the proposed procedure is only sketched in this section. The full details can be found in [13].

[2]The complexity related to additions is neglected as it is much smaller than that required for multiplications.

[3]Recall that the training algorithm is conveniently performed by off-the-shelf stochastic gradient descent algorithms, that are particularly efficient thanks to the use of stochastic gradient descent methods and of the backpropagation algorithm [4].

The approach employs a BB approach, in which the bound is carefully selected to speed up convergence. Specifically, we successively partition the set $[\mathbf{0},\boldsymbol{P}] = [0,P_i]^L$ into L-dimensional hyper-rectangles of the form

$$\mathcal{M}^k = \{\boldsymbol{p}:\ r_i^{(k)} \leq p_i \leq s_i^{(k)}, \forall\, i=1,\ldots,L\} \triangleq [\boldsymbol{r}^{(k)},\boldsymbol{s}^{(k)}]. \quad (10)$$

For each $\mathcal{M}^k$, an upper-bound of the WSEE is obtained as:

$$\begin{aligned}
&\max_{\boldsymbol{p}\in\mathcal{M}^k} \sum_{i=1}^{L} w_i \frac{\log\left(1+\frac{\alpha_i p_i}{1+\sum_{j\neq i}\beta_{i,j}p_j}\right)}{\mu_i p_i + P_{c,i}} \\
&\leq \sum_{i=1}^{L} w_i \max_{\boldsymbol{p}\in\mathcal{M}^k} \frac{\log\left(1+\frac{\alpha_i p_i}{1+\sum_{j\neq i}\beta_{i,j}p_j}\right)}{\mu_i p_i + P_{c,i}} \\
&= \sum_{i=1}^{L} w_i \max_{r_i^{(k)}\leq p_i\leq s_i^{(k)}} \underbrace{\frac{\log\left(1+\frac{\alpha_i p_i}{1+\sum_{j\neq i}\beta_{i,j}r_j^{(k)}}\right)}{\mu_i p_i + P_{c,i}}}_{:=\overline{\mathrm{EE}}_i(p_i,\mathcal{M}^k)} \quad (11)
\end{aligned}$$

It is seen that the bound is obtained by maximizing the individual EE of each user with respect to $\boldsymbol{p}$ over $\mathcal{M}^k$. This clearly provides a very accurate bound. In addition, such a maximization entails a negligible computational complexity, because $\overline{\mathrm{EE}}_i(p_i,\mathcal{M}^k)$ can be seen to be a strictly pseudo-concave function of p_i [3]. Thus, its global maximizer is obtained by setting to zero its first-order derivative, which yields

$$\frac{\alpha_i(\mu_i p_i + P_{c,i})}{1+\sum_{j\neq i}\beta_{i,j}r_j^{(k)}+\alpha_i p_i} = \mu_i \ln\left(1+\frac{\alpha_i p_i}{1+\sum_{j\neq i}\beta_{i,j}r_j^{(k)}}\right), \quad (12)$$

While Equation (12) can be solved numerically by any root finding algorithm, e.g. with Newton-Raphson's or Halley's method, we have observed that these methods might suffer from numerical problems since $\frac{d}{dp_i}\overline{\mathrm{EE}}_i$ tends rapidly to zero as $p_i \to \infty$. Instead, a numerically more stable approach is to observe that the unique solution of (12) is given by

$$\widetilde{p}_i^{(k)} = \frac{1}{\widetilde{\alpha}_i}\left(\frac{\frac{\widetilde{\alpha}_i}{\mu_i}P_{c,i}-1}{W_0\left(\left(\frac{\widetilde{\alpha}_i}{\mu_i}P_{c,i}-1\right)e^{-1}\right)}-1\right), \quad (13)$$

where $\widetilde{\alpha}_i = \frac{\alpha_i}{1+\sum_{j\neq i}\beta_{i,j}r_j^{(k)}}$ and $W_0(\cdot)$ is the principal branch of the Lambert W function.

Finally, equipped with (13), an adaptive BB procedure can be devised, which, as confirmed by the numerical results in Section V, exhibits a remarkable complexity-performance trade-off.

