electronics

MDPI

Article

A Deep Learning Based Transmission Algorithm for Mobile Device-to-Device Networks

Tae-Won Ban [†,‡] **and Woongsup Lee** [‡,*]

Department of Information and Communication Engineering, Gyeongsang National University, Gyeongnam 53064, Korea; twban35@gnu.ac.kr

* Correspondence: wslee@gnu.ac.kr

† Current address: Marine Science Bldg. 807, Tongyeong, Gyeongnam 53064, Korea.

‡ These authors contributed equally to this work.

Received: 31 October 2019; Accepted: 14 November 2019; Published: 16 November 2019

Abstract: Recently, device-to-device (D2D) communications have been attracting substantial attention because they can greatly improve coverage, spectral efficiency, and energy efficiency, compared to conventional cellular communications. They are also indispensable for the mobile caching network, which is an emerging technology for next-generation mobile networks. We investigate a cellular overlay D2D network where a dedicated radio resource is allocated for D2D communications to remove cross-interference with cellular communications and all D2D devices share the dedicated radio resource to improve the spectral efficiency. More specifically, we study a problem of radio resource management for D2D networks, which is one of the most challenging problems in D2D networks, and we also propose a new transmission algorithm for D2D networks based on deep learning with a convolutional neural network (CNN). A CNN is formulated to yield a binary vector indicating whether to allow each D2D pair to transmit data. In order to train the CNN and verify the trained CNN, we obtain data samples from a suboptimal algorithm. Our numerical results show that the accuracies of the proposed deep learning based transmission algorithm reach about 85%∼95% in spite of its simple structure due to the limitation in computing power.

Keywords: D2D; mobile caching; CNN; deep learning; transmission algorithm

1. Introduction

According to a recent study, it was predicted that the total amount of Internet traffic will increase threefold over five years from 2017 to 2022 and that mobile Internet traffic will increase sevenfold for the same time period [1]. More specifically, video traffic will be growing more steeply than other types of traffic. Video traffic, which accounted for about 75% of total Internet traffic in 2017, will account for about 82% in 2022. Another interesting study showed that the most popular 50 videos account for almost 80% of the total amount of views for YouTube [2]. Thus, the mobile caching network has been attracting much attention as a new approach to cope effectively with the explosively growing mobile Internet traffic [3–7].

A probabilistic caching scheme with a low complexity for minimizing the caching failure probability was proposed [3]. It was shown that the density of successful reception can be maximized by optimally placing files on caching servers according to varying channel conditions [4]. The concept of collaboration in mobile caching was proposed in [5,6]. The collaboration distance was optimized in [5], and a tradeoff between collaboration distance and interference was investigated in [6]. A joint non-convex problem for resource scheduling and power allocation in a wireless caching network was formulated, and an algorithm was designed based on two decomposed convex problems [7]. Contrary to conventional approaches such as multiple antennas and heterogeneous networks to improve the spectral efficiency of cellular networks, the mobile caching network can dramatically reduce the traffic load, especially for core and backhaul networks, and is based on D2D communication. D2D communication can shorten the distance between transmitters and receivers compared to conventional cellular communications. The shortened distance between end points can reduce the end-to-end latency and power consumption and can enhance data rates. Motivated by these potentials, many previous studies have investigated mobile D2D communications [8–22].

In this paper, we also investigate the problem of radio resource management for D2D networks, which is one of most challenging problems. We consider a cellular overlay D2D network where there is no mutual interference between D2D links and cellular links. To improve the spectral efficiency, all D2D devices are allowed to share given radio spectra. Thus, we need to choose an optimal set of D2D links to transmit data considering the interference among D2D devices. Most of the previous studies adopted heuristic or mathematical approaches to propose algorithms for resource management for D2D networks, which continuously cause complexities that are repeated every scheduling decision, and the total complexities will be thus tremendous if accumulated. We propose a new algorithm based on deep learning, which has been widely used in various fields due to its potential. Especially, many studies have demonstrated that deep learning can be successfully exploited in the communication field [23–28]. We formulate a CNN to design an algorithm that chooses D2D links to transmit data. Even though the CNN based algorithm might result in unignorable complexities in the learning process, they are only caused during the learning process, and no more complexities are thus required in the scheduling process. In addition, the complexity of the supervised deep learning is bounded and predictable compared to deep reinforcement learning (DRL). We also design a sub-optimal scheme to obtain data samples required to train the designed CNN and verify the trained CNN. Ninety percent of the obtained data are used to train the designed CNN, and the remaining 10% is used to verify the trained CNN. An early terminating learning scheme with an adaptive learning rate is used to avoid over-fitting of the CNN with limited computing power. We analyze the performance of the CNN in terms of accuracy and average sum rate and compare them with those of the sub-optimal scheme. Our numerical results confirm that the CNN has accuracies of 85%∼95%, which indicates that the CNN can yield 85%∼95% identical scheduling results to the sub-optimal scheme.

The rest of this paper is organized as follows. A detailed discussion of relevant studies about D2D communications is provided in Section 2. In Section 3, our D2D network model and wireless channel model are described. In Section 4, a sub-optimal scheme to obtain data samples to train and verify our neural network is described and a deep learning based scheduling scheme is also proposed. The performance of the proposed scheme is analyzed in terms of accuracy and average sum rate in Section 5. Finally, the conclusions of this paper are drawn in Section 6.

