

Article

Efficient Gradient Updating Strategies with Adaptive Power Allocation for Federated Learning over Wireless Backhaul

Yunji Yang, Yonggi Hong and Jaehyun Park *

Division of Smart Robot Convergence and Application Engineering, Department of Electronic Engineering, Pukyong National University, Busan 48513, Korea; yyj26@pukyong.ac.kr (Y.Y.); yongki0503@hanmail.net (Y.H.)

* Correspondence: jaehyun@pknu.ac.kr

Abstract: In this paper, efficient gradient updating strategies are developed for the federated learning when distributed clients are connected to the server via a wireless backhaul link. Specifically, a common convolutional neural network (CNN) module is shared for all the distributed clients and it is trained through the federated learning over wireless backhaul connected to the main server. However, during the training phase, local gradients need to be transferred from multiple clients to the server over wireless backhaul link and can be distorted due to wireless channel fading. To overcome it, an efficient gradient updating method is proposed, in which the gradients are combined such that the effective SNR is maximized at the server. In addition, when the backhaul links for all clients have small channel gain simultaneously, the server may have severely distorted gradient vectors. Accordingly, we also propose a binary gradient updating strategy based on thresholding in which the round associated with all channels having small channel gains is excluded from federated learning. Because each client has limited transmission power, it is effective to allocate more power on the channel slots carrying specific important information, rather than allocating power equally to all channel resources (equivalently, slots). Accordingly, we also propose an adaptive power allocation method, in which each client allocates its transmit power proportionally to the magnitude of the gradient information. This is because, when training a deep learning model, the gradient elements with large values imply the large change of weight to decrease the loss function.

Keywords: federated learning; image classification; wireless backhaul; aggregated gradient updating



check for updates

Citation: Yang, Y.; Hong, Y.; Park, J. Efficient Gradient Updating Strategies with Adaptive Power Allocation for Federated Learning over Wireless Backhaul. *Sensors* **2021**, *21*, 6791. <https://doi.org/10.3390/s21206791>

Academic Editor: Leopoldo Angrisani

Received: 23 August 2021
Accepted: 10 October 2021
Published: 13 October 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

Recently, deep neural networks (DNNs) or convolutional neural networks (CNNs) have been widely applied to complicated signal processing, such as classification tasks and signal regression problems, due to their outstanding performances in nonlinear adaptability and feature extraction ([1–3] and references therein) and are also extended to the distributed sensing systems (e.g., the object recognition using distributed micro-Doppler radars in [4] and the data driven digital healthcare applications [5–7]). In the distributed sensing systems, centralized training strategies may be adopted to train their common DNN or CNN modules by sharing their sensing data. However, due to the data-size and the privacy issues of the locally collected data, the centralized training is not desirable, especially when the capacity of the backhaul link for the data exchange is limited.

The federated learning approach has been extensively investigated as an alternative distributed machine learning method [8,9] where, rather than sharing their locally collected dataset, the clients report the stochastic gradient information (minimizing the loss function with respect to their local dataset) to the main server. The main server then aggregates the stochastic gradient information and broadcast it to the clients. Accordingly, to achieve the unbiased stochastic gradient at the main server, the training data sampling methods are investigated [10,11]. Furthermore, in [12], to reduce the communication overhead of transmitting the updated gradient information (proportional to the number of weights in

the DNNs and CNNs), an efficient weight aggregation protocol for federated learning is proposed and in [13], the structured updating method is proposed for the communication cost reduction. However, they assume that the stochastic gradient information is perfectly transferred from the multiple clients to the main server without any distortion.

In the federated learning process, when the clients are connected with wireless-connected clients, local gradient information needs to be transferred from the distributed clients to the server over the wireless backhaul link and can be distorted due to wireless channel fading. In [14–17], for the wireless backhaul, the federated learning strategies are proposed for the MNIST hand-writing image classification and the associated wireless resources are efficiently optimized. In [14,15], the average of the local stochastic gradient vectors is recovered at the server when the pre-processed local gradient vectors are transferred from the clients. In [16], the compressive sensing approach is proposed to estimate the local gradient vectors at the server. In [17], joint communication and federated learning model is developed, where the resource allocation and the client selection methods are proposed such that the packet error rates of the communication links between server and clients are optimized. We note that most of the previous works have focused on the estimation of local stochastic gradient vectors at the server.

In this paper, we also consider the federated learning system, where distributed clients are connected to the server via wireless backhaul link and develop efficient training strategies for the federated learning over wireless backhaul link. Differently from the previous works, where the average of the local stochastic gradient vectors (i.e., the equal-weight combining) is recovered at the server, we propose an efficient gradient updating method, in which the local gradients are combined such that the effective signal-to-noise ratio (SNR) is maximized at the server. In addition, we also propose a binary gradient updating strategy based on thresholding in which the round associated with all channel having small channel gains is excluded from federated learning. That is, when the backhaul links for all clients have channel gain smaller than a pre-defined threshold simultaneously, the server may have severely distorted gradient vectors, which can be avoided through the proposed updating with thresholding. Furthermore, because each client has limited transmission power, it is effective to allocate more power on the channel slots carrying specific important information, rather than allocating power equally to all channel resources (equivalently, slots). Accordingly, we also propose an adaptive power allocation method, in which each client allocates its transmit power proportionally to the magnitude of the gradient information. This is because, when training a deep learning model, the gradient elements with large values imply the large change of weight to decrease the loss function.

Through the extensive computer simulations, it can be found that the proposed gradient updating methods improve the federated learning performance over the wireless channel. Specifically, due to the distortion over wireless channel, the classification accuracy of the equal-weight combining decreases drastically as the rounds of the federated learning increase. In contrast, the proposed effective SNR maximizing scheme with thresholding exhibits the accuracy performance which is comparable to that for the federated learning over the error-free backhaul link. We note that, as the threshold level increases, the federated learning is performed stably, because the highly distorted gradient update vector due to small channel gain can be discarded by a large threshold level. However, the large threshold level may incur the gradient updating delay, but the adaptive power allocation strategy can improve the trade-off between the federated learning performance and the learning delay due to the threshold level.

The rest of this paper is organized as follows. In Section 2, the system model for the federated learning system with the wireless backhaul is presented in which the distributed clients have a common CNN module for the handwriting character recognition. In Section 4, gradient updating methods are proposed. In addition, the adaptive power allocation method is also developed considering the importance of the gradient information. In Section 5, we provide several simulation results and in Section 6, we give our conclusions.

