

Article

A Federated Incremental Learning Algorithm Based on Dual Attention Mechanism

Kai Hu ^{1,2} , Meixia Lu ¹ , Yaogen Li ¹ , Sheng Gong ¹ , Jiasheng Wu ³ , Fenghua Zhou ⁴ , Shanshan Jiang ⁵ 
and Yi Yang ^{1,2,*} 

- ¹ School of Automation, Nanjing University of Information Science and Technology, Nanjing 210044, China; 001600@nuist.edu.cn (K.H.); 20191223036@nuist.edu.cn (M.L.); 20201249100@nuist.edu.cn (Y.L.); 20211249032@nuist.edu.cn (S.G.)
- ² Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAET), Nanjing University of Information Science and Technology, Nanjing 210044, China
- ³ School of Economics and Business Management, Nanjing University of Science and Technology, Nanjing 210094, China; 20181223081@nuist.edu.cn
- ⁴ China Air Separation Engineering Co., Ltd., Hangzhou 310000, China; zhoufh@cnasec.com
- ⁵ School of Management Science and Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China; jss@nuist.edu.cn
- * Correspondence: yangyi@nuist.edu.cn; Tel.: +86-187-5287-8094

Abstract: Federated incremental learning best suits the changing needs of common Federal Learning (FL) tasks. In this area, the large sample client dramatically influences the final model training results, and the unbalanced features of the client are challenging to capture. In this paper, a federated incremental learning framework is designed; firstly, part of the data is preprocessed to obtain the initial global model. Secondly, to help the global model to get the importance of the features of the whole sample of each client, and enhance the performance of the global model to capture the critical information of the feature, channel attention neural network model is designed on the client side, and a federated aggregation algorithm based on the feature attention mechanism is designed on the server side. Experiments on standard datasets CIFAR10 and CIFAR100 show that the proposed algorithm accuracy has good performance on the premise of realizing incremental learning.

Keywords: federated learning; dual attention mechanism; incremental learning



Citation: Hu, K.; Lu, M.; Li, Y.; Gong, S.; Wu, J.; Zhou, F.; Jiang, S.; Yang, Y. A Federated Incremental Learning Algorithm Based on Dual Attention Mechanism. *Appl. Sci.* **2022**, *12*, 10025. <https://doi.org/10.3390/app121910025>

Academic Editor: Vincent A. Cicirello

Received: 30 August 2022

Accepted: 30 September 2022

Published: 6 October 2022

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



Copyright: © 2022 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

Federated learning (FL) can make full use of all data while keeping participants' data confidential to train a better global model than the local model that each participant trains separately using their own data. Google proposed an FL algorithm for mobile terminals in 2016 [1], each client trains each local model, then aggregates all local models to obtain a better global model. The process of exchanging model information between clients is carefully designed so that clients can learn the private data content of other clients. When the global model is obtained, the data information sources seem to be integrated; this is the core idea of Federated Learning.

After the concept of FL was proposed by Google, H. Brendan McMahan et al. proposed a practical method for the FL of deep networks based on iterative model averaging in 2017 [2]. This algorithm uses relatively small communication rounds to train a high-quality model, which is the classic federated averaging algorithm. In the later period, many federated learning algorithms were further developed, and many excellent branch algorithms were formed [3–9]. Liu, Y. et al. proposed a federated transfer learning framework that can be flexibly adapted to various multi-party secure machine learning tasks in 2018 [10]. The framework enables the target domain to build a more flexible and efficient model by leveraging many tags in the source domain. Yang, Q. et al. proposed to build a data network

based on a federation mechanism in 2019 [11]; it can allow knowledge sharing without compromising user privacy. Zhuo HH and others proposed a federated reinforcement learning framework in 2020 [12]; this framework adds Gaussian interpolation to shared information to protect data security and privacy. Peng Y et al. proposed a federated semi-supervised learning method for network traffic classification in 2021 [13]; it can effectively protect data privacy and does not require a large amount of labeled data.

Traditional FL can only be trained in a batch setting, classification information for all samples is known beforehand, model's classify ability is also fixed. In the face of new tasks and data, it has to re-train totally. Among the many directions of FL at present, research on incremental tasks is rare [14,15]. Luo [16] pointed out that traditional data processing technologies have outdated models, weakened generalization capabilities, and do not consider the security of multi-source data. They proposed an online federated incremental learning algorithm for blockchain in 2021. During incremental learning, there will be a problem of unbalanced client samples. Clients with a large sample size have greater weight and significantly impact the final model training results.

Given the above discussion, to dynamically handle the increase of resources without retraining, reduce the impact of large sample clients on the final model training results, and the problem of feature imbalance when model aggregation is elusive. We propose a federated incremental learning algorithm based on a dual attention mechanism. This algorithm can dynamically handle the increase in resources without retraining while mining the characteristics of the overall client-side samples; it can enhance the capture performance of the model training server for the key information of client-side features. The contributions of this paper are as follows:

(1) We design a federated incremental learning framework. First, the framework randomly sampling the same number of samples from each client, to ensure the balance of pre-training samples, and trains with the federated averaging model to obtain the preliminary period global model on the server. Then the iCaRL strategy [17] is applied to the traditional FL framework; this strategy can classify samples according to the nearest mean rule, use preferential sample selection based on herd behavior, perform representation learning for knowledge distillation, and dynamically handle resource increases without retraining. Therefore, the federated incremental learning framework can take the dynamic changes in training tasks and keep the data confidential.