2. Related Work

Interesting algorithms to select optimal communication modes for D2D devices were studied for cellular aided D2D networks [8–10]. It was assumed in [8] that mobile devices can select a communication mode among a mode using dedicated resources, a mode reusing cellular resources, and a conventional cellular mode, while a D2D mode using dedicated resources was not considered in [9]. An adaptive mode selection of potential D2D devices was formulated as a follower evolutionary game, and an evolutionary stable strategy was considered to be the solution [10]. The authors investigated how to manage or mitigate the cross-interference between cellular and D2D communications in cellular underlay D2D networks [11–13]. In cellular underlay D2D networks, D2D devices share the same radio resource with cellular mobile devices. The authors of [11] showed that D2D devices can avoid the harmful interference from cellular networks if they decode signaling messages broadcast by cellular base stations (BSs) and take advantage of the information for radio resource management embedded in the decoded signaling messages. A hybrid mechanism based on fractional frequency reuse (FFR) scheme and an almost blank sub-frame (ABS) scheme was proposed to reduce the interference caused by cellular networks to D2D networks [12], which is expected to be very effective especially in cell edge areas. In [13], it was shown that a game theory based formulation for resource block allocations can have at least one Nash equilibrium point, and a distributed power control scheme was also proposed to minimize the cross-interference between cellular and D2D networks.

In addition, various problems of radio resource management for D2D networks have been widely investigated [14–21]. A channel aware scheduling scheme for D2D links was proposed in [14]. It was shown in [15] that if a BS can acquire the perfect information of channel gains of all communication links, multi-user diversity (MUD) gain can optimize the performance of D2D networks without significantly deteriorating the performance of cellular networks. Contrary to [14,15], the authors of [16] attempted to reduce the interference caused by D2D networks to cellular networks, and they proposed a simple heuristic algorithm because of the tremendous complexity of an optimal algorithm. The interference relationships among D2D and cellular communication links were modeled as an interference graph and a joint resource allocation scheme, yielding a near-optimal solution with low computational complexity [17]. An efficient bandwidth allocation scheme to maximize the utility of both D2D users and cellular users was proposed in [18]. A distributed algorithm with low complexity was also proposed because the original allocation problem was NP-hard. The convergence of the proposed distributed algorithm was proven in a static environment. The authors of [19] investigated how to form spectrum sharing partners between D2D links and cellular links optimally in cellular underlay D2D networks considering the cross-interference. Centralized and distributed algorithms for cellular overlay D2D networks were proposed in [20]. It was shown that a distributed algorithm can significantly reduce the signaling overhead with a marginal loss in performance, compared to a centralized algorithm. A two phase resource sharing algorithm was designed in such a way that its computational complexity could be adapted according to the network condition. In the first phase, the initial set of candidate channels is adaptively determined, and Lagrangian dual decomposition is used to determine the optimal power for D2D devices, maximizing the network sum rate in the second phase [21]. Finally, the authors in [22] proposed a novel peer-to-peer (P2P) protocol based on D2D communication, which combines the conventional application layer P2P protocol and the routing and scheduling schemes in lower layers.

On the other hand, an overview of the state-of-the-art deep learning architectures and algorithms relevant to the network traffic control systems was provided [23]. Deep neural network (DNN) based channel estimation and signal detection in orthogonal frequency-division multiplexing (OFDM) was studied [24]. A DNN enabled millimeter wave massive multiple-input multiple-output framework for effective hybrid precoding was also proposed [25]. A radio resource allocation algorithm for cognitive satellite communications was proposed by leveraging multi-objective deep reinforcement learning (DRL) and artificial neural network ensembles [26]. An energy efficient DRL based algorithm for unmanned aerial vehicle (UAV) control was proposed [27]. Despite recent improvements, DNNs tend to be easily over-fitted, while DRL faces several challenges. For DRL, a policy must be inferred by trial-and-error interaction with the environment, and agents must deal with long range time dependencies, which is known as the credit assignment problem [28].

3. Network Model

We investigated a cellular overlay D2D communication network with $2N$ mobile devices, as illustrated in Figure 1. We had no mutual interference between D2D links and cellular links because a dedicated radio resource was allocated for D2D devices, while all D2D devices shared radio spectra for higher spectral efficiency. If a mobile device wishes to receive data, it must be associated with another mobile device storing the data. We assumed that the process for associations was beyond the scope of this paper. We have N associations in Figure 1 because each mobile device was already associated with another mobile device. Although all N pairs were allowed to share radio spectra, the overall performance of the network could be enhanced by optimally choosing D2D pairs among N pairs due to the interference. In this paper, we thus focus on how to choose an optimal set of D2D pairs among N pairs. h_{ji} denotes a channel coefficient between a transmitter i and a receiver

j, where $1 \leq i \leq N$ and $1 \leq j \leq N$. We assumed a semi-static Rayleigh fading channel model. Thus, h_{ji} $\forall i, j$ was distributed by a complex Gaussian distribution following $\sim \mathcal{CN}(0, 1)$. All channel coefficients were independent and identically distributed (i.i.d.). It was assumed that all channels between transmitters and receivers were reciprocal because a time division duplexing (TDD) scheme was considered, and thus, $h_{ji} = h_{ij}$ $\forall i, j$. In addition, we assumed that all channels were semi-static. Thus, h_{ji} was static during one frame period and varied randomly each frame period. If $i = j$, h_{ji} denotes the gain of the channel for the ith pair's data transmission. Otherwise, it denotes the gain of an interference channel. All mobile devices' transmission power was identical and denoted by P_t. We considered a greedy source model as a traffic demand model where all transmitters had infinite packets to transmit. A greedy source model is a simple packet data model that is effective in analyzing the maximum throughput or data rate without guaranteeing any quality-of-service. In this paper, we focus on verifying the feasibility of deep learning in D2D networks and investigating the accuracy of the deep learning based algorithms. Even though more practical spatial and mobility models such as stochastic geometry and Manhattan models were not considered for simplicity, they will be able to be considered in our future work once the feasibility of deep learning has been verified in D2D networks.

