

## Article

# Quantum-Driven Energy-Efficiency Optimization for Next-Generation Communications Systems

Su Fong Chien <sup>1,†</sup> , Heng Siong Lim <sup>2,†</sup> , Michail Alexandros Kourtis <sup>3,†</sup>, Qiang Ni <sup>4,†</sup>, Alessio Zappone <sup>5,†</sup> and Charilaos C. Zarakovitis <sup>3,4,6,\*,†</sup>

- <sup>1</sup> MIMOS Berhad, Technology Park Malaysia, Kuala Lumpur 57000, Malaysia; sf.chien@mimos.my  
<sup>2</sup> Faculty of Engineering and Technology, Multimedia University, Melaka 75450, Malaysia; hslim@mmu.edu.my  
<sup>3</sup> National Centre for Scientific Research “DEMOKRITOS” (NCSR), Institute of Informatics and Telecommunications, 153 10 Athens, Greece; akis.kourtis@iit.demokritos.gr  
<sup>4</sup> Department of Computing and Communications, Lancaster University, Lancaster LA1 4YW, UK; q.ni@lancaster.ac.uk  
<sup>5</sup> Department of Electrical and Information Engineering, University of Cassino and Southern Lazio, 03043 Cassino, Italy; alessio.zappone@unicas.it  
<sup>6</sup> AXON LOGIC P.C., Innovation Department, 142 31 Athens, Greece  
\* Correspondence: c.zarakovitis@axonlogic.gr; Tel.: +30-6944-6962-21  
† These authors contributed equally to this work.

**Abstract:** The advent of deep-learning technology promises major leaps forward in addressing the ever-enduring problems of wireless resource control and optimization, and improving key network performances, such as energy efficiency, spectral efficiency, transmission latency, etc. Therefore, a common understanding for quantum deep-learning algorithms is that they exploit advantages of quantum hardware, enabling massive optimization speed ups, which cannot be achieved by using classical computer hardware. In this respect, this paper investigates the possibility of resolving the energy efficiency problem in wireless communications by developing a quantum neural network (QNN) algorithm of deep-learning that can be tested on a classical computer setting by using any popular numerical simulation tool, such as Python. The computed results show that our QNN algorithm can be indeed trainable and that it can lead to solution convergence during the training phase. We also show that the proposed QNN algorithm exhibits slightly faster convergence speed than its classical ANN counterpart, which was considered in our previous work. Finally, we conclude that our solution can accurately resolve the energy efficiency problem and that it can be extended to optimize other communications problems, such as the global optimal power control problem, with promising trainability and generalization ability.

**Keywords:** energy efficiency; quantum computing; quantum deep neural networks; quantum entanglement; quantum machine learning; quantum superposition; resource optimization



**Citation:** Chien, S.F.; Lim, H.S.; Kourtis, M.A.; Ni, Q.; Zappone, A.; Zarakovitis, C.C. Quantum-Driven Energy-Efficiency Optimization for Next-Generation Communications Systems. *Energies* **2021**, *14*, 4090. <https://doi.org/10.3390/en14144090>

Academic Editor: Edmundas Kazimieras Zavadskas

Received: 30 May 2021  
Accepted: 25 June 2021  
Published: 6 July 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 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 (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The intelligence of resolving complex problems in engineering, science, finance, etc., has become increasingly dependent on machine learning methods and, especially, deep-learning approaches driven by artificial neural networks (ANNs). In the communications engineering regime, for example, the studies in [1,2] recently utilized ANN deep-learning to resolve the non-deterministic polynomial-time (NP-hard) problem of energy efficiency by means of training an ANN architecture with a set of optimal power allocations so as to enable the system to learn how to autonomously predict the optimal power control policy with respect to the varying wireless channel conditions.

Yet, one can exploit classical ANN learning to improve quantum tasks [3,4], such as adaptive quantum computation [5], quantum metrology [6], modeling non-Markovian quantum processes [7] and coherent transport of quantum states [8], or exploit quantum algorithms to speed up classical learning processes [9,10], or exploit quantum computing

devices to carry out learning tasks with quantum data [11,12]. These options are mainly available due to the ability of quantum deep learning in performing computationally intensive tasks by amalgamating the versatility of neural networks with the computing power of quantum neurons [13,14]. As such, quantum deep-learning based on ANN rationalism has become an interesting new research topic with industry and academia to seek more sophisticated ways in exploring the potentials of quantum-oriented solutions.

Within this direction, we notice that the artificial perceptron unit from which ANNs are constructed cannot extend to the actual capacity of the quantum perceptron. This is because the ANN perceptron activation functions are non-linear in nature and do not exactly match with the linear quantum operators. That means that, although classical ANN training can exploit the advantages of quantum information, its complexity can be further improved upon embedding it into quantum neural network (QNN) formalism [13–17], which dictates that training can be performed using a layer-by-layer step and without having to consider all layers, as done with classical ANNs. Notably, the literature reports quite limited QNN architectures that are proven to train classical systems based on quantum deep learning. For instance, the QNN subroutine in [13] attains storing intermediate results in quantum random access memory, which speeds up the training time compared to ANN algorithms. The study in [14], models the quantum neurons as natural generalizations of perceptrons in feed-forward QNN machines, while authors in [15] use a qubit-circuit setup to facilitate the system to learn classical ANN data assisted by a quantum algorithm. Furthermore, the recent study in [16] examines the complexity of the machine learning dynamics in QNNs by proposing parameterized quantum circuits (e.g., hybrid quantum–classical frameworks) that can approximate the deviation from the actual optimal points explicitly and establish the ANN complexity as a function of the parameters of QNN. Finally, another recent study in [17] focuses on inverted harmonic oscillators to analyze the behavior of QNN complexity at different quantum timescales encoded in dissipation, scrambling and asymptotic regimes, which serve as useful guidelines for generalizing ANNs and evaluating QNNs.

Bearing the quantum technology landscape and its perspective, this paper introduces QNN modeling tailored to optimize the power control of the energy-efficiency problem in wireless communications. We recall that the energy-efficiency problem is a type of sum-of-ratios problem with NP-hard computational complexity, meaning that it belongs to the hardest class of problems to resolve and integrate in a practical system setting. Particularly, we initially show how the optimal power control policy can be approached using a classical ANN, where learning is typically addressed over a series of input, output and hidden layers. We then present the proposed QNN deep-learning modeling, which facilitates the computation of the parameter matrices of each perceptron unitary operator in a layer-by-layer fashion (i.e., without any need to rely on additional unitary operators of the full quantum system) and contributes in a threefold manner as summarized in the following.

1. The proposed QNN model computes the trace over all qubits, accounting for the total number of perceptrons over all system layers, meaning that it introduces a comparable training step size to the ANN, which typically computes the trace of limited qubits, accounting for the perceptron that acts on the running and previous layers. Therefore, we show through experiments that the convergence speed of our QNN is comparable to its ANN analogue for deriving the optimal power control policy of the energy-efficiency problem.
2. To calculate the training function at each QNN training step, our model formulates the non-unitary stochastic transition mapping of the overall system by considering the system parameters of two layers only, meaning that the size of the parameter matrices depends only on the width of the network, rather than the width of each layer as required in ANN.
3. Due to its small step size and lightweight mapping, our QNN training process can be approached with similar algorithmic logic as in the classical ANN systems (i.e., by following a quantum analogue of back-propagation). Therefore, we develop a highly

practicable QNN-oriented deep-learning algorithm that can be tested in a classical computer setting using any popular numerical simulation tool, such as Python.

To our knowledge, this study is the first to consider resolving the energy efficiency problem using QNN deep learning in an effort to provide incentives for exploring the quantum perspective in different and more complex communications problems.