(2) The dual attention mechanism is added to the federated incremental learning framework. A channel attention neural network model is designed on the client-side and used as the FL's local model. This model adds the SE module [18] based on the classic Graph Convolutional Neural (GCN) network, which can help the model to obtain the importance of the features of the overall samples of each client during model training and can effectively reduce the influence of noise. In the global model, a federated aggregation algorithm based on the feature attention mechanism is designed to provide appropriate attention weights for each local model. This weight corresponds to the model parameters of each layer of the neural network. The attention weight value is used as the aggregation coefficient, which can enhance the global model's capture performance for the key features' key information.

This paper is organized as follows: The Section 2 introduces the relevant background information; the Section 3 elaborates on our proposed algorithm; the Section 4 is the experiment performance of this algorithm, and these results are discussed; the Section 5 summarizes the paper.

2. Background

2.1. Federated Averaging Algorithm

In the classic FL algorithm, the federated averaging algorithm is generally used for model training of federated learning. The federated averaging algorithm is mainly model averaging. In the federated averaging algorithm, each client locally performs stochastic gradient descent on the existing model parameters \bar{w}_t using local data [19], the updated

model parameter $\omega_{t+1}^{(k)}$ is sent to the server, the server aggregates the received model parameters, that is, uses a weighted average of the received model parameters, and the updated parameters $\bar{\omega}_{t+1}$ is sent to each client. This method is called model averaging [20]. Finally, the server checks the model parameters, and if it converges, the server sends a signal to each participant to stop the model training.

$$\omega_{t+1}^{(k)} = \bar{\omega}_t - \eta \nabla_k \tag{1}$$

$$\bar{\omega}_{t+1} = \sum_{k=1}^K \frac{n_k}{n} \omega_{t+1}^{(k)} \tag{2}$$

In Formula (1), η is the learning rate, ∇_k is the local gradient update of the k th participant, n_k is the local data volume of the k th participant, n is the local data volume of all participants, $\omega_{t+1}^{(k)}$ are the parameters of the local model of the k th participant at this time, and $\bar{\omega}_{t+1}$ are the aggregated global model parameters.

2.2. The Basic Structure of Federated Learning

Federated learning is an algorithm that does not need to directly fuse multi-party data for training and only needs to encrypt and exchange client model parameters to train a high-performance gl. Therefore, federated learning can meet the requirements of user privacy protection and data security. Figure 1 is an example of a federated learning architecture that includes a coordinator. In this scenario, the coordinator is an aggregation server, which can send the initial random model to each client. The clients use their respective models to train the model and send the model weight updates to the aggregation server. After that, the aggregation server aggregates the model updates received from the client and sends the aggregated model updates back to the client. Under this architecture, the client’s origin is always stored locally, which can protect user privacy and data security [21–26].

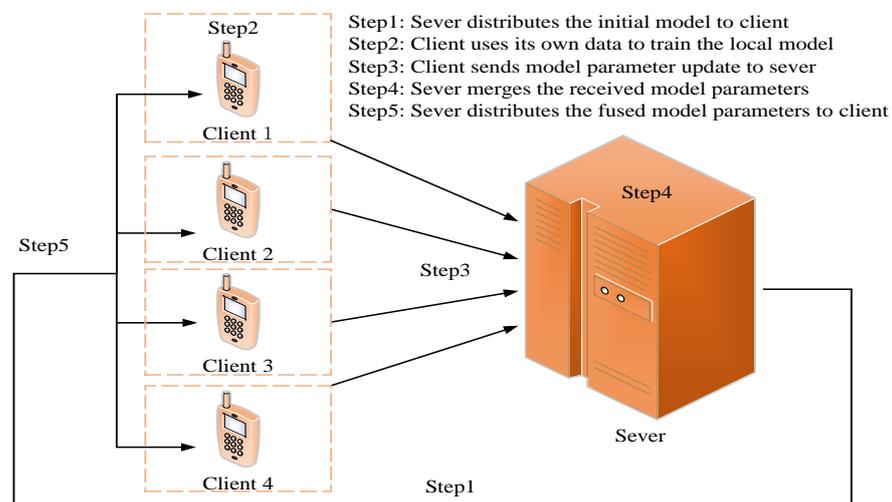


Figure 1. The federated learning framework proposed in this paper.

