


Article

# Improving Energy Efficiency Fairness of Wireless Networks: A Deep Learning Approach

Hoon Lee <sup>1</sup>, Han Seung Jang <sup>2,\*</sup>  and Bang Chul Jung <sup>3,\*</sup>

<sup>1</sup> Department of Information and Communications Engineering, Pukyong National University, 45 Yongso-ro, Nam-gu, Busan 48513, Korea; hlee@pknu.ac.kr

<sup>2</sup> School of Electrical, Electronic Communication, and Computer Engineering, Chonnam National University, 50, Daehak-ro, Yeosu 59626, Korea

<sup>3</sup> Department of Electrical Engineering, Chungnam National University, 99 Daehak-ro, Yuseong-gu, Daejeon 34134, Korea

\* Correspondence: hsjang@jnu.ac.kr (H.S.J.); bcjung@cnu.ac.kr (B.C.J.);  
Tel.: +82-61-659-7234 (H.S.J.); +82-42-821-6580 (B.C.J.)

Received: 8 October 2019; Accepted: 5 November 2019; Published: 11 November 2019



**Abstract:** Achieving energy efficiency (EE) fairness among heterogeneous mobile devices will become a crucial issue in future wireless networks. This paper investigates a deep learning (DL) approach for improving EE fairness performance in interference channels (IFCs) where multiple transmitters simultaneously convey data to their corresponding receivers. To improve the EE fairness, we aim to maximize the minimum EE among multiple transmitter–receiver pairs by optimizing the transmit power levels. Due to fractional and max-min formulation, the problem is shown to be non-convex, and, thus, it is difficult to identify the optimal power control policy. Although the EE fairness maximization problem has been recently addressed by the successive convex approximation framework, it requires intensive computations for iterative optimizations and suffers from the sub-optimality incurred by the non-convexity. To tackle these issues, we propose a deep neural network (DNN) where the procedure of optimal solution calculation, which is unknown in general, is accurately approximated by well-designed DNNs. The target of the DNN is to yield an efficient power control solution for the EE fairness maximization problem by accepting the channel state information as an input feature. An unsupervised training algorithm is presented where the DNN learns an effective mapping from the channel to the EE maximizing power control strategy by itself. Numerical results demonstrate that the proposed DNN-based power control method performs better than a conventional optimization approach with much-reduced execution time. This work opens a new possibility of using DL as an alternative optimization tool for the EE maximizing design of the next-generation wireless networks.

**Keywords:** energy efficiency fairness; wireless networks; power control; interference channels; deep learning; unsupervised learning; deep neural networks

## 1. Introduction

For the last decade, energy efficiency (EE) of wireless communication systems, which measures the effectiveness of the power consumption utilized for the data transmission, has been emerging as an important performance metric for the next-generation networking scenarios including wireless sensor networks, device-to-device communications, and internet-of-things (IoT) systems [1–10]. The EE metric was introduced in [1], and it is defined as the ratio of achievable data rate to total power consumption. The fractional programming framework was employed in [1,2] to address the EE maximization problems in various configurations. Extending these works, the EE maximizing

cooperative communication strategies were investigated in the multi-cell multi-user networks [3–10]. Weighted sum EE metrics was intensively studied in [4–7]. Zero forcing (ZF)-based coordinated beamforming schemes were proposed in [4], which maximize the sum EE of the multiple heterogeneous networks. To further increase the EE performance, the works in [5–7] relaxed the ZF assumption and presented optimization algorithms for identifying efficient precoding solution without the ZF condition. Due to the multi-cell and multi-user interference terms, the weighted sum EE maximization problems are no longer convex and cannot be tackled by the off-the-shelf convex optimization software. Hence, handling the non-convexity of the EE maximization tasks has been an important issue in the multi-cell environment. The authors of [5,6] employed the fractional programming framework for the weighted EE maximization problem in the multi-cell systems and provided alternating optimization algorithms where blocks of the optimization variables are determined in an iterative manner. However, such an alternating calculation loses the globally optimality of the weighted EE maximization problems. This issue was addressed in [7] by applying the branch-and-bound (BnB) method, which iteratively searches lower and upper bounds of the original problem and converges the globally optimal point. However, the iterative bound searches and solution calculations of the BnB algorithm would result in high computational complexity for the fast fading environment where the beamforming solution may not be timely obtained for the real-time communication services.

The sum EE maximization would not be an appropriate design approach since it cannot guarantee the quality of the EE for heterogeneous wireless devices. In particular, in the multi-cell systems, the EE of each cell was defined as the sum EE of all intra-cell users [4–7]. Obviously, such a design leads to highly unbalanced EE performance among multiple cells and mobile users, and cannot guarantee the EE performance of each wireless devices. To overcome this issue, the EE fairness has been suggested as an alternative performance measure of the sum EE metric [3,8–10]. The fairness of the EE can be examined by the minimum EE performance among multiple wireless devices. Thus, the EE fairness optimization has been typically expressed as the maximization task of the minimum EE performance, resulting in the max-min formulations. The max-min nature of the EE fairness formulation ensures that all the wireless devices achieve the same level of EE performance. However, the maximization of the EE fairness would be more difficult than that of the sum EE performance due to the non-smooth and non-convex objective functions. The Dinkelhach method was adopted in [8] along with the alternating optimization process for solving the EE fairness maximization problem in the multi-cell networks. To reduce the computational complexity of the alternating optimization, the successive convex approximation (SCA) framework was applied to maximize the minimum EE performance [3,9,10]. Among those, the conic approximation strategy provided in [3] has been revealed as the most effective solution in terms of both the performance and the computational complexity.

The previous works have mostly focused on the traditional optimization techniques, and thus they relied on the convex optimization software such as the CVX. The optimization approaches are powerful for identifying model-based calculation algorithms with guaranteed convergence. However, the convex optimization software generally requires iterative computations for each channel realization, implying that the algorithms in [3,8–10] would not be practical for the fast fading environment in which the channel statistics would change before computing the solution. Hence, the conventional optimization approaches might not be a good solution for handling the EE fairness maximization problems in real-time communication scenarios. In addition, due to the non-convexity of the EE formulation, the optimization algorithms in [3,8–10] cannot guarantee the globally optimality for their convergence point. It is thus not easy to identify the globally optimal performance based on the traditional optimization techniques.

To tackle these difficulties, deep learning (DL) techniques [11,12] have recently applied in the design of wireless communication systems [13–17] and resource management algorithms [18–23]. The data-driven optimization capability of the DL has been shown to be efficient for addressing model-free and non-convex formulations in the wireless communication networks. The possibility of the DL was investigated in [13–17] for a design of end-to-end physical layer communication modules.

Meanwhile, there have been intensive studies on using deep neural networks (DNNs) to solve wireless resource allocation problems [18–23]. The authors of [18] proposed a supervised learning method which learns the input–output behavior of an existing power control scheme, the weighted minimum mean square error (WMMSE) algorithm, to maximize the sum rate performance in an interference channel (IFC) setup. The method in [18] achieves a similar performance of the WMMSE algorithm with much reduced time complexity. However, the supervised learning nature in [18] can only produce the DNN simply memorizing the locally optimal solution of the WMMSE algorithm. Hence, it is not easy to handle the non-convexity of the resource allocation problems and the supervised learning approach generally yields the degraded sum rate performance.

Unsupervised learning-based power control schemes have recently been investigated [19–23] to overcome the limitations of the supervised training strategy. The unsupervised learning does not need the training label data, i.e., the solution obtained by the WMMSE as in [18], and finds an effective resource management solution by itself. The DNNs are employed to approximate the unknown optimal solution directly, whose approximation accuracy is guaranteed by the universal approximation theorem. As a result, it is possible to identify the globally optimal resource allocation solution by utilizing well-designed DNNs. In [19], the sum rate and the sum EE maximization problems were tackled via the DNNs trained in an unsupervised manner. It was shown that the unsupervised learning techniques perform better than the WMMSE algorithm which achieves a locally optimal solution for the sum rate maximization problem. In addition, the computational complexity of the DNN-based power control scheme is much lower than that of the WMMSE method requiring the iterative calculations. Similar results have been observed in [20–23], where the possibilities of the DNN for solving the non-convex sum rate maximization problem in cognitive radio and device-to-device communication setups. However, since these methods are restricted to the sum performance optimization, it is still not clear whether the DL-based approach is valid for the EE fairness maximization task with the fractional and non-smooth objective function.

