Journal of Network Intelligence ©2024 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **9**, Number **1**, February **2024**

FedQT: A Federated Learning Model Compression Algorithm Based on Quicksort and Top_Avg Pruning

Jin-Quan Zhang

College of Computer Science and Engineering
Shandong University of Science and Technology
Qingdao, 266590, China
tjzhangjinquan@126.com

Hui-Xin Xu

College of Computer Science and Engineering
Shandong University of Science and Technology
Qingdao, 266590, China
xhxin949@163.com

Yun-Shen Ma

College of Computer Science and Engineering
Shandong University of Science and Technology
Qingdao, 266590, China
1151490933@qq.com

Li-Na Ni*

College of Computer Science and Engineering
Shandong University of Science and Technology
Qingdao, 266590, China
nln@163.com

*Corresponding author: Li-Na Ni

Received March 17, 2023, revised May 25, 2023, accepted August 14, 2023.

Abstract. *With the advent of the fourth industrial revolution, data ushered in explosive growth. Federated learning can protect users' privacy and raw data from being known by third parties. Its client data is only trained locally and will not be uploaded to the server. However, the need to continuously transmit between the server and the client in federated learning increases communication overhead and communication delay, and the consumption of resources is very large. In order to improve the communication efficiency of the federation, we proposed an FedQT federated learning model compression algorithm based on pruning and quantization. Firstly, the fast sorting is used to preprocess the trained parameters. Secondly, the middle part of the processed parameters is averaged to obtain the pruning threshold* $avg(\omega)$*. Then, according to the obtained new threshold standard, the model performs Top_Avg pruning based on median comparison. Finally, the* $k-means$ *algorithm is used to cluster the parameters for further compression. In the simulation experiment, FedQT is compared with uncompressed and other compression algorithms, and the time complexity is analyzed. It is proved that FedQT can compress the trained model parameters to improve its communication efficiency while reducing the time overhead. The model can be compressed by more than one thousand times, and the model accuracy can be maintained above 97%, which is enough to prove that FedQT algorithm has good performance in the compression framework of federated learning.*

Keywords: Federated learning, Model compression, Sparse, Quantization

1. **Introduction.** In the era of big data, more and more mobile devices into our lives, whether it is conventional mobile phones, computers or for disease detection and early warning of health wearable devices are infiltrated into all aspects of people's lives [1]. From mobile phone browsing records to health detection information in wearable devices to application usage records in computers. All aspects of human life can be recorded by electronic data, but the privacy protection problem is becoming more and more serious [2]. For example, in different medical institutions, for different diseases have their own patients and cases, if you want to study the disease by accessing the disease information of such patients, it will inevitably lead to privacy leakage of user data [3]; what's more, an attacker can withdraw all the privacy information of a patient through the only partial data feature. In the case of more and more IoT devices, how to achieve secure transmission between devices is a major problem. Some scholars use identity authentication and other protocols [4, 5] to achieve, but how to set a reasonable protocol is difficult, and there are still some security risks.

Federated learning [6] was first proposed by Google in 2016. The original purpose was to solve the problem of data islands. After verifying its good performance on privacy protection, it was widely used in many large amounts of data and privacy protection in complex client scenarios. It can also be combined with identity authentication technology to achieve better protection [7]. Federated learning has stricter supervision on privacy. Different from the traditional way that the server needs to collect all client data when training the model, federated learning framework only needs to transmit locally trained model parameters between the client and the server to reduce the amount of transmission and avoid the transmission of all local data, thereby reducing the risk of data privacy leakage during transmission [8, 9, 10]. However, frequent iterative training makes federated learning very dependent on the transmission between server and client. Especially for the local clients with limited computing power and transmission bandwidth, the huge resource consumption caused by the transmission process has become a major factor hindering its development.

In order to solve such problems, this paper mainly makes the following contributions in pruning, and uses quantization to further compress the volume of the model after pruning:

1. We propose an algorithm based on QSTop_Avg, the main content of which is to sort the parameters by quick sorting, and take the value of the middle part after sorting to obtain a more scientific pruning threshold;
2. On the basis of quick sorting, a pruning standard of $avg(\omega)$ is proposed, and the average value of the middle part after sorting is calculated to obtain a fairer pruning threshold. The pruning method of median value comparison is used to further reduce the energy consumption in the pruning process and improve the training efficiency of the model.

2. **Related Works.** Model compression in federated learning has incomparable advantages as a better optimization method, and this section will introduce the research status of domestic and foreign scholars and related reserve knowledge.

2.1. **Research status.** The model parameters of client transmission in federated learning are very large and complex. If this is not improved, it will have a great impact on the efficiency of federated learning. With the deepening of research, the drawbacks of federated learning requiring a lot of resource consumption have also emerged. How to effectively improve the efficiency of federated learning without reducing the performance of the model has become a hot research topic in current federated learning [11, 12, 13].

Sattle et al. [14] proposed an efficient federated distillation method which can reduce the amount of communication necessary to achieve fixed performance targets by more than two orders of magnitude when compared to FD, and by more than four orders of magnitude when compared to parameter averaging based techniques like federated averaging. Liu et al. [15] proposed an adaptive pruning and analyzed the convergence rate and learning delay of FL system mathematically. Then, by jointly optimizing the pruning ratio and spectrum allocation, the optimization problem is formulated, and the algorithm has a good performance in the wireless federated learning scheme. Young et al. [16] designed a rate distortion framework to quantize and optimize the weights after training to improve compression at any quantization bitrate.