The practical significance of our work stands paramount for creating new versions of truly distributed and decentralized network intelligence that can spread across the Cloud to the Edge, down to reaching end-user devices. In that sense, Things, for example, that nowadays cannot cope with learning and training tasks due to their limited computational capacity, could gain inference via more lightweight QNN learning and training algorithms embedded on Things. Indeed, as proved in [18–20] and verified in [13–17], QNN algorithms have the potential to speed up learning and training over classical ANN algorithms because QNNs do not involve large dimensional data in both supervised and unsupervised learning approaches. That means by having the same computing capacity available at, for example, a Thing, an Edge Node, a server, etc., we can resolve more complex and process-intensive tasks by simply switching from ANN to QNN architectural models. This could further foster advancements on a variety of use cases, such as connected vehicles, industrial Internet of Things, smart homes and cities, etc., where the need for distributing intelligence and computing emerges. In the forthcoming Thing-to-Cloud era, the human–human, human-to-machine and machine-to-machine interactions are expected to require faster computing facilities for resource management, where QNN models can become the mean for resolving such complex problems in real-time fashion.

The remainder of the paper is as follows. Section 2 formulates the network system modeling and the energy efficiency problem of power control. Sections 3 and 4 present the optimization procedure of the problem at hand, using a generalized ANN approach and the proposed QNN paradigm, respectively. Section 5 evaluates our findings using numerical simulations, with Section 6 concluding the paper.

## 2. System Model and Problem Formulation

To identify the key parameters of the proposed QNN training process with respect to those utilized in ANN approaches, we draw our focus to a widely-adopted system modeling (e.g., [1,21,22]) illustrated in Figure 1. Particularly, our modeling structures the uplink transmissions of a multi-cell interference network of total bandwidth  $B$  and  $L$  with a total number of users serviced by  $M$  total number of Base Stations (BSs), each one equipped with  $N$  antennas. Assuming standard interference effects between the channels of user  $i = 1, 2, \dots, L$  and user  $j \neq i$  at each BS  $m = 1, 2, \dots, M$ , and by recalling the Shannon capacity theorem, we can describe the maximum achievable data rate of user  $i$  at its associated BS  $m_i$ :

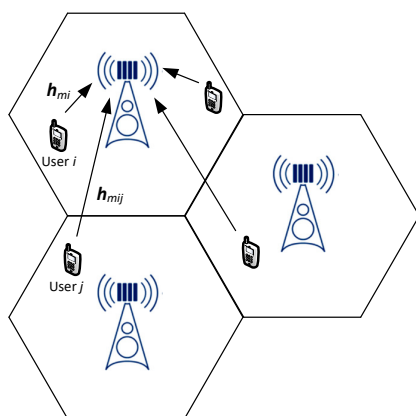
$$R_i = B \cdot \log_2 \left( 1 + \frac{(\|\mathbf{h}_{m_i}\|^2 \cdot p_i) / \sigma_i^2}{1 + \sum_{j \neq i} (|\mathbf{h}_{m_i} \cdot \mathbf{h}_{m_{ij}}|^2 \cdot p_j) / \sigma_i^2} \right), \quad (1)$$

where (i)  $\mathbf{h}_{m_{ij}} \in \mathbb{C}^N$  represents the channel association between the  $j$ -th user and the BS  $m_i$ , (ii)  $\sigma_i^2$  is the thermal power of the zero-mean circularly symmetrical complex Gaussian channel noise, and (iii)  $p_i$  and  $p_j$  are the transmitted powers from the  $i$ -th and  $j$ -th user to BS  $m_i$ , respectively. The operators  $\|\cdot\|$ ,  $|\cdot|$ ,  $(\cdot)^T$   $(\cdot)^\dagger$  denote the  $L^2$ -norm, absolute value, matrix transpose and the Hermitian transpose, respectively, while that typeset in bold (e.g.,  $\mathbf{h}$ ) expresses variables in matrix form. In view of (1), we formulate the most inclusive yet hardest-to-optimize expression between the available energy efficiency metrics, so-called the weighted sum energy efficiency (WSEE), as follows [23,24]:

$$WSEE_i = \frac{w_i \cdot R_i}{\mu_i \cdot p_i + P_{c,i}}, \quad (\text{in bit per Joule}) \quad (2)$$

where (i)  $\mu_i$  is the coefficient of the power amplifier, (ii)  $P_{c,i}$  is the total circuit power consumption of user  $i$  and its associated BS  $m_i$ , and (iii)  $w_i \geq 0$  is the priority weight of each  $(i, m)$  link between users and BSs, i.e., considering that some users may have the prerequisite for a higher data rate than others, with  $\mathbf{w} = \{w_1, \dots, w_i, \dots, w_j, \dots, w_L\}$ . More sophisticated modelings of the channel interference, as well as (later on) of the circuit power consumption are available in the relevant literature and can be considered to enrich the data rate expression in (1) and the energy efficiency expression in (2), respectively; yet, such insights are out of the scope of this work, which aims to describe the methodology of applying the new QNN approach on generic wireless communication system settings. Using (2) and by setting the transmitted power of each user as subject to an average transmitted power threshold, i.e.,  $p_i \leq P_i$  with  $\mathbf{p} = [p_1, \dots, p_i, \dots, p_j, \dots, p_L]$  and  $\mathbf{P} = [P_1, \dots, P_i, \dots, P_j, \dots, P_L]$ , we formulate the (genetic) WSEE optimization problem as follows:

$$\begin{aligned} & \max_{\mathbf{p}} \sum_{i=1}^L WSEE_i \\ & \text{subject to: } 0 \leq p_i \leq P_i, \forall i = 1, \dots, i, \dots, j, \dots, L. \end{aligned} \quad (3)$$



**Figure 1.** Illustration of the considered system modeling drawn on the uplink data transmissions of a multi-cell interference network consisting of multiple base stations and mobile users.

Problem (3) is a type of sum-of-ratios problem with non-deterministic polynomial-time hardness (NP-hard) in its computational complexity, meaning that it belongs to the hardest class of problems to resolve explicitly and integrate in practice. Traditionally, such complexities are tracked by transforming problems such as (3) into simpler forms, using variable relaxation methods [23,24], which, on the other hand, introduce considerable truncation errors that compromise the convergence and accuracy of their solutions. To bypass such issues, the recent studies of [1,2,21,25] approach NP-hard types of problems with ANN modeling to obtain their solutions by using the trained neural network. For the sake of clarity, in the following Section 3, we build such an ANN model by elaborating on the parameters of the considered WSEE system modeling.

### 3. Artificial Neural Network Architecture for the Energy Efficiency Problem

The aim is to resolve problem (3) by deriving the power allocations  $\mathbf{p}$  as closed-form functions of the propagation channels  $\mathbf{h}_{m_i}$ ,  $\mathbf{h}_{m_{ij}}$  and the power threshold parameters  $\mathbf{P}$  so that the computation of the optimal power allocation  $\mathbf{p}^*$  can have negligible complexity, i.e., to track the fading channel variations and update in real time. To do so, we compact the nominator and denominator of the data rate expression in (1) by defining the channel realization vectors  $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_i, \dots, \alpha_L]$  with  $\alpha_i = \|\mathbf{h}_{m_i}\|^2 / \sigma_i^2$ ,  $\beta_{ij} = |\mathbf{h}_{m_i} \cdot \mathbf{h}_{m_{ij}}|^2 / \sigma_i^2$ , and re-structure the WSEE problem (3) using the map as follows:

$$\mathcal{F} : \mathbf{a} = (\alpha_i, \beta_{ij}, P_i)_{ij} \in \mathbb{R}^{L \cdot (L+1)} \mapsto \mathbf{p}^* \in \mathbb{R}^L, \quad (4)$$