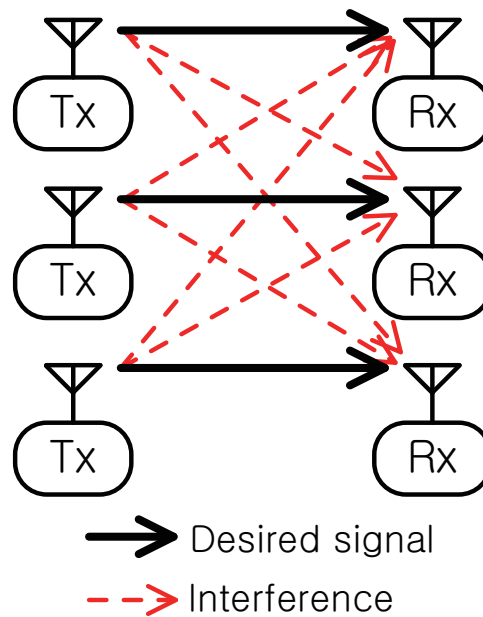
In this paper, we revisit the EE fairness maximization problem for the IFC, where the globally optimal solution is generally not available due to the non-convexity [3]. Our target is to apply the DL technique for improving the EE fairness performance of existing optimization approaches as well as reducing the computational complexity. We introduce a fully-connected DNN structure for approximating a mapping from the channel vector to the optimal power control solution. The original EE fairness maximization problem determining unknown solution computation strategy is reformulated into the training task of the DNN such that it can produce an effective power allocation solution. However, the state-of-the-art DL libraries, which are based on the gradient descent (GD) method, would not be feasible in our formulation due to the non-smooth max-min EE objective function. We thus derive a modification of the GD update rule for the EE fairness maximization problem. An unsupervised training algorithm is proposed which does not require any information regarding the optimal power control policy. The test performance of the trained DNN was examined through intensive numerical simulations. It was verified that the proposed DNN-based power control scheme performs better than the best known performance achieved by the SCA framework [3]. In addition, the numerical results prove that the DNN can significantly reduce the execution time of the conventional SCA method.

This paper is organized as follows. Section 2 describes the system model for the IFC and formulates the EE fairness maximization problem. The proposed DNN approach and the unsupervised training strategy are presented in Section 3. We provide numerical results of the trained DNNs in Section 4 and compare its performance with traditional optimization approaches. Finally, the paper is terminated with conclusions in Section 5.

Throughout this paper, we employ uppercase boldface letters, lowercase boldface letters, and normal letters for matrices, vectors, and scalar quantities, respectively. Sets of  $m$ -by- $n$  complex-valued and real-valued matrices are denoted by  $\mathbb{C}^{m \times n}$  and  $\mathbb{R}^{m \times n}$ , respectively. The expectation operation over random variable  $X$  is represented as  $\mathbb{E}_X[\cdot]$ .

## 2. System Model

We consider an IFC in Figure 1 in which multiple transmitters convey information to multiple receivers at the same time over the same frequency band. Without loss of the generality, it is assumed that there exist total  $K$  transmitter–receiver pairs. Let  $\mathcal{K} \triangleq \{1, \dots, K\}$  be the set of indices of the transmitters and receivers, i.e., index  $k$  can refer to transmitter  $k$  or receiver  $k$ . Each receiver is supported by its corresponding transmitter, and the signals leaked from each transmitter to other receivers act as interference degrading the communication performance. Such a scenario prevails in practical wireless networks including multi-cell communication systems, IoT, and device-to-device networks.



**Figure 1.** Schematic diagram of interference channels with three ( $K = 3$ ) transmitter–receiver pairs.

Throughout the paper, quasi-static frequency-flat fading is assumed, but the channel coefficient would vary at each transmission time block. Denoting  $g_{ij} \in \mathbb{C}$  as the complex channel coefficient between transmitter  $i$  and receiver  $j$  ( $i, j \in \mathcal{K}$ ), the received signal  $y_i \in \mathbb{C}$  at receiver  $i$  is written by

$$y_i = \sqrt{p_i}g_{ii}x_i + \sum_{j \in \mathcal{K} \setminus i} \sqrt{p_j}g_{ji}x_j + n_i, \quad (1)$$

where  $x_i \sim \mathcal{CN}(0,1)$  stands for the transmitted symbol at transmitter  $i$ ,  $n_i \sim \mathcal{CN}(0,1)$  equals the Gaussian noise, and  $p_i$  is the transmit power of transmitter  $i$ . From Equation (1), the data rate achieved at receiver  $i$  is written by

$$R_i(\mathbf{H}) = \log \left( 1 + \frac{h_{ii}p_i(\mathbf{H})}{1 + \sum_{j \in \mathcal{K} \setminus i} h_{ji}p_j(\mathbf{H})} \right). \quad (2)$$

where  $h_{ij} \triangleq |g_{ij}|^2$  is the channel gain between transmitter  $i$  and receiver  $j$ ,  $\mathbf{H} = \{h_{ij}, \forall i, j \in \mathcal{K}\} \in \mathbb{C}^{K \times K}$  indicates the channel matrix collecting all the channel gains. Here, the transmit power  $p_i(\mathbf{H}), \forall i \in \mathcal{K}$ , is now represented as a function of the channel gain matrix  $\mathbf{H}$  to be optimized for each channel realization  $\mathbf{H}$ .

The EE of transmitter  $i$  is defined as the ratio of the achievable rate  $R_i(\mathbf{H})$  to its total power consumption  $Q_i(\mathbf{H})$  which includes the overall power related to the signal transmission and circuit operations. The total power consumption  $Q_i(\mathbf{H})$  at transmitter  $i$  can be characterized as [3,10]

$$Q_i(\mathbf{H}) = \frac{1}{\zeta} p_i(\mathbf{H}) + P_C, \quad (3)$$

where  $\zeta \in (0, 1]$  accounts for the efficiency of the power amplifier and  $P_C$  denotes the static operation power of circuitry such as the mixer, the cooling systems, the power supply, the frequency synthesizer, the digital-to-analog converter, etc. [1]. Then, the EE  $\eta_i(\mathbf{H})$  of transmitter  $i$  is expressed as

$$\eta_i(\mathbf{H}) = \frac{R_i(\mathbf{H})}{Q_i(\mathbf{H})} = \frac{\log \left( 1 + \frac{h_{ii} p_i(\mathbf{H})}{1 + \sum_{j \in \mathcal{K} \setminus i} h_{ji} p_j(\mathbf{H})} \right)}{\frac{1}{\zeta} p_i(\mathbf{H}) + P_C}. \quad (4)$$

In a practical IFC setup, due to the heterogeneous nature of wireless propagation, the EE performance of multiple transmitter–receiver pairs is highly asymmetric. Hence, we desire to balance the EE performance by maximizing the minimum EE averaged over an arbitrary given fading distribution  $\mathbf{H}$ . The instantaneous minimum EE  $\eta_{\min}(\mathbf{H})$  for a given channel realization  $\mathbf{H}$  can be represented as

$$\eta_{\min}(\mathbf{H}) = \min_{i \in \mathcal{K}} \eta_i(\mathbf{H}) = \min_{i \in \mathcal{K}} \frac{\log \left( 1 + \frac{h_{ii} p_i(\mathbf{H})}{1 + \sum_{j \in \mathcal{K} \setminus i} h_{ji} p_j(\mathbf{H})} \right)}{\frac{1}{\zeta} p_i(\mathbf{H}) + P_C} \quad (5)$$

Then, the problem for maximizing the average minimum EE  $\mathbb{E}_{\mathbf{H}}[\eta_{\min}(\mathbf{H})] = \mathbb{E}_{\mathbf{H}}[\min_{i \in \mathcal{K}} \eta_i(\mathbf{H})]$  is formulated as

$$\max_{\mathbf{p}(\mathbf{H})} \mathbb{E}_{\mathbf{H}}[\min_{i \in \mathcal{K}} \eta_i(\mathbf{H})] \quad (6)$$

$$\text{subject to } 0 \leq p_i(\mathbf{H}) \leq P_i, \forall i \in \mathcal{K}, \quad (7)$$

where  $\mathbf{p}(\mathbf{H}) \triangleq \{p_i(\mathbf{H}), \forall i \in \mathcal{K}\} \in \mathbb{R}^{N \times 1}$  is the concatenation of the transmit power of all the transmitters and the constraint in Equation (7) indicates the transmit power constraint with  $P_i$  being the maximum allowable transmit power budget at transmitter  $i$ .

In general, Equation (6) is a non-convex problem [24], and it is highly difficult to identify the globally optimal solution  $\mathbf{p}^*(\mathbf{H})$ . To address the non-convexity of Equation (6), theoretical optimization techniques have been applied in recent literature [3,10]. In particular, the SCA framework [25] is adapted in [3,10], which tackle a series of convex approximations of Equation (6) in an iterative manner. The monotonic convergence and the complexity analysis of these SCA-based algorithms have been intensively studied. However, the conventional methods [3,10] suffer from several implementation challenges. First, due to the coupled variables in the objective function of Equation (6), the SCA framework cannot guarantee the quality of the convergence points of the SCA algorithms. To be specific, the conventional methods can only satisfy the necessary optimality condition, and, obviously, this limits the performance of the solution since they generally converges to poor local optimum and saddle points. Next, the iterative nature of the SCA framework incurs expensive and repetitive calculations for the algorithms in [3,10], resulting in the expensive computation cost for the transmitters. This becomes severe in the fast fading scenario where the channel coefficients  $\mathbf{H}$  changes before computing the solution of Equation (6) through the iterations.