2.3. Class Incremental Learning

Class incremental methods [27–32] learn from non-stationary distributed data streams, and these methods should be suitable for a large number of tasks without adding excessive computation and memory. Their goal is to use old knowledge to improve the learning of new knowledge (forward transfer), and to use new data to improve performance on previous tasks (backward transfer). During each training phase, the learner only has access to data for one task [33]. The task consists of multiple classes, allowing the learner to process

the training data for the current task numerous times during training. A typical class increment setup consists of tn task sequences.

$$ts = [(C^1, D^1), (C^2, D^2), \dots, (C^{tn}, D^{tn})] \tag{3}$$

where C is the class, D is the data, and each task ts is represented by a set of classes and training data. Today, most classifiers for incremental learning are trained with a cross-entropy loss, the cross-entropy of all classes for the current task. The loss is calculated as follows:

$$l_c(x, y, \theta^{ts}) = \sum_{j=1}^{N^{ts}} y_j \log \frac{\exp(h_j)}{\sum_{j=1}^{N^{ts}} \exp(h_j)} \tag{4}$$

where x is the input feature of the training sample, and $y \in 0, 1^{N^{ts}}$ is a truth label vector corresponding to x_i . N^{ts} is used to represent the total number of classes for the ts task, $N^{ts} = \sum_{i=1}^{ts} |C^i|$, data $D^{ts} = (x_1, y_1), (x_2, y_2), \dots, (x_{m^{ts}}, y_{m^{ts}})$. We consider incremental learners that are deep networks parameterized by weights θ , and we further split the neural network into a feature extractor f with weights φ and a linear classifier g with weights Z according to $h(x) = g(f(x; \varphi); Z)$. But in this case, since softmax normalization is performed on all classes seen in all previous tasks, errors during training will be backpropagated from all outputs, including the output of those classes that do not correspond to the current task. Therefore, we can only consider the network output belonging to the class in the current task, and define the cross-entropy loss as follows, because this loss only considers the softmax-normalized prediction of the class in the current task; therefore errors are only backpropagated from the probabilities associated with these classes in task ts [34].

$$l_{c^*}(x, y, \theta^{ts}) = \sum_{j=1}^{|C^{ts}|} y_{N^{ts-1}+j} \log \frac{\exp(h_{N^{ts-1}+j})}{\sum_{j=1}^{|C^{ts}|} \exp(h_{N^{ts-1}+j})} \tag{5}$$

3. Federated Incremental Learning Algorithm

In addition to data privacy protection in FL, dealing with the dynamic changes in training tasks is also essential to research content. For example, in recommender systems, user data will be updated dynamically. The traditional training model is in the face of new tasks and data. Retraining all the old and new data will cost much training. Therefore, whether the user data can be dynamically updated has become the key to measuring the pros and cons of an algorithm. We need a federated learning algorithm that can cope with the dynamic changes in training tasks and keep the data confidential. In FL, scholars have found that large-sample client weight parameters are large and have a significant impact on the final model training. The imbalance of importance, and the problem that the importance of features is difficult to capture. This paper introduces a dual attention mechanism in the federated incremental learning framework. We hope to reduce the impact of considerable sample client data noise on the global model, improve the ability of the global model to capture essential client features, and achieve more excellent business value. The idea of this algorithm is as follows:

(1) As traditional federated learning can only be trained in a batch setting and needs to be retrained in the face of new tasks and data, more flexible strategies are required to handle large-scale and dynamic real-world object classification situations. This paper proposes a federated incremental learning framework that can cope with the dynamic changes in training tasks and keep the data confidential. We combine the care strategy with a federated learning framework, and dynamically classify samples based on the nearest mean rule to handle resource increases without retraining dynamically. Therefore, the framework can reduce the cost of data storage and model retraining when adding classes, and also reduce the risk of data leakage due to model gradient updates.

(2) Aiming at the problem of sample imbalance in each client, the client with a large sample size greatly significantlucenes the final model training result. On the federated

incremental learning framework, this paper designs a channel attention neural network model on the client-side and uses it as a local model for federated learning. This model adds the SE module based on the classical graph convolutional neural network, which can help the model to obtain the importance of the features of the respective overall samples of each client during model training and can effectively reduce the influence of noise.

(3) For traditional federated learning, the initial parameters of the global model are randomized, and the initial randomized parameters will affect the convergence speed of the global model. A pre-training module is added to the federated incremental learning framework, and the same number of samples are extracted from each client as pre-training data. The federated averaging model is used for training to obtain a global model on the server, which can speed up the model convergence. In addition, extracting the same number of samples from each client can ensure the balance of pre-training samples, to the impact of large sample clients on the global model.

(4) The federated learning design aims to jointly train a high-quality global model for the client. Still, when the client data is unbalanced, the significant sample client weight parameter significantly impacts the final model training result. On the federated incremental learning framework, this paper designs a federated aggregation algorithm based on the feature attention mechanism in the global model to provide appropriate attention weights for each local model. This weight corresponds to the model parameters of each layer of the neural network, and the attention weight value is used as the aggregation coefficient, which can enhance the capture performance of the global model for key feature information.