In addition to the above improved methods, sattler et al. [17] proposed sparse binary compression (SBC), which combines the existing communication delay and gradient sparse techniques with novel binarization methods and optimal weight update coding to push the compression gain to a new limit. Sattler et al. [18] proposed that STC extends the existing gradient sparse compression technology, it is a communication protocol that compresses upstream and downstream communications through sparseness, networking, error accumulation, and optimal golomb coding. Can achieve more efficient communication.

In summary, the purpose of model compression is to reduce the size of the model, improve the computational efficiency and communication efficiency of the model, while maintaining the performance and generalization ability of the model. In federated learning, the methods of optimizing the volume size usually include: model compression, knowledge distillation, low-rank decomposition, etc. Knowledge distillation [19] uses the knowledge of larger models (teacher models) to guide the training of smaller models (student models), but traditional knowledge distillation often has the problem of relatively low learning efficiency of student models. Low-rank decomposition [20] refers to a sparse convolution kernel matrix that combines dimension and low-rank constraints, but its matrix decomposition operation cost is high, and layer-by-layer decomposition is not conducive to global parameter compression, and the convergence efficiency is low, so it is gradually forgotten. However, no matter which compression method is used, more computing resources and time costs are needed to maintain high accuracy. Some methods also have a negative impact on the accuracy of the model.

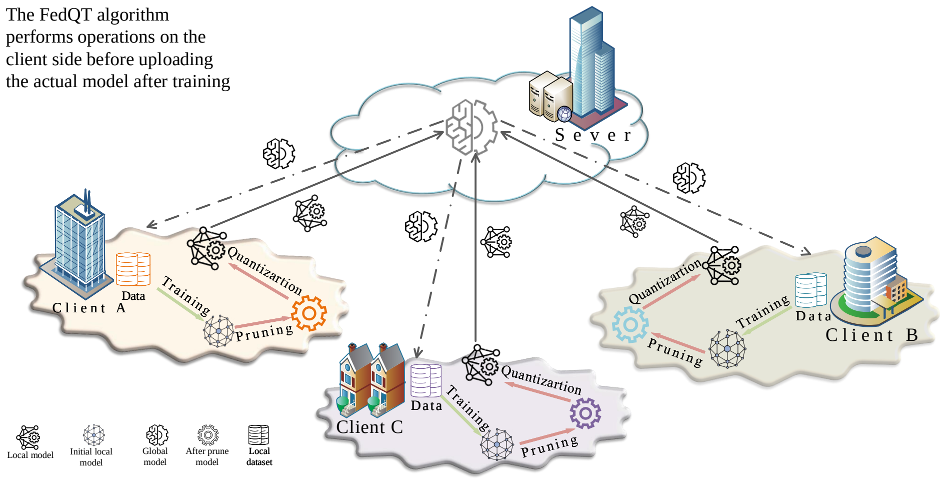
The FedQT algorithm performs operations on the client side before uploading the actual model after training

Figure 1. FedQT Algorithm in Federated Learning

In order to solve the problem of time consumption and accuracy loss in model compression. This paper proposes FedQT algorithm. After the fast sorting operation of the parameters, the average value of the parameters in the middle part is selected as the standard of the pruning threshold. According to the threshold, the Top_Avg pruning algorithm based on median comparison is used to prune the hierarchical parameters, and then the clustering algorithm is used to quantify the pruning parameters. The parameters are mapped to the centroids of different clusters, and the model parameters are compressed while maintaining accuracy, thereby effectively reducing the amount of parameter transmission and improving the efficiency of federated learning. The application of FedQT algorithm in federated learning is shown in Figure 1.

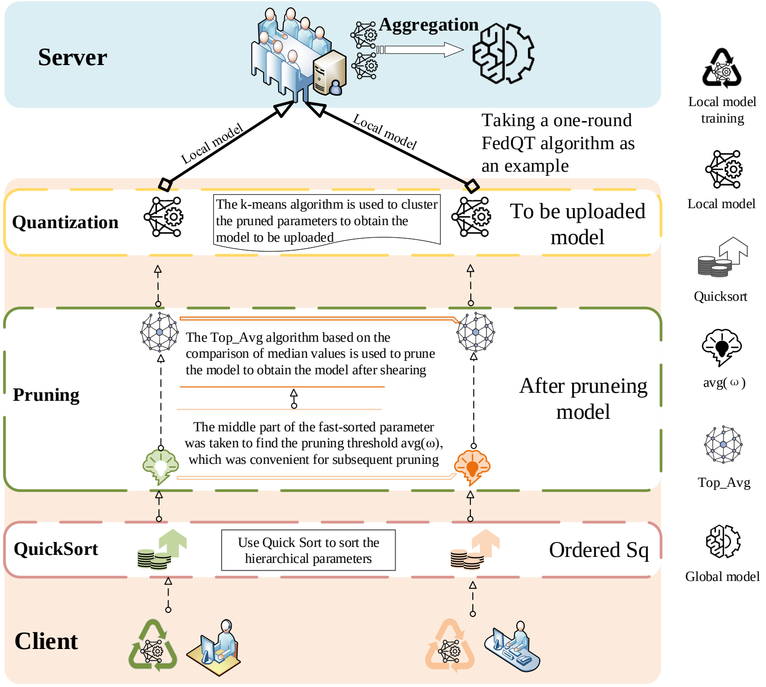
Server
Aggregation
Local model
Local model
Taking a one-round FedQT algorithm as an example
Quantization
The k-means algorithm is used to cluster the pruned parameters to obtain the model to be uploaded
To be uploaded model
Pruning
The Top_Avg algorithm based on the comparison of median values is used to prune the model to obtain the model after shearing
The middle part of the fast-sorted parameter was taken to find the pruning threshold avg(ω), which was convenient for subsequent pruning
After pruneing model
QuickSort
Use Quick Sort to sort the hierarchical parameters
Ordered Sq
Client
Local model training
Local model
Quicksort
avg(ω)
Top_Avg
Global model