To overcome such limitations of traditional optimization approaches, we propose a DL-based power control scheme for the minimum EE maximization problem in Equation (6). The DL-based resource management techniques have been recently investigated [18–23]. Existing studies have mostly

focused on the sum rate maximization problem [18,20–23], which has been known to be simpler than our formulation in Equation (6). Although the sum EE maximization was considered in [19], it is not clear whether the max-min EE formulation can be indeed addressed by the DL algorithms. Therefore, the expression power of the DNNs for solving the EE fairness problem in Equation (6) has not been adequately examined in the literature. It is worth noting that, since the training of DNNs are typically based on the gradient descent method [12,26], it would not be straightforward to verify the feasibility of the state-of-the-art DL libraries for addressing the non-smooth optimization problem in Equation (6). Therefore, maximizing the EE fairness performance is a nontrivial task through the existing sum metric training rules [18–23].

### 3. Proposed DL-Based Power Control Scheme

In this section, we present a DL approach to solve the non-convex EE fairness problem in Equation (6) efficiently. The basics of the DL are briefly introduced, followed by the proposed DL-based solution.

#### 3.1. DL Basics

As illustrated in Figure 2, we consider a fully-connected DNN with an input layer,  $L$  hidden layers, and an output layer. The DNN computes an output vector  $\mathbf{x}_{L+1} \in \mathbb{R}^{N_{L+1} \times 1}$  of length  $N_{L+1}$  for a given input feature  $\mathbf{x}_0 \in \mathbb{R}^{N_0 \times 1}$  of length  $N_0$ . The input vector  $\mathbf{x}_0$  is regarded as the training data sampled from the training set  $\mathcal{X} \triangleq \{\mathbf{x}_0\}$  collected in advance. When the distribution of  $\mathbf{x}_0$  is available, e.g., the Rayleigh fading, the training data can be randomly generated from the known probability density function. The end-to-end operation of the DNN is characterized by a mapping  $\phi: \mathbb{R}^{N_0 \times 1} \rightarrow \mathbb{R}^{N_{L+1} \times 1}$ . For convenience, let us denote layer 0 and  $L+1$  as the input and output layers, respectively, whereas hidden layers are represented by layer  $l$  for  $l = 1, \dots, L$ . Each layer  $l$  for  $l = 1, \dots, L, L+1$  produces its output vector  $\mathbf{x}_l \in \mathbb{R}^{N_l \times 1}$  of length  $N_l$  from its input  $\mathbf{x}_{l-1} \in \mathbb{R}^{N_{l-1} \times 1}$  of length  $N_{l-1}$ , which is obtained from the preceding layer  $l-1$ . To be specific, the output  $\mathbf{x}_l$  of layer  $l$  is given by

$$\mathbf{x}_l = \sigma_l(\mathbf{W}_l \mathbf{x}_{l-1} + \mathbf{b}_l), \quad (8)$$

where  $\sigma_l(\cdot)$  stands for an activation function at layer  $l$ , and  $\mathbf{W}_l \in \mathbb{R}^{N_l \times N_{l-1}}$  and  $\mathbf{b}_l \in \mathbb{R}^{N_l \times 1}$  denote a weight matrix and a bias vector at layer  $l$ , respectively. From Equation (8), it can be seen that the calculation of each layer is expressed as linear matrix multiplication and addition followed by the activation function. The activation function  $\sigma_l(\cdot)$  is an element-wise function which introduces the non-linearity to the DNN. Possible candidates for the activation functions are the sigmoid function,  $\text{sig}(z) \triangleq \frac{1}{1+e^{-z}}$ , rectified linear unit (ReLU) function,  $\text{ReLU}(z) \triangleq \max\{0, z\}$ , etc. Since the choice of the activation functions is directly related to the expression capability of the DNN, we need to carefully determine efficient functions  $\{\sigma_l(\cdot), \forall l\}$  by examining the validation performance.

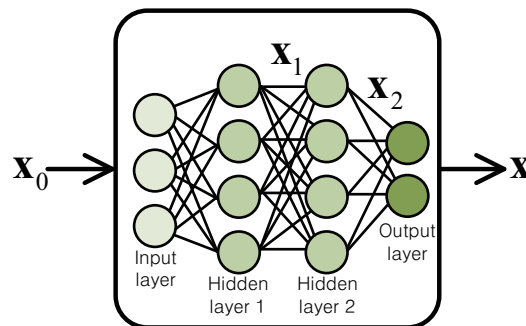


Figure 2. Fully-connected DNN with two ( $L = 2$ ) hidden layers.

Finally, the output  $\mathbf{x}_{L+1}$  of the DNN is obtained through  $L + 1$  successive computations in Equation (8). When  $\Theta \triangleq \{\mathbf{W}_l, \mathbf{b}_l, \forall l\}$  includes the set of the weight matrices and the bias vectors of the DNN, the end-to-end calculation of the DNN can be written as

$$\mathbf{x}_{L+1} = \sigma_{L+1}(\mathbf{W}_{L+1}\sigma_L(\mathbf{W}_L \cdots \sigma_1(\mathbf{W}_1\mathbf{x}_0 + \mathbf{b}_1) \cdots + \mathbf{b}_L) + \mathbf{b}_{L+1}), \quad (9)$$

$$\triangleq \phi(\mathbf{x}_0; \Theta), \quad (10)$$

where the mapping  $\mathbf{x}_{L+1} = \phi(\mathbf{x}_0; \Theta)$  characterizes the end-to-end operation of the DNN. The aim of the DL is to identify the parameter set  $\Theta$  minimizing (or maximizing) a certain objective function  $\mathcal{C}(\Theta)$  specifying targets and applications of the DNN. The objective function includes the performance of the training samples  $\mathbf{x}_0$  by measuring the average performance over the training set  $\mathcal{X}$ . In general, the objective function  $\mathcal{C}(\Theta)$  can be written by

$$\mathcal{C}(\Theta) = \mathbb{E}_{\mathcal{X}}[c(\mathbf{x}_0, \mathbf{x}_{L+1}; \Theta)], \quad (11)$$

where the function  $c(\mathbf{x}_0, \mathbf{x}_{L+1}; \Theta)$  stands for the objective of each training sample  $\mathbf{x}_0 \in \mathcal{X}$ . The minimization of  $\mathcal{C}(\Theta)$  with respect to  $\Theta$  is regarded as the training of the DNN. Due to non-convexity of the activation functions and serial calculations in Equation (8), the solution to minimize  $\mathcal{C}(\Theta)$  cannot be obtained in a closed-form expression. To this end, the state-of-the-art DL techniques employ the mini-batch stochastic gradient descent (SGD) method for determining an efficient DNN parameter  $\Theta$  [12]. An iterative update rule of the mini-batch SGD at training iteration  $t$  is expressed as

$$\begin{aligned} \Theta^{[t]} &= \Theta^{[t-1]} - \alpha \nabla_{\Theta} \mathcal{C}(\Theta^{[t-1]}), \\ &= \Theta^{[t-1]} - \alpha \mathbb{E}_{\mathcal{X}}[\nabla_{\Theta} c(\mathbf{x}_0, \mathbf{x}_{L+1}; \Theta^{[t-1]})], \\ &\simeq \Theta^{[t-1]} - \alpha \frac{1}{S} \sum_{\mathbf{x}_0 \in \mathcal{S}} \nabla_{\Theta} c(\mathbf{x}_0, \mathbf{x}_{L+1}; \Theta^{[t-1]}), \end{aligned} \quad (12)$$

where  $\Theta^{[t]}$  is the DNN parameter obtained at iteration  $t$ ,  $\alpha > 0$  denotes the learning rate, and  $\nabla_{\Theta} \mathcal{C}(\Theta^{[t]})$  accounts for the gradient of  $\mathcal{C}(\Theta)$  evaluated at  $\Theta = \Theta^{[t]}$ . In Equation (12), the expected gradient  $\mathbb{E}_{\mathcal{X}}[\nabla_{\Theta} c(\mathbf{x}_0, \mathbf{x}_{L+1}; \Theta)]$  is approximated by its empirical mean  $\frac{1}{S} \sum_{\mathbf{x}_0 \in \mathcal{S}} \nabla_{\Theta} c(\mathbf{x}_0, \mathbf{x}_{L+1}; \Theta)$  measured over a mini-batch set  $\mathcal{S} \subset \mathcal{X}$  with  $S \triangleq |\mathcal{S}|$  training samples. By using commercial graphics processing unit (GPUs), parallel and fast implementation of Equation (12) is possible. After the training, the generalization performance of the trained DNN  $\Theta$  is examined by using a test set consisting of the unseen data during the training step.