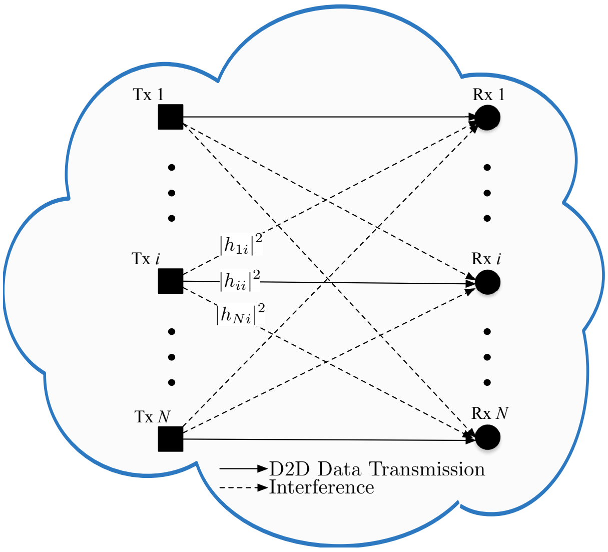


Figure 1. An illustration of a D2D communication network with N pairs.

If we define $\mathbb{T}$ and $\mathbb{U}$ as a transmission set consisting of the pairs of devices that will be allowed to transmit data and a universal set consisting of all N pairs, respectively, then $\mathbb{T} \subset \mathbb{U} \triangleq \{1, 2, \cdots, N\}$. For a given $\mathbb{T}$, the received signal-to-interference plus noise power ratio (SINR) for the i-pair in the given $\mathbb{T}$, γ_i, can be calculated as:

$$\gamma_i = \frac{P_t |h_{ii}|^2}{P_n + \sum_{j \in \mathbb{T}, j \neq i} P_t |h_{ij}|^2}, \; i \in \mathbb{T}, \tag{1}$$

where P_n denotes a Gaussian thermal noise power. If the numerator and denominator of (1) are both divided by P_n, (1) can be rewritten as:

$$\gamma_i = \frac{\Gamma |h_{ii}|^2}{1 + \sum_{j \in \mathbb{T}, j \neq i} \Gamma |h_{ij}|^2}, \; i \in \mathbb{T}, \tag{2}$$

where Γ is defined as $\frac{P_t}{P_n}$ and denotes a transmission power of the signal-to-noise power ratio (SNR). Then, the sum rate for the given $\mathbb{T}$ can be easily calculated as:

$$R_{\mathbb{T}} = \sum_{i \in \mathbb{T}} \log_2 (1 + \gamma_i). \tag{3}$$

We can find an optimal set of pairs, $\mathbb{T}^*$, to maximize the sum rate as follows:

$$\mathbb{T}^* = \arg\max_{\mathbb{T}\subset\mathbb{U}} R_{\mathbb{T}}. \tag{4}$$

4. Proposed Deep Learning Based Scheme

Supervised deep learning algorithms continuously train neural networks to minimize the error of the output of the neural networks and the target solution. An extensive amount of data is thus required for training. In this paper, we repeated the training of our neural network toward optimal solutions given in (4) and obtained data for the repeated trainings from extensive channel realizations. In addition, we should verify whether the algorithms have been over-fitted by using extra channel realizations different from those used for training. If the whole channel gains are available, we can find the optimal combination given in (4) based on the brute-force searching algorithm. However, the brute-force searching algorithm will cause a tremendous computational complexity, especially as N increases. Thus, we formulated a sub-optimal scheme as an alternative to obtain data samples required to train our deep learning algorithm.

4.1. A Sub-Optimal Scheme to Obtain Data Samples for Training

The main concept of the sub-optimal scheme was proposed in our previous study [20]. It was shown that the sub-optimal scheme can achieve comparable sum rates to the brute-force searching scheme with an extremely low computational complexity. The sub-optimal scheme is described in Algorithm 1. In this paper, we used the sub-optimal scheme to obtain data samples instead of an optimal scheme merely because of the complexity of the optimal scheme. However, using the sub-optimal scheme does not cause any change in the proposed algorithm, nor does it limit the contributions of this paper. For given N pairs, the brute-force scheme requires a maximum of 2^N iterations, while the sub-optimal scheme only requires a maximum of N iterations. In the sub-optimal scheme, N pairs of mobile devices are sorted according to their channel gains in descending order, ignoring interference channels. The sorted pairs are re-indexed by $\hat{i}$, $\hat{1} \le \hat{i} \le \hat{N}$. Thus, the sorted pairs satisfy:

$$|h_{\hat{1}\hat{1}}|^2 \ge |h_{\hat{2}\hat{2}}|^2 \ge \cdots \ge |h_{\hat{N}\hat{N}}|^2. \tag{5}$$

Algorithm 1 A sub-optimal algorithm to obtain training samples.

```
Sort $|h_{ii}|^2$ in descending order
Initialize: $\mathbb{T} = \emptyset$ and $R_0 = 0$
for $k = 1$ to $N$ do
    for $i = 1$ to $k$ do
        Calculate the SINR for the $\hat{i}^{\text{th}}$ pair, $\gamma_{\hat{i}}$
    end for
    $R_k = \sum_{i=1}^{k} \log_2(1 + \gamma_{\hat{i}})$
    if $R_{k-1} \le R_k$ then
        $\mathbb{T} = \mathbb{T} \cup \{\hat{k}\}$
    else
        break
    end if
end for
```

In the $k(1 \le k \le N)^{\text{th}}$ iteration, the sub-optimal scheme calculates $R_k = \sum_{\hat{i}=\hat{1}}^{\hat{k}} \log_2(1 + \gamma_{\hat{i}})$, which is the sum rate when the k pairs $\hat{1}$ through $\hat{k}$ transmit data simultaneously, and compares it with

R_{k-1}. If the calculated sum rate is greater than or equal to the sum rate obtained in the previous iteration, i.e., $R_{k-1} \leq R_k$, the pair $\hat{k}$ is allowed to transmit data and added to $\mathbb{T}$. Thus, $\mathbb{T}$ is updated by $\mathbb{T} = \mathbb{T} \cup \{\hat{k}\}$, and the algorithm moves on to the next iteration. Otherwise, the algorithm is terminated. Finally, the pairs included in the transmission set $\mathbb{T}$ are allowed to transmit data simultaneously as soon as the algorithm is terminated early before N iterations or stops after completing N iterations.