2. System Model

In Figure 1, we consider the federated learning systems with wireless backhaul, where the L multi-clients have their own datasets to train each local network. Here, a common neural network model is shared for all clients and it is trained through the federated learning over wireless backhaul connected to the main server. The common neural network is designed for the classification problem, in which the label \hat{d} is induced from the network output for the l th client's measured data with the label d , $\mathbf{S}_d^{(l)}$. That is,

$$\begin{aligned} \hat{d} &= \arg \max_{d=1, \dots, D} \mathbf{x}_{out}[d], \\ \text{s.t. } \mathbf{x}_{out} &= \mathbf{f}(\mathbf{S}_d^{(l)}; \boldsymbol{\theta}) \in \mathbb{R}^{D \times 1} \end{aligned} \quad (1)$$

where $\mathbf{f}; \boldsymbol{\theta}$ denotes the non-linear neural network function with the model parameter $\boldsymbol{\theta} (\in \mathbb{R}^{P \times 1})$ that gives the estimate of the categorical label probability vector as its output vector. Here, P denotes the number of weights in the common neural network model and $\mathbf{x}_{out}[d]$ is the d th element of the vector \mathbf{x}_{out} . We note that the size of the model parameter (P) is determined by the structure of the neural network model. Specifically, in the case of a convolutional layer with K $K_{f1} \times K_{f2}$ filters, the number of weights is given as $K_{f1} \times K_{f2} \times K + K$ that accounts for the kernel size ($K_{f1} \times K_{f2}$), the number of kernels (K) and the number of biases (K). In the case of a single fully-connected layer, the number of weights is calculated as $N_{in} \times N_{nr} + N_{nr}$, where N_{in} and N_{nr} denote the input size and the number of neurons, respectively. See also Section 2.1. We note that, because the collected data at each client are generally of a large dimension with private security issues, it is not desirable to report the collected data to the server. Furthermore, the large dimension of the data may cause the significant burden on the typical backhaul link to transmit a number of training datasets. Instead, the neural network model $\mathbf{f}; \boldsymbol{\theta}$ will be shared over all clients and $\boldsymbol{\theta}$ can be locally trained with the data obtained from each client. By denoting $\boldsymbol{\theta}^{(l)}$ as the model parameter trained at the l th client, $\boldsymbol{\theta}^{(l)}$ is reported to the server through the wireless uplink backhaul for the federated learning. The associated federated learning strategies and power allocation over the wireless backhaul will be discussed in more detail in Section 4.

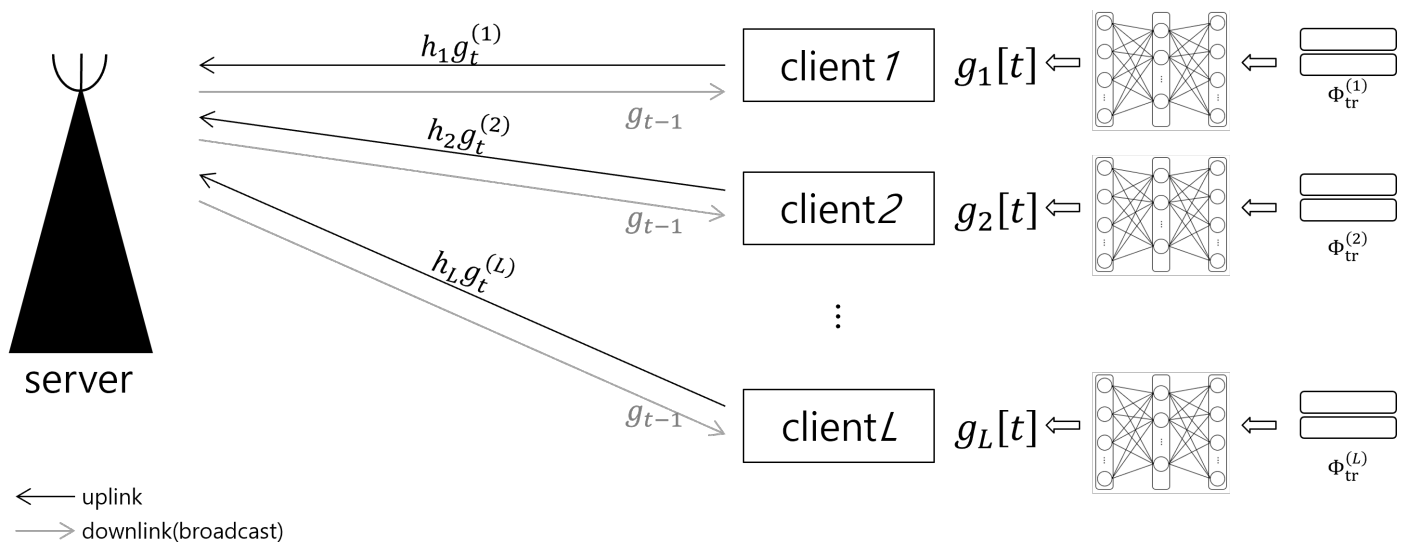


Figure 1. Federated Learning over wireless communication system for target classification.

2.1. CNN Architecture for Handwriting Character Recognition

Throughout the paper, multiple clients have a common neural network for the handwriting character recognition. Specifically, a typical CNN module is considered for the character image classification as in Figure 2, but the proposed federated learning strategy

can be applied to other CNN models. The non-linear neural network function $\mathbf{f}(\mathbf{S}_i^{(l)}; \boldsymbol{\theta})$ in (1) is composed of an input layer, convolutional layers, activation layers, max pooling layer, a fully-connected layer, and an output layer. See Section 4 for the specific values of the hyperparameters of CNN module.

Layers of 2D-CNN
Input layer $width \times height \times (samples)$
Convolutional Layer filter size $K_{f11} \times K_{f12}$ Number of filters K_1
Rectified Linear Unit
Max Pooling Layer
Convolutional Layer filter size $K_{f21} \times K_{f22}$ Number of filters K_2
Rectified Linear Unit
Max Pooling Layer
Convolutional Layer filter size $K_{f31} \times K_{f32}$ Number of filters K_3
Rectified Linear Unit & Dropout
Max Pooling Layer
Fully connected Layer
Rectified Linear Unit & Dropout
Fully connected Layer
Classification Layer

Figure 2. CNN module for handwriting character recognition.

– *Convolution Layer*: The handwriting image matrix, $\mathbf{S}_i^{(l)} \in \mathbb{R}^{N_{width} \times N_{height}}$ is exploited as the input of the convolution layers. In addition, each element of their output is computed through the convolution operation with a $K_{f_{i1}} \times K_{f_{i2}}$ filter (equivalently, kernel) for i th layer. Specifically, the output of the i th convolution layer can be given as:

$$X_{(i)}[m, n, k] = \sum_{p=0}^{K_{f_{i1}}-1} \sum_{q=0}^{K_{f_{i2}}-1} \sum_{k=0}^{K_{i-1}} f_a \left(X_{(i-1)}[m+p, n+q, k] W_{(i)}[p, q, k] \right) + b_{(i)}[k],$$

where $X_{(i-1)}[m, n, k]$ is the (m, n, k) th element of $\mathbf{X}_{(i-1)} \in \mathbb{R}^{m_{i-1} \times n_{i-1} \times k_{i-1}}$, the input of the i th layer and $f_a(\cdot)$ is an activation function. In addition, $W_{(i)}[p, q, k]$ is the (p, q, k) th element of the filter matrix $\mathbf{W}_{(i)}$ at the i th layer and $b_{(i)}[k]$ is the k th element of a bias vector $\mathbf{b}_{(i)}$. Throughout the paper, rectified linear unit (ReLU) function is used as the activation function, which is given as

$$f_a(\mathbf{X}_{(i)}) = \max(\mathbf{0}, \mathbf{X}_{(i)}).$$

– *Max pooling layer*: In the pooling layer, to reduce the dimension of the input data without losing useful information, the elements of the input are down-sampled [18]. In the Max pooling layer, after dividing the input matrix into multiple blocks, the maximum value in each block is sampled and forwarded to the dimension-reduced output matrix.