### 3.2. DL Approach for EE Fairness Maximization

We provide the DL approach for solving the EE fairness problem in Equation (6). A key idea is to approximate unknown optimal solution  $\mathbf{p}^*(\mathbf{H})$  of Equation (6) through a DNN. We construct a DNN-based optimization framework, as shown in Figure 3. We first sample numerous channel realizations  $\{\mathbf{H}\}$  from the training set  $\mathcal{H}$  for constructing a mini-batch set  $\mathcal{S}$  at each iteration of the SGD algorithm in Equation (12). The training set  $\mathcal{H} = \{\mathbf{H}\}$  consists of real measurement channel values or can be considered as a known probability density function of the fading environment such as the Rayleigh distribution.

The channel gain matrix  $\mathbf{H} \in \mathbb{C}^{N \times N}$  is first vectorized into  $\mathbf{h} \in \mathbb{C}^{N^2 \times 1}$  to be passed to the fully-connected DNN in Equation (9). The channel gain vector  $\mathbf{h}$  is then processed by  $L$ -layer DNN  $\phi(\mathbf{H}; \Theta)$  by setting  $\mathbf{x}_0 = \mathbf{H}$  in Equation (9). Then, the output of the DNN  $\mathbf{x}_{L+1}$  is utilized as the transmit

power solution, i.e.,  $\mathbf{x}_{L+1} = \mathbf{p}(\mathbf{H})$ . In other words, the solution computation rule  $\mathbf{p}(\mathbf{H})$  is alternatively calculated using the DNN as

$$\mathbf{p}(\mathbf{H}) \simeq \phi(\mathbf{H}; \Theta). \tag{13}$$

From Equation (13), we approximate the optimal transmit power  $\mathbf{p}^*(\mathbf{H})$ , which is generally unknown and intractable, through the fully-connected DNN  $\phi(\mathbf{H}; \Theta)$ . However, the expression power of the DNN is still unclear for representing the optimal solution. The following theorem provides rigorous and mathematical basis for the DNN approximation in Equation (13).

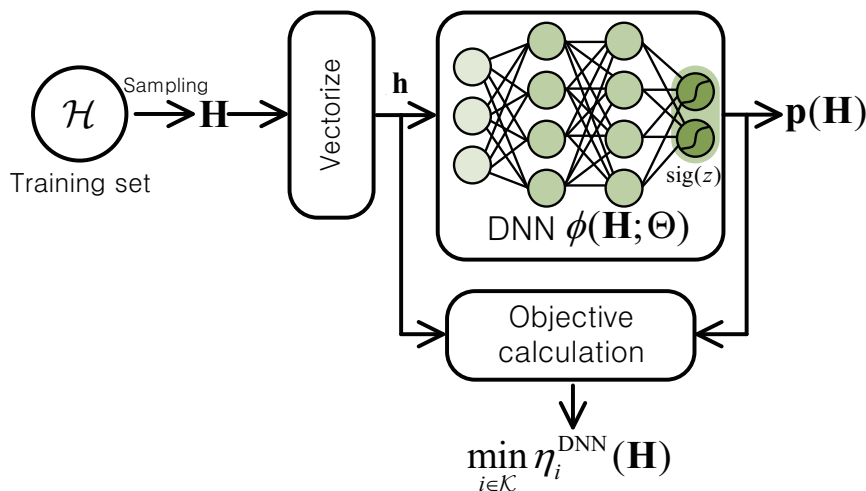


Figure 3. Proposed DNN-based EE fairness maximization.

**Theorem 1.** For a compact set  $\mathcal{H} = \{\mathbf{H}\}$ , there exists a  $L$ -layer DNN  $\phi(\mathbf{H}; \Theta)$ , which is realized with sigmoid activation functions and sufficiently large  $L$ , can approximate any continuous-valued function  $\mathbf{p}(\mathbf{H})$  over  $\mathbf{H} \in \mathcal{H}$  within an arbitrary small worst-case error  $\epsilon > 0$ . Mathematically, it is given as

$$\sup_{\mathbf{H} \in \mathcal{H}} \|\mathbf{p}(\mathbf{H}) - \phi(\mathbf{H}; \Theta)\| \leq \epsilon. \tag{14}$$

The proof of Theorem 1 is given in [18,27]. The above theorem, which is well-known as the *universal approximation theorem* [27], verifies the existence of the structure and the parameter set of the DNN  $\phi(\mathbf{H}; \Theta)$  that can act as a universal approximator for any continuous function  $\mathbf{p}(\mathbf{H})$ . A recent work [28] proved that the universal approximation theorem holds for any Lebesgue-integrable function  $\mathbf{p}(\mathbf{H})$  with  $\int_{\mathcal{H}} |\mathbf{p}(\mathbf{H})| d\mathbf{H} < \infty$  and a fully-connected DNN having the ReLU activation functions. More precisely, the result in [28] can be expressed as

$$\int_{\mathcal{H}} |\mathbf{p}(\mathbf{H}) - \phi(\mathbf{H}; \Theta)| d\mathbf{H} < \epsilon. \tag{15}$$

The Lebesgue-integrable property of  $\mathbf{p}(\mathbf{H})$  along with the universal approximation theorem in Equation (15) reveals that our DNN  $\phi(\mathbf{H}; \Theta)$  can successfully approximate non-continuous but Lebesgue-integrable mappings such as the indicator function. Since Equations (14) and (15) hold for any mapping  $\mathbf{p}(\mathbf{H})$  defined over the compact set  $\mathcal{H}$ , the universal approximation is also fulfilled for the optimal  $\mathbf{p}^*(\mathbf{H})$ . Therefore, the DNN approximation approach in Equation (13) is accurate for the unknown optimal solution  $\mathbf{p}(\mathbf{H})$  for the EE fairness maximization problem regardless of its continuity.

Although the universal approximation theorem guarantees the quality of the DNN approximation in Equation (13), it does not state how to determine the DNN parameter satisfying Equations (14) and (15). Hence, we need to derive an efficient training strategy determining  $\Theta$  for the EE fairness



maximization. To this end, by plugging Equation (13) into Equation (6), the original optimization problem in Equation (6) is first reformulated into

$$\max_{\Theta} \mathbb{E}_{\mathbf{H}} \left[ \min_{i \in \mathcal{K}} \eta_i^{\text{DNN}}(\mathbf{H}; \Theta) \right] \quad (16)$$

$$\text{subject to } 0 \leq [\phi(\mathbf{H}; \Theta)]_i \leq P_i, \forall i \in \mathcal{K}, \quad (17)$$

where  $[\mathbf{z}]_i$  indicates the  $i$ th element of a vector  $\mathbf{z}$  and  $\eta_i^{\text{DNN}}(\mathbf{H}; \Theta)$  denotes the EE of transmitter  $i$  in Equation (4) evaluated via the DNN solution  $\mathbf{p}(\mathbf{H}) = \phi(\mathbf{H}; \Theta)$ , which is defined as

$$\eta_i^{\text{DNN}}(\mathbf{H}; \Theta) = \frac{\log \left( 1 + \frac{h_{ii}[\phi(\mathbf{H}; \Theta)]_i}{1 + \sum_{j \in \mathcal{K} \setminus i} h_{ji}[\phi(\mathbf{H}; \Theta)]_j} \right)}{\frac{1}{\varepsilon} [\phi(\mathbf{H}; \Theta)]_i + P_C}. \quad (18)$$

Compared to the original formulation in Equation (6) whose optimization variable is the intractable function  $\mathbf{p}(\mathbf{H})$ , we now optimize the DNN parameter set  $\Theta$  in the DNN training problem in Equation (16), which can be handled by the state-of-the-art DL optimization tool, e.g., the mini-batch SGD algorithm. However, since the SGD method has been typically developed for unconstrained training applications, the transmit power constraint in Equation (17) cannot be straightforwardly tackled by existing DL libraries. To overcome this difficulty, we construct the DNN  $\phi(\mathbf{H}; \Theta)$  such that it always produces a feasible solution for Equation (17). We set the activation function of the output layer  $\sigma_{L+1}(\mathbf{z})$  as a scaled sigmoid function as

$$\sigma_{L+1}(\mathbf{z}) = \mathbf{P} \odot \text{sig}(\mathbf{z}), \quad (19)$$

where  $\mathbf{P} = [P_1, \dots, P_K]^T \in \mathbb{R}^{K \times 1}$  denotes the stacked vector of the transmit power budget  $P_i, \forall i \in \mathcal{K}$ , and  $\odot$  accounts for the element-wise multiplication operator. Thanks to the fact  $[\text{sig}(\mathbf{z})]_i \in [0, 1]$ , the output activation in Equation (19) is guaranteed to yield the feasible transmit power solution  $[\phi(\mathbf{H}; \Theta)]_i \in [0, P_i]$ .