4.2. A Proposed Scheme Based on Convolutional Neural Networks

The architecture of our CNN for deep learning is shown in Figure 2 and consists of two hidden convolution layers. The first convolution layer consists of 256 convolution filters with an $N \times N$ input matrix. The input matrix consists of channel coefficients and is denoted by $[h_{ji}]_{1 \leq j \leq N, 1 \leq i \leq N}$. Each convolution filter is initialized by the Xavier normal initializer [29]. The width and height of the output of a convolution filter can both be calculated by:

$$O = \frac{N - K + 2P}{S} + 1, \tag{6}$$

where O is the width and height of the output of a convolution filter, N is the input size, K is the kernel (filter) size, P is the number of paddings, and S is the stride. In the first convolutional layer, it was assumed that the kernel size of each convolution filter was 5 × 5 with a stride of one, and we did not pad zeros; thus, $K = 5$, $S = 1$, and $P = 0$. Based on (6), the height and width of our first convolutional layer is given by:

$$O_1 = N - 4. \tag{7}$$

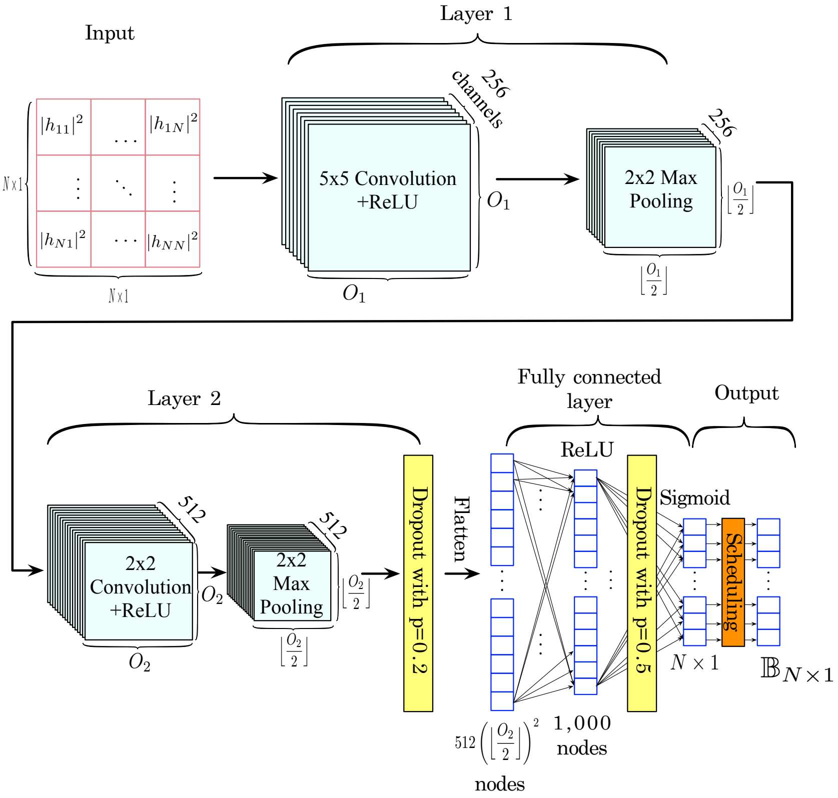


Figure 2. Proposed convolutional neural network for D2D communication networks.

Each convolution filter was activated by a rectified linear unit (ReLU) function, which returned the element-wise $\max(x, 0)$ for a given input x. The output of each convolution filter was followed by a 2 × 2 max pooling layer. A 2 × 2 max pooling layer performed down-sampling operations along the spatial dimensions by applying a max filter to non-overlapping sub-regions. For each of the regions represented by the filter, the maximum value of that region would be output. Thus, each element of the output matrix would be the maximum value of a region in the original input. If O_1 is odd, the 2 × 2 max pool will be only applied to the $(O_1 - 1) \times (O_1 - 1)$ matrix except for the last column and row. Otherwise, it will be applied to the $O_1 \times O_1$ matrix. Thus, the width and height of the output of the 2 × 2 max pool layer is given by $\left\lfloor \frac{O_1}{2} \right\rfloor$, which can be calculated as:

$$\left\lfloor \frac{O_1}{2} \right\rfloor = \left\lfloor \frac{N}{2} - 2 \right\rfloor = \left\lfloor \frac{N}{2} \right\rfloor - 2, \tag{8}$$

if O_1 is replaced by (7). The final output size of the first layer was $\left(\left\lfloor \frac{N}{2} \right\rfloor - 2 \right) \times \left(\left\lfloor \frac{N}{2} \right\rfloor - 2 \right) \times 256$. The second convolution layer consisted of 512 convolution filters. Each filter was also initialized by the Xavier normal initializer and $K = 2$. We also assumed that $S = 1$ and $P = 0$. The input size of the second convolutional layer was the output size of the first max pooling layer, which is given in (8). If N is replaced by $\left\lfloor \frac{O_1}{2} \right\rfloor$ in (6), then the width and height of the output of each convolution filter in the second layer are given as:

$$O_2 = \left\lfloor \frac{O_1}{2} \right\rfloor - 1 = \left\lfloor \frac{N-4}{2} \right\rfloor - 1 = \left\lfloor \frac{N}{2} \right\rfloor - 3. \tag{9}$$