– *Flatten, Fully-Connected (FC) layer*: The flatten layer is used for changing the shape of output of convolution layer into the vector which is used as the input of FC layer. We note that, in the case of a single fully-connected layer with N_{in} input elements and N_{nr} neurons, the number of weights is given as $N_{in} \times N_{nr} + N_{nr}$. In the FC layer, the output of convolution layer is associated with a proper loss function such that the label is correctly identified after the training.

Throughout the paper, the cross entropy (CE) is used as the loss function which is given as

$$L_{CE}(\mathbf{x}_{out}, \bar{\mathbf{L}}_d; \boldsymbol{\theta}) = - \sum_{i=1}^D \bar{\mathbf{L}}_d[i] \log(\mathbf{x}_{out}[i]), \quad (2)$$

where $\mathbf{x}_{out} \in \mathbb{R}^{D \times 1}$ is the output of FC and $\bar{\mathbf{L}}_d$ is a label one-hot encoded vector of size D that has zeros in all elements except the d th element, which is assigned a value of 1. Then, by using the local training datasets $(\Phi_{tr}^{(l)} = \{\mathbf{S}_{d,tr}^{(l)}, \bar{\mathbf{L}}_{d,tr}\}_{t=1}^{N_{tr}})$ at the l th client, the network function parameter can be updated as:

$$\boldsymbol{\theta}_t^{(l)} \leftarrow \boldsymbol{\theta}_{t-1}^{(l)} + \mathbf{g}_{t-1}^{(l)}, \quad (3)$$

where $\mathbf{g}_{t-1}^{(l)} (\in \mathbb{R}^{P \times 1})$ denotes the gradient such that the loss function is minimized for the local training datasets $\Phi_{tr}^{(l)}$ and is given as $\mathbf{g}_{t-1}^{(l)} \triangleq -\eta \nabla_{\boldsymbol{\theta}} L_{CE}(\mathbf{x}_{out}, \bar{\mathbf{L}}_{d,tr}; \boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{t-1}^{(l)}}$ with a learning rate, η .

2.2. Signal Model for Wireless Backhaul

As in Figure 1, the clients are connected to the server through the wireless backhaul link. For the federated learning, the model parameters aggregated at the server are broadcast at each iteration of training phase through the wireless downlink channel, while the model parameters trained at the l th client are reported to the server through the wireless uplink backhaul link. Throughout the paper, we focus only on the uplink phase of multiple access channel and assume that the broadcast channel for the downlink phase is error-free, as done in [15,16,19].

Assuming that the clients and the server have a single antenna for the backhaul link, when total B channel resources with narrowband signal bandwidth are available (Here, we note that the channel resources may be given in the frequency axis or may be given in the time axis.), the received signal at server for the t th round of the gradient update can be given as

$$y_t[b] = \sum_{l=1}^L h_{l,t}[b] x_{l,t}[b] + n_t[b], \quad (4)$$

for $b = 1, \dots, B$, where $x_{l,t}[b]$ is the precoded transmit signal of the l th client at the b th channel resource with $E[|x_{l,t}[b]|^2] = 1$ for the t th round. Here, $h_{l,t}[b]$ and $n_t[b]$ denote the aggregated Rayleigh fading channel and the zero-mean additive white Gaussian noise (AWGN) at the b th channel resource, respectively. That is, $h_{l,t}[b]$ follows a Gaussian distribution with a zero-mean and a variance $\sigma_{h_l}^2$ (that is, $h_{l,t}[b] \sim \mathcal{N}(0, \sigma_{h_l}^2)$). Likewise, $n_t[b] \sim \mathcal{N}(0, \sigma_n^2)$. In addition, the wireless channel is constant over each round of federated learning process, but changes independently from round to round. By concatenating $y_t[b]$ in (4), the received signal at server can be vectorized as:

$$\mathbf{y}_t = \begin{bmatrix} y_t[1] \\ \vdots \\ y_t[B] \end{bmatrix} = \sum_{l=1}^L \mathbf{H}_{l,t} \mathbf{x}_{l,t} + \mathbf{n}_t, \quad (5)$$

where $\mathbf{H}_{l,t} = \text{diag}\{h_{l,t}[1], \dots, h_{l,t}[B]\}$ and

$$\mathbf{x}_{l,t} = \begin{bmatrix} x_{l,t}[1] \\ \vdots \\ x_{l,t}[B] \end{bmatrix}, \quad \mathbf{n}_t = \begin{bmatrix} n_t[1] \\ \vdots \\ n_t[B] \end{bmatrix}.$$

Here, $\text{diag}\{a_1, \dots, a_B\}$ denotes a $B \times B$ diagonal matrix having its diagonal elements as a_1, \dots, a_B .

3. Federated Learning for Handwriting Character Recognition

Note that, as in (3), the CNN parameter $\theta^{(l)}$ can be trained with the local training datasets at each client, which limits the adaptability of the CNN due to the lack of the globally measured data. Accordingly, to train their parameters globally, federated learning strategy is exploited, known as an efficient learning strategy suitable to the multi-clients environment such as our system model shown in Figure 1.

Specifically, during the t th round of the training phase, each client receives the gradient of the model parameter \mathbf{g}_{t-1} from the server via the backhaul link. Then, by exploiting \mathbf{g}_{t-1} instead of $\mathbf{g}_{t-1}^{(l)}$ in (3) the network function parameter can be updated as:

$$\theta_t^{(l)} \leftarrow \theta_{t-1}^{(l)} + \mathbf{g}_{t-1}. \quad (6)$$

We note that \mathbf{g}_{t-1} is the globally aggregated gradient computed at the server, which tends to minimize the loss function with respect to the data collected at all clients. Then, each client can compute its next local gradient $\mathbf{g}_t^{(l)}$ such that the local loss function is minimized for the locally collected datasets $\Phi_{tr}^{(l)}$. Then, the locally updated gradient vector is reported to the server via the backhaul link. The server can then aggregate the local gradient vector to get \mathbf{g}_t as:

$$\mathbf{g}_t = \mathbf{f}_g(\mathbf{g}_t^{(l)}, l = 1, \dots, L), \quad (7)$$

where the function $\mathbf{f}_g(\cdot)$ represents the gradient aggregation function. In [20], the Federated Averaging technique (i.e., equal weight combining) is proposed which is given as:

$$\mathbf{g}_t = \frac{1}{L} \sum_{l=1}^L \mathbf{g}_t^{(l)}. \quad (8)$$

The aggregated gradient \mathbf{g}_t is again broadcast to the multi-clients and exploited to update the neural network model at each client. The above described steps are repeated for a given number of rounds, T .

At the beginning of the training phase, the server needs to initialize the global model parameters and, throughout the paper, the parameters are initialized based on He normal weight initialization method [21], which is advantageous when used with ReLU activation function. Based on the above description, generalized federated learning process is summarized in Algorithm 1.

Algorithm 1 Generalized federated learning train process.