3.1. Federated Incremental Learning Framework

In addition to data privacy protection in FL, dealing with the dynamic changes in training tasks is also essential to research content. For example, in recommender systems, user data will be updated dynamically. The traditional training model is in the face of new tasks and data. Retraining all the old and new data will cost much training. Therefore, whether the user data can be dynamically updated has become the key to measuring the pros and cons of an algorithm. We need a federated learning algorithm that can cope with the dynamic changes in training tasks and keep the data confidential. Because incremental learning can continuously learn new concepts in the data stream [35–37], we consider introducing incremental learning in federated learning to help each client first train data when there are only a few classes initially, and then add classes can be gradually added for learning. Still, incremental learning has the problem of historical forgetting. Therefore, we consider adding a new training strategy iCaRL strategy to incremental learning to classify according to the nearest mean rule of samples, use preferential sample selection based on herd behavior, and perform representation learning for knowledge extraction and prototype rehearsal, without retraining. Increases in dynamic processing resources. Adding incremental learning to traditional federated learning can reduce the cost of data storage and model retraining when adding classes and reduce the risk of data leakage due to model gradient updates.

In addition, in traditional federated learning, the initial parameters of the global model are all randomly generated numbers, which will affect the convergence speed of the model. To speed up the convergence speed of the model, this paper considers adding a pre-training module. Generally, the pre-training module uses $n\%$ of all data as pre-training data. However, this method can easily expand the influence of clients with large sample sizes on the global model, especially the influence of non-important features [38–47] of large samples in incremental learning. We plan to extract the same number of samples from each client as pre-training data to ensure the balance of pre-training samples and reduce the impact of large-sample clients on the global model. The federated incremental learning framework is shown in Figure 2.

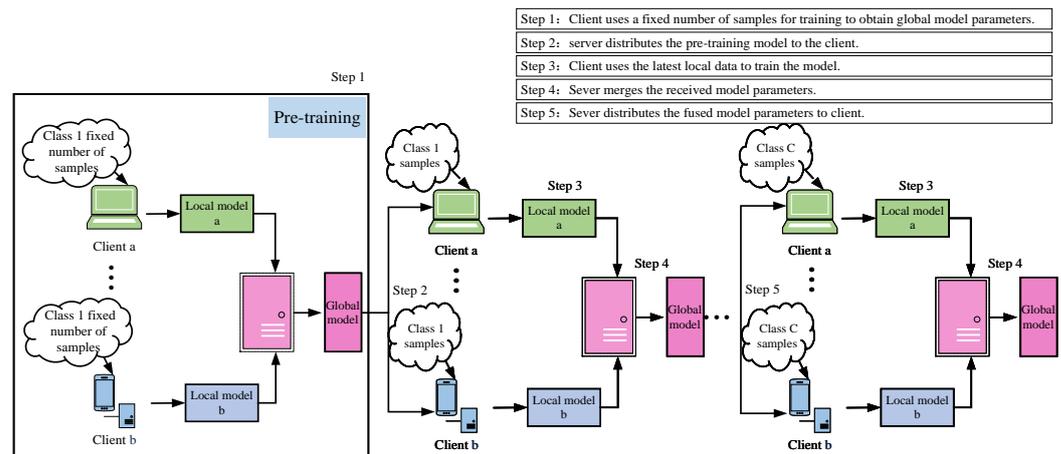


Figure 2. Federated incremental learning framework.

The client first uses a fixed number of samples of class 1 for training and aggregation to obtain pre-trained global model parameters, which can speed up the convergence of the model and reduce the impact of non-important features in large-sample client data on the global model. The global model parameters are distributed to the client, and the client uses the remaining samples of class 1 for training. After the training, the client model parameters of the t th communication are sent to the server, and the server sends the fused model parameters to the client. Then the client uses the class 2 samples and some old data to form a training set for training, uses the feature extractor to extract feature vectors from the old and new data, and calculates the respective average feature vectors. The predicted value is brought into the loss function of the combination of distillation and classification loss for optimization. The client model parameters at the $(t + 1)$ th communication are obtained and sent to the server, so storage until the incremental learning of all classes is completed.

3.2. Dual Attention Mechanism Module

The original design concept of federated learning aims to jointly train high-quality global models on the client side without revealing privacy. For example, mobile shopping malls, banks, and other apps with high-quality customer information can recommend suitable shopping and financial products through joint model training without revealing user information. Many large enterprises have trained high-performance global models to help small and medium-sized enterprises in their later development. However, due to the imbalance of samples between clients, there will be a problem that the noise generated by clients with a large sample size when participating in the training will also significantly impact the results of the final trained global model. To this end, given the problem of large samples and considerable noise, consider adding a channel attention mechanism on the client-side, perform feature compression in the spatial dimension to obtain a feature map with a global receptive field, and then learn through fully connected layers (FC) relationship between channels. Finally, multiplying the learned weight coefficients of each channel with all the elements of the corresponding channel can help to minimize the impact of noise on the final model training results while obtaining the characteristics of the respective overall samples of each client. At the same time, federated learning joins incremental learning [48,49] to perform dynamic task training. Learning at the same time simultaneously increments, will lead to the imbalance of sample importance and the problem that the importance of features is difficult to capture. For this reason, it is difficult to capture the imbalance of features. We consider adding a feature attention mechanism [50,51] when the global model is aggregated, to enhance the capture performance of the model training for critical feature information. Therefore, we consider adding a dual attention mechanism module that simultaneously channels attention and features attention to the federated learning framework. The dual attention mechanism module is shown in Figure 3.