where  $\mathbf{p}^*$  is the vector with elements of the optimal power allocation coefficients of the  $L$  users corresponding to the  $M$  BS indexing, and  $\mathbb{R}$  is the set of real numbers. At this point, we recall the ANN universal approximation property, which yields the input–output relationship of a feed-forward neural network with fully-connected layers, capable of emulating any continuous function [25]. Indeed, as  $\mathbf{p} \geq 0$ , the WSEE function in (2) is continuous, which facilitates approximating the map in (4) with an ANN model of fully-connected feed-forward layers, i.e., an input layer based on the realization of the parameter vector  $\mathbf{a}$  and an output layer based on the power allocation vector  $\mathbf{p}$  that, in essence, stands as an estimate of the optimal power allocation vector  $\mathbf{p}^*$  corresponding to input  $\mathbf{a}$ . Note that due to feed-forward, not only are these two input–output layers created, but also  $K$  additional “hidden” neural layers are generated as well. To enumerate all the neural layers, we label the input layer as *layer-0*, the first “hidden” layer as *layer-1* with  $N_1$  neurons, the  $k$ -th “hidden” layer as *layer- $k$*  with  $N_k$  neurons, and so on, until the output *layer- $K + 1$* , which includes  $N_{K+1}$  neurons, as illustrated in Figure 2. In such a fully-connected ANN system, each layer  $k = 1, \dots, K + 1$  has  $N_k$  neurons in total, with each neuron  $n$  computing the following:

$$\zeta_k(n) = f_{nk}(\gamma_{nk}^T \cdot \zeta_{k-1} + \delta_{nk}), \quad (5)$$

where (i)  $\zeta_k = (\zeta_k(1), \dots, \zeta_k(N_{k+1}))$  represents the  $N_{k+1} \times 1$  output vector of layer  $k$ , (ii)  $\gamma_{nk} \in \mathbb{R}^{N_{k-1}}$  the neuron-dependent weights, (iii)  $\delta_{nk} \in \mathbb{R}$  the neuron-dependent bias terms, and (iv)  $f_{nk}$  is the activation function of neuron  $n$  in layer- $k$ . From (5), we see that although each neuron can perform simple operations, by combining the processing of multiple neurons in a single ANN, the system can perform complex tasks and obtain the overall input–output map in (4), which can approximate virtually any type of optimization function. The challenge, however, is how to configure the weights  $\{\gamma_{nk}\}$  and bias terms  $\{\delta_{nk}\}$  for achieving the desired approximation accuracy. This can be resolved by training the considered ANN system using a supervised procedure, where a certain training set of optimal power allocations can be deployed as the collection of channel realizations  $\{\mathbf{a}_n\}_{n=1}^{N_T}$  and the corresponding optimal power allocations  $\{\mathbf{p}_n^*\}_{n=1}^{N_T}$  of  $N_T$  training tuples  $(\mathbf{a}_n, \mathbf{p}_n^*)$ . In other words, the training set contains  $\{\mathbf{p}_n^*\}_{n=1}^{N_T}$  examples of desired power allocation vectors corresponding to some possible configurations of system parameters  $\mathbf{a}_n$ , whereas, by exploiting these examples, the system can learn to predict the power allocation for new realizations of  $\mathbf{a}_n$  that are not included in the running training set. More specifically, by setting  $\mathcal{L}$ , the error measure between the actual and desired outputs in (5), the ANN training problem can be written as follows:

$$\min_{\gamma, \delta} \frac{1}{N_T} \sum_{n=1}^{N_T} \mathcal{L}(\mathbf{p}_n(\gamma, \delta), \mathbf{p}_n^*). \quad (6)$$

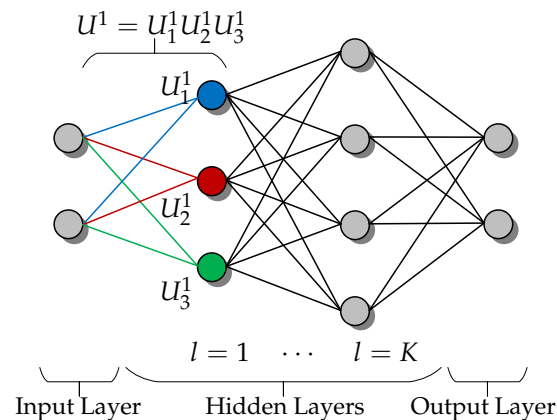
Problem (6) can be straightforwardly resolved by the recently-announced techniques based on the stochastic gradient descent (SGD) method [1,2,21,25]. That is, for generating the training set in (6), the generic WSEE maximization problem (3) should be reformulated by means of normalizing its maximum transmit power coefficients  $\{p_i\}$ , which requires transforming the variable  $p_i$  into  $p_i \rightarrow \tilde{p}_i \cdot P_i$  for all  $i = 1, \dots, L$  for constraining these power coefficients to lie within the closed interval  $[0, 1]$ . Using the variable transformation  $\tilde{p}_i \cdot P_i$ , the WSEE expression in (2) becomes  $\tilde{E}E_i(\tilde{p}_i) = \frac{\tilde{R}_i}{\mu_i \cdot P_i \cdot \tilde{p}_i + P_{c,i}}$  and the generic WSEE problem (3) can be rewritten into its equivalent form (i.e., suitable for applying SGD) as follows:

$$\max_{\tilde{\mathbf{p}}} \sum_{i=1}^L w_i \frac{B \cdot \log\left(1 + \frac{\tilde{\alpha}_i \tilde{p}_i}{1 + \sum_{j \neq i} \tilde{\beta}_{ij} \tilde{p}_j}\right)}{\mu_i \cdot P_i \cdot \tilde{p}_i + P_{c,i}} \quad (7)$$

subject to :  $0 \leq \tilde{p}_i \leq 1, \forall i = 1, 2, \dots, L,$

where  $\tilde{\alpha}_i = \alpha_i \cdot P_i$ ,  $\tilde{\beta}_{ij} = \beta_{ij} \cdot P_j$ , for all  $i = 1, \dots, L$ . In problem (7), the normalized ANN training set is  $S_T = \{(\tilde{\mathbf{a}}_n, \tilde{\mathbf{p}}_n^*) | n = 1, \dots, N_T\}$  with the parameter vector  $\tilde{\mathbf{a}} = (\tilde{\alpha}_i, \tilde{\beta}_{ij}, P_i)_{ij}$

to be able to be conveniently resolved, using the improved branch-and-bound method proposed in [21]. Note that the importance in reformulating the initial problem (3) into (7) is that for any value of the maximum power vectors  $P_1, \dots, P_L$ , the obtained optimal solution ensures that the transmit powers are lying within  $[0,1]$ . Consequently, problem (7) will have much weaker dependency between the optimal solution and the maximum power constraints, compared to the problem (3). Besides this, problem (7) gives a simpler structure of the optimal solution as a function of the input parameters  $\mathbf{a}$ , among which we find the maximum power constraints  $P_1, \dots, P_L$ , which simplifies the task of learning the optimal power allocation rule.



**Figure 2.** Illustration of the channel transition between two layers during the ANN training process. To attain a stable state (i.e., the optimal result) the ANN needs access to multiple layers (i.e., input/output/hidden layers) at any given time.

After the training phase, the resulting ANN input–output relationship provides an approximation of the optimal power allocation policy, which can be written in closed-form by considering the composition of the affine combinations and activation functions of all neurons. As shown in [1,21], such an approach provides accurate results with limited complexity, which makes NP-hard problem optimization (such as ANN-driven WSEE optimization) able to be applied in close-to-real-time, i.e., far faster and reliably than conventional relaxations discussed previously in Section 2.

#### 4. Deep Quantum Neural Network Architecture for the Energy Efficiency Problem

By exploiting the quantum advantage [19], it is anticipated that QNN will be able to solve the WSEE optimization problem (3) with comparable or better efficacy than its classical ANN counterpart. This is because upon realizing the ANN training process as a variational quantum circuit built on a continuous variable architecture, the training process can be enabled to encode quantum information in continuous degrees of freedom, i.e., by exploiting the amplitudes of the propagation channels  $\mathbf{h}_{m_i}$  and  $\mathbf{h}_{m_{ij}}$ . That means that the feed-forward neural network can switch its building blocks from classical to quantum neurons, which, in turn, enterprise the computations of the power allocations by considering not only the system parameters of their own  $k$  layer, but all the  $K + 1$  layers. Such a feature is known as universal quantum computation [26] and enables fast optimization with minimum memory demands because the number of the required qubits depends only on the width of the network at hand.