- 1: Initialize θ_0 based on He normal weight initialization method
 - 2: $\mathbf{g}_0 \leftarrow \mathbf{0}$
 - 3: **for** $t \leftarrow 1$ to T **do**
 - 4: (Clients) $\theta_t^{(l)} \leftarrow \theta_{t-1}^{(l)} + \mathbf{g}_{t-1}$
 - 5: (Clients) Update $\mathbf{g}_t^{(l)}$ from the datasets $\Phi_{tr}^{(l)}$
 - 6: (Clients) Report $\mathbf{g}_t^{(l)}$ to the server via the backhaul link
 - 7: (Server) $\mathbf{g}_t \leftarrow \mathbf{f}_g(\mathbf{g}_t^{(l)}, l = 1, \dots, L)$ as in (7)
 - 8: (Server) Broadcast \mathbf{g}_t to multi-clients
 - 9: **end for**
-

Differently from the centralized learning, the datasets collected by each client are not necessarily reported to the main server in Algorithm 1. We note that, in many cases, data sharing is not free from security, regulatory and privacy issues [8]. We also note that the communication cost for the centralized learning depends on the number/size of the

collected data [22,23]. In contrast, the communication cost for the federated learning is independent with the data size, but depends on the CNN architecture (specifically, the number of weights in the CNN).

4. Gradient Updating and Adaptive Power Allocation Strategies for the Federated Learning over Wireless Backhaul

In line 6 of Algorithm 1, multi-clients should report their local gradient vectors $\mathbf{g}_t^{(l)}$ through the backhaul link with B channel resources at each round. Specifically, each client should design the transmit signal $\mathbf{x}_{l,t}$ to transmit $\mathbf{g}_t^{(l)}$ in (5). In addition, the server should estimate $\hat{\mathbf{g}}_t^{(l)}$ from the received signal \mathbf{y}_t in (5).

4.1. Linear Gradient Estimation for Federated Learning over Wireless Backhaul

To avoid the inter-channel interference over the wireless backhaul link, conventional orthogonal multiple access method with linear precoding is considered in which the wireless resource blocks are orthogonally allocated to each client. Specifically, by letting $\bar{B} = \frac{B}{L}$, which is assumed to be an integer, \mathbf{x}_l can be given as

$$\mathbf{x}_{l,t} = \begin{bmatrix} \mathbf{0}_{\bar{B}(l-1) \times \bar{B}} \\ \mathbf{I}_{\bar{B}} \\ \mathbf{0}_{(B-\bar{B}l) \times \bar{B}} \end{bmatrix} \Psi_{\bar{B} \times \bar{P}} \bar{\mathbf{g}}_t^{(l)}, \quad (9)$$

where $\Psi_{\bar{B} \times \bar{P}}$ is a predefined pseudo-random matrix satisfying the restricted isometry property (RIP) condition [24] and unitary such as:

$$\Psi_{\bar{B} \times \bar{P}}^H \Psi_{\bar{B} \times \bar{P}} = \frac{1}{\|\bar{\mathbf{g}}_t^{(l)}\|^2} \mathbf{I}_{\bar{P}}. \quad (10)$$

Note that $\mathbf{g}_t^{(l)} (\in \mathbb{R}^{P \times 1})$ in (3) is split into multiple \bar{P} dimensional vectors, $\bar{\mathbf{g}}_t^{(l)}$ and each split vector is transmitted through \bar{B} wireless resources.

Then, (5) can be rewritten as:

$$\bar{\mathbf{y}}_{l,t} = \bar{\mathbf{H}}_{l,t} \Psi_{\bar{B} \times \bar{P}} \bar{\mathbf{g}}_t^{(l)} + \bar{\mathbf{n}}_t, \quad (11)$$

where $\bar{\mathbf{H}}_{l,t} = \text{diag}\{h_{l,t}[B(l-1)+1], h_{l,t}[B(l-1)+2], \dots, h_{l,t}[Bl]\}$ and

$$\bar{\mathbf{y}}_{l,t} = \begin{bmatrix} y_t[\bar{B}(l-1)+1] \\ \vdots \\ y_t[l\bar{B}] \end{bmatrix}, \quad \bar{\mathbf{n}}_t = \begin{bmatrix} n_t[\bar{B}(l-1)+1] \\ \vdots \\ n_t[l\bar{B}] \end{bmatrix}.$$

When $\bar{B} \geq \bar{P}$, $\mathbf{g}_t^{(l)}$ can be estimated from (11) by exploiting the linear estimation methods such as zero-forcing or MMSE estimation. That is, ZF estimate of $\mathbf{g}_t^{(l)}$ can be given as:

$$\hat{\mathbf{g}}_t^{(l)} = (\bar{\mathbf{G}}_{l,t}^H \bar{\mathbf{G}}_{l,t})^{-1} \bar{\mathbf{G}}_{l,t}^H \bar{\mathbf{y}}_{l,t}, \quad (12)$$

where $\bar{\mathbf{G}}_{l,t} = \bar{\mathbf{H}}_{l,t} \Psi_{\bar{B} \times \bar{P}}$. When $\bar{B} < \bar{P}$ and $\mathbf{g}_t^{(l)}$ is sparse, compressive sensing approach such as basis pursuit or orthogonal matching pursuit algorithms [25,26] can be applied to estimate $\mathbf{g}_t^{(l)}$.

4.2. Proposed Gradient Updating Method Using Maximal Ratio Combining and Thresholding

From (12), the server can estimate the gradient reported from the l th client, $\hat{\mathbf{g}}_t^{(l)}$. Note that, because the channel gain of the wireless backhaul link is varying over the round during the federated learning process. The ill-conditioned channel with small channel gain may increase the estimation error and distort the gradient information associated with the

l th client. Accordingly, in what follows, we propose two gradient update methods based on the channel gain, $\bar{\mathbf{H}}_{l,t}$.

4.2.1. Gradient Update by Maximum Ratio Combining

Note that the estimate of $\mathbf{g}_t^{(l)}$ is more reliable for larger channel gain. To see this, by considering a simple case with $\bar{B} = \bar{P}$, we can rewrite (12) as:

$$\begin{aligned}\hat{\mathbf{g}}_t^{(l)} &= \mathbf{\Psi}_{\bar{B} \times \bar{P}}^{-1} (\bar{\mathbf{H}}_{l,t}^H \bar{\mathbf{H}}_{l,t})^{-1} \bar{\mathbf{H}}_{l,t}^H \bar{\mathbf{y}}_{l,t} \\ &= \bar{\mathbf{g}}_t^{(l)} + \mathbf{\Psi}_{\bar{B} \times \bar{P}}^{-1} (\bar{\mathbf{H}}_{l,t}^H \bar{\mathbf{H}}_{l,t})^{-1} \bar{\mathbf{H}}_{l,t}^H \bar{\mathbf{n}}_t.\end{aligned}\quad (13)$$

Accordingly, the mean squared estimation error is proportional to $\frac{\sigma_n^2}{\|\bar{\mathbf{H}}_{l,t}\|_F^2}$. Equivalently, the effective SNR can be given as $\frac{\|\bar{\mathbf{H}}_{l,t}\bar{\mathbf{g}}_t^{(l)}\|_F^2}{\sigma_n^2}$. Therefore, when updating the aggregated gradient at the server from $\mathbf{g}_t^{(l)}$, $l = 1, \dots, L$, instead of (8), we can exploit the weighted sum of $\mathbf{g}_t^{(l)}$ as

$$\mathbf{g}_t = \sum_{l=1}^L w_t^{(l)} \hat{\mathbf{g}}_t^{(l)}, \quad (14)$$

where the weight $w_t^{(l)}$ that maximizes the effective output SNR can be derived as:

$$w_t^{(l)} = \frac{\|\bar{\mathbf{H}}_{l,t}\|_F^2}{\sum_{l=1}^L \|\bar{\mathbf{H}}_{l,t}\|_F^2}, \quad (15)$$

which is denoted as the maximum ratio combining (MRC) weights and allows the gradient vector that has undergone a better channel to contribute more to the aggregated gradient at the server. This is because it is more reliable and less-distorted through the wireless backhaul link, as observed from (13). To the best of our knowledge, the gradient update strategy by channel-based MRC in federated learning system with wireless backhaul has not been considered before.