Using the output activation in Equation (19), we can remove the transmit power constraint in Equation (17). Then, the problem in Equation (16) can be simplified into an unconstrained training task as

$$\max_{\Theta} \mathbb{E}_{\mathbf{H}} \left[ \min_{i \in \mathcal{K}} \eta_i^{\text{DNN}}(\mathbf{H}; \Theta) \right]. \quad (20)$$

The unconstrained DL solvers can be applied to address the problem in Equation (20). At the  $t$ th iteration, the mini-batch SGD update rule in Equation (12) for Equation (20) is expressed as

$$\Theta^{[t]} = \Theta^{[t-1]} + \alpha \frac{1}{S} \sum_{\mathbf{H} \in \mathcal{S}} \nabla_{\Theta} \left[ \min_{i \in \mathcal{K}} \eta_i^{\text{DNN}}(\mathbf{H}; \Theta^{[t-1]}) \right], \quad (21)$$

where  $\mathcal{S} = \{\mathbf{H}\}$  with  $S \triangleq |\mathcal{S}|$  is the mini-batch set of the channel gain matrix  $\mathbf{H}$  and the sign of the update direction of Equation (21) changes since our target is to maximize the minimum EE  $\min_{i \in \mathcal{K}} \eta_i(\mathbf{H})$ . In Equation (21), the gradient  $\nabla_{\Theta} [\min_{i \in \mathcal{K}} \eta_i^{\text{DNN}}(\mathbf{H}; \Theta)]$  would not be available due to the non-smooth minimum operation. This issue can be handled via a simple modification of the update rule based on the sub-gradient method [24] developed for addressing non-differential functions. Let  $\hat{i}$  be the index of the transmitter having the smallest EE performance, i.e.,

$$\hat{i} \triangleq \arg \min_{i \in \mathcal{K}} \eta_i^{\text{DNN}}(\mathbf{H}; \Theta). \quad (22)$$

Then, the sub-gradient  $\partial_{\Theta}[\min_{i \in \mathcal{K}} \eta_i^{\text{DNN}}(\mathbf{H}; \Theta)]$  with respect to  $\Theta$  is obtained as [29]

$$\partial_{\Theta} \left[ \min_{i \in \mathcal{K}} \eta_i^{\text{DNN}}(\mathbf{H}; \Theta) \right] = \nabla_{\Theta} \eta_i^{\text{DNN}}(\mathbf{H}; \Theta). \quad (23)$$

Finally, the proposed training rule can be expressed as

$$\Theta^{[t]} = \Theta^{[t-1]} + \alpha \frac{1}{S} \sum_{\mathbf{H} \in \mathcal{S}} \nabla_{\Theta} \eta_i^{\text{DNN}}(\mathbf{H}; \Theta^{[t-1]}). \quad (24)$$

### 3.3. Training and Implementation

A training algorithm for solving Equation (20) is presented in Algorithm 1. Before the training, we need to construct the training set  $\mathcal{H} = \{\mathbf{H}\}$  which contains numerous channel gain matrices. It is worth noting that the training set can be the probability space since the distribution of the channel  $\mathbf{H}$  has been well investigated and accurately modeled, e.g., the Rayleigh and the Rician fading. Hence, similar to [13], we can generate infinitely large number of training samples during the iterations of the SGD algorithm. An efficient initialization of the DNN parameter  $\Theta$  can be found by the Xavier initialization strategy [30]. At each training iteration, we sample a random mini-batch set  $\mathcal{S}$  from the known distribution of the channel. Then, the DNN parameter is updated based on the mini-batch sub-gradient method in Equation (24). We repeat these iterative updates during the predefined training iteration  $T$ . One may apply the early stopping technique [11] by observing validation performance of the DNN evaluated over validation samples which are independently generated with the training data.

---

#### Algorithm 1 Mini-batch SGD training algorithm.

---

Construct the training set  $\mathcal{H}$ .

Initialize  $t = 0$ , the DNN parameter  $\Theta^{[0]}$ , and the number of iterations  $T$ .

**for**  $t = 1 : T$  **do**

Sample a mini-batch channel set  $\mathcal{S} \subset \mathcal{H}$ .

Update  $\Theta^{[t]} \leftarrow \Theta^{[t-1]} + \alpha \frac{1}{S} \sum_{\mathbf{H} \in \mathcal{S}} \nabla_{\Theta} \eta_i^{\text{DNN}}(\mathbf{H}; \Theta^{[t-1]})$ .

**end for**

Obtain the trained DNN parameter  $\Theta^{[T]}$ .

---

The proposed DL method in Algorithm 1 does not require any knowledge of the optimal solution  $\mathbf{p}^*(\mathbf{H})$  during the training step. Thus, the proposed DL training strategy is regarded as an unsupervised learning policy [19–23]. Notice that the supervised learning approach [18] needs to generate the labeled data, i.e., the solution obtained from existing optimization algorithm, for all the channel samples  $\mathbf{H}$  in the training set  $\mathcal{H}$ . Since the proposed unsupervised learning method does not depend on the solution of the original formulation in Equation (6), the training of the DNN can be done in a much simpler way compared to the supervised training algorithms [18].

The training process might require intensive computations for updating a large number of weights and biases with numerous training samples. However, since we can obtain the training data in advance, the DNN training task can be performed offline before the real-time communication. Once the DNN is trained, its real-time calculations are replaced with the linear matrix multiplications in Equation (8) whose computational complexity is much lower than that of the iterative optimization methods such as the SCA algorithm [3]. Thus, in terms of the real-time execution time, the proposed DNN-based power control scheme would be more efficient than the conventional algorithm. This is further discussed through numerical results in Section 4. The implementation of the proposed DL-based power control scheme can be achieved by means of memory units at the transmitters. Specifically, the trained DNN parameter  $\Theta$ , which is a collection of trained weight matrices and bias vectors, is stored to memory

units of each transmitter. Then, based on the channel matrix  $\mathbf{H}$  each transmitter  $i$  can compute its transmit power  $p_i(\mathbf{H})$  as  $p_i(\mathbf{H}) = [\phi(\mathbf{H}; \Theta)]_i$ .

The DNN trained over a certain distribution of  $\mathbf{H}$  might not be applied to other channel models. To this end, we train multiple DNNs for all possible analytical channel models in advance. For instance, two DNNs are individually trained with the Rayleigh and the Rician fading setups. In the real-time communication, the transmitters, e.g., the eNodeBs in the LTE systems, can determine the channel statistics based on the feedback information of the receivers. Hence, the transmitters can estimate the distribution of  $\mathbf{H}$  and can select one DNN that matches with the real-time channel statistics. The transmitters can also check the change in the long-term channel statistics, and the dynamic switching of the DNN is possible in practice. Thanks to the high-speed memory units, the delay incurred by the DNN switching is negligible compared to the overall communication duration. As a result, the proposed DNN approach can be directly applied to a more realistic scenario when the long-term characteristics of the channels vary without retraining processes.

#### 4. Numerical Results

In this section, we present numerical results evaluating the maximized minimum (max-min) EE performance of the proposed DNN scheme. Unless otherwise stated, a DNN with  $L = 6$  hidden layers was considered. The dimension of the hidden layer was set to 50K, whereas the output layer dimension was fixed to  $K$  to produce  $K$ -dimensional transmit power solution  $\mathbf{p}(\mathbf{H}) \in \mathbb{R}^{K \times 1}$ . The ReLU function  $\text{ReLU}(z) = \max\{0, z\}$ , which is a popular choice for hidden activation functions [11–13,20], was adopted for all hidden layers. We applied the batch normalization technique [31] after each hidden and output layer to accelerate the training process. Unless otherwise stated, the learning rate  $\alpha$  and the mini-batch size  $S$  were set to  $\alpha = 0.0003$  and  $S = 5000$ , respectively. The early stopping [11] was employed with  $T = 25,000$  training iterations and  $10^6$  validation samples.

We utilized the Adam method [26] for the SGD algorithm. At each iteration  $t$ , the update rules of the moving average of the gradient  $\mathbf{m}^{[t]}$ , the second moment  $\mathbf{v}^{[t]}$ , and the DNN parameter  $\Theta^{[t]}$  are given, respectively, by [26]

$$\begin{aligned}\mathbf{m}^{[t]} &= \beta_1 \mathbf{m}^{[t-1]} + (1 - \beta_1) \mathbf{g}^{[t-1]}, \\ \mathbf{v}^{[t]} &= \beta_2 \mathbf{v}^{[t-1]} + (1 - \beta_2) \mathbf{g}^{[t-1]} \odot \mathbf{g}^{[t-1]}, \\ \Theta^{[t]} &= \Theta^{[t-1]} - \eta \frac{\sqrt{1 - \beta_2^t}}{1 - \beta_1^t} \mathbf{m}^{[t]} \odot (\sqrt{\mathbf{v}^{[t]}} + \varrho),\end{aligned}$$

where  $\mathbf{g}^{[t]}$  is defined as the instantaneous gradient of the cost function,  $\odot$  denotes the element-wise division, and  $\sqrt{\mathbf{v}^{[t]}}$  indicates the element-wise square root of the vector  $\mathbf{v}^{[t]}$ . Two hyper parameters  $\beta_1 \in [0, 1)$  and  $\beta_2 \in [0, 1)$  adjust the decaying rates of the moving average of the first and the second momentums of the gradient, respectively. Following [26], we fixed  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$ . A positive parameter  $\varrho > 0$  was introduced to avoid numerical errors. For the initialization,  $\mathbf{m}^{[0]}$  and  $\mathbf{v}^{[0]}$  were set to the zero vectors.