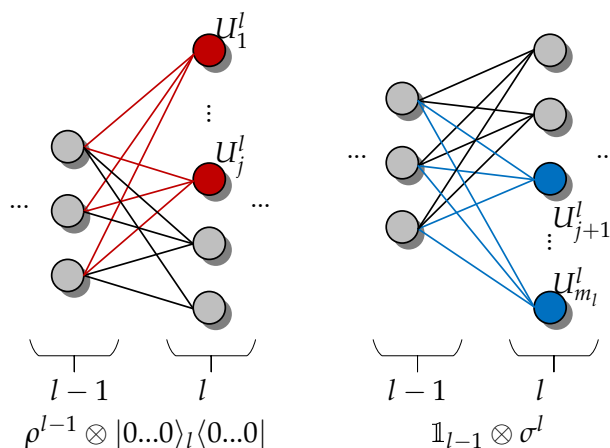
In this respect, we model the QNN training as a quantum circuit of multiple quantum perceptrons, where each perceptron is deemed to be a general unitary operator acting on  $m$  input qubits and  $n$  output qubits. That means that each perceptron can (i) be built using  $L$  hidden layers of qubits, (ii) act on an initial input state  $\rho^{in}$  of qubits, and (iii) produce an output state  $\rho^{out}$  of qubits that can be represented by the following:

$$\rho^{out} \equiv tr_{in,hidden} \left( \mathcal{U} \left( \rho^{in} \otimes |0 \dots 0\rangle_{out,hidden} \langle 0 \dots 0| \right) \mathcal{U}^\dagger \right), \tag{8}$$

where  $\mathcal{U} = U^{out} U^L U^{L-1} \dots U^1$  is the QNN quantum circuit and  $\mathcal{U}^\dagger$  is the layer-specific perceptron expressed as the unitary operator of the product of the qubits between adjusting layers, e.g., perceptron  $U^l$  reflects to the qubits between layer- $(l - 1)$  and layer- $l$ , with  $l = 1, \dots, L$ . Note that, as perceptrons are arbitrary unitary operators, they may not commute within the QNN, e.g., as shown in Figure 3, meaning that the order of their operations can be defined as the composition of a sequence of completely positive layer-to-layer  $\varepsilon^l$  transition maps, i.e., the following:

$$\rho^{out} = \varepsilon^{out} \left( \varepsilon^L \left( \dots \varepsilon^2 \left( \varepsilon^1 \left( \rho^{in} \right) \dots \right) \right) \right), \text{ with} \tag{9}$$

$$\varepsilon^l \left( X^{l-1} \right) \equiv tr_{l-1} \left( \prod_{j=m_l}^1 \mathcal{U}_j^l \left( X^{l-1} \otimes |0 \dots 0\rangle_l \langle 0 \dots 0| \right) \prod_{j=1}^{m_l} \mathcal{U}_j^{l\dagger} \right).$$



**Figure 3.** Illustration of the channel transition between two layers during the QNN training process. To attain a stable state (i.e., the optimal result) the QNN needs to only access two layers at any given time, which greatly reduces the memory requirements compared to ANNs, which require access to more than two (multiple) layers each time (recall Figure 2 for example).

For example,  $U_j^l$  in (9) is the  $j$ -th perceptron acting on previous layers  $l - 1$  and the current layer  $l$ , while  $m_l$  is the total number of perceptrons acting on layers  $l - 1$  and  $l$ . On this basis, the QNN mapping in (9) facilitates the information data to propagate from the input to the output, which naturally works as a feed-forward neural network; thus, it corresponds to the classical ANN mapping in (4) of the generic WSEE problem (3). More importantly, adopting the mapping in (9), the QNN training can be performed with similar algorithmic logic as that of the classical ANN system based on the mapping in (4), i.e., by following a quantum analogue of the so-called *back-propagation algorithm* [27]. Recall that back-propagation aims at facilitating the training phase to obtain the output density matrix  $\rho_x^{out}$  as closer to the desired output  $|\psi_x^{out}\rangle$ , which corresponds to the input  $|\psi_x^{in}\rangle$ . As such, we integrate back-propagation of (9) by setting the training function as follows:

$$C = \frac{1}{N} \cdot \sum_{x=1}^N \langle \psi_x^{out} | \rho_x^{out} | \psi_x^{out} \rangle, \tag{10}$$

which is known in the field of quantum information theory as *the fidelity function* [27]. Particularly, function  $C$  in (10) helps to (i) estimate the perceptrons as quantum states averaged over the training data, and (ii) quantify the difference between the  $\rho^{out}$  output and a certain (desired) output by varying between 0 (worst estimation) and 1 (best estimation). For example,  $C$  can be minimized by updating the perceptrons unitary operators in the QNN training (9), i.e.,  $U \rightarrow e^{ie\mathbf{K}}U$ , where  $\mathbf{K}$  is the matrix with all system parameters of the

corresponding perceptron unitary operator and  $\varepsilon$ , the step size of the SGD algorithm. In view of (10), we can define the QNN learning rate  $\eta$  and the update matrices  $\mathbf{K}_j^l$  for each perceptron  $j$  at layer  $l$  as follows:

$$\mathbf{K}_j^l = \eta \cdot \frac{2^{m_{l-1}}}{N} \cdot \sum_{x=1}^N \text{tr}_{rest} M_j^l, \quad \text{with} \quad (11)$$

$$M_j^l = \left[ \prod_{i=j}^1 \mathcal{U}_i^l (\rho_x^{l-1,l}) \prod_{i=1}^j \mathcal{U}_i^{l\dagger}, \prod_{i=j+1}^{m_l} \mathcal{U}_i^{l\dagger} (\mathbb{I}_{l-1} \otimes \sigma_x^l) \prod_{i=m_l}^{j+1} \mathcal{U}_i^l \right],$$

$$\rho_x^{l-1,l} = \rho_x^{l-1} \otimes |0 \dots 0\rangle_l \langle 0 \dots 0|,$$

$$\sigma_x^l = \mathcal{F}^{l+1} (\dots \mathcal{F}^{out} (|\psi_x^{out}\rangle \langle \psi_x^{out}|) \dots)$$

and  $\mathcal{F}^l$  is the join channel to  $\varepsilon^l$ , which is the transition channel from layer  $l + 1$  to layer  $l$ , e.g., as illustrated in Figure 3.

Relying on (10), (11) and the feed-forward rationale of the SGD algorithm, we present in Algorithm 1 the pseudo-code to structurize the QNN deep-learning that derives the optimal power allocations of the WSEE problem (3). Notice the key contribution of Algorithm 1 that computes the trace over all qubits (i.e., in line 9) independently from  $\mathcal{U}_j^l$ , which gives rise to two important features for QNN operation. First, recalling that one of the merits of SGD rationale is that the parameter matrices  $K$  can be computed layer-by-layer without any need to rely on additional unitary operators of the full quantum system [14], Algorithm 1 has a smaller step size  $\varepsilon$  than the ANN training in (6), which speeds up QNN convergence toward deriving the optimal WSEE power allocations. Second, by coordinating (10) with SGD, the computation of (9) at each step of Algorithm 1 involves only two layers, meaning that the size of the  $K$  matrices in the calculation of (10) depends only on the width of the network, rather than the width of each layer as required by the respective ANN training function in (5). This latter feature results in considerable memory savings and makes the proposed QNN deep-learning process highly practicable to apply in a realistic system setting and scaling. To demonstrate the practicability of our new QNN approach, the following Section 5 evaluates Algorithm 1 using benchmarks and simulations.