V. Numerical results

We consider the uplink of a wireless interference network in which $L=4$ single-antenna UEs are placed in a square area with edge 2 km and communicate with 4 access points placed at coordinates $(0.5, 0.5)$ km, $(0.5, 1.5)$ km, $(1.5, 0.5)$ km, $(1.5, 1.5)$ km, equipped with $n_R = 2$ antennas each. The path-loss is modeled following [14], with carrier frequency 1.8 GHz and power decay factor 4.5, while fast fading terms are modeled as proper complex Gaussian random variates with unit variance. The circuit power consumption and power amplifier inefficiency terms are equal to $P_{c,i} = 1\,\mathrm{W}$ and $\mu_i = 4$ for all $i=1,\ldots,L$, respectively. The noise power at each receiver is $\sigma^2 = F\mathcal{N}_0 B$, wherein $F = 3\,\text{dB}$ is the receiver noise figure, $B = 180\,\text{kHz}$ is the communication bandwidth, and $\mathcal{N}_0 = -174\,\text{dBm/Hz}$ is the noise spectral density. All users have the same maximum transmit powers $P_1, \dots, P_L = P_{\text{max}}$.

The proposed ANN-based solution of Problem (4) is implemented through a feedforward ANN with $K+1$ fully-connected layers, having $K = 5$ hidden layers with 128, 64, 32, 16, 8 neurons, respectively. In order to generate the training set, Problem (4) needs to be solved for different realizations of the vector $\tilde{\boldsymbol{a}} = (\tilde{\alpha}_i, \tilde{\beta}_{i,j}, P_{\text{max}})_{i,j}$. When doing this, the use of realistic numbers for the receive noise power and propagation channels leads to coefficients $\{\tilde{\alpha}_i, \tilde{\beta}_{i,j}\}_{i,j}$ with quite a large magnitude. This is known to cause numerical problems to the gradient-descent training algorithm. Classical normalization approaches are infeasible in this scenario due to the channel coefficients being unbounded. Instead, we reduce their magnitude by a logarithmic transform of the parameter vector $\tilde{\boldsymbol{a}}$. A similar problem occurs for the output powers, which in some cases might be close to zero due to the normalization by P_{max}. The problem here is that the training algorithms for the ANN treat very small numbers as zero leading to numerical instabilities and, hence, bad training results for these values. This issue is also resolved by expressing the output powers in logarithmic scale. However, computing logarithms for values close to zero might cause numerical problems in itself. In order to avoid this issue, a suitable approach is to clip logarithmic values approaching $-\infty$ at $-M$ for $M > 0$. In our experiments, $M = 20$ worked well.[4] Thus, the considered normalized training set is

$$\mathcal{S}_T = \{(\log_{10} \tilde{\boldsymbol{a}}_n, \max\{-20, \log_{10} \tilde{\boldsymbol{p}}_n^*\}) \,|\, n = 1, \dots, N_T\},$$

where all functions are applied element-wise to the vectors in the training set.

Our experiments verify that the widely used class of ReLU activation function performs well in this application. Specifically, the first hidden layer has an exponential linear unit (ELU) activation, to compensate for the logarithmic conversion in the training set. This choice, together with the logarithmic normalization of the data set, has proven itself essential for good training performance. The following hidden layers alternate ReLU and ELU activation functions, while the output layer deploys a linear activation function. This choice of activation functions has shown best training performance among several evaluated configurations, including using only ReLU or only ELU hidden layers. The choice of linear output layer seems to contrast with the fact that the transmit powers need to be constrained in the interval $[0, 1]$. However, enforcing this constraint directly in the output activation function might mislead the ANN. Indeed, it could lead to low training errors simply thanks to the use of cut-off levels in the activation function, instead of being the result of proper adjustment of the hidden layer weights and biases.

A. Training Performance

The ANN is trained over 100 epochs with batches of size 128 and shuffling of the training data before each epoch. Initialization is performed by Keras [15] with default parameters. Problem (7) is solved by the NAdam optimizer with the squared error as loss function.[5] The training set is generated from 2,000 independent and identically distributed (i.i.d.) realizations of UEs' positions and propagation channels. Each user is randomly placed in the service area and channels are then generated according to the channel model described above. Each UE i is associated to the access point towards which it enjoys the strongest effective channel α_i. For each channel realization, we apply the BB algorithm outlined in Section IV to solve (4) for $P_{\text{max}} = -30, \dots, 20\,\text{dB}$ in $1\,\text{dB}$ steps with relative tolerance 0.01. This yields a training set of 102,000 samples. Validation and test sets are generated in the same way from 200 and 10,000 i.i.d. channel realizations, respectively, resulting in 10,200 and 510,000 samples.