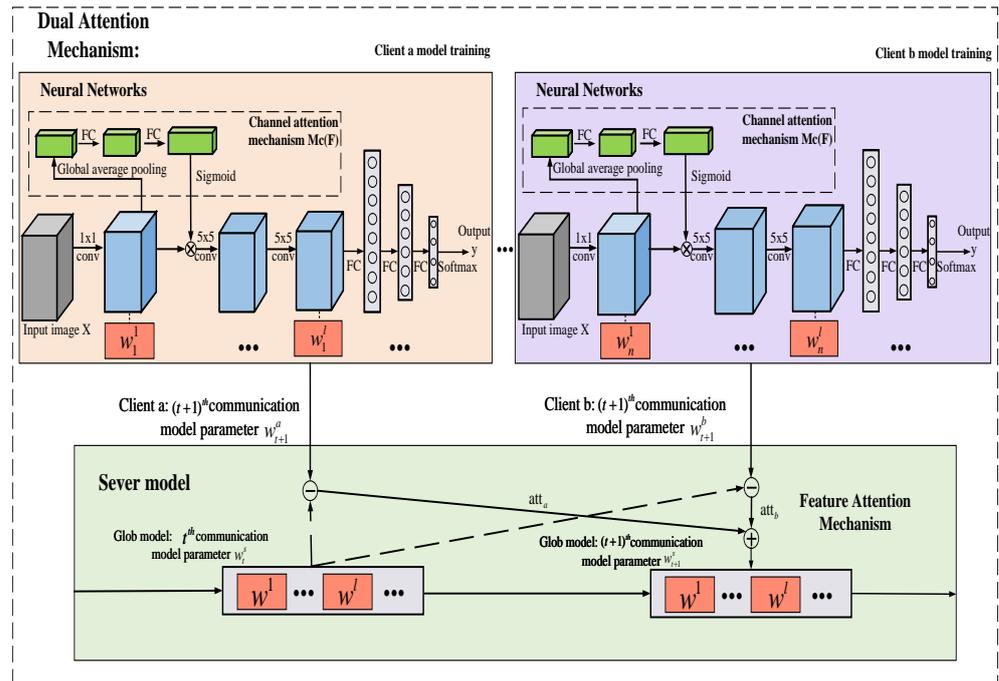


Figure 3. Dual attention mechanism module.

In Figure 3, the first dual attention mechanism module is to add a channel attention mechanism to the client to help obtain the characteristics of the overall samples of each client while minimizing the impact of noise on the final model training results. There are many models of the local model of the client, such as LSTM, CNN, and other algorithms combined with the SE channel attention module. The architecture of CNN combined with channel attention is shown in Figure 4. The specific process of the client-side neural network model is shown in Figure 5.

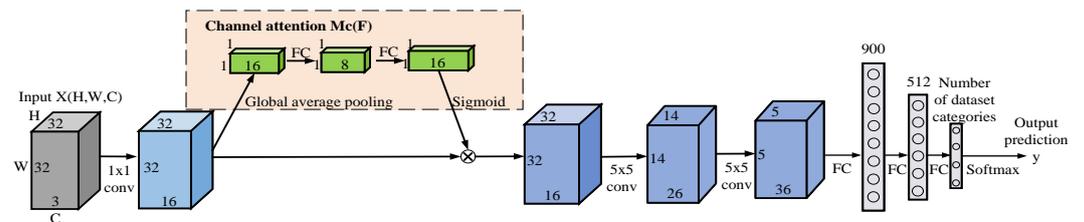


Figure 4. Neural network model.

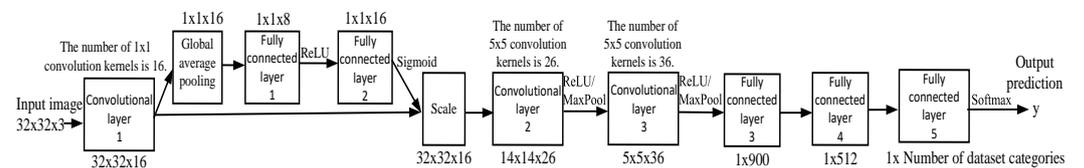


Figure 5. Neural network model flow chart.

As shown in Figure 4, the SE module is added after the first convolutional layer of the local model, because the number of channels in the layers too far behind is too large, which is easy to cause overfitting. If the feature map is too small, we use Improper operation will introduce a large proportion of non-pixel information. More importantly, close to the classification layer, the effect of attention is more sensitive to the classification results, and it is easy to affect the decision of the classification layer. Therefore, after placing it in the first convolutional layer, each layer of the convolutional network has 16 convolution

kernels, and each convolution kernel corresponds to a feature channel. Channel attention can allocate resources between each convolution channel. The final output is obtained by learning the degree of dependence of each channel and adjusting different feature maps according to the degree of dependence. Therefore, we can add the channel attention mechanism to help focus on the overall characteristics of the sample and reduce the impact of noise on the final model training results. As shown in the figure, the input image size is 32×32 , that is, $M = 32$, the number of channels C is 3, and it is input to the convolution layer 1. The convolution layer contains 16 convolution kernels, and the convolution kernel sizes 1×1 , that is, $\text{kernel} = 1$, and no pixels are filled around the input image matrix, that is, $P = 0$. We can think of the convolution kernel as a sliding window, which slides forward with a set step size, set the step size to 1, that is, $bu = 1$. According to the output calculation Formula (6) of the convolutional layer, the size of the output image can be calculated as 32×32 , that is, $Ne = 32$, and the number of channels C is 16.