As in the first convolution layer, each convolution filter was also activated by a ReLU function, and the output of each filter was down-sampled by a 2 × 2 max pooling layer. The width and height of the output of the 2 × 2 max pooling layer is given by $\left\lfloor \frac{O_2}{2} \right\rfloor$, which can be calculated as:

$$\left\lfloor \frac{O_2}{2} \right\rfloor = \left\lfloor \frac{\left\lfloor \frac{N}{2} \right\rfloor - 3}{2} \right\rfloor = \left\lfloor \frac{\frac{N}{2} - 3}{2} \right\rfloor = \left\lfloor \frac{N-6}{4} \right\rfloor, \tag{10}$$

where the second equality is valid because:

$$\left\lfloor \frac{x+m}{n} \right\rfloor = \left\lfloor \frac{\lfloor x \rfloor + m}{n} \right\rfloor \tag{11}$$

for any positive integer n [30]. The output size of the second max pooling layer was $\left(\left\lfloor \frac{N-6}{4} \right\rfloor \right) \times \left(\left\lfloor \frac{N-6}{4} \right\rfloor \right) \times 512$. The outputs of the max pooling layer were dropped out with a probability $p = 0.2$ to prevent the neural network from over-fitting. Thus, randomly selected neurons were ignored with a probability of 0.2 during training. The outputs were flattened to a one-dimensional array with the size of $512 \left(\left\lfloor \frac{N-6}{4} \right\rfloor \right)^2 \times 1$ and were reduced to 1000 × 1 by a fully connected layer, which had a ReLU as an activation function. The 1000 × 1 array went through another drop-out layer with $p = 0.5$. It was reduced to an $N \times 1$ array by another connected layer. Finally, the output of the fully connected layer was activated by the sigmoid function. The sigmoid function defined by $S(x) = \frac{1}{1+e^{-x}}$ for a given input x can be interpreted as a probability in many applications because $0 \leq S(x) \leq 1$. The output activated by the sigmoid function is denoted by $\mathbb{P}$, and the ith element of the $\mathbb{P}$, $\mathbb{P}[i]$, can be interpreted as the probability that the ith D2D pair is allowed to transmit data. Our scheduler determined if each D2D pair i would be allowed to transmit data based on the corresponding $\mathbb{P}[i]$. Thus, $\mathbb{B}[i]$ indicating whether to allow the ith D2D pair to transmit data can be determined as:

$$\mathbb{B}[i] = \begin{cases} 1, & \text{if } \mathbb{P}[i[] \geq 0.5 \\ 0, & \text{otherwise} \end{cases}, \; 1 \leq i \leq N. \tag{12}$$

Our proposed neural network was repeatedly trained to enhance the performance of scheduling by reducing the error between $\mathbb{B}$ and the result obtained by the sub-optimal scheme.

5. Numerical Results

In this section, we analyze the performance of the proposed scheme based on a CNN by using Python and Tensorflow. We obtained 100,000 data samples from the sub-optimal scheme, of which 90,000 samples were used to train the neural network proposed in Section 4 to increase the accuracy of scheduling, and the remaining 10,000 samples were used to verify whether the trained neural network was well fitted by testing the accuracy of scheduling based on the neural network. The size of batch was set to 100, and the number of epochs was 100. We thus needed 900 iterations for each epoch to train our neural network. Over-fitting is always a challenging problem for neural networks. Although large learning rates increase the learning speed of neural networks, they can easily cause over-fittings. On the contrary, small learning rates that can prevent neural networks from over-fitting slow down the learning speed of neural networks, and tremendous computing power is thus required to train neural networks. In this paper, we used an early terminating learning scheme with an adaptively decreasing learning rate. The learning rate for the *i*th epoch can be given by:

$$r(i) = \frac{r_{\text{init}}}{1 + i \times d}, \; 1 \leq i \leq 100, \tag{13}$$

where r_{init} and d denote an initial learning rate and a non-negative number to control the decaying speed, respectively. If $d = 0$, learning rates are constant for all epochs. Figure 3 shows learning rates given in (13) for $r_{\text{init}} = 10^{-3}$ and $d = \in \{0.1, 0.2, \cdots, 1.0\}$. We began to train the neural network with r_{init}, which was relatively large to speed up the trainings. However, we decreased the learning rate gradually to prevent over-fitting of neural networks as i increased. Thus, $r(i)$ in Figure 3 decreased as i increased. $r(i)$ decreased more sharply as d increased. In this paper, we used $r_{\text{init}} = 10^{-3}$ and $d = 1$ for our neural network. In addition, our training procedure could be automatically terminated if there was no improvement for three epochs to reduce the training time.