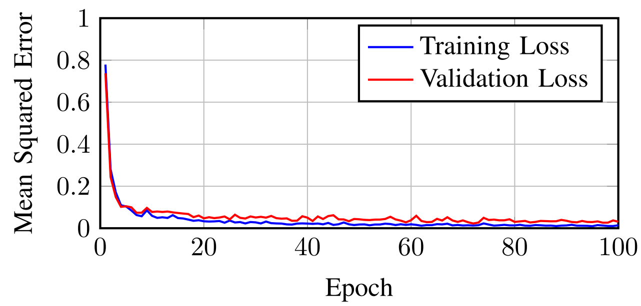

Fig. 1. Training and validation loss.

Considering training, validation, and test sets, 622,200 data samples were generated, which required solving the NP-hard problem (4) 622,200 times. Computing this data set took 8.4 CPU hours on Intel Haswell nodes with Xeon E5-2680 v3 CPUs running at 2.50 GHz. The mean and median times per sample are 48.7 ms and 4.8 ms, respectively, which shows the effectiveness of the proposed BB algorithm, and supports the argument that offline generation of a training set for the proposed ANN-based power control method is quite affordable.

The average training and validation losses for the final ANN, averaged over 10 realizations of the network, are shown in Fig. 1. It can be observed that both errors quickly approach a very small value that is of the same order of magnitude as the numerical tolerance of the training data. Moreover, neither of the losses increases over time which leads to the conclusion that the adopted training procedure fits the training data well, without underfitting or overfitting.

B. Testing Performance

The average performance of the final ANN on the test set is reported in Fig. 2. This test set is never used during training and, thus, the ANN has no information about it except for its statistical properties gathered from the training set (and, possibly, the validation set due to hyperparameter tuning). It can be seen from Fig. 2 that the gap to the optimal value is virtually non-existent. This is confirmed by the relative approximation error that has mean and median values of 0.0133 and 0.00739, respectively. Note that using a test set based on 10,000 channel

[4]Note that, although using a logarithmic scale, the transmit powers are not expressed in dBW, since the logarithmic values are not multiplied by 10. Thus $-M = -20$, corresponds to -200 dBW.

[5]Source code and data sets are available from `https://github.com/bmatthiesen/deep-EE-opt`.

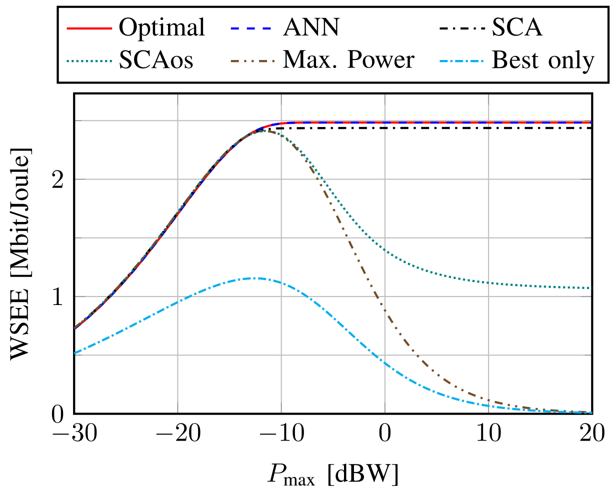

Fig. 2. Performance on the test set compared to the global optimal solution, first-order optimal solutions, and fixed power allocations.

scenarios means that this performance is what is obtained by using the trained ANN for 10,000 channel coherence times. This confirms that the training phase needs to be performed only sporadically, as argued in Section III-A.