$$Ne = \frac{M - \text{kernel} + 2P}{bu} + 1 \quad (6)$$

After that, we perform global average pooling on it, perform feature compression on the output image obtained through convolutional layer one along the spatial dimension, and turn each two-dimensional feature channel into an actual number, which has a global effect to some extent. The receptive field and t output dimension match the input's feature channels put, and the number of channels C is 16. After global average pooling, the feature dimension $C = 16$ is reduced to the dimension $C = 8$ through the fully connected layer 1, and then activated by ReLU. Then the dimension $C = 8$ is raised back to the original dimension 16 through the fully connected layer 2, which is used here. The two fully connected layers can better fit the complex correlation between channels with more nonlinearities, and significantly reduce the number of parameters and computation. After the fully connected layer 2, a sigmoid activation function is used to obtain a normalized weight between 0 and 1, and finally, a Scale operation is used to weight the normalized channel attention weight to the features of each channel; it can help focus on the overall feature importance of the sample and reduce the impact of noise on the final model training results. The Scale operation refers to the channel-by-channel weighting of the previous features through multiplication to complete the re-calibration of the original feature map in the channel dimension. After that, we continue using convolutional layer 2 and convolutional layer 3 for feature extraction on the noise-reduced feature map. And adding add activation function ReLU and MaxPool between convolutional layer 2 and convolutional layer 3. The activation function can extract helpful feature information and make the model more discriminative, and the pooling layer can avoid overfitting by down-sampling. Finally, because two or more fully connected layers can solve the nonlinear problem satisfactorily, we input the features of the convolutional layer 3 into the fully connected layer 3 and the fully connected layer 4 to obtain a 512-dimensional vector, then go through the full connection layer. Connection layer 5 receives a c -dimensional vector (c is the number of classes in the current dataset), then inputs it into the softmax classifier for classification, and outputs the predicted value y .

As shown in Figure 3, the second dual attention mechanism module considers that incremental learning, when learning at the same time as incremental classification learning, it the problem of unbalanced features and difficulty in capturing important featurefeaturesd a feature-space attention mechanism to enhance the grasping performance of key information. Feature attention mechanism based global aggregation is shown in Figure 6.

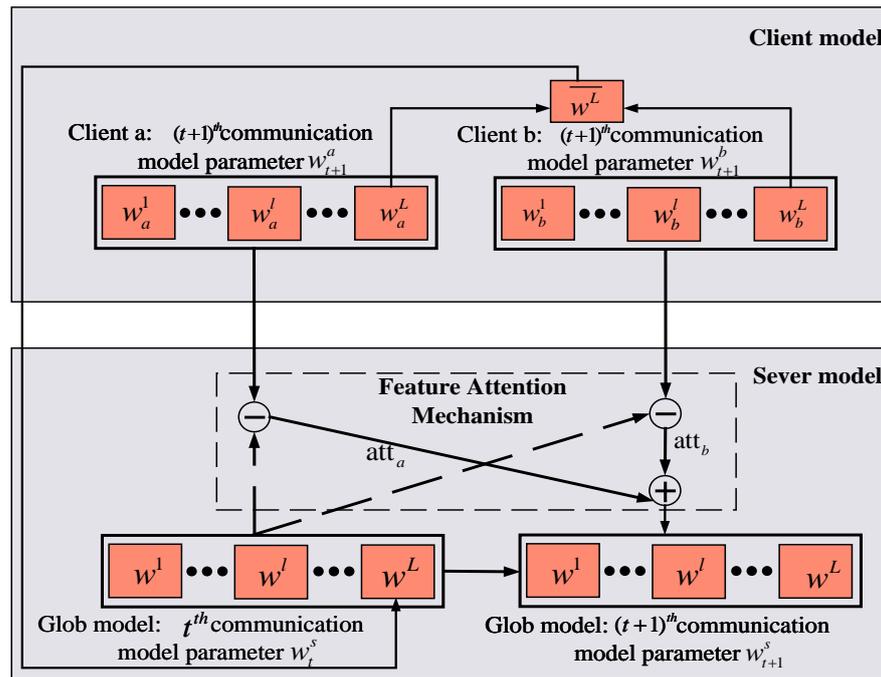


Figure 6. Feature attention mechanism based global aggregation.