4.2.2. Binary Gradient Update by Thresholding

When the backhaul links for all clients have small channel gain simultaneously, the server may receive severely distorted gradient vectors even though it exploits the MRC strategy, such as (14). Accordingly, we propose a method in which the round associated with all channel having small channel gains is excluded from federated learning. Specifically, if $\sum_{l=1}^L \|\bar{\mathbf{H}}_{l,t}\|_F^2 < \epsilon$, the associated gradient is not updated at the server, where ϵ is a pre-defined constant. Based on the above description, the proposed federated learning process is summarized in Algorithm 2.

Algorithm 2 Proposed federated learning train process.

- 1: Initialize θ_0 based on He normal weight initialization method
 - 2: $\mathbf{g}_0 \leftarrow \mathbf{0}$
 - 3: **for** $t \leftarrow 1$ to T **do**
 - 4: (Clients) $\theta_t^{(l)} \leftarrow \theta_{t-1}^{(l)} + \mathbf{g}_{t-1}$
 - 5: (Clients) Update $\mathbf{g}_t^{(l)}$ from the datasets $\Phi_{tr}^{(l)}$
 - 6: (Clients) Report $\mathbf{g}_t^{(l)}$ to the server via the backhaul link
 - 7: **if** $\sum_{l=1}^L \|\bar{\mathbf{H}}_{l,t}\|_F^2 < \epsilon$ **then**
 - 8: (Server) $\mathbf{g}_t \leftarrow \sum_{l=1}^L w_t^{(l)} \hat{\mathbf{g}}_t^{(l)}$ as in (14)
 - 9: **else**
 - 10: (Server) $\mathbf{g}_t \leftarrow \mathbf{g}_{t-1}$
 - 11: **end if**
 - 12: (Server) Broadcast \mathbf{g}_t to multi-clients
 - 13: **end for**
-

4.3. Adaptive Power Allocation Strategy Based on the Gradient Information

When the transmission power of each client is limited, rather than allocating power equally to all channel resources (equivalently, slots), it is effective to allocate more power on the channel slots carrying specific important information. Note that, when training a deep learning model, the gradient elements with large values imply the large change of weight to decrease the loss function. Accordingly, because $\mathbf{g}_i^{(l)} (\in \mathbb{R}^{P \times 1})$ in (3) is split into multiple \bar{P} dimensional vectors, $\bar{\mathbf{g}}_i^{(l)}$ in (9), each client allocate its transmit power proportionally to the magnitude of $\bar{\mathbf{g}}_i^{(l)}$ in our proposed power allocation strategy. Assuming that $\bar{N} = P/\bar{P}$ is an integer and then, the number of multiple split vectors is given as \bar{N} . The adaptive power allocation strategy can be accomplished by setting:

$$\mathbf{\Psi}_{\bar{B} \times \bar{P}}^H \mathbf{\Psi}_{\bar{B} \times \bar{P}} = \frac{\bar{N}}{\|\bar{\mathbf{g}}_i^{(l)}\|^2} \mathbf{I}_{\bar{P}}. \quad (16)$$

We note that the constraint of (10) allows the equal power to be used when transmitting the split vector $\bar{\mathbf{g}}_i^{(l)}$, while the constraint of (16) allows the power to be used in proportion to the magnitude of $\bar{\mathbf{g}}_i^{(l)}$ at each transmission, exhibiting the same total transmit power as in (10). In addition, the power allocation as (16) has not been considered in the conventional federated learning methods over wireless channels.

5. Experiment Results

To see the validation of the proposed federated learning train strategy discussed in Section 4, we develop the CNN module for handwriting character recognition having the architecture in Figure 2. Specifically, the CNN module has three two-dimensional convolutional layers and the values for the hyperparameters exploited in the computer simulations are summarized in Table 1. Then, the number of elements in the gradient vector $\mathbf{g}^{(l)}$ is given as 5.26×10^4 . The CNN module is shared by three clients connected to the server over the wireless channel. Throughout the simulations, we exploit the handwriting MNIST dataset where $N_{width} = N_{height} = 28$. In addition, three clients are considered and the received SNR at the server is defined as:

$$SNR_{rec} = \frac{\sum_{l=1}^L \sigma_{h_l}^2}{L\sigma_n^2}, \quad (17)$$

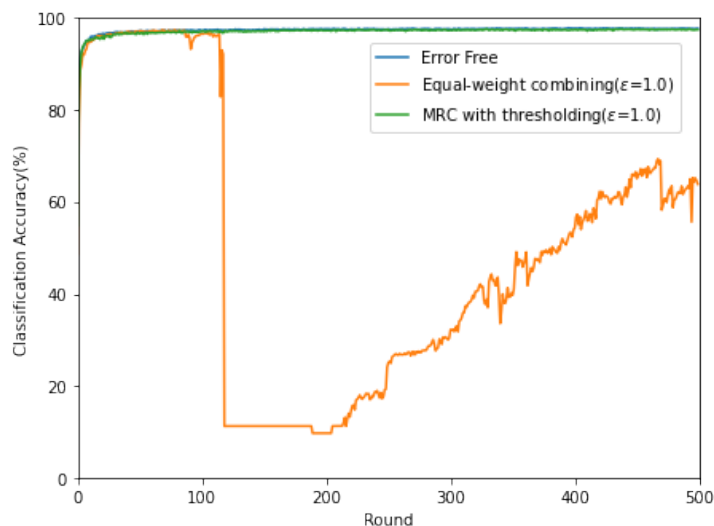
where σ_n^2 is the variance of the AWGN. In addition, we split the gradient vector into multiple vectors having 128 elements (i.e., $\bar{P} = 128$ in (9)).

Table 1. The values for the hyperparameters of the CNN module for handwriting character recognition.

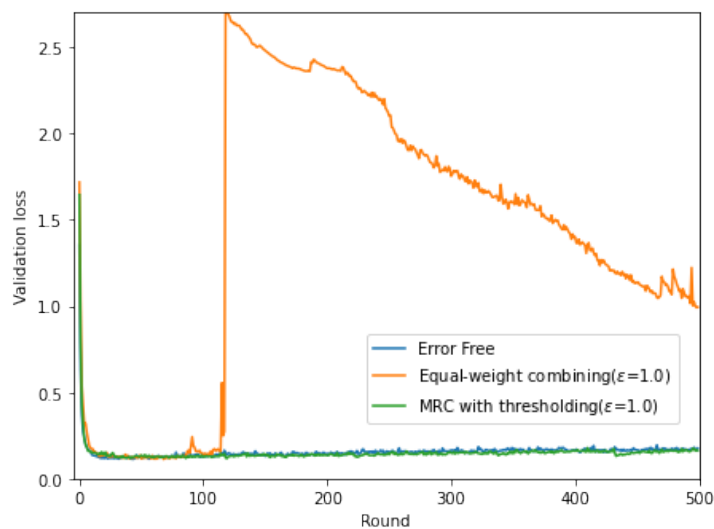
	Values
The number of layers	3
The number of filters at each layer	3
Filter size (The 1st layer), $K_{f11} \times K_{f12}$	(1 × 16)
Filter size (The 2nd layer), $K_{f21} \times K_{f22}$	(16 × 32)
Filter size (The 3rd layer), $K_{f31} \times K_{f32}$	(32 × 64)
Optimizer	ADAM optimizer [27]
Learning rate, η	0.001