In addition to near-optimal approximation performance and low computational complexity, the proposed ANN-based approach also outperforms several baseline approaches. Specifically, we have included a comparison with the following benchmarks:

- **SCAos:** A first-order optimal, gradient-descend based method presented in [16]. For each value of $P_{\max}$, the algorithm initializes the transmit power to $p_i = P_{\max}$, for all $i = 1, \ldots, L$.
- **SCA:** The same algorithm as before, but in this case a double-initialization approach is used. Specifically, at $P_{\max} = -30\,\mathrm{dBW}$ once again maximum power initialization is used. However, for all values of $P_{\max} > -30\,\mathrm{dBW}$, the algorithm is ran twice, once with the maximum power initialization, and once initializing the transmit powers with the optimal solution obtained for the previous $P_{\max}$ value. Then, the power allocation achieving the better WSEE value is retained.
- **Max. Power:** All UEs transmit at maximum power, i.e. $p_i = P_{\max}$, for all $i = 1, \ldots, L$. This strategy is known to perform well in interference networks for low $P_{\max}$ values.
- **Best only:** Only one UE is allowed to transmit, specifically that with the best effective channel. This approach is motivated for high $P_{\max}$ values, as a naive way of nulling out multi-user interference.

The results show that the proposed ANN-based approach outperforms all benchmarks. Only the SCA approach performs comparably to the proposed ANN-based approach. However, as described above, this method relies on a sophisticated initialization rule, which requires to solve the WSEE maximization problem twice and for the complete range of $P_{\max}$ values. This is clearly not suitable for obtaining a "one-shot" solution, i.e. when the WSEE needs to be maximized only for one specific value of $P_{\max}$. Moreover, it requires some calibration depending on the channel statistics since it performs well provided the $P_{\max}$ range is sufficiently far away from the WSEE saturation region. Thus, the SCA approach from [16] has quite a higher complexity than the ANN-based method, and performs slightly worse. In conclusion, we can argue that the ANN approach is much better suited to online power allocation than state-of-the-art approaches.

VI. Conclusions

This work has proposed an online energy-efficient power control method for high-mobility networks. It was shown that an ANN can be used to solve challenging energy-efficient power control problems, providing a closed-form expression that approximates the relationship between the system propagation channels and the optimal power allocation rule. Moreover, it was shown that large training sets can be generated offline by means of a newly proposed BB procedure, which significantly reduces the complexity of the overall power control method. Numerical results indicate that near-optimal performance can be obtained, also when a slight mismatch exists between training and testing conditions.

References

[1] "NGMN 5G white paper," NGMN Alliance, Tech. Rep., Mar. 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 J. Sel. Areas Commun.*, vol. 34, no. 5, 2016.
[3] A. Zappone and E. Jorswieck, *Energy Efficiency in Wireless Networks via Fractional Programming Theory*, ser. Found. Trends Commun. Inf. Theory. Now Publishers, 2015, vol. 11, no. 3-4.
[4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
[5] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.
[6] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks," 2017. [Online]. Available: http://arxiv.org/pdf/1710.02913.pdf
[7] Y. S. Nasir and D. Guo, "Deep reinforcement learning for distributed dynamic power allocation in wireless networks," 2018. [Online]. Available: https://arxiv.org/abs/1808.00490
[8] J. Fang, X. Li, W. Cheng, Z. Chen, and H. Li, "Intelligent power control for spectrum sharing: A deep reinforcement learning approach," 2017. [Online]. Available: https://arxiv.org/abs/1712.07365
[9] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5438–5453, Oct. 2018.
[10] F. Liang, C. Shen, W. Yu, and F. Wu, "Towards optimal power control via ensembling deep neural networks," 2018. [Online]. Available: http://arxiv.org/abs/1807.10025
[11] W. Lee, M. Kim, and D. Cho, "Deep power control: Transmit power control scheme based on convolutional neural network," *IEEE Commun. Lett.*, vol. 22, no. 6, pp. 1276–1279, Jun. 2018.
[12] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function," *Neural Netw.*, vol. 6, pp. 861–867, 1993.
[13] B. Matthiesen, A. Zappone, E. A. Jorswieck, and M. Debbah, "Deep learning for optimal energy-efficient power control in wireless interference networks," 2018. [Online]. Available: https://arxiv.org/abs/1812.06920
[14] G. Calcev *et al.*, "A wideband spatial channel model for system-wide simulations," *IEEE Trans. Veh. Technol.*, vol. 56, no. 2, March 2007.
[15] F. Chollet *et al.*, "Keras," https://keras.io, 2015.
[16] B. Matthiesen, Y. Yang, and E. A. Jorswieck, "Optimization of weighted individual energy efficiencies in interference networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Barcelona, Spain, Apr. 2018.