Figure 2. FedQT Algorithm Structure Diagram

As shown in Figure 2, under the premise of ensuring model performance and data privacy, our FedQT algorithm first sorts the hierarchical parameters quickly, and uses the sorted sequence to obtain the average value of the middle part using the Top-k algorithm. The mean value is the pruning threshold $avg(\omega)$. On the basis of order, we use the idea similar to binary search to perform Top_Avg pruning operation, and finally use the traditional k-means algorithm to quantify. The entire FedQT compression process can reduce the size and computational burden of the model as much as possible. The focus of this paper is to propose a new pruning threshold standard and pruning method to select the weight in the pruning process more fairly. The optimization of the quantization part will be carried out in the subsequent research, which is not repeated here.

2.2. **Preparation work.** This section will give a brief overview of the relevant knowledge and define of federated learning model compression.

2.2.1. *Related definitions.* Pruning is a common neural network model compression technique, which is also widely used in federated learning. It can prune the model by removing the weight parameters that have little effect on the aggregation results, thereby reducing the size and computational complexity of the model. It also helps to avoid problems such as over-fitting.

There seem to be many pruning algorithms, but pruning the network model is recognized as an effective pruning method. The standard pruning process, such as Figure 3 (left 1), is an extended pruning process.

It mainly includes three parts: training, pruning and trimming.

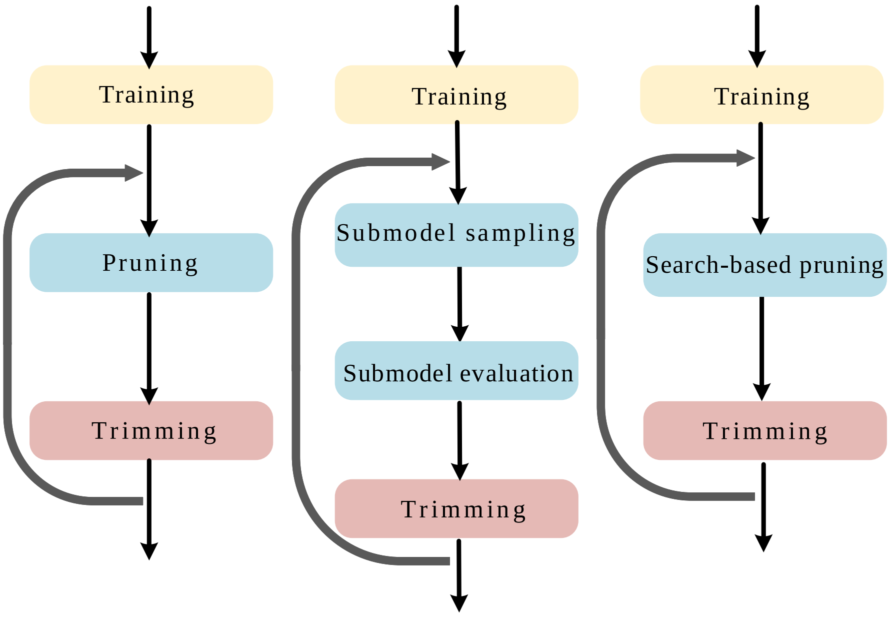

FIGURE 3. Pruning Process

(1)**Training**: That is training the network model;

(2)**Pruning**: Various pruning algorithms such as fine-grained pruning, vector pruning, kernel pruning and filter pruning can be performed here. Note that the network model structure needs to be evaluated after general pruning;

(3)**Trimming**: This is a necessary step to restore the expression ability of the model affected by the pruning operation.

Quantization is to further optimize the model parameter pruning. The conventional quantization generally uses the clustering algorithm to quantify the model of each tensor to avoid the tilt of the parameters after pruning, and use the centroid of each cluster to more fairly reflect the distribution of tensor values, thereby reducing the occupation of memory space and making the distribution of parameters more fair and representative. It should be noted that pruning and quantization may reduce the accuracy of the model, so a balanced choice between accuracy and model size is needed. In this paper, the traditional clustering algorithm is used to quantify. The specific work of quantization in model compression is the direction of our future research.

2.2.2. *Equation description.* This section explains some definitions and related formulas involved in the proposed model compression algorithm. S_t is a subset of m clients, randomly selected in round t, and the subscript m refers to the client of this subset in round t (the number of clients in $|S_t| = m$ is controlled by element C). In this paper, (w_t - w_{t+1}, m) is called Δw, where $\Delta w = \{\Delta w^1, \Delta w^2, \Delta w^3, \ldots, \Delta w^m\}$ is the weight and deviation of the neural network model used for training, and they are used to pass the update message of each client to the server.