In Figure 3 (respectively, Figure 4), we evaluate the classification accuracy and CE loss of the conventional gradient updating method based on the equal-weight combining and the proposed updating method based on MRC, discussed in Section 4.2 for high SNR ($SNR_{rec} = 15$ dB) (respectively, low SNR ($SNR_{rec} = -10$ dB)). For comparison purposes, the performance of the federated learning with error-free backhaul link is also evaluated. Here, the channel gain of each client is set as $\sigma_{h_l}^2 = \{0.3, 1.0, 3.0\}$ and the threshold level in

given as $\epsilon = 1.0$, and this value was experimentally determined. For the local training of the commonly shared CNN module, ADAM optimizer is adopted [27] at each client with a fixed learning rate, $\eta = 0.001$.



(a)

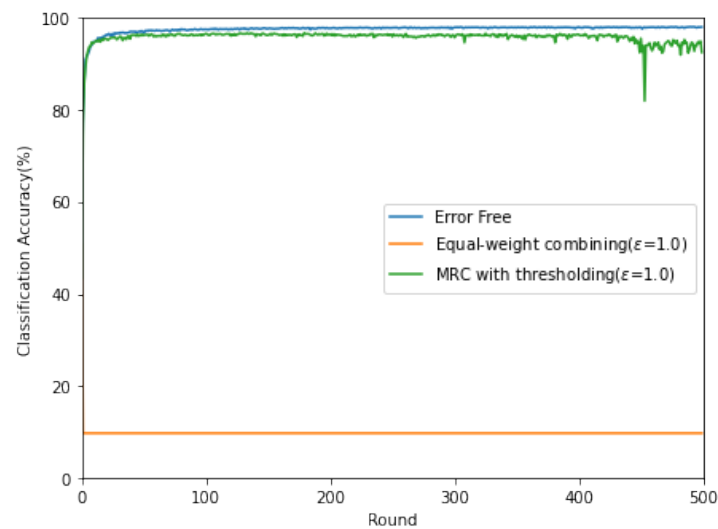


(b)

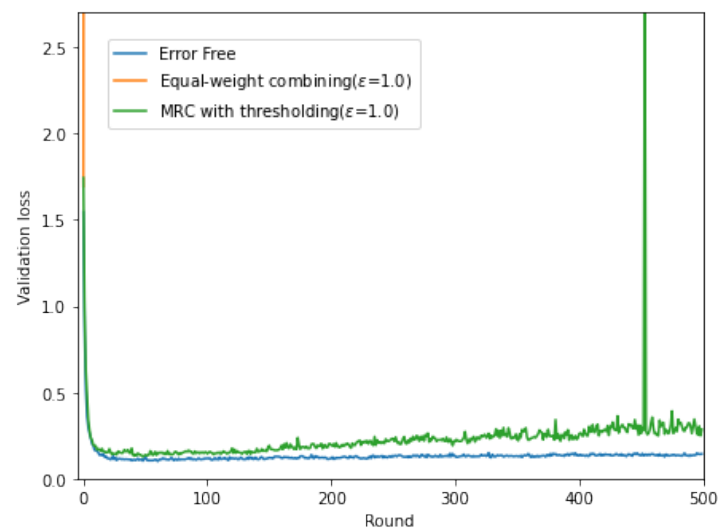
Figure 3. (a) Classification accuracy and (b) CE loss curves at $SNR_{rec} = 15$ dB.

From Figure 3, when the backhaul link is perfect and noise free, the classification accuracy increases in proportion to the rounds and the accuracy up to 0.97 can be achieved. In contrast, due to the channel fading and noise in the wireless backhaul link, training does not proceed stably when the conventional equal-weight combining is exploited. In Round 120, there is a sharp increase at the loss curve from 0.28 to 2.75, resulting in the decrease in the accuracy from 0.92 to 0.11. In contrast, the performance of the proposed updating method based on MRC in Section 4.2 exhibits a similar performance to that with the perfect backhaul link. In Figure 4, it can be found that, for low SNR, the classification accuracy of the equal-weight combining is not improved as the rounds increases and is below 0.15. In addition, the associated CE loss goes to infinity. At low SNR, it is difficult to recover the distortion caused over the wireless backhaul link when transmitting the gradient for model update. Especially, when there is channel distortion, the equal-weight combining

does not reflect the received SNR in the gradient update and fails to train the distributed CNN modules. Interestingly, the updating method based on MRC and thresholding shows unstable peak in the CE loss, but it can avoid the CE loss divergence and improve the classification accuracy as the learning round increases.



(a)

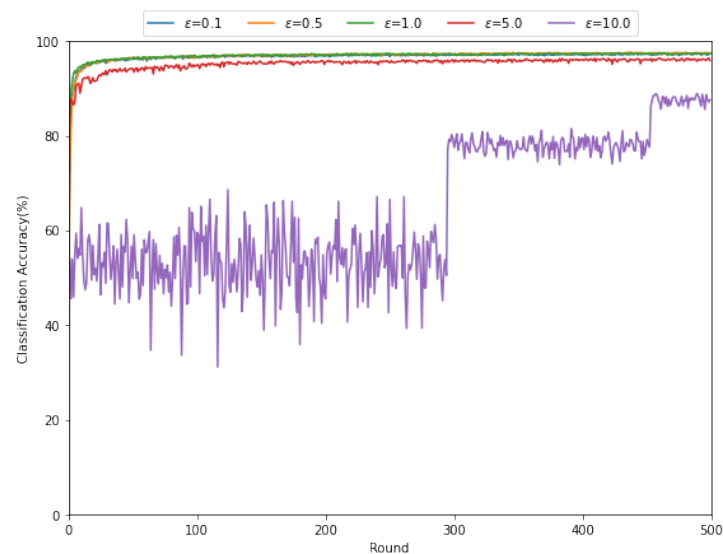


(b)

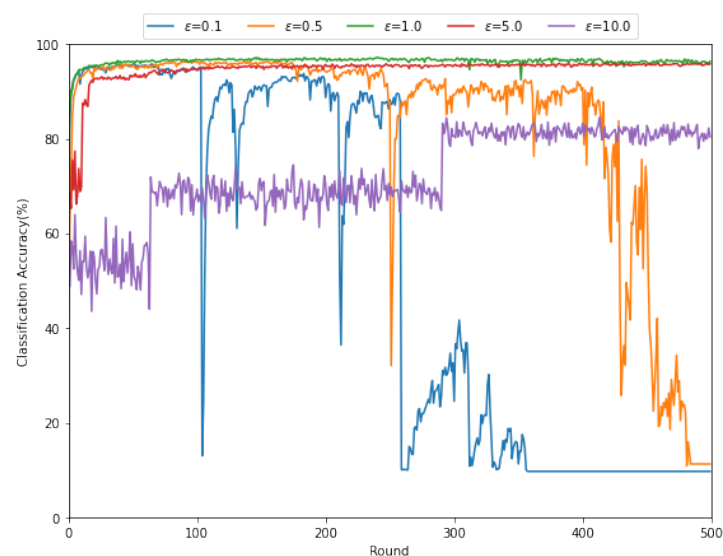
Figure 4. (a) Classification accuracy and (b) CE loss curves at $SNR_{rec} = -10$ dB.