---

**Algorithm 1:** Pseudo-code of the proposed quantum deep-learning process.

---

- 1 **Initialization:**
  - 2 Initialize Unitary  $\mathcal{U}_j^l$  randomly  $\forall j, l$
  - 3 **Feed-forward:**
  - 4 For each batch of  $(|\psi_x^{in}\rangle, |\psi_x^{out}\rangle), \forall x \in \text{Batchsize}, l \in L$
  - 5 (i) Apply the channel  $\varepsilon^l$  to the output state of layer  $l - 1$
  - 6 (ii) Tensor  $\rho_x^{l-1}$  with layer  $l$  in state  $|0 \dots 0\rangle_l$  and apply  $\mathcal{U}^l = \mathcal{U}_{m_l}^l \dots \mathcal{U}_1^l$ :
  - 7 (iii) Trace out layer  $l - 1$  and store  $\rho_x^l$
  - 8 **Update Network:**
  - 9 (i) Compute the parameter matrices  $K_j^l$
  - 10 (ii) Update each unitary  $\mathcal{U}_j^l$  in which  $\mathcal{U} \rightarrow e^{i\varepsilon K} \mathcal{U}$
  - 11 **Repeat:**
  - 12 Go to step-3 until the training function reaches its maximum.
- 

## 5. Numerical Evaluations

Before evaluating our findings, we recall that it is practicably impossible to simulate deep QNN learning algorithms using classical computers for more than a handful of qubits, due to the exponential growth of Hilbert space [18–20]. Therefore, our evaluations will follow similar rationalism with the evaluations of the relevant studies in [13–17], where the QNN performance is examined over a series of benchmarks to evaluate the size and



number of the QNN training steps, e.g., if the algorithms can attain a stable state (i.e., derive the optimal result) in fast time (i.e., in as few as possible and short training steps).

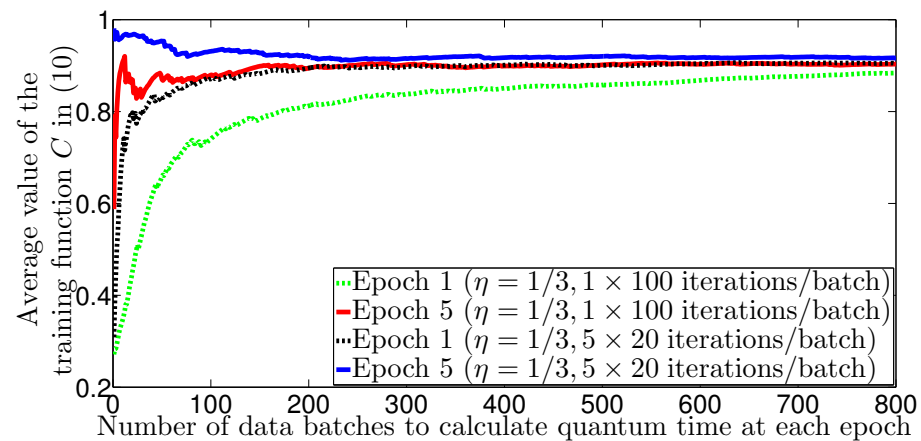
We also recall that a striking feature of Algorithm 1 is that the parameter matrices can be calculated layer-by-layer without ever having to apply the unitary corresponding to the full quantum circuit on all the constituent qubits of the QNN in one go. In other words, we need only to access two layers at any given time (as shown in Figure 3), which greatly reduces the memory requirements of Algorithm 1 compared to ANNs, where access to more than two layers are required (as shown in Figure 2) with the respective increase in memory requirements. Hence, the size of the matrices in our calculation only scale with the width of the network, enabling us to train deep QNNs in a considerably less resource-costly and faster way than ANNs.

As such, to simulate the QNN deep-learning process in (11) toward optimizing the WSEE power allocation problem (3), we use a classical desktop computer with typical processing capacity (i.e., Intel i7 with 16 GB of RAM), which, in principle, is not designed to handle more than a handful of qubits (as stated previously). That means that, with mainstream equipment, we can execute Algorithm 1 for a four-layer QNN system of 16 neurons in total (i.e., 4 neurons for the input layer, 8 neurons for the two hidden layers and 4 neurons for the output layer), while the addition of more neurons is subject to future quantum-based equipment. Yet, even under conventional equipment, we demonstrate that such a four-layer QNN system can attain much higher performance than ANN systems of a bigger size, i.e., consisting of seven layers and 28 neurons in total. In addition, for fair simulation comparisons, we deliberate over a realistic system setting similar to [21], which provides a generic Matlab coding of the so-called globally optimal energy efficient power control framework in interference networks.

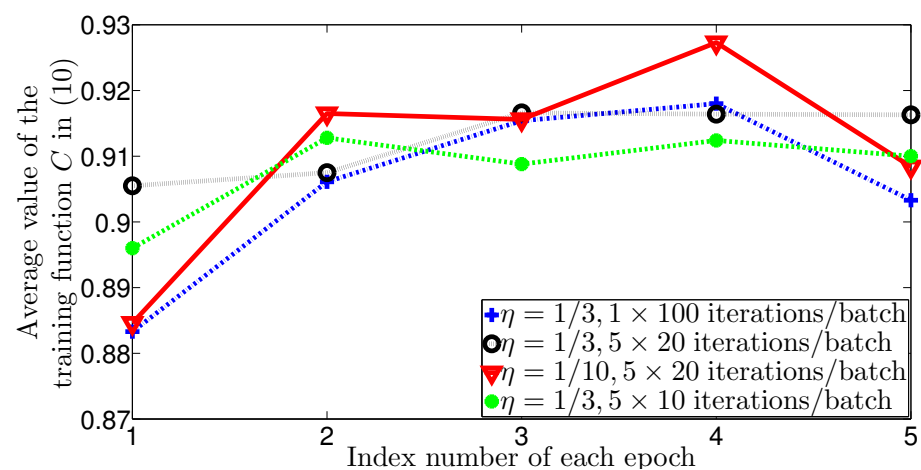
On this basis, we translate the codes provided by [14] (available online at [28]) to Python coding for producing a *training module* composed of 102,000 training samples for  $P_{max} = -30, \dots, -20$  dB in 1 dB step and 10,200 validation samples, i.e., 10% of the overall training samples. In the produced training module, we split the total number of iterations  $N_T$  into  $R$  rounds so that the parameter matrices  $\mathbf{K}_j^l$  in (11) can be re-initialized at the beginning of each feed-forward round of execution. Such matrix re-initialization ensures that the QNN training function  $C$  in (10) can increase with a smaller step size (i.e., more rapidly) than the ANN training function  $\mathcal{L}(\mathbf{p}_n(\gamma, \delta), \mathbf{p}_n^*)$  in (6) because each matrix  $\mathbf{K}_j^l$  includes all the necessary unitary operators of the full quantum system, rather than additional unitary operators corresponding to each perceptron as done in ANN. Having defined the rationale of our system setting, we proceed by updating the perceptron unitary, using  $\mathbf{U} \rightarrow e^{i\epsilon\mathbf{K}} \cdot \mathbf{U}$  and performing a random re-initialization and reshuffling with each batch size of 128 samples of data. We arrange [5,4,3,2] neurons over the considered four-layer QNN architecture, which consequences to 5 input perceptrons with 32 states and 2 output perceptrons with 4 states. In addition, we arrange the original information data over 17 sets of input data, and set zeros for the rest of the input state so that there is no need to add offset data for the output perceptrons, as the original output set is exactly equal to 4. Notice that this evaluation architecture is equivalent to the four-layer classical training process architecture with [32,16,8,4] neurons, which, without loss of generality, converges in  $5 \times 800 \times 100 = 400,000$  iterations.

In Figure 4 we investigate the role of the parameter matrices  $\mathbf{K}_j^l$  in the QNN training process. Particularly, we set  $N_T = 100$ , learning rate  $\eta = 1/3$ , step size  $\epsilon = 0.1$  and 100 iterations to study the effect of  $\mathbf{K}_j^l$  on the value of the training function when it is initialized once and multiple times per step. We observe that when  $\mathbf{K}_j^l$  is initialized, Algorithm 1 can always produce lower cost values than those with 5 re-initializations across the number of batches, either for the first epoch or the fifth epoch. We also observe that by initializing multiple matrices  $\mathbf{K}_j^l$ , Algorithm 1 can attain its maximum cost value much faster than by initializing a single matrix for the whole training session. As result, the training function converges toward a steady state (optimal power allocation result) after 300 batches of data for multiple re-initializations.