Definition 2.1. *Model parameter update* [21]*:ω represents the parameters of the FL model, the data set is distributed on K clients, and the data size of each client in federated learning is n_{k}. The initial global model is ω^0 expressed as the model parameters after participating in the client training. The new weights $\omega^{\mathrm{t+1}}$ are calculated according to Equation* (1)*, and then the updated model parameters are used for the next round of training.*

$$w^{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_k^{t+1} \tag{1}$$

$$w \leftarrow w - \eta \nabla \ell(w; b) \tag{2}$$

If the client is selected by the server, the data set B is divided, and then the E round training is performed locally according to Equation (2)*, and the model parameters after the client completes the training are uploaded to the server.*

In the pruning algorithm, a good pruning threshold can better reflect the fairness of pruning. In this paper, a Top_Avg algorithm and a new parameter of avg(ω) is proposed.

Definition 2.2. *Pruning threshold: According to the values in each layer of tensor, it is sorted into three parts, and the average value of the middle part is used as the parameter avg(ω).*

$$\mathrm{avg}(\omega) = \frac{\sum Top\_z - \sum Top\_(z - med)}{med} \tag{3}$$

In the above Equation (3)*, avg(ω) can be obtained, where z in Top_z can be taken as 0.3n, that is, not only the first z values are obtained, but the middle part of med is averaged to represent the weight level of the parameters in this layer.*

Definition 2.3. *Top_Avg pruning: After obtaining the mean avg(ω) of the parameters of the model, it means that the Top_Avg algorithm has a standard to measure whether a parameter needs to be pruned. In order to more clearly describe the pruning process of Top_Avg in this paper, the pruning process is formulated, as shown in Equation* (4)*:*

$$Top\_Avg(\omega) = \begin{cases} \omega_i, & if\ \omega_i > avg(\omega) \\ 0, & if\ \omega_i < avg(\omega) \end{cases} \tag{4}$$

The pruning module is applied to each layer to determine whether the weight is greater than the obtained avg(ω), which allows the pruning and sparse processing of parameters to be regarded as the parameters selected by the user. Each tensor can be represented by w_i (where i is the i-th value in a tensor).

3. System model.

In this section, we will explain the structure of our proposed FedQT algorithm. The detailed steps in the algorithm are explained.

3.1. FedQT Framework.

In the pruning and quantization process summary, the smaller weights in the model are usually pruned and the model parameters are further quantified, thereby reducing the number and volume of parameters in the model, thereby reducing the storage and calculation costs of the model.

On the basis of traditional pruning, a QSTop_Avg model pruning algorithm is proposed. The core idea is to propose a new pruning threshold standard: $avg(\omega)$ based on the introduction of fast sorting. The federated learning model compression algorithm based on QSTop_Avg is mainly to prune the model that has been trained by the client in the local model client. After the subsequent quantization operation, the weight distribution in the response tensor is more reasonable and the volume of the model is changed.

Figure 4 takes the federated learning model of two clients as an example to briefly summarize the application of QSTop_Avg algorithm in federated learning. The data of different clients can get a model parameter after local training. The FedQT algorithm is to further compress the processed model parameters. Firstly, the trained model parameters are quickly sorted, and the required values are selected after the sorting is completed. The selected values are used to obtain the required threshold: $avg(\omega)$, obtained threshold is compared with the data (as shown in the orange part of Figure 4). After that, the k-means algorithm is used to quantify the model to ensure fairness and further compress the volume of the model. The pruned and quantized model can achieve better compression rate, which can not only reduce the efficiency of the model transmission process, but also simplify the subsequent operation of the model and reduce resource consumption.

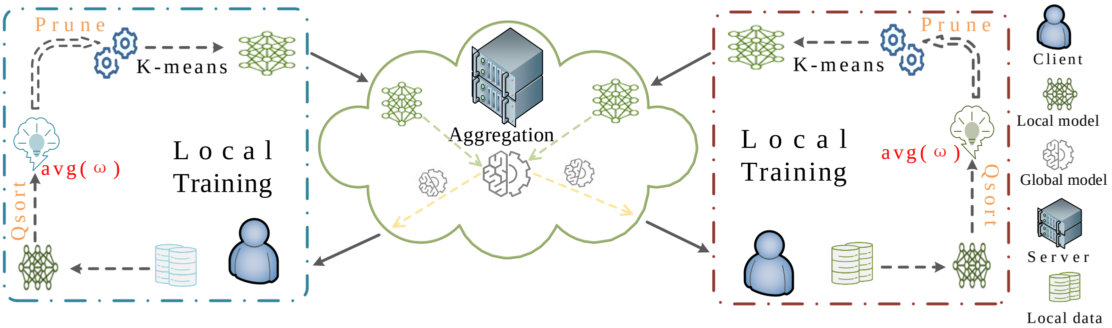

FIGURE 4. QSTop_Avg Algorithm Farmework

3.2. **Arithmetic statement.** The FedQT algorithm compresses the model volume by pruning and quantifying the weight parameters. Its main operation exists in the client, regardless of the problem of device heterogeneity and data heterogeneity, focusing on improving the communication efficiency by reducing the model volume.