All the channel gains for the training, validation, and testing steps were generated with the Rayleigh fading. The transmit power constraint  $P_i$  for each transmitter  $i$  was fixed as  $P_1 = \dots = P_K = P$  and the power amplifier efficiency  $\zeta$  was given as  $\zeta = 0.35$  [3]. The training was implemented in Python with Tensorflow on a PC with Intel Core i7 CPU @3.60 GHz, 32 GB of RAM, and GeForce RTX 2080 GPU.

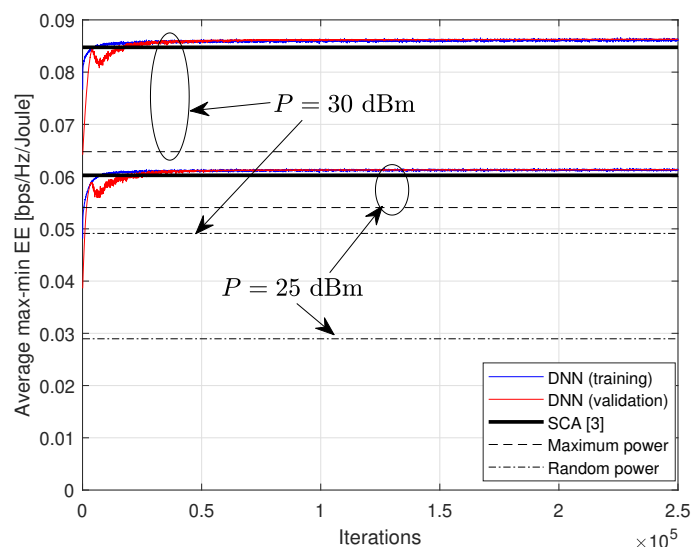
##### 4.1. DNN Validation

In this subsection, we validate our hyper-parameter setting for the DNN training such as the number of hidden layers and the mini-batch size. In Figure 4, we first present the convergence of

the training and the validation performance of the proposed DNN with  $K = 2$  and  $P_C = 33$  dBm for  $P = 25$  and 30 dBm. As reference, we consider the following three baselines.

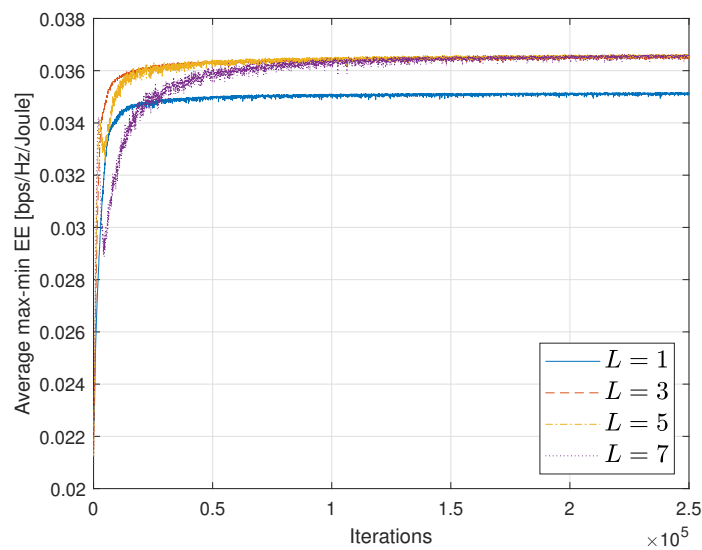
- SCA [3]: The SCA-based iterative optimization algorithm in [3] was implemented with the CVX and MOSEK solver.
- Maximum power: The transmit power was set to its maximum value as  $p_i(\mathbf{H}) = P, \forall i \in \mathcal{K}$ .
- Random power: A random transmission policy was employed where  $p_i(\mathbf{H})$  is uniformly distributed over  $[0, P]$ .

It is shown that the performance of the DNN evaluated over the training and the validation sets converges well, implying that the proposed SGD update (Equation (24)) is effective for the max-min formulation in Equation (6). It is interesting to see that the proposed DNN method performs better than other baselines, especially the conventional SCA-based optimization approach [3]. Due to the non-convex nature of Equation (6), the SCA algorithm developed in [3] cannot guarantee the global optimality and converges to the KKT stationary point of Equation (6). On the contrary, the proposed DL approach is able to identify the global optimum for Equation (6) as discussed in the universal approximation theorem [27,28]. This infers that the DNN could handle the non-convexity of the EE fairness maximization problem with the data-driven SGD update rule in Equation (24).



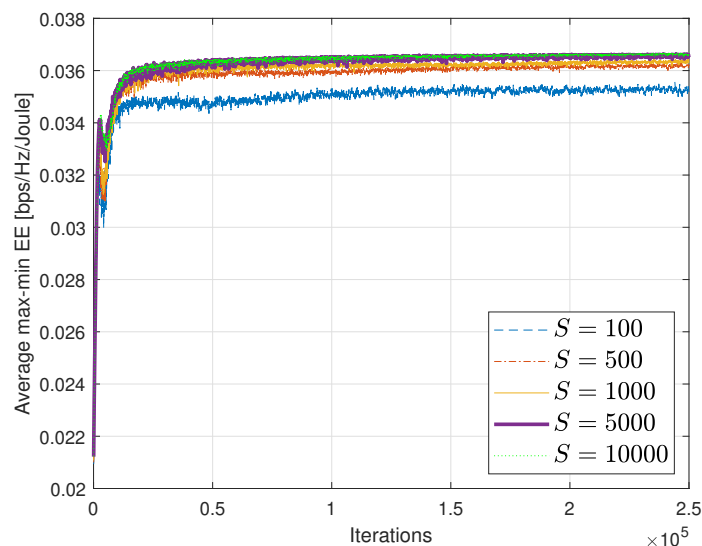
**Figure 4.** Convergence of training and validation performance with  $K = 2$  and  $P_C = 33$  dBm for different  $P$ .

Figure 5 illustrates the convergence behavior of the validation performance with  $K = 4$ ,  $P_C = 33$  dBm, and  $P = 30$  dBm for various number of hidden layers  $L$ . The convergence performance of a DNN with a single hidden layer, i.e.,  $L = 1$ , is worse than other configurations, whereas the performance of DNNs with  $L = 3, 5$ , and 7 hidden layers show similar performance. However, it is observed that the convergence speed of the case with  $L = 7$  is slower than that of  $L = 3$  and 5. This is because the DNN with such a large number of hidden layers overfits to its training data [11]. Hence, the validation performance of the DNN with  $L = 7$  would not that be good compared to other cases. We thus fixed  $L = 5$  for the rest of the simulation.



**Figure 5.** Convergence of validation performance with  $K = 4$ ,  $P_C = 33$  dBm, and  $P = 30$  dBm for different  $L$ .

In Figure 6, we plot the convergence performance with different mini-batch size  $S$ . The DNN trained with a small mini-batch set ( $S = 100$  and  $500$ ) converges to a poor performance. We can see that increasing the mini-batch size is beneficial to improve the validation performance of the DNN since the empirical expectation of the sub-gradient in Equation (24) becomes more accurate as  $S$  grows. However, a large mini-batch size needs intensive SGD computation in the training step. Thus, we need to choose an appropriate  $S$  for achieving a good trade-off between the performance and the training complexity. In Figure 6, we can see that  $S = 5000$  is an efficient choice for our scenario.