As shown in the figure above, to enhance the capture performance of key information and solve the problem that the respective features are unbalanced and the importance is difficult to capture in the incremental learning with the incremental learning of classification and learning at the same time, this paper introduces the feature attention mechanism for the model aggregation of the client, we improve model performance by capturing the importance of neural network layers in multiple local models. This mechanism can automatically consider the weight of the relationship between the server model and the client. In iterative training, continuously updating the parameters reduces the weighted distance between the server and the client model, and the expected distance between the server and the client model is minimized. The optimization objective is calculated as shown in Formula (7).

$$J(\omega_t^s, \omega_{t+1}^1, \dots, \omega_{t+1}^K) = \arg \min_{\omega_{t+1}^s} \sum_{k=1}^n \left[\frac{1}{2} att_k D(\omega_t^s, \omega_{t+1}^k)^2 \right] \quad (7)$$

Among them, ω_t^s is the model parameter of the server in the t th communication, and ω_{t+1}^k is the model parameter of the client k in the $(t + 1)$ th communication. $D(\cdot, \cdot)$ is the distance between the two sets of neural parameters calculated using the Euclidean distance formula, att_k is the important weight of the client model. The hierarchical soft attention method is used to capture the hierarchical importance of the neural network in multiple local models, and it is aggregated into the global model as feature attention to achieve the optimal server and client. The distance between the models is minimized, the expected function of the previous formula is derived to obtain the gradient, and K clients perform gradient descent to update the parameters of the global model.

$$g = \sum_{k=1}^K att_k (\omega_t^s - \omega_{t+1}^k) \quad (8)$$

$$\omega_{t+1}^s = \omega_t^s - \alpha g \quad (9)$$

The importance weight of the client model att_k is calculated by hierarchical soft attention. The server-side model is used as a query value, and the client-side model is used as a key value to calculate the attention score of each layer in the neural network. The attention formula is shown in Equation (10). It should be noted that due to the

incremental learning, the number of neurons output by the last layer of the model fully connected is the number of dataset classes, which changes dynamically, resulting in loading the local model parameters at this time to the server. When calculating the attention score of each layer together with the global model in the last communication, there will be a problem of weight mismatch. Therefore, before calculating the attention score of each layer, we need to average the weights of the last fully connected layer in all client models, and then assign them to the previous layer parameters of the global model of the latest communication.

$$att_k^l = softmax(\|\omega^l - \omega_k^l\|_p) = \frac{exp(\|\omega^l - \omega_k^l\|_p)}{\sum_{k=1}^K exp(\|\omega^l - \omega_k^l\|_p)} \quad (10)$$

$$\omega^L = (\omega_a^L + \dots + \omega_b^L) / K \quad (11)$$

where K is the number of clients, ω^l is the model parameter of the l th layer of the server, ω_k^l is the model parameter of the l th layer of the k th local client, $l \in [1, L]$. We take the p -norm of the difference between the matrices as the similarity value of the query and key values of the l th layer and use the softmax function for the similarity value to obtain the attention value of the l th layer of client k , and the feature attention of the entire client $att_k = \{att_k^1, att_k^2, \dots, att_k^L\}$.

4. Experimental Analysis

CPU: AMD R5-3600, memory: 16G DDR4, GPU: NVIDIA Geforce RTX2070S, operating system: 64-bit Windows 10; the experimental framework is the Pytorch open-source framework. Stochastic gradient descent is used as the learning rate; the initial learning rate is 0.2, the weight attenuation coefficient is set to 0.00001, the training batch size is 128, the number of local clients is 2, and each class of the Cifar10 and Cifar100 datasets is randomly divided into 2 copies, and sent to each local client in class increment. The local client performs incremental learning for Cifar10 according to 2 classes and 5 classes, and incremental learning for Cifar100 according to 10 classes, 20 classes, and 50 classes. This paper's experiments mainly verify the influence of the two structures on accuracy.

This experiment uses the classic CNN as the network model in the architecture of this article (CNN construction model code reference: <https://github.com/jhjiezhang/FedAvg/blob/main/src/models.py>, accessed on 1 March 2022). It compares it with the Federated Averaging model, which uses the same structure of test set CNN. The result of the experiment is an average of 10 times.

The datasets used in this experiment are two public datasets, CIFAR-10 and CIFAR-100. CIFAR-10 is a small dataset for recognizing ubiquitous objects organized by Hinton students Alex Krizhevsky and Ilya Sutskever. The dataset contains a total of 10 classes of RGB color images: airplanes, cars, birds, cats, deer, dogs, frogs, horses, boats, and trucks. The image size is 32×32 , and there are a total of 50,000 training images and 10,000 test images in the dataset. The CIFAR100 dataset has 100 classes. Each class has 600 color images of size 32×32 , of which 500 are used as training set and rest 100 are used as test set. The 100 class is composed of 20 classes (each class contains 5 subclasses). To better evaluate the model, set TP to represent the actual class, that is, the positive samples that the model correctly predicts, and FP to represent the actual actualized class, that is, the positive samples that are predicted to be negative by the model. The accuracy formula is as follows:

$$Accuracy = \frac{TP}{TP + FP} \quad (12)$$

4.1. Ablation Experiment—Rationality Analysis of Pre-Training Module

This experiment is an ablation experiment. To only use the CNN model combined with the pre-training module, instead of the model framework of the second innovation in this algorithm, the traditional FedAvg aggregation method is used to verify the effect of the CNN model combined with the pre-training on the accuracy improvement.