In this paper, we propose a new federated learning model compression algorithm(FedQT) based on QSTop_Avg pruning. First, after quickly sorting the client parameters, we take the *med* values between them, and use the Equation (3) to find the corresponding pruning threshold $avg(\omega)$, which is used as a measure of whether the parameters need to be pruned. In addition, there is a need for one-by-one comparison in the pruning process. In the case of consuming a lot of time resources, the median comparison method is used to reduce the time overhead of the model pruning process. The fundamental idea is similar to the binary search method, but it is not exactly the same, in general. That is a new standard for calculating the pruning threshold is proposed in the FedQT algorithm, which further guarantees the accuracy of pruning. The QSTop_Avg algorithm in FedQT is shown in Algorithm 1:

FedQT algorithm is mainly used for the optimization of model compression in federated learning, and the specific process is represented by Algorithm 1. The focus of this paper is to innovate and optimize the pruning algorithm. The specific content of the quantization is not the focus of this paper, so it is not in the postscript. The algorithm proposed in this paper is aimed at the weight parameters of the hierarchy. Therefore, after obtaining the hierarchy parameters, the parameters are processed by quick sorting, and then the average value of the middle *med* positions is obtained. The value is used as the threshold for pruning, and the Top_Avg standard is used for pruning. Experiments show that the FedQT algorithm has better performance in federated learning model compression.

Algorithm 1: QSTop_Avg: model pruning algorithm based on Top_Avg

```
Input: weight parameter $\omega$
Output: after prune weight parameter $\omega$
procedure Pruning
  Set the temporary parameter $n$, set $S_q$, the number of matrix layers $CN$
for ($CN\,! = 0$) do
    $S_q \leftarrow QSort.S_q$
    $avg(\omega) = \frac{\sum Top\_z - \sum Top\_(z-med)}{med}$
    $n \leftarrow length.S_q$
    for ($i \leqslant n$) do
        if ($\omega_i < avg(\omega)$) then
            $\omega_i$=0
        $i++$
    $CN--$
end procedure
```

4. **Algorithm design.** In this section, we will introduce our proposed FedQT algorithm in detail. We divide it into two parts: pruning based on QSTop_Avg and the overall FedQT algorithm, and explain the algorithm's steps.

4.1. **Model pruning.** Model pruning is a technique to compress model size and improve model efficiency by reducing parameters in the model. In federated learning, model pruning can also be applied to reduce the computational and storage resource consumption of the model on local devices, thereby improving the efficiency and scalability of the federated learning algorithm.

The traditional Top-k sparse algorithm uses the k value as the standard for pruning, so the local sorting method will accelerate the efficiency of the algorithm. In the process of pruning the parameters of the hierarchy, if only the first k weights are taken, it is possible to ignore the influence of some weights on the final model aggregation, and the selection of k values also requires a lot of experimental support. Considering that in the parameters of each layer, the average value of the middle part is more representative of its own level, the pruning based on the threshold value of the middle part can better reflect the fairness of pruning. After sorting, the med values of the middle part are found, and the value of $avg(\omega)$ is obtained. Using the Top_Avg algorithm to prune, the effect will be more balanced.

Based on the idea of divide-and-conquer method, fast permutation was invented by british computer scientist Tony Hoare. Quick sorting is an efficient global sorting algorithm. The core idea is to divide a sequence into two subsequences, where all elements of one subsequence are smaller than those of the other subsequence, and then recursively sort the two subsequences.

Table 1. Comparison of Time Complexity

Algorithm	**Time Complexity**
Quick sort	$O(nlogn)$
Heap sort	$O(nlogk)$
Bubble sort	$O(nk)$

In order to obtain better pruning criteria, FedQT algorithm needs the value of the middle part of the parameter to set the pruning threshold. Therefore, it is necessary to obtain the maximum value of the first $n/3$ and the value of the first $2n/3$ (n is the total number of parameters in this layer) to take the middle part, and the time complexity will be lower on the basis of orderly parameters. The advantages of fast sorting are fast speed, simple implementation and suitable for various data types and data sizes, quick sort is introduced here.

In terms of time complexity, the performance of fast sorting time complexity and bubble sorting time complexity is affected by the size of k and $logn$. k is the top k values in n, in the worst case, k and n will be equal, at this time $2^k > n$ that $logn < k$ that time complexity $O(nlogn) < O(nk)$. Through simple derivation and Table 1, it can be proved that the time complexity of fast sorting in general state is better than that of bubble sorting, and FedQT algorithm needs to use median comparison for pruning in the future, so fast sorting is more suitable for this scenario.

5. **FedQT algorithm.** In order to ensure the fairness in the pruning process, this paper proposes a new measure of model pruning: Top_Avg. After obtaining the value of the middle part of the parameter, it can be calculated to obtain a more fair average $avg(\omega)$. Next, the small weight connection is pruned: the parameters whose ownership weight is lower than the threshold $avg(\omega)$ are pruned; after pruning, the parameters are quantified.

Algorithm 2: k-means [22]

```
procedure Quantization
Input: parameter $\omega$
Output: the divided clusters $C = C_1, C_2, ..., C_k$
$C = \emptyset (1 \leqslant i \leqslant k)$
for $(j = 1, 2, ..., m)$ do
    Calculate the Euclidean distance $d_{ji}$ between sample $x_i$ and mean solar time vector $\mu_i$
    $\lambda_j = argmin_{i \in 1,2,...,k} d_{ji}$
    $C_{\lambda_j} = C_{\lambda_j} \cup x_j$
for $(i = 1, 2, ..., k)$ do
    $\mu'_i = \frac{\sum_{x \in (C_i)} x}{|(C_i)|}$
    if $\mu_i ! = \mu'_i$ then
        $\mu_i = \mu'_i$
    keep $\mu_i$
end procedure
```

The value of a tensor can be converted into multiple clusters, and the centroid of these clusters is used as the best representative to avoid parameter tilt. The quantization here uses the k-means [22] clustering algorithm, formulaic expression is shown in Equation (5).

$$\mathrm{k-means}(\Delta\omega) = \{\forall \omega_i \in \Delta\omega_m, \text{ let } \omega_i \leftarrow c_j\} \tag{5}$$

In Equation (5), c is used to represent the centroid of the cluster, and c_j is used to represent the jth predicted centroid. The traditional k-means algorithm is used to quantify the parameters after pruning. The euclidean distance is used to calculate the distance between the sample and the mean vector. The k-means algorithm is shown in Algorithm 2. In this paper, the FedQT algorithm using pruning and quantization is shown in Algorithm 3.