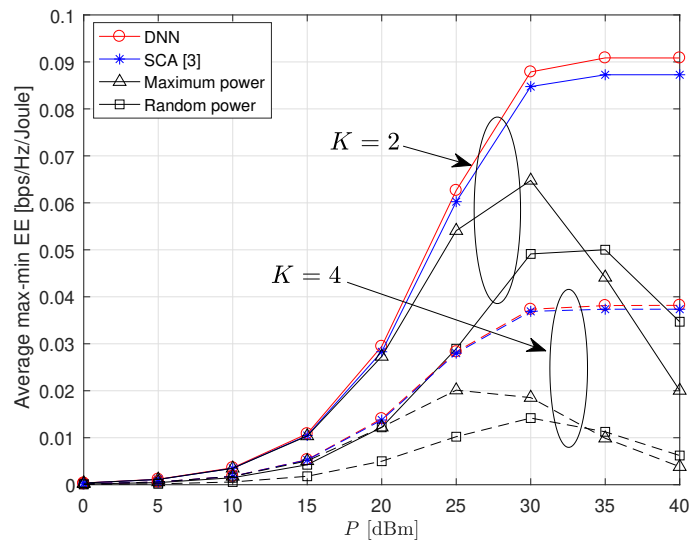


**Figure 6.** Convergence of validation performance with  $K = 4$ ,  $P_C = 33$  dBm, and  $P = 30$  dBm for different  $S$ .

#### 4.2. Performance Comparison

In this subsection, we focus on evaluating the test performance of the trained DNN and comparison with the baseline methods. Figure 7 depicts the average max-min EE performance with respect to the transmit power budget  $P$  with  $P_C = 33$  dBm for  $K = 2$  and  $4$ . It is clear that the

proposed DNN-based power control scheme is superior to other baselines. In particular, the proposed DNN method performs better than the conventional optimization approach [3] regardless of  $K$  and  $P$ .



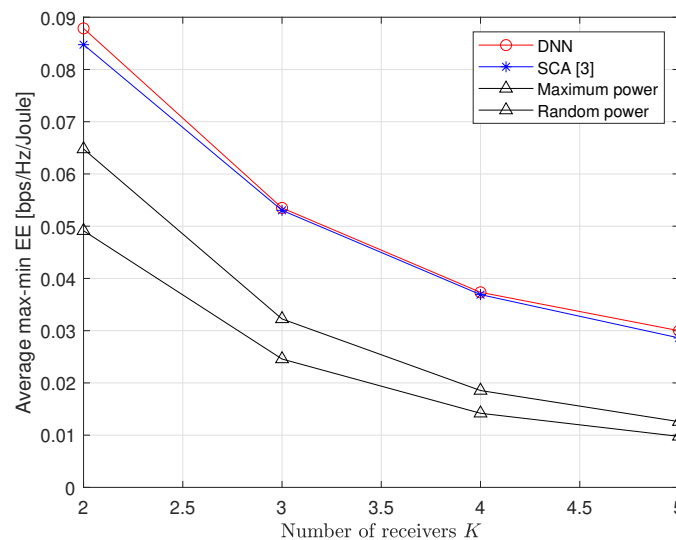
**Figure 7.** Average max-min EE performance as a function of  $P$  with  $P_C = 33$  dBm for  $K = 2$  and 4.

Table 1 shows the average CPU running time of the SCA method [3] and the proposed DNN scheme. Notice that the computations of the SCA method are online tasks which should be performed for each channel realization, whereas the training of the DNN can be performed offline with the aid of the training data collected in advance. Since the proposed DL approach does not require any retraining processes, the offline training complexity does not affect the online execution time of the trained DNN. Therefore, for fair comparison, the complexity of the DNN was evaluated only for the online testing step for producing the power control solution via successive linear operations in Equation (9). We implemented the test calculation of the trained DNN using Matlab and CPU and Matlab with CVX carried out the SCA algorithm. The CPU simulation was performed on a PC with Intel Core i5 CPU and 8 GB of RAM. We can see that the proposed DNN remarkably reduces the CPU running time compared to that of the SCA algorithm [3], which iteratively calculates the solution for each channel realization. This is because the real-time computation of the trained DNN is simply realized by the cascaded matrix multiplications and additions in Equation (9). Therefore, the power control solution of the proposed DNN method can be easily obtained compared to the iterative SCA algorithm [3]. It is also worth noting that the complexity of the proposed DL method only depends on the number of receivers  $K$  but not the transmit power constraint  $P$  since the proposed DNN structure relies only on  $K$ . The results in Table 1 infer that the proposed DNN not only improves the EE fairness performance but also reduces the energy required for the optimization tasks at the transmitters. Therefore, the proposed DL-based power control strategy could provide high EE performance in terms of both the communication and the computations.

**Table 1.** Average CPU Running Time [s].

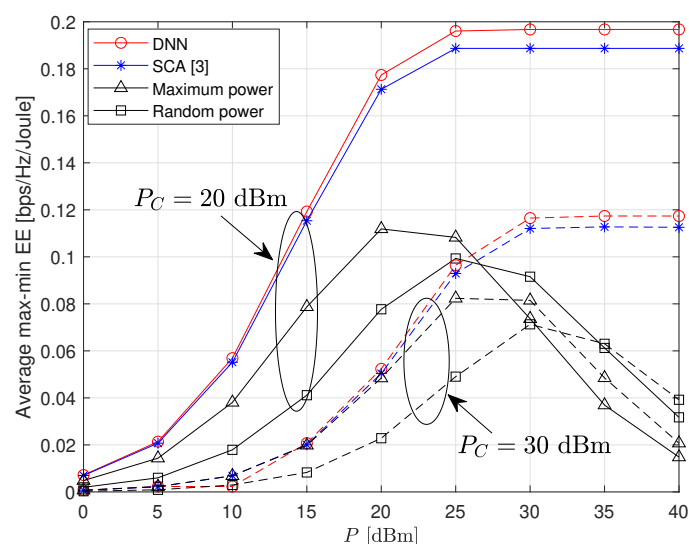
	SCA [3]		DNN
	$P = 10$ dBm	$P = 20$ dBm	
$K = 2$	0.79	1.67	$6.00 \times 10^{-4}$
$K = 3$	0.97	2.64	$9.85 \times 10^{-4}$
$K = 4$	1.10	3.13	$1.24 \times 10^{-3}$

Figure 8 presents the average max-min EE performance with respect to the number of receivers  $K$ . Both the proposed DNN scheme and the conventional SCA algorithm show a similar performance for a large  $K$ . However, as observed in Table 1, the computational complexity of the DNN is much lower than that of the conventional SCA approach. We thus conclude that the proposed DNN scheme still has advantages over the method in [3] even with a small performance gain.



**Figure 8.** Average max-min EE performance as a function of  $K$  for  $P = 30$  dBm and  $P_C = 33$  dBm.

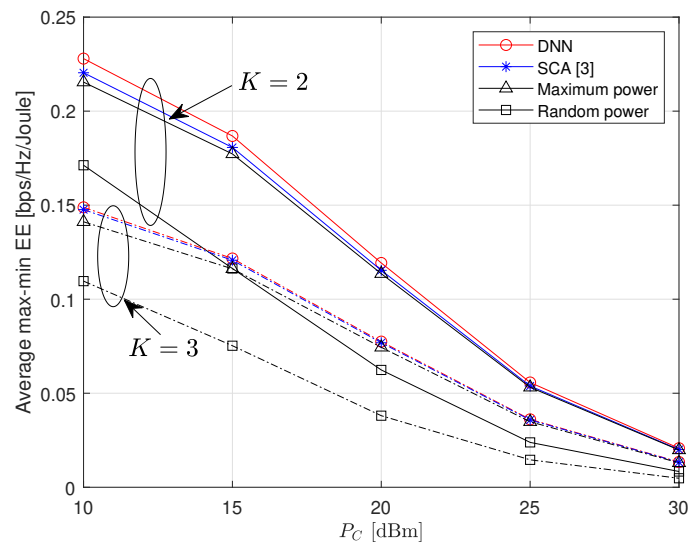
Figure 9 provides the average max-min EE performance by changing  $P$  with various circuit power  $P_C$ . It is revealed that the performance gain achieved by the proposed DNN scheme gets larger as  $P$  increases. We can also see that the DNN is still powerful for a large circuit power with  $P_C = 30$  dBm.



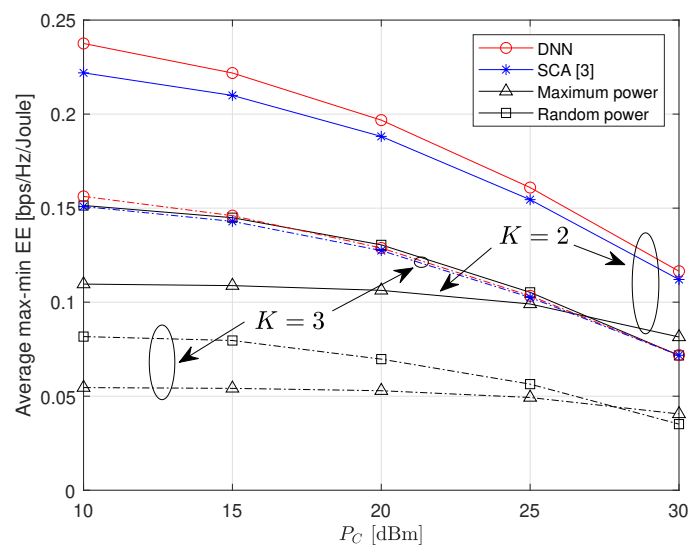
**Figure 9.** Average max-min EE performance as a function of  $P$  with  $K = 2$  for  $P_C = 20$  and 30 dBm.