In Figure 5, we evaluate the classification accuracy for various threshold levels ϵ with (a) $SNR_{rec} = 15$ dB and (b) $SNR_{rec} = -10$ dB when the updating method with MRC and thresholding in Section 4.2.2 is exploited. From Figure 5a, at high SNR, the federated learning can be well operated through the gradient updating method with MRC and thresholding, regardless of the threshold levels. However, for $\epsilon = 10.0$, the accuracy does not effectively increase as the learning round increases. That is, for a larger threshold level, more local gradient vectors transferred through the wireless channel can be discarded. In Figure 5b, it can be found that the classification performance is more sensitive to the threshold level at low SNR compared to the high SNR case. Specifically, as ϵ is larger, the federated learning is performed stably. This is also because the gradient update vector containing the amplified noise due to small channel gain can be discarded for large ϵ . We

note that the large ϵ may incur the gradient updating delay, which leads the trade-off between the federated learning performance and the learning delay.



(a)

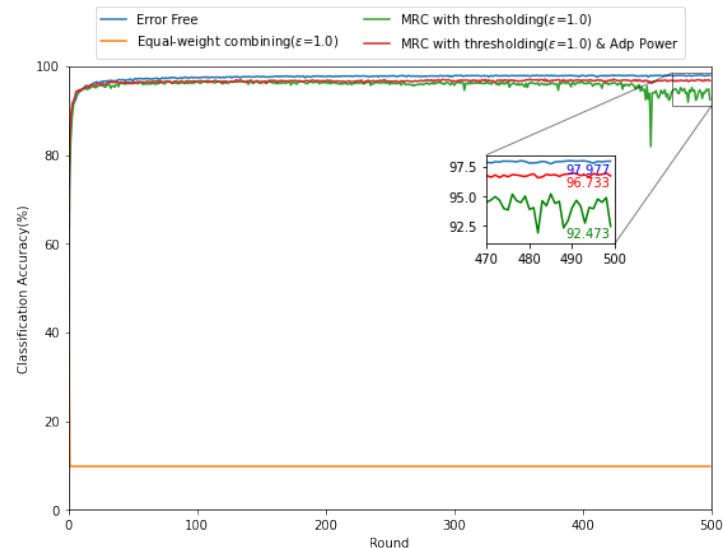


(b)

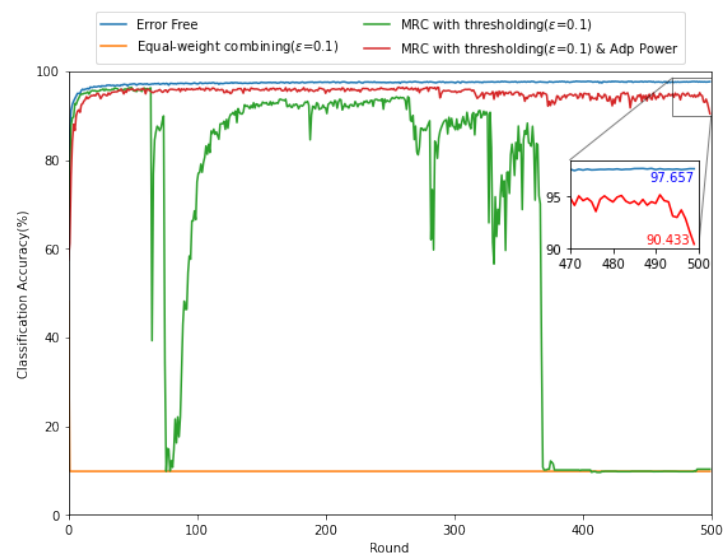
Figure 5. Classification accuracy according to various threshold levels when the updating method with MRC and thresholding in Section 4.2.2 are exploited for (a) $SNR_{rec} = 15$ dB and (b) $SNR_{rec} = -10$ dB.

In Figure 6, to validate the adaptive power allocation strategy in Section 4.3, we evaluate the classification accuracy of various gradient updating methods with/without the adaptive power allocation strategy when the received SNR is low with different threshold levels (i.e., (a) $\epsilon = 1.0$ and (b) $\epsilon = 0.1$). It can be found that the accuracy of the MRC based gradient updating method with $\epsilon = 1.0$ in Figure 6a is more stable compared to that with $\epsilon = 0.1$ in Figure 6b, which coincides with the observation in Figure 5. Interestingly, by exploiting the adaptive power allocation strategy jointly with the MRC based gradient updating method in Figure 6a, the accuracy can be improved by 96.7% and it is comparable to the performance with error-free backhaul link. In addition, from Figure 6b, the adaptive power allocation strategy drastically stabilizes the federated learning performance during

the learning process over wireless channel even for small $\epsilon = 0.1$. Accordingly, the adaptive power allocation strategy improves the trade-off between the federated learning performance and the learning delay due to the threshold level discussed in Figure 5.



(a)



(b)

Figure 6. Comparison of classification accuracy with (a) $\epsilon = 1.0$ and (b) $\epsilon = 0.1$ for $SNR_{rec} = -10$ dB.

In Tables 2 and 3, the confusion matrices for the test dataset are evaluated after the federated learning is completed, where the proposed gradient updating method (Table 2) and the conventional updating method (Table 3) are, respectively, exploited. From Table 2, the proposed gradient updating method shows the classification accuracy of 0.9 or more for all labels. However, from Table 3, the CNN module trained through the conventional gradient updating method over wireless channel misclassifies most test data with specific labels.

Table 2. Confusion matrix for hand writing character recognition of the proposed gradient updating method.

Predicted Label	True Label									
	0	1	2	3	4	5	6	7	8	9
0	0.984	0.001	0.020	0.004	0.005	0.006	0.008	0.003	0.022	0.007
1	0	0.979	0.002	0	0.005	0.001	0.002	0.002	0.001	0.001
2	0	0.004	0.943	0.005	0	0	0	0.015	0.004	0
3	0	0.001	0.006	0.963	0	0.006	0	0	0.007	0.007
4	0	0.001	0.001	0	0.929	0	0.002	0.001	0.005	0.004
5	0	0.001	0	0.019	0.001	0.955	0.005	0.001	0.003	0
6	0.010	0.005	0.003	0	0.016	0.015	0.980	0	0.007	0.001
7	0.001	0.001	0.016	0.007	0.001	0.003	0	0.949	0.007	0.016
8	0	0.007	0.009	0.002	0.001	0.011	0.001	0.003	0.915	0.006
9	0.005	0	0.001	0	0.042	0.003	0.001	0.026	0.029	0.961

Table 3. Confusion matrix for hand writing character recognition of equal-weight combining based gradient updating method.