Algorithm 3: FedQT: a federated learning model compression algorithm based on quicksort and Top_Avg pruning

```
procedure Compression
Input:  parameter $\omega$, centroids $c_j(|c_j|$ means that we have four centroids)
Output:  After pruning and quantization weight parameter $\omega$
 Pruning
      $\omega \leftarrow QSTop\_Avg(\omega)$
 Quantization
      $\omega \leftarrow k$ - $means(\omega)$
 for each Centroid of $\omega_{\mathrm{m}} \in \omega$ do
    $\omega_i \leftarrow c_j$
end procedure
```

The QSTop_Avg pruning algorithm is combined with the k-means quantization algorithm to prune the model parameters trained by the local client to prevent the tilt of the weight parameters. The first half of the algorithm uses our newly proposed pruning method and the threshold of pruning. On this basis, pruning is carried out to achieve more scientific pruning, and some weight values with relatively small influence but not negligible are retained. In the second half, the traditional k-means quantization method is used to quantify the parameters that have been pruned. Based on the center c_j of 4, better results can be obtained. After the compression processing of the FedQT algorithm, it can be proved by experiments that our algorithm can ensure that the loss of model parameters is as small as possible. In the case, the model volume is compressed to improve the efficiency of federated learning.

6. **Experiment.**

In the following, we verify the effect of our algorithm in federated learning by setting up the simulation experiment and comparing our proposed FedQT with the better performance federated learning compression algorithm.

6.1. **Experimental preparation.**

In this section, the effect of the proposed FedQT algorithm in federated learning model compression is verified by simulation experiments, and the results are analyzed and introduced to prove the effectiveness of the FedQT algorithm.

Table 2. Parameter Configuration

Parameter	Value	Description
C	10%	participation in training client ratio
B	16	client local training data batch size
E	10	client local training times
C_j	4	The number of centroids in quantization
σ	5	Dirichlet distribution parameter

In order to verify the effect of the QSTop_Avg algorithm proposed in this paper, a standardized data set MNIST [23] and a fundus retinal OCTMNIST data set [24] were used in the simulation experiment. The FedQT algorithm was compared with FedAvg [25] and Fedzip [26] algorithms, and the data set was processed with a data processing method based on dirichlet distribution [27]. The accuracy of the parameters after pruning and the loss value after compression are discussed. In order to control the variables to better verify the feasibility of the test, the equipment heterogeneity problem in the federation is not considered, that is the computing power and communication efficiency between each client are the same. The meaning and value of the parameters used in the experiment are shown in Table 2.

6.2. **Analysis of experimental results.** In this section, the FedQT algorithm is simulated and compared with other algorithms. The CNN neural network model is used, and the standard data set MNIST and the fundus retina-related data set OCTMNIST are selected for experiments and analysis.

The FedQT algorithm can compress the model volume after pruning, and the single-machine simulation experiment is used for comparison. In federated learning, FedAvg allocates weights according to the local data volume of each client for federated learning, which is a classical federated averaging algorithm. Fedzip uses three processes of pruning, quantization and coding to compress the model. Although it has better compression efficiency, it also has accuracy loss.

Table 3. Comparison of Algorithms on MNIST and OCTMNITS

Dataset	**Methods**	**Convergence Speed**	**Train**	**Test**	**Loss**	**Compressibility**
OCTMNIST	FedAvg	baseline	98.74	97.52	0.015	1×
	Fedzip	same	98.16	96.77	0.033	1065×
	FedQT	slightly high	98.52	97.03	0.024	1032×
MNIST	FedAvg	baseline	98.63	96.41	0.137	1×
	Fedzip	same	98.27	96.03	0.231	1065×
	FedQT	slightly high	98.43	96.23	0.164	1032×

The FedQT algorithm performs well on model compression (as can be seen from Table 3). The accuracy on MNIST training set and test set is slightly lower than the uncompressed FedAvg algorithm, but it performs better for Fedzip, which also applies the compression algorithm. In contrast, the accuracy loss caused by FedQT is small, and the volume of model parameters in the transmission process is also reduced. There is not much difference between the compression ratio and the Fedzip algorithm after pruning, quantization and coding. Although the compression rate of the Fedzip algorithm is smaller, in essence, our proposed FedQT algorithm has been able to achieve about a thousand times compression. This proves that the FedQT algorithm is suitable for the compression of the federated learning model, and verifies the effectiveness of the FedQT algorithm in federated learning compression.

Figure 5 shows the different losses caused by the FedAvg, Fedzip and FedQT algorithms during training, and the standard loss is used here, which is a comparison metric commonly used in federated learning model compression. According to the above figure results, the FedQT algorithm proposed in this paper can obtain lower loss, and its initial loss is almost consistent with the FedAvg algorithm without compression, in the first 40 rounds of training, the loss has been decreasing, and after about 100 rounds of training, its loss is more in line with FedAvg, it can show that the algorithm is fairer in the pruning process.