To further examine the impact of the circuit power, we present the average max-min EE performance in Figure 10 for  $K = 2$  and 3. It is shown that the proposed DL method performs well regardless of the circuit power  $P_C$ . We observe that the gain achieved by the proposed DL scheme over the conventional SCA algorithm gets larger as  $P_C$  decreases. The DNN scheme becomes more

powerful in the high regime of  $P = 30$  dBm (Figure 10b). Such a performance gain obtained by the DL approach is significantly meaningful with much reduced computational complexity.



(a)  $P = 15$  dBm



(b)  $P = 30$  dBm

**Figure 10.** Average max-min EE performance as a function of  $P_C$  for  $K = 2$  and 3.

## 5. Conclusions

This paper studies a DL method for improving the EE fairness performance in the IFC systems where multiple transmitters simultaneously communicate with their corresponding receivers. In particular, the maximization task of the minimum EE is investigated to determine energy-efficient transmit power control strategy at the transmitters. This results in non-convex, fractional, and non-smooth max-min EE formulations. Although the EE fairness maximization has been recently addressed by using the SCA framework, such a traditional optimization approach suffers from high computational complexity for the iterations and the optimality loss. To tackle these issues, we present a DNN-based power control scheme where the unknown optimal solution is approximated by carefully designed DNNs. The universal approximation theorem guarantees the optimality of the



proposed DNN approach; however, it is still unclear how to identify an efficient DNN parameter set for maximizing the EE fairness performance. We propose an unsupervised learning algorithm such that the DNN is trained to directly maximize the minimum EE performance by observing numerous channel samples during the training steps. A careful analysis on the SGD update rule for the non-smooth objective function is presented for the proposed training algorithm. Intensive numerical results verify the effectiveness of the proposed DL-based power control method over the conventional SCA algorithm. It was observed that the EE fairness performance can be improved by the proposed DL method. In addition, it is shown that the DNN can shift the online calculations of the SCA method to its offline training step. As a result, the real-time CPU execution time of the proposed DL scheme is much lower than that of the SCA algorithm.

**Author Contributions:** H.L., H.S.J., and B.C.J. conceived and designed the experiments; H.L. performed the numerical simulations; H.L. and H.S.J. analyzed the data; B.C.J. contributed analysis tools; B.C.J. took part in the validation and analysis of results; and H.L., H.S.J., and B.C.J. wrote the paper.

**Funding:** This research received no external funding.

**Acknowledgments:** This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2019R1F1A1060648), in part by the NRF grant funded by the Korea government Ministry of Science and ICT (No. 2019R1F1A1061023), and in part by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2019-2017-0-01635) supervised by the IITP (Institute for Information & communications Technology Promotion).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Isheden, C.; Chong, Z.; Jorswieck, E.; Fettweis, G. Framework for link-level energy efficiency optimization with informed transmitter. *IEEE Trans. Wirel. Commun.* **2012**, *11*, 2946–2957. [[CrossRef](#)]
2. Zappone, A.; Jorswieck, E. Energy efficiency in wireless networks via fractional programming theory. *Found. Trends Commun. Inf. Theory* **2015**, *11*, 185–396. [[CrossRef](#)]
3. Nguyen, K.-G.; Vu, Q.-D.; Juntti, M.; Tran, L.-N. Distributed solutions for energy efficiency fairness in multicell MISO downlink. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 6232–6247. [[CrossRef](#)]
4. Xu, Z.; Yang, C.; Li, G. Y.; Liu, Y.; Xu, S. Energy-efficient CoMP precoding in heterogeneous networks. *IEEE Trans. Signal Process.* **2014**, *64*, 1005–1017. [[CrossRef](#)]
5. He, S.; Huang, Y.; Jin, S.; Yang, L. Coordinated beamforming for energy efficient transmission in multicell multiuser systems. *IEEE Trans. Commun.* **2013**, *61*, 4961–4971. [[CrossRef](#)]
6. He, S.; Huang, Y.; Yang, L.; Ottersten, B. Coordinated multicell multiuser precoding for maximizing weighted sum energy efficiency. *IEEE Trans. Signal Process.* **2014**, *62*, 741–751. [[CrossRef](#)]
7. Tervo, O.; Tran, L.-N.; Juntti, M. Optimal energy-efficient transmit beamforming for multi-user MISO downlink. *IEEE Trans. Signal Process.* **2015**, *63*, 5574–5588. [[CrossRef](#)]
8. He, S.; Huang, Y.; Jin, S.; Yu, F.; Yang, L. Max-min energy efficient beamforming for multicell multiuser joint transmission systems. *IEEE Commun. Lett.* **2013**, *17*, 1956–1959. [[CrossRef](#)]
9. Du, B.; Pan, C.; Zhang, W.; Chen, M. Distributed energy-efficient power optimization for CoMP systems with max-min fairness. *IEEE Commun. Lett.* **2014**, *18*, 999–1002. [[CrossRef](#)]
10. Nguyen, K.-G.; Tran, L.-N.; Tervo, O.; Vu, Q.-D.; Juntti, M. Achieving energy efficiency fairness in multicell MISO downlink. *IEEE Commun. Lett.* **2015**, *19*, 1426–1429. [[CrossRef](#)]
11. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
12. LeCun, Y.; Yoshua, B.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)] [[PubMed](#)]
13. O’Shea, T.; Hoydis, J. An introduction to deep learning for the physical layer. *IEEE Trans. Cog. Commun. Netw.* **2017**, *3*, 563–575. [[CrossRef](#)]
14. Dörner, S.; Cammerer, S.; Hoydis, J.; Brink, S. Deep learning-based communication over the air. *IEEE J. Sel. Top. Signal Process.* **2018**, *12*, 132–143. [[CrossRef](#)]
15. Lee, H.; Lee, I.; Lee, S. H. Deep learning based transceiver design for multi-colored VLC systems. *Opt. Express* **2018**, *26*, 6222–6238. [[CrossRef](#)] [[PubMed](#)]
16. Lee, H.; Lee, I.; Quek, T.Q.S.; Lee, S.H. Binary signaling design for visible light communication: A deep learning framework. *Opt. Express* **2018**, *26*, 18131–18142. [[CrossRef](#)] [[PubMed](#)]

17. Lee, H.; Lee, S.H.; Quek, T.Q.S.; Lee, I. Deep learning framework for wireless systems: Applications to optical wireless communications. *IEEE Commun. Mag.* **2019**, *57*, 35–41. [[CrossRef](#)]
18. Sun, H.; Chen, X.; Shi, Q.; Hong, M.; Fu, X.; Sidiropoulos, N.D. Learning to optimize: Training deep neural networks for interference management. *IEEE Trans. Signal Process.* **2018**, *66*, 5438–5453. [[CrossRef](#)]
19. Lee, W.; Cho, D.-H.; Kim, M. Deep power control: Transmit power control scheme based on convolutional neural network. *IEEE Commun. Lett.* **2018**, *22*, 1276–1279. [[CrossRef](#)]
20. Lee, H.; Lee, S.H.; Quek, T.Q.S. Constrained deep learning for wireless resource management. In Proceedings of the ICC 2019—2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–6.
21. Lee, W.; Resource allocation for multi-channel underlay cognitive radio network based on deep neural network. *IEEE Commun. Lett.* **2018**, *22*, 1942–1945. [[CrossRef](#)]
22. Lee, W.; Kim, M.; Cho, D.-H. Transmit power control using deep neural network for underlay device-to-device communication. *IEEE Wirel. Commun. Lett.* **2019**, *8*, 141–144. [[CrossRef](#)]
23. Lee, W.; Kim, M.; Cho, D.-H. Deep learning based transmit power control in underlaid device-to-device communication. *IEEE Sys. J.* **2019**, *13*, 2551–2554. [[CrossRef](#)]
24. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.
25. Hong, M.; Razaviyayn, M.; Luo, Z.-Q.; Pang, J.-S. A unified algorithmic framework for block-structured optimization involving big data: With applications in machine learning and signal processing. *IEEE Signal Process. Mag.* **2016**, *33*, 57–77. [[CrossRef](#)]
26. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the the 3rd International Conference for Learning Representations. Represent. (ICLR), San Diego, CA, USA, 7–9 May 2015.
27. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [[CrossRef](#)]
28. Lu, Z.; Pu, H.; Wang, F.; Hu, Z.; Wang, L. The expressive power of neural networks: A view from the width. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 6231–6239.
29. Boyd, S.; Xiao, L.; Mutapcic, A. *Subgradient Methods*; Technical Report EE364b; Stanford Univ: Stanford, CA, USA, 2014.
30. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
31. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariance shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).