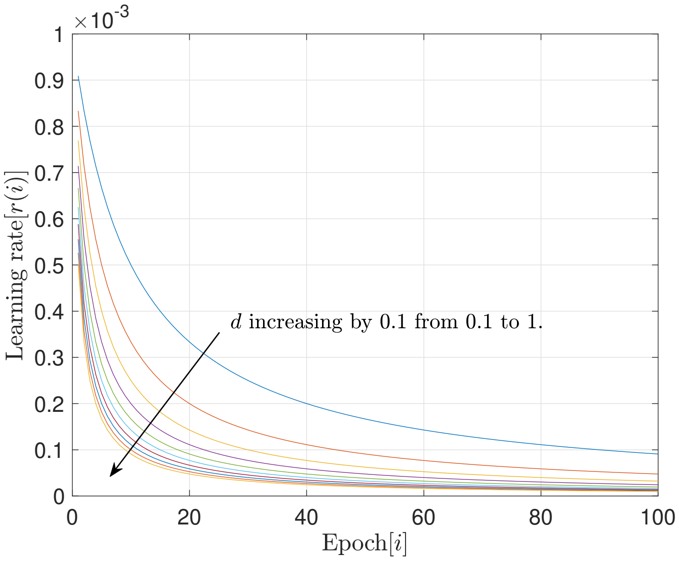
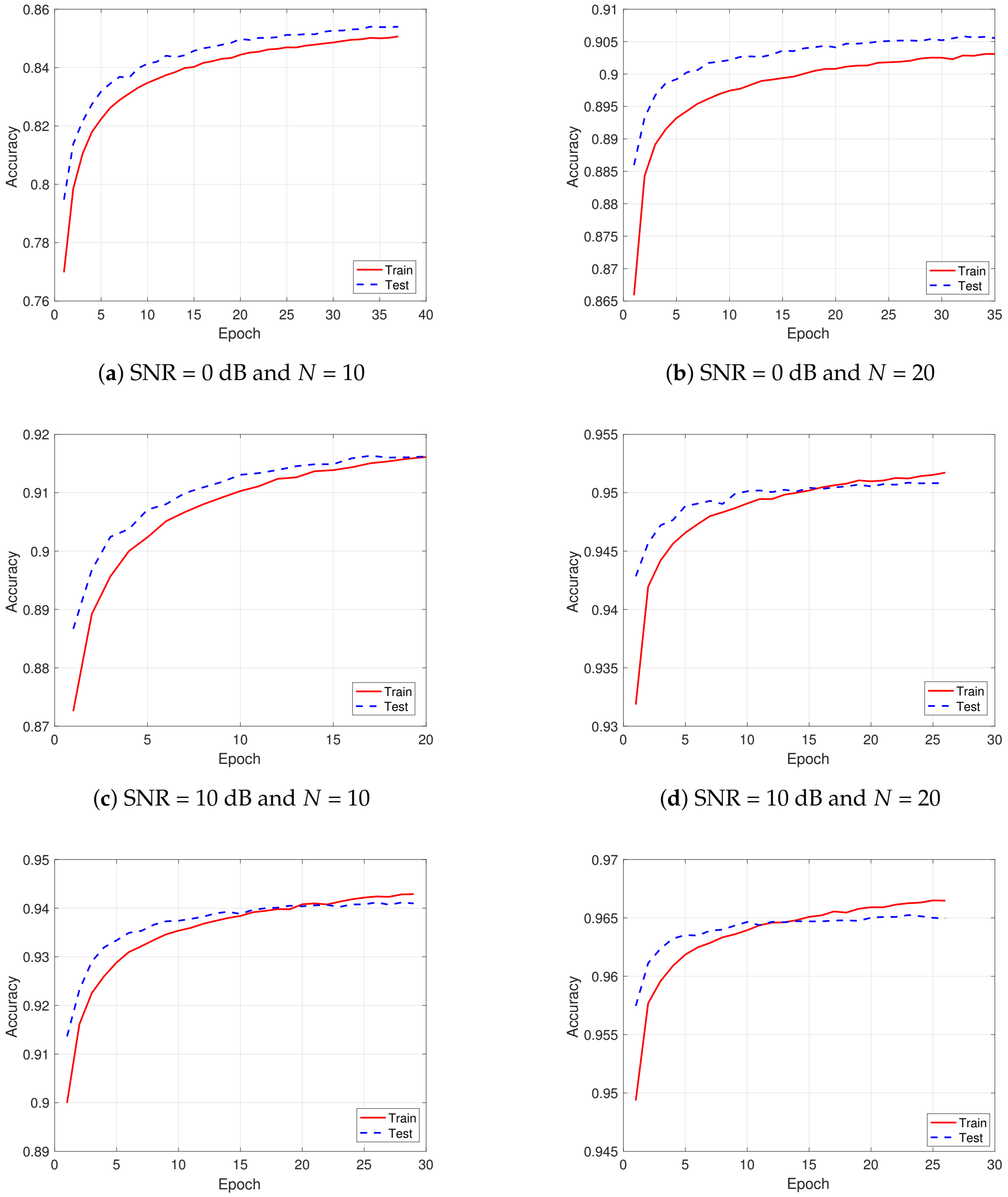


Figure 3. A decaying learning rate as an epoch increases. $r_{\text{init}} = 10^{-3}$, and d increases by 0.1 from 0.1 to one.

Figure 4 shows the accuracy of the proposed neural network. SNR was set to 0 dB, 10 dB, or 20 dB, and N was set to 10 or 20. The accuracy was measured by comparing $\mathbb{B}$ obtained from the neural network with that of the sub-optimal scheme. It was clearly shown that the accuracies were enhanced as the number of epochs for training increased regardless of SNR and N. The trainings were terminated at different epochs due to the early terminating learning scheme. For SNR $=0$ dB, when $N=10$ and $N=20$, the trainings were terminated at the 37th and 35th epochs, respectively. For SNR $=10$ dB, when $N=10$ and $N=20$, the trainings were terminated at the 20th and 26th epochs, respectively. For SNR $=20$ dB, when $N=10$ and $N=20$, the trainings were terminated at the 29th and 26th epochs, respectively. It was shown that the accuracy of the proposed neural network based scheduling improved as N or SNR increased. There were many more training data than test data. Thus, it took more time to stabilize the accuracy for the training data. It was shown that as the epoch increased, the accuracy for the training data became higher than for the test data. It was also confirmed that no over-fitting was observed.