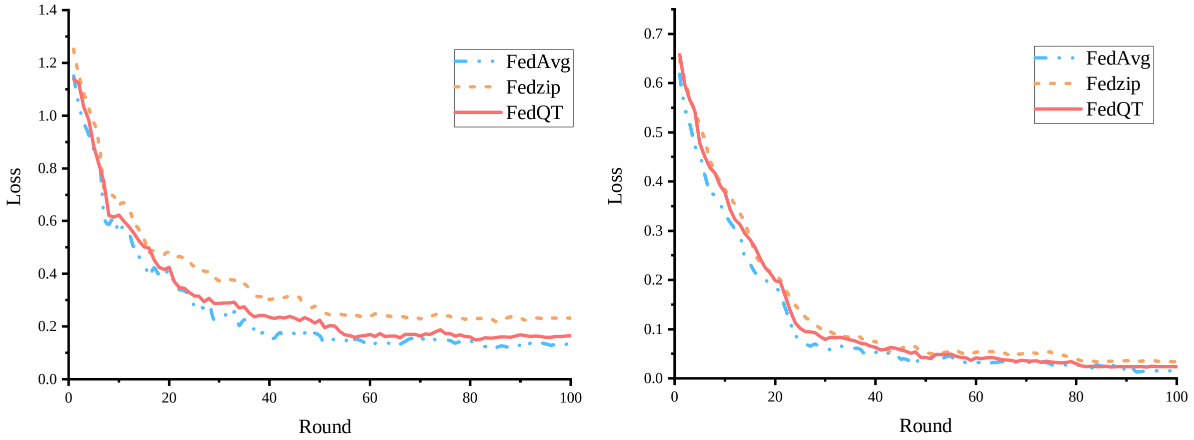

Figure 5. Comparison of Loss Between MNIST and OCTMNIST

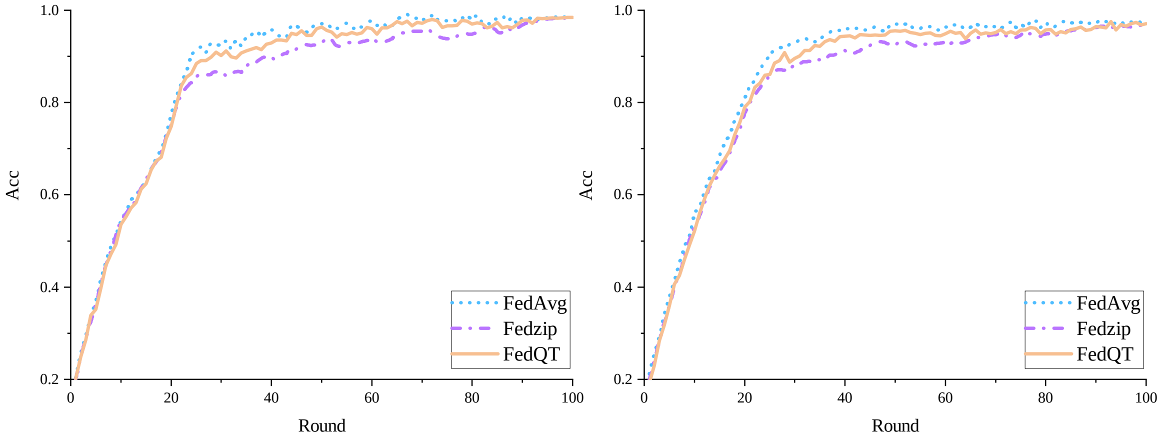

Figure 6. Comparison of Accuracy Between MNIST and OCTMNIST

From Figure 6, it is not difficult to see that the accuracy of FedQT on MNIST dataset can reach more than 97%, and the accuracy on OCTMNIST (medical retinal image dataset) can also reach 96%. After 100 rounds of iteration, its accuracy gradually stabilized. It is believed that the gradual convergence begins after 25 rounds of iteration, and stable convergence can be achieved after 100 rounds of iteration. The problem of data accuracy degradation may be alleviated by considering data enhancement.

7. Conclusion.

In this paper, in order to solve the problem of model volume and redundancy in federated learning, a FedQT model compression algorithm based on Top_Avg pruning and quantization algorithm is proposed to reduce the volume of the model. Experiments show that our FedQT algorithm can effectively improve the model communication efficiency of federated learning, and compared with the previous algorithm, it can maintain better convergence speed and lower accuracy loss. That is, the FedQT algorithm can have a better compression rate and a smaller loss for the federated learning model.

In the future work, we will continue to further optimize the communication efficiency of the model in federated learning, and consider how to further reduce the huge consumption caused by its operation and the security problems in the process of frequent communication.

Acknowledgments. This work was supported in part by the National Science Foundation of Shandong Province under the Grant ZR2022LZH014.

REFERENCES

[1] A. Alex, and L.-C. Chen. "Perceived security of BYOD devices in medical institutions," *International Journal of Medical Informatics*, vol. 168, pp. 1-7, ISSN 1386-5056, 2022.

[2] J. Alemany, E.-D. Val, and A.García-Fornes. "A review of privacy decision-making mechanisms in online social networks," *ACM Computing Surveys (CSUR)*, vol. 55, no. 2, pp. 1-32, 2022.