Figures 7 and 8 are the test results of the Incre-FL algorithm and the Icarl-FedAvg algorithm with the pre-training module added to the two test sets, where the x-axis represents the number of classes, the y-axis represents the test accuracy. The number of different classes During the incremental learning process, the average accuracy of the algorithm in this paper is slightly better than the Icarl-FedAvg algorithm. The accuracy of the CIFAR10 dataset when the classification increments are 2 and 5 reaches 43.45% and 63.16%. The accuracy rates on the CIFAR100 dataset with classification increments of 10, 20, and 50 reach 30.03%, 39.35%, and 39%. The algorithm proposed in this paper has made a series of improvements, and finally, the algorithm’s formance per has been improved to a certain extent. The influence of the pre-training module on the Incre-FL algorithm is shown in Table 1. We add the pre-training module to speed up the convergence of the model. The module plays a certaspecific in improving the accuracy of image classification tasks.

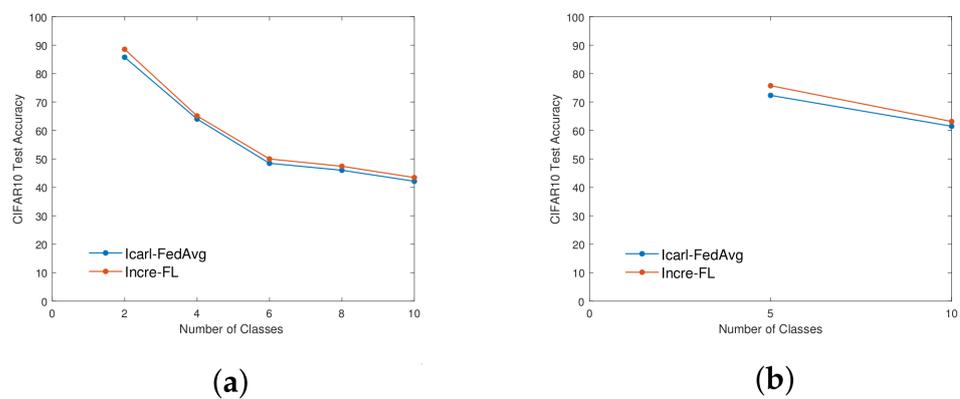


Figure 7. The accuracy curve of the pre-training module in the test set Cifar10. (a) Accuracy graph when the class increment is 2. (b) Accuracy graph when the class increment is 5.

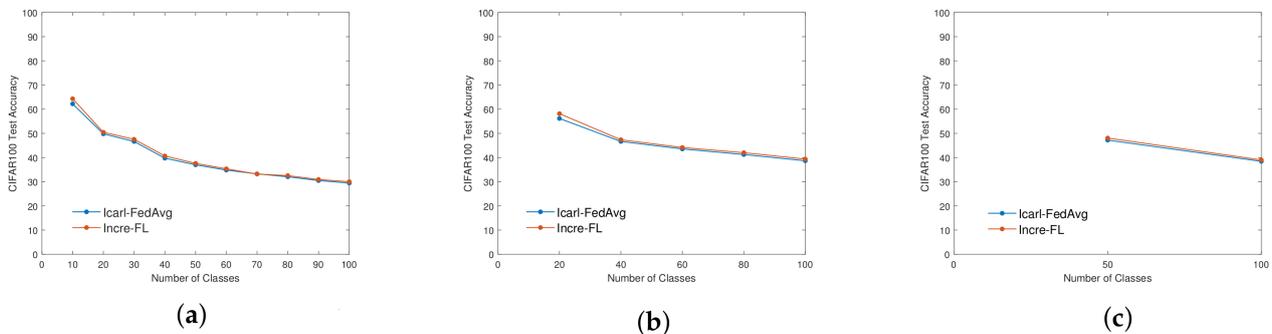


Figure 8. The accuracy curve of the pre-training module in the test set Cifar100. (a) Accuracy graph when the class increment is 10. (b) Accuracy graph when the class increment is 20. (c) Accuracy graph when the class increment is 50.

Table 1. The influence of the pre-training module.

Dataset	Increments/Test Classes	Model	Accuracy (%)
CIFAR10	2/10	Icarl-FedAvg	42.14
		Incre-FL	43.45
	5/10	Icarl-FedAvg	61.49
		Incre-FL	63.16

Table 1. *Cont.*

Dataset	Increments/Test Classes	Model	Accuracy (%)
CIFAR100	10/100	Icarl-FedAvg	29.44
		Incre-FL	30.03
	20/100	Icarl-FedAvg	38.65
		Incre-FL	39.35
	50/100	Icarl-FedAvg	38.42
		Incre-FL	39.00

4.2. Ablation Experiment—Rationality Analysis of Dual Attention Mechanism Module

There is a problem of sample imbalance in each client in federated learning. A client with a large sample size has a significant weight parameter, which significantly impacts the final model training result