In Figure 5, we examine the impact of the learning rate for the wireless communications data set. From the figure we see that the learning rate  $\eta = 1/10$  is too small to get stable unitary matrices  $\mathbf{U}$  compared to  $\eta = 1/3$  after 3 training epochs. We also see that by re-initializing multiple  $K$  matrices for 50 and 100 iterations within the training module, we can attain stable results (i.e., the WSEE optimal powers) after 3 epochs only (considerably fast). However, we notice that with a smaller learning rate, there are chances to attain higher cost value at the expense of increasing the epoch time, which may consume a longer total training time.



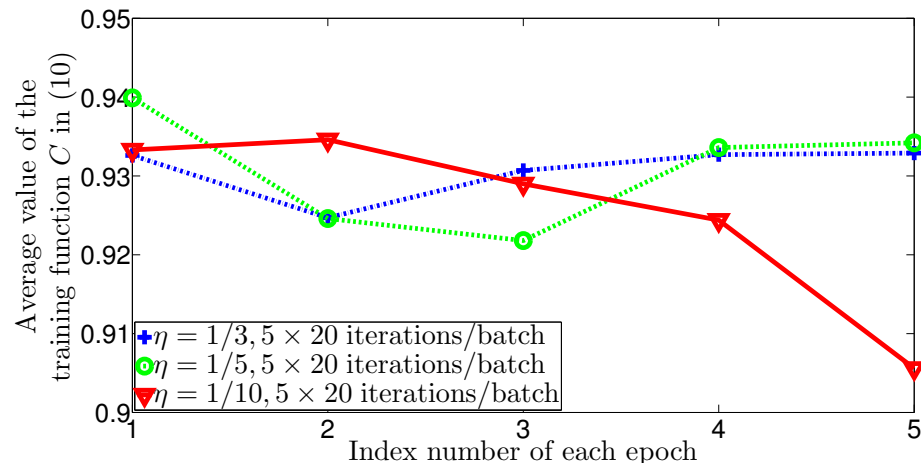
**Figure 4.** Average value of the training function versus the number of data batches to calculate the quantum time for each epoch. We show that the proposed training function in (10) can indeed converge toward a steady state (optimal power allocation result) after 300 batches of data re-initializations.



**Figure 5.** Average value of the training function versus the index number of each epoch. We show that Algorithm 1 can reach the stable results (i.e., the WSEE optimal powers) considerably fast (after 3 epochs only). We also show that Algorithm 1 becomes more accurate by increasing the learning rate from  $\eta = 1/10$  to  $\eta = 1/3$ , which is expected and matches with the relevant algorithms' behavior in [13–17].

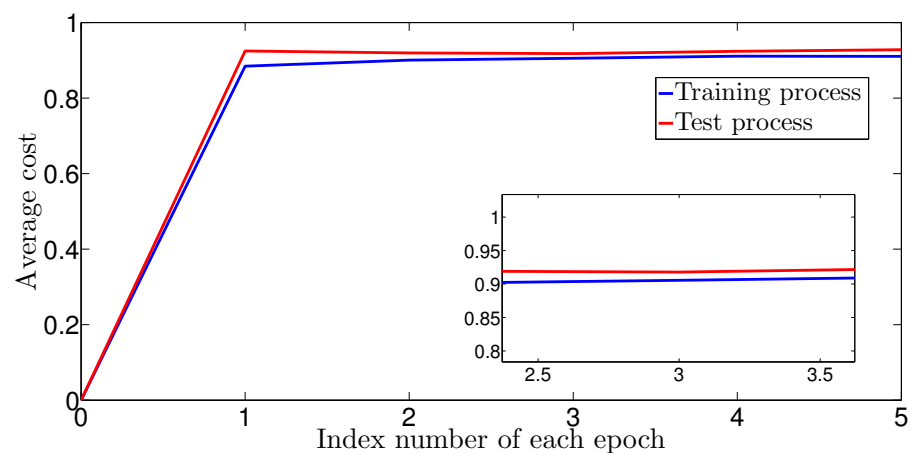
In Figure 6, we extend the globally optimal power control framework of [14] by adding a training perceptron unitary operator to test the reliability of the proposed QNN architecture. For this experiment, we use 10,200 samples of new data for validation to observe that the trained perceptron unitary operator with learning rate of either  $\eta = 1/3$  or  $\eta = 1/5$  can provide better and more stable cost values after 4 epochs. We also notice from Figure 6 that the learning rate  $\eta = 1/10$  is too small for Algorithm 1 to reach a stable state in a short time, which matches with our previous observations in Figure 5 and implies that

there is an optimal value of  $\eta$  to be chosen during the training process so that Algorithm 1 can reach its actual minimum time of convergence. Nevertheless, even with small  $\eta$ , the results of Figure 6 show that the training cost value and the validation cost value are almost comparable, which confirms the reliability of the proposed QNN architecture.

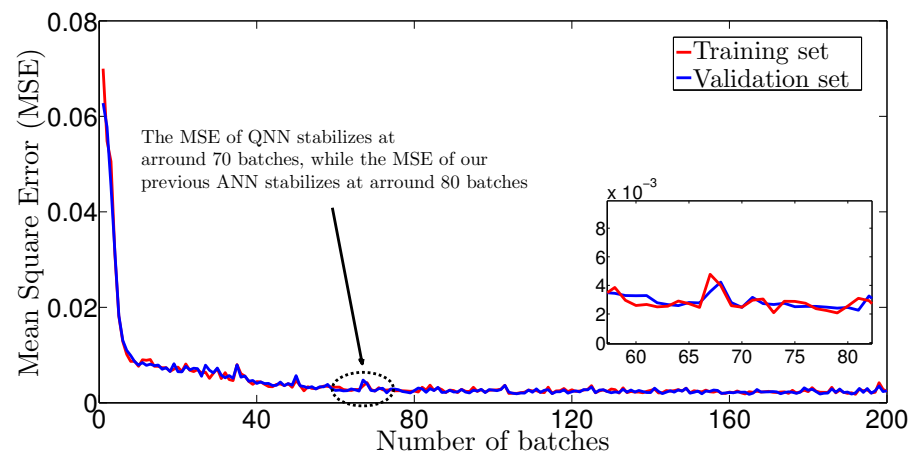


**Figure 6.** Average value of the training function versus the index number of each epoch, using the training perceptron unitary operator. We show that besides optimizing the WSEE problem, Algorithm 1 can be also generalized to optimize the power control problem in [14]—although the two problems differ radically, Algorithm 1 can resolve both of them reliably and robustly, i.e., by requiring a similar training step size of  $\eta = 1/3$  and comparable implementation time between 3 and 4 epochs.