[3] Z.-H. Lv, and P. Piccialli. "The Security of Medical Data on Internet Based on Differential Privacy Technology," *Association for Computing Machinery*, vol. 21, no. 55, pp. 1-18, ISSN 1533-5399, 2021.

[4] T.-Y. Wu, Q. Meng, L. Yang, S. Kumari, and M. Pirouz. "Amassing the Security: An Enhanced Authentication and Key Agreement Protocol for Remote Surgery in Healthcare Environment," *Computer Modeling in Engineering and Sciences*, vol. 134(1), pp. 317-341, 2023.

[5] L.-N. Ni, B.-G. Ni, Y.-C. Tang, and J.-Q. Zhang. "DACSC: Secure Authentication Protocol Based on Dynamic Authentication Credentials and IntelSGX in Cloud Computing Environments," *Journal of Network Intelligence*, 2023.

[6] N.-K. Ray, D. Puthal, and D. Ghai. "Federated learning," *IEEE Consumer Electronics Magazine*, vol. 10, no. 6, pp. 106-107, 2021.

[7] M.-T. Wu, Q. Teng, S. Huda, Y.-C. Chen, and C.-M. Chen. "A Privacy Frequent Itemsets Mining Framework for Collaboration in IoT Using Federated Learning," *ACM Transactions on Sensor Networks*, vol. 19(2), no. 27, pp. 1-15, 2023.

[8] Y.-X. Fei, M. Fan, Y.-M. Zhu, and J.-K. Hu. "A Comprehensive Survey of Privacy-Preserving Federated Learning: A Taxonomy, Review, and Future Directions," in *ACM Computing Surveys (CSUR)*, doi: 10.1145/3460427, vol. 54(6), pp. 1–36, 2022.

[9] S. Zhao, S. Chen, and Z. Wei. "Statistical Feature-based Personal Information Detection in Mobile Network Traffic," *Wireless Communications and Mobile Computing*, pp. 1-17, 2022.

[10] S. Zhang and Y. Zhang. "Privacy leakage vulnerability detection for privacy-preserving computation services," in *IEEE International Conference on Web Services (ICWS)*, pp. 219-228, 2022.

[11] Y. Chen, W.-S. Gan, Y.-D. Wu, and P. S. "Privacy-preserving federated mining of frequent itemsets," in *Information Sciences*, vol. 625, pp. 504-520, ISSN: 0020-0255, 2023.

[12] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang, and D. Niyato. "Federated learning for 6g communications: Challenges, methods, and future directions," *China Communications*, vol. 17, no. 9, pp. 105-118, 2020.

[13] J. Xu, W. Du, Y. Jin, W. He, and R. Cheng. "Ternary compression for communication-efficient federated learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 3, pp. 1162-1176, 2020.

[14] F. Sattler, A. Marban, R. Rischke and W. Samek. "CFD: Communication-Efficient Federated Distillation via Soft-Label Quantization and Delta Coding," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 4, pp. 2025-2038, 2021.

[15] S. Liu, G. Yu, R. Yin, and J. Yuan. "Adaptive network pruning for wireless federated learning," *IEEE Wireless Communications Letters*, vol. 10, no. 7, pp. 1572-1576, 2021.

[16] S.-I. Young, W. Zhe, D. Taubman and B. Girod. "Transform Quantization for CNN Compression," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 5700-5714, 2022.

[17] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek. "Sparse binary compression: Towards distributed deep learning with minimal communication," in *International Joint Conference on Neural Networks (IJCNN).IEEE*, pp. 1-8, 2019.

[18] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek. "Robust and communication-efficient federated learning from non-iid data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3400-3413, 2019.

[19] A. Alkhulaifi, F. Alsahli, and I. Ahmad. "Knowledge distillation in deep learning and its applications," *PeerJ Computer Science*, vol. 7, pp. 1-19, 2021.

[20] V. Saragadam, R. Balestriero, A. Veeraraghavan, and R.-G. Baraniuk. "Deeptensor: Low-rank tensor decomposition with deep network priors," *arXiv preprint arXiv:2204.03145*, 2022.

[21] M.-D. Nguyen, S.-M. Lee, Q.-V. Pham, D.-T. Hoang, D.-N. Nguyen, and W.-J. Hwang. "Hcfl: A high compression approach for communication-efficient federated learning in very large scale iot networks," *IEEE Transactions on Mobile Computing*, pp. 1-13, 2022.

[22] Y. Cheng, and Y. Yu. “Mathematical Modelling of IoT-Based Health Monitoring System,” *Computational and Mathematical Methods in Medicine*, pp. 1-7, 2022.

[23] L. Deng. “The mnist database of handwritten digit images for machine learning research [best of the web],” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141-142, 2012.

[24] J. Yang, R. Shi, and B. Ni. “Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis,” in *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI). IEEE*, pp. 191-195, 2021.

[25] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B.A. -Arcas. “Communication-efficient learning of deep networks from decentralized data,” in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 1273-1282, 2012.

[26] A. Malekijoo, M.-J. Fadaeieslam, H. Malekijou, M. Homayounfar, F.-L. Shabdiz, and R. Rawassizadeh.“Fedzip: A compression framework for communication-efficient federated learning,” *arXiv preprint arXiv:2102.01593*, 2021.

[27] R.-H. Hijazi. “Analysis of compositional data using Dirichlet covariate models,” *American University*, 2003.