(**a**) SNR = 0 dB and N = 10

(**b**) SNR = 0 dB and N = 20

(**c**) SNR = 10 dB and N = 10

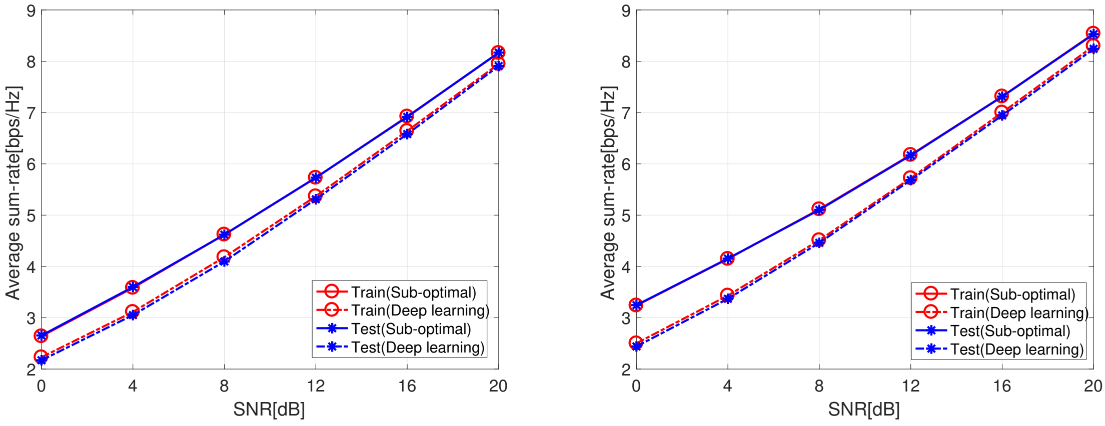
(**d**) SNR = 10 dB and N = 20

(**e**) SNR = 20 dB and N = 10

(**f**) SNR = 20 dB and N = 20

Figure 4. Accuracy of the deep learning based scheme when SNR$\in$ {0 dB, 10 dB, 20 dB} and N = 10 or 20.

Figure 5 shows the average sum rates for both the training samples and test samples, obtained by the proposed neural network and the sub-optimal scheme, respectively. SNR $\in$ {0 dB, 4 dB, $\cdots$, 20 dB}, and N was set to 10 or 20. As shown in Figure 2, the number of convolution layers, filters per layer, and filter size that were used in this paper were all restricted due to the limitation in computing power. Thus, all average sum rates of the neural network were lower than those of the sub-optimal scheme for both the training samples and test samples, regardless of SNR and N. Fortunately, however, the difference of average sum rates between the neural network and the sub-optimal scheme decreased as SNR increased. For the neural network, no significant difference of the average sum rates between the training samples and the test samples was observed, which showed that over-fitting was efficiently prevented thanks to the learning scheme with adaptive learning rates.



(**a**) N = 10. (**b**) N = 20.

Figure 5. Average sum rates of the deep learning based scheme and sub-optimal scheme for various SNR values when N = 10 or 20.

6. Conclusions

In this paper, we investigated D2D communication networks, which are attractive for offloading mobile Internet traffic from core networks and can significantly enhance the quality of communications and spectral efficiency by reducing end-to-end communication ranges between transmitters and receivers, compared to mobile cellular communication networks. The performance of D2D communication networks was closely related to how transmissions of D2D pairs are scheduled. In this paper, we adopted a new approach to schedule transmissions in D2D communication networks efficiently using supervised learning based on a CNN. The CNN consisted of two convolution layers and a fully connected layer. We used a sub-optimal scheme instead of an optimal scheme to obtain samples for supervised learning because an optimal scheme achieving the maximal performance requires a tremendous computational complexity. Ninety percent of the obtained samples were used to train the neural network to achieve the same scheduling results as the sub-optimal scheme, while the remaining 10% of the obtained samples were used to test whether the trained neural network was over-fitted. To overcome our limitation in computing power, we adopted an early terminating learning scheme with an adaptive learning rate where the training procedure was automatically terminated if no improvement was observed for three epochs, and a learning rate that began with quite a large value exponentially decreased as the epoch increased. Our extensive numerical results showed that the neural network could yield about 85%$\sim$95% accuracies, which indicated that 85%$\sim$95% of scheduling decisions from the neural network were identical to the scheduling decisions from the sub-optimal scheme, which was the target algorithm. Especially when SNR $=$ 20 dB and N $=$ 20, the accuracy of the neural network approached about 97%, and the average sum rate of the trained neural network was also about 97% of the sub-optimal scheme for both the training samples and the test samples.

Author Contributions: The authors contributed equally.

Funding: This work was supported by the Institute for Information & communications Technology Promotion (IITP) grant funded by the Korean government (MSIT) (No. 2015- 0-00820, A research on a novel communication system using storage as wireless communication resource).