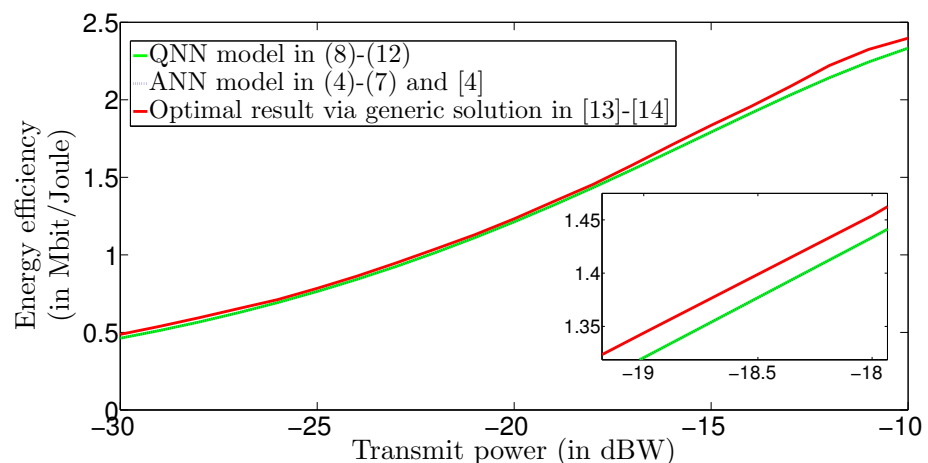
In Figures 7–9, we examine the optimization accuracy and implementation time of the proposed QNN model in (8)–(11) in comparison to the ANN model in (4)–(7). For these experiments, we generated a training set from 2000 independent and identically distributed (i.i.d.) realizations of users' positions and propagation channels. We considered each user to be randomly placed in the service area and that each user  $i$  is associated to the BS  $m$  with the strongest effective channel  $\{h_{m_i}\}$ . For each channel realization, we apply Algorithm 1 to solve the WSEE problem (3) for  $P_i = -30, \dots, 20$  dB in 1 dB step, which yields a training set of 102,000 samples. Note that for the comparison between QNN and ANN, besides the training set, we also need to define a validation set, which is commonly used during training to estimate the generalization performance of the QNN, i.e., the performance on a data set that the QNN was not trained on [19].



**Figure 7.** Average cost versus the number of batches. The average cost measures the quantum fidelity, e.g., the QNN output and the desired output (the closed is better). We show that the training and validation sets quickly approach each other with a very small value of the same order of magnitude, e.g., in 3 epochs.



**Figure 8.** Mean square error versus the number of epochs. We show that by feeding 800 batches of training data, our QNN model requires 70 epochs to stabilize its MSE—we recall that by using the same training set of 800 batches, our ANN model in [21] requires 70 epochs to stabilize its MSE. Consequently, the performance of the QNN model is comparable with the performance of the ANN model for resolving the WSEE problem (3).



**Figure 9.** Energy efficiency versus transmit power. We show that the QNN model in (8)–(11) derives similar energy efficiency results to the ANN model in (4)–(7) and [21], while both QNN and ANN training models deviate at approximately 0.05 Mbit/Joule, on average, from the actual optimal WSEE solution given in [23,24].

Having defined these training and validation sets, we plot in Figure 7 the average cost versus the number of epochs and in Figure 8 the mean square error (MSE) versus the number of epochs needed for WSEE optimization. From Figure 7, we observe that both the errors of the training and validation sets quickly approach a very small value of the same order of magnitude (in about 3 epochs), and that 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. Furthermore, we feed our QNN with 800 batches of training data to observe in Figure 8 that it needs approximately 70 epochs to stabilize its MSE (i.e., to reach the optimal WSEE result)—instead, in our previous study ([21], Figure 2, p. 10), we feed with 800 batches of training data a classical ANN model to show that it can resolve the WSEE problem in 80 epochs. These results confirm that, given the same training data, the implementation speed of our QNN model is comparable and slightly faster than the implementation speed of the classical ANN model to resolve the same WSEE problem with similar optimization accuracy.

Finally, Figure 9 illustrates a comparison regarding the accuracy and actual optimization result between our QNN model and our classical ANN model [21]. Particularly, in

Figure 9, we plot the energy efficiency (in bit/Joule) versus the transmitted power (in dBW) by following the setting mentioned previously and by taking as the evaluation metric the generic WSEE optimization method introduced in [23,24]. We observe that within the region of  $-30 \leq P_i \leq -18$  dBW, both QNN and ANN derive almost the same results as the actual optimal solution method in [23,24], while for the power region  $-18 \leq P_i \leq -10$  dBW, both the QNN and ANN training models deviate by approximately 0.05 Mbit/Joule on average from the actual WSEE solution. Nevertheless, the main point is that our QNN is proved to be as accurate as the ANN model. Therefore, by recalling our previous findings in Figures 7 and 8, where we show that our feedforward QNN has slightly better trainability than its ANN analogue, we can conclude that there is potential for QNNs to perform faster training when applied on similar to WSEE problems, without showcasing overfit or underfit issues and by using classical computer hardware.

## 6. Conclusions

This paper studied the perspective of adapting quantum learning and training logic in wireless communication networks by proposing a new quantum neural network architecture for resolving the power control in energy efficiency optimization problems. We formulated the generic energy efficiency problem and approached it, using both classical and quantum neural network models. We conducted the design of the learning and training analysis for both these models to clarify their rationalism, with architectural differences and similarities. Furthermore, we proposed Algorithm 1 as the tangible mean of applying our QNN model, and conducted a series of benchmarks to evaluate its size and the number of training steps. We performed simulations to show that our algorithm can attain a stable state (i.e., derive the optimal result) in fast time (i.e., short training step), and that it can be generalized to other, more complicated problems, such as the power control problem in [21], with comparable implementation time and accuracy. We also showed through simulations that our QNN model can attain the same accuracy with its ANN analogue by featuring slightly faster trainability for this specific energy efficiency problem. Based on these results, we believe that the perspective of quantum-oriented learning and training can play a significant role in the next-generation communications era in terms of decentralizing intelligence and saving computing and memory resources. The fact that our QNN model can run using classical computer hardware in a comparable manner to its ANN analogue model provides the incentive for future studies to clarify further links between quantum and communications engineering and improve QNNs to address other communications problems.

**Author Contributions:** Conceptualization, S.F.C.; methodology, S.F.C.; software, S.F.C. and H.S.L.; validation, M.A.K., Q.N. and A.Z.; formal analysis, S.F.C., H.S.L. and C.C.Z.; investigation, S.F.C., H.S.L. and C.C.Z.; resources, M.A.K., Q.N. and A.Z.; data curation, S.F.C. and H.S.L.; writing—original draft preparation, S.F.C. and C.C.Z.; writing—review and editing, C.C.Z., H.S.L. and S.F.C.; visualization, S.F.C., H.S.L. and C.C.Z.; All authors have read and agreed to the published version of the manuscript.

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

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The authors would like to thank Robert Salzmann at the Center for Quantum Information and Foundations, University of Cambridge, for his technical assistance.

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

## References

1. Zappone, A.; Renzo, M.D.; Debbah, M. Wireless networks design in the era of deep learning: Model-based, AI-based, or both? *IEEE Trans. Commun.* **2019**, *67*, 7331–7376. [[CrossRef](#)]
2. Zappone, A.; Renzo, M.D.; Debbah, M.; Lam, T.T.; Qian, X. Model-aided wireless artificial intelligence: Embedding expert knowledge in deep neural networks towards wireless systems optimization. *IEEE Vehic. Technol. Mag.* **2019**, *14*, 60–69. [[CrossRef](#)]

3. Aïmeur, E.; Brassard, G.; Gambs, S. Machine Learning in a Quantum World. In *Advances in Artificial Intelligence*; Lamontagne, L., Marchand, M., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4013.
4. Marquardt, F. Machine learning and quantum devices. *SciPost Phys. Lect. Notes* **2021**, *29*, 21.
5. Tiersch, M.; Ganahl, E.J.; Briegel, H.J. Adaptive quantum computation in changing environments using projective simulation. *Sci. Rep.* **2015**, *5*, 12874. [[CrossRef](#)] [[PubMed](#)]
6. Lovett, N.B.; Crosnier, C.; Perarnau-Llobet, M.; Sanders, B.C. Differential evolution for many-particle adaptive quantum metrology. *Phys. Rev. Lett.* **2013**, *110*, 220501. [[CrossRef](#)] [[PubMed](#)]
7. Banchi, L.; Grant, E.; Rocchetto, A.; Severini, S. Modelling non-Markovian quantum processes with recurrent neural networks. *New J. Phys.* **2018**, *20*, 123030. [[CrossRef](#)]
8. Porotti, R.; Tamascelli, D.; Restelli, M. Coherent transport of quantum states by deep reinforcement learning. *Commun. Phys.* **2019**, *2*, 61. [[CrossRef](#)]
9. Aïmeur, E.; Brassard, G.; Gambs, S. Quantum speed-up for unsupervised learning. *Mach. Learn.* **2013**, *90*, 261. [[CrossRef](#)]
10. Paparo, G.D.; Dunjko, V.; Makmal, A.; Martin-Delgado, M.A.; Briegel, H.J. Quantum speedup for active learning agents. *Phys. Rev. X* **2014**, *4*, 031002. [[CrossRef](#)]
11. Amin, M.H.; Andriyash, E.; Rolfe, J.; Kulchytskyy, B.; Melko, R. Quantum Boltzmann machine. *Phys. Rev. X* **2018**, *8*, 021050. [[CrossRef](#)]
12. Sentís, G.; Monràs, A.; Muñoz-Tapia, R.; Calsamiglia, J.; Bagan, E. Unsupervised classification of quantum data. *Phys. Rev. X* **2019**, *9*, 041029. [[CrossRef](#)]
13. Allcock, J.; Hsieh, C.; Kerenidis, I.; Zhang, S. Quantum Algorithms for Feed-Forward Neural Networks. *arXiv* **2019**, arXiv:1812.03089.
14. Beer, K.; Boundarenko, D.; Farrelly, T.; Osbotne, T.J.; Salzmann, R.; Wolf, R. Efficient Learning for Deep Quantum Neural Networks. *arXiv* **2019**, arXiv:1902.10445.
15. Gyongyosi, L.; Imre, S. Training Optimization for Gate-Model Quantum Neural Networks. *J. Nat. Res.* **2019**, *9*, 12679. [[CrossRef](#)]
16. Choudhury, S.; Dutta, A.; Ray, D. Chaos and Complexity from Quantum Neural Network: A study with Diffusion Metric in Machine Learning. *arXiv* **2021**, arXiv:2011.07145.
17. Bhattacharyya, A.; Chemissany, W.; Haque, S.S.; Murugan, J.; Yan, B. The Multi-faceted Inverted Harmonic Oscillator: Chaos and Complexity. *arXiv* **2021**, arXiv:2007.01232.
18. Lloyd, S.; Mohseni, M.; Rebentrost, P. Quantum algorithms for supervised and unsupervised machine learning. *arXiv* **2013**, arXiv:1307.0411.
19. Abbas, A.; Sutter, D.; Zoufal, C.; Lucchi, A.; Figalli, A.; Woerner, S. The power of quantum neural networks. *arXiv* **2020**, arXiv:2011.00027.
20. Zhang, Y.; Ni, Q. Recent Advances in Quantum Machine Learning. *Wiley J. Quantum Eng.* **2020**, *2*, 1–20. [[CrossRef](#)]
21. Matthiesen, B.; Zappone, A.; Besser, K.L.; Jorswieck, E.A.; Debbah, M. A Globally Optimal Energy-Efficient Power Control Framework and its Efficient Implementation in Wireless Interference Networks. *IEEE Trans. Signal Process.* **2020**, *68*, 3887–3902. [[CrossRef](#)]
22. Lee, W.; Kim, M.; Cho, D. Deep Power Control: Transmit Power Control Scheme Based on Convolutional Neural Network. *IEEE Commun. Lett.* **2018**, *22*, 1276–1279. [[CrossRef](#)]
23. Zarakovitis, C.C.; Ni, Q.; Spiliotis, J. New Energy Efficiency Metric With Imperfect Channel Considerations for OFDMA Systems. *IEEE Wirel. Commun. Lett.* **2014**, *3*, 473–476. [[CrossRef](#)]
24. Zarakovitis, C.C.; Ni, Q.; Spiliotis, J. Energy-Efficient Green Wireless Communication Systems with Imperfect CSI and Data Outage. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 3108–3126. [[CrossRef](#)]
25. Leshno, M.; Lin, V.Y.; Schocken, S. Multilayer Feed-Forward Networks with a Non-polynomial Activation Function can Approximate any Function. *Neural Netw.* **1993**, *6*, 861–867. [[CrossRef](#)]
26. Beer, K.; Bondarenko, D.; Farrelly, T. Training deep quantum neural networks. *Nat. Commun.* **2020**, *11*, 808. [[CrossRef](#)]
27. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information*; Cambridge Univ. Press: Cambridge, UK, 2010.
28. Generic Matlab Coding for Simulating Quantum Deep Learning Processes. Available online: <https://github.com/R8monaW/DeepQNN> (accessed on 23 February 2021).

## Short Biography of Authors



**Chien Su Fong** is a principal researcher at MIMOS Berhad. He received the B.Sc. and M.Sc. degrees from the University of Malaya, Malaysia, in 1995 and 1998, respectively, and the Ph.D. degree from Multimedia University, Malaysia, in 2002. He has been serving as a TPC member or a reviewer for ICACCI, ICP, SETCAC, ISCIT, GNDS, Ad Hoc Networks, and CNCT. He has published several of tens of conference and refereed journal papers and holds few patents. His current research interests include green communications, optimization, applications of bio-inspired algorithm, machine learning, neural networks, and quantum computing algorithms. He is one of the editors-in-chief of Bio-Inspired Computation in Telecommunications.



**Heng Siong Lim** received the BEng (Hons) degree in electrical engineering from Universiti Teknologi Malaysia in 1999, and the MEngSc and PhD degrees from Multimedia University in 2002 and 2008, respectively, where he is currently an Associate Professor with the Faculty of Engineering and Technology. His research interests include signal processing and receiver design for wireless communications.



**Michail Alexandros Kourtis** received his PhD from the University of the Basque Country (UPV/EHU) in 2018 and from 2012 is a Research Associate at NCSR “Demokritos” working on various H2020 research projects. His research interests include Network Function Virtualization, 5G and Network Slicing.



**Qiang Ni** received the B.Sc., M.Sc., and Ph.D. degrees from the Huazhong University of Science and Technology (HUST), China, all in engineering. He is currently a Professor and the Head of the Communication Systems Group with InfoLab21, School of Computing and Communications, Lancaster University, Lancaster, U.K. He has published over 150 papers. His main research interests lie in the area of future generation communications and networking, including green communications and networking, cognitive radios, heterogeneous networks, 5G, energy harvesting, IoT, and vehicular networks. He was an IEEE 802.11 Wireless Standard Working Group Voting Member and a Contributor to the IEEE Wireless Standards.



**Alessio Zappone** obtained his Ph.D. degree in electrical engineering in 2011 from the University of Cassino and Southern Lazio, Cassino, Italy. Afterwards, he has been with TU Dresden, Germany, from 2012 to 2016. From 2017 to 2019 he was the recipient of an Individual Marie Curie fellowship for experienced researchers, carried out at CentraleSupélec, Paris, France. He is now a tenured professor at the university of Cassino and Southern Lazio, Italy. He is an IEEE senior member, serves as senior area editor for the IEEE Signal Processing Letters, and served twice as guest editor for the IEEE Journal on Selected Areas on Communications. He chairs the RIS special interest group REFLECTIONS, activated within the Signal Processing for Computing and Communications Technical Committee.



**Charilaos C. Zarakovitis** received the BSc degree from the Technical University of Crete, Greece, in 2003, the M.Sc and G.C.Eng degrees from the Dublin Institute of Technology, Ireland, in 2004 and 2005, and the M.Phil and Ph.D degrees from Brunel University, London, UK, in 2006 and 2012, respectively, all in electronic engineering. In Academia, he has worked as Senior Researcher at (i) Infolab21, Lancaster University UK, (ii) 5GIC, University of Surrey UK, and (iii) MNLab, National Centre for Scientific Research “DEMOKRITOS” Greece. In Industry, he has worked as Chief R&D Engineer at AXON LOGIC Greece, and as R&D Engineer at MOTOROLA UK, INTRACOM S.A Greece, and NOKIA-SIEMENS-NETWORKS UK. His research interests include Deep Learning, Quantum Neural Networks, green communications, bioinspired and game-theoretic decision-making systems, network optimization, cognitive radios, Ad-Hoc Clouds, among others.