Predicted Label	True Label									
	0	1	2	3	4	5	6	7	8	9
0	0.010	0.004	0.012	0.004	0.011	0.026	0.010	0.002	0.005	0.001
1	0	0	0	0	0	0	0	0	0	0
2	0.977	0.875	0.961	0.877	0.932	0.777	0.971	0.842	0.919	0.943
3	0	0	0	0	0	0	0	0	0	0
4	0.009	0.085	0.026	0.113	0.039	0.172	0.017	0.126	0.059	0.036
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0.025	0.001	0.002	0.016	0	0	0.022	0	0.020
8	0.004	0.011	0	0.004	0.002	0.026	0.002	0.007	0.017	0.001
9	0	0.001	0	0	0	0	0	0	0	0

6. Conclusions

In this paper, efficient gradient updating strategies are developed for federated learning when distributed clients are connected to the server via a wireless backhaul link. That is, a common CNN module is shared for all the distributed clients and it is trained through the federated learning over wireless backhaul connected to the main server. During the training phase, local gradients need to be transferred from the distributed clients to the server over a wireless noisy backhaul link. To overcome the distortion due to wireless channel fading, an effective SNR maximizing gradient updating method is proposed, in which the gradients are combined such that the effective SNR is maximized at the server. In addition, when the backhaul links for all clients have small channel gain simultaneously, the server may have severely distorted gradient vectors. Accordingly, we propose a binary gradient updating strategy based on thresholding in which the round associated with all channels having small channel gains is excluded from federated learning, which results in the trade-off between the federated learning performance and the learning delay. Due to the channel fading and noise in the wireless backhaul link, training does not proceed stably with the conventional equal-weight combining especially at low SNR. In contrast, the updating method based on MRC and thresholding improves the classification accuracy as the learning round increases by avoiding the CE loss divergence. Finally, we also propose an adaptive power allocation method, in which each client allocates its transmit power proportionally to the magnitude of the gradient information. Note that the gradient elements with large values imply the large change of weight to decrease the loss function. Through the computer simulations, it is confirmed that the adaptive power allocation strategy can improve the trade-off between the federated learning performance and the learning delay due to the threshold level.

Author Contributions: Conceptualization, Y.Y. and J.P.; methodology, Y.Y. and J.P.; software, Y.Y., Y.H. and J.P.; validation, J.P.; formal analysis, J.P.; investigation, Y.Y. and Y.H.; writing—original draft preparation, Y.Y.; writing—review and editing, J.P.; supervision, J.P.; project administration, J.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea, funded by the Ministry of Education (2018R1D1A1B07043786).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

- Zhang, K.; Zuo, W.; Gu, S.; Zhang, L. Learning Deep CNN Denoiser Prior for Image Restoration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
- Hadhrani, E.A.; Mufti, M.A.; Taha, B.; Werghe, N. Transfer learning with convolutional neural networks for moving target classification with micro-Doppler radar spectrograms. In Proceedings of the 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China, 26–28 May 2018; pp. 148–154. [\[CrossRef\]](#)
- Tsakatakis, G.; Aidini, A.; Fotiadou, K.; Giannopoulos, M.; Pentari, A.; Tsakalides, P. Survey of Deep-Learning Approaches for Remote Sensing Observation Enhancement. *Sensors* **2019**, *19*, 3929. [\[CrossRef\]](#) [\[PubMed\]](#)
- Kim, Y.; Moon, T. Human Detection and Activity Classification Based on Micro-Doppler Signatures Using Deep Convolutional Neural Networks. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 8–12. [\[CrossRef\]](#)
- Rieke, N.; Hancox, J.; Li, W.; Milletari, F.; Roth, H.R.; Albarqouni, S.; Bakas, S.; Galtier, M.N.; Landman, B.A.; Maier-Hein, K.; others. The future of digital health with federated learning. *NPJ Digit. Med.* **2020**, *3*, 1–7. [\[CrossRef\]](#) [\[PubMed\]](#)
- Chaabene, S.; Bouaziz, B.; Boudaya, A.; Hökelmann, A.; Ammar, A.; Chaari, L. Convolutional Neural Network for Drowsiness Detection Using EEG Signals. *Sensors* **2021**, *21*, 1734. [\[CrossRef\]](#) [\[PubMed\]](#)
- Nafea, O.; Abdul, W.; Muhammad, G.; Alsulaiman, M. Sensor-Based Human Activity Recognition with Spatio-Temporal Deep Learning. *Sensors* **2021**, *21*, 2141. [\[CrossRef\]](#) [\[PubMed\]](#)
- Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol. (TIST)* **2019**, *10*, 1–19. [\[CrossRef\]](#)
- Konečný, J.; McMahan, B.; Ramage, D. Federated optimization: Distributed optimization beyond the datacenter. *arXiv* **2015**, arXiv:1511.03575.
- Bottou, L. Large-scale machine learning with stochastic gradient descent. In Proceedings of the COMPSTAT'2010, Paris, France, 22–27 August 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 177–186.
- Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; Chandra, V. Federated learning with non-iid data. *arXiv* **2018**, arXiv:1806.00582.
- Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H.B.; Patel, S.; Ramage, D.; Segal, A.; Seth, K. Practical secure aggregation for privacy-preserving machine learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 1175–1191.
- Konečný, J.; McMahan, H.B.; Yu, F.X.; Richtárik, P.; Suresh, A.T.; Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv* **2016**, arXiv:1610.05492.
- Amiri, M.M.; Gündüz, D. Federated Learning Over Wireless Fading Channels. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 3546–3557. [\[CrossRef\]](#)
- Amiri, M.M.; Gündüz, D. Machine Learning at the Wireless Edge: Distributed Stochastic Gradient Descent Over-the-Air. In Proceedings of the 2019 IEEE International Symposium on Information Theory (ISIT), Paris, France, 7–12 July 2019; pp. 1432–1436. [\[CrossRef\]](#)
- Jeon, Y.S.; Amiri, M.M.; Li, J.; Poor, H.V. A Compressive Sensing Approach for Federated Learning Over Massive MIMO Communication Systems. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 1990–2004. [\[CrossRef\]](#)
- Chen, M.; Yang, Z.; Saad, W.; Yin, C.; Poor, H.V.; Cui, S. A Joint Learning and Communications Framework for Federated Learning Over Wireless Networks. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 269–283. [\[CrossRef\]](#)
- Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [\[CrossRef\]](#)
- Zhu, G.; Wang, Y.; Huang, K. Broadband Analog Aggregation for Low-Latency Federated Edge Learning. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 491–506. [\[CrossRef\]](#)
- McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.

21. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
22. Xiong, J.; Bi, R.; Zhao, M.; Guo, J.; Yang, Q. Edge-Assisted Privacy-Preserving Raw Data Sharing Framework for Connected Autonomous Vehicles. *IEEE Wirel. Commun.* **2020**, *27*, 24–30. [[CrossRef](#)]
23. Kang, D.; Ahn, C.W. Communication Cost Reduction with Partial Structure in Federated Learning. *Electronics* **2021**, *10*, 2081. [[CrossRef](#)]
24. Candès, E.J. The restricted isometry property and its implications for compressed sensing. *C. R. Math.* **2008**, *346*, 589–592. [[CrossRef](#)]
25. Kim, S.; Yun, U.; Jang, J.; Seo, G.; Kang, J.; Lee, H.; Lee, M. Reduced Computational Complexity Orthogonal Matching Pursuit Using a Novel Partitioned Inversion Technique for Compressive Sensing. *Electronics* **2018**, *7*, 206. [[CrossRef](#)]
26. Chen, S.; Donoho, D. Basis pursuit. In Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 31 October–2 November 1994; Volume 1, pp. 41–44.
27. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.