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

References

1. Cisco, Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022, Whitepaper. 2018. Available online: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html (accessed on 30 October 2019).
2. Yang, P.; Zhang, N.; Zhang, S.; Yu, L.; Zhang, J.; Shen, X. Content Popularity Prediction Towards Location-Aware Mobile Edge Caching. *IEEE Trans. Multimed.* **2019**, *21*, 915–929. [CrossRef]
3. Kang, H.J.; Kang, C.G. Mobile Device-to-Device (D2D) Content Delivery Networking: A design and Optimization Framework. *IEEE J. Commun. Netw.* **2014**, *16*, 568–577. [CrossRef]
4. Malak, D.; Al-Shalash, M.; Andrews, J.G. Optimizing Content Caching to Maximize the Density of Successful Receptions in Device-to-Device Networking. *IEEE Trans. Commun.* **2016**, *64*, 4365–4380. [CrossRef]
5. Golrezaei, N.; Mansourifard, P.; Molisch, A.F.; Dimakis, A.G. Base-Station Assisted Device-to-Device Communications for High-Throughput Wireless Video Networks. *IEEE Trans. Wirel. Commun.* **2014**, *13*, 3665–3676. [CrossRef]
6. Golrezaei, N.; Dimakis, A.G.; Molisch, A.F. Scaling Behavior for Device-to-Deivce Communications with Distributed Caching. *IEEE Trans. Inf. Theory* **2014**, *60*, 286–4298. [CrossRef]
7. Zhang, L.; Xiao, M.; Wu, G.; Li, S. Efficient Scheduling and Power Allocation for D2D-Assisted Wireless Caching Networks. *IEEE Trans. Commun.* **2016**, *64*, 2438–2452. [CrossRef]
8. Doppler, K.; Yu, C.; Ribeiro, C.B.; Jänis, P. Mode selection for Device-to-Device Communication underlaying an LTE-Advanced Network. In Proceedings of the 2010 IEEE Wireless Communication and Networking Conference, Sydney, Australia, 18–21 April 2010.
9. Hakola, S.; Chen, T.; Lehtomäki, J.; Koskela, T. Device-to-Device (D2D) Communication in Cellular Network—Performance Analysis of Optimum and Practical Communication Mode Selection. In Proceedings of the 2010 IEEE Wireless Communication and Networking Conference, Sydney, Australia, 18–21 April 2010.
10. Zhu, K.; Hossain, E. Joint Mode Selection and Spectrum Partitioning for Device-to-Device Communication: A Dynamic Stackelberg Game. *IEEE Trans. Wirel. Commun.* **2015**, *14*, 1406–1420. [CrossRef]
11. Peng, T.; Lu, Q.; Wang, H.; Xu, S.; Wang, W. Interference avoidance mechanisms in the hybrid cellular and device-to-device systems. In Proceedings of the 2009 IEEE 20th International Symposum on Personal, Indoor and Mobile Radio Communications, Tokyo, Japan, 13–16 September 2009.
12. Kim, J.; Karim, N.A.; Cho, S. An Ineterference Mitigation Scheme of Device-to-Device Communications for Sensor Networks Underlying LTE-A. *Sensors* **2017**, *17*, 1088.
13. Katsinis, G.; Tsiropoulou, E.E.; Papavassiliou, S. Multicell Interference Management in Device-to-Device Underlay Cellular Networks. *Future Ineternet* **2017**, *9*, 44. [CrossRef]
14. Wu, X.; Tavildar, S.; Shakkottai, S.; Richardson, T.; Li, J.; Laroia, R.; Jovicic, A. FlashLinQ: A Synchronous Distributed Scheduler for Peer-to-Peer Ad Hoc Networks. *IEEE/ACM Trans. Netw.* **2013**, *21*, 1215–1228. [CrossRef]
15. Janis, P.; Koivunen, V.; Ribeiro, C.; Korhonen, J.; Doppler, K.; Hugl, K. Interference-aware resource allocation for device-to-device radio underlaying cellular networks. In Proceedings of the IEEE 69th Vehicular Technology Conference, Barcelona, Spain, 26–29 April 2009.
16. Zulhasnine, M.; Huang, C.; Srinivasan, A. Efficient resource allocation for device-to-device communication underlaying LTE network. In Proceedings of the 2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications, Niagara Falls, ON, Canada, 11–13 October 2010; pp. 368–375.
17. Zhang, R.; Cheng, X.; Yang, L.; Jiao, B. Interference Graph-Based Resource Allocation (InGRA) for D2D Communications Underlaying Cellular Networks. *IEEE Trans. Veh. Technol.* **2015**, *64*, 3844–3850. [CrossRef]

18. Zhang, J.; Wu, G.; Xiong, W.; Chen, Z.; Li, S. Utility-Maximization Resource Allocation for Device-to-Device Communication Underlaying Cellular Networks. In Proceedings of the 2013 IEEE Globecom Workshop, Atlanta, GA, USA, 9–13 December 2013; pp. 623–628.
19. Wang, L.; Tang, H.; Wu, H.; Stüber, G.L. Resource Allocation for D2D Communications Underlay in Rayleigh Fading channels. *IEEE Trans. Veh. Technol.* **2017**, *66*, 1159–1170. [CrossRef]
20. Ban, T.-W.; Jung, B.C. On the Link Scheduling for Cellular-Aided Device-to-Device Networks. *IEEE Trans. Veh. Technol.* **2016**, *65*, 9404–9409. [CrossRef]
21. Esmat, H.H.; Elmesalawy, M.M.; Ibrahim, I.I. Adaptive Resource Sharing Algorithm for Device-to-Device Communications Underlaying Cellular Networks. *IEEE Commun. Lett.* **2016**, *20*, 530–533. [CrossRef]
22. Wu, Y.; Wang, S.; Liu, W.; Guo, W.; Chu, X. Iunius: A Cross-Layer Peer-to-Peer System with Device-to-Device Communications. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 7005–7017. [CrossRef]
23. Fadlullah, Z.M.; Tang, F.; Mao, B.; Kato, N.; Akashi, O.; Inoue, T.; Mizutani, K. State-of-the-Art Deep Learning: Evolving Machine Intelligence toward Tomorrow's Intelligent Network Traffic Control Systems. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2432–2455. [CrossRef]
24. Ye, H.; Li, G.Y.; Juang, B. Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems. *IEEE Wirel. Commun. Lett.* **2018**, *7*, 114–117. [CrossRef]
25. Huang, H.; Song, Y.; Yang, J.; Gui, G.; Adachi, F. Deep-Learning-Based Millimeter-Wave Massive MIMO for Hybrid Precoding. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3027–3032. [CrossRef]
26. Ferreira, P.V.R.; Paffenroth, R.; Wyglinski, A.M.; Hackett, T.M.; Bilén, S.G.; Reinhart, R.C.; Mortensen, D.J. Multiobjective Reinforcement Learning for Cognitive Satellite Communications Using Deep Neural Network Ensembles. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 1030–1041.
27. Liu, C.H.; Chen, Z.; Tang, J.; Xu, J.; Piao, C. Energy-Efficient UAV Control for Effective and Fair Communication Coverage: A Deep Reinforcement Learning Approach. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 2059–2070. [CrossRef]
28. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep Reinforcement Learning: A Brief Survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. [CrossRef]
29. Hanin, B.; Rolnick, D. How to Start Training: The Effect of Initialization and Architecture. In Proceedings of the 32nd Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018.
30. Graham, R.L.; Knuth, D.E.; Patashnik, O. Concrete Mathematics:: A Foundation for Computer Science. *Comput. Phys.* **1989**, *3*, 106–107. [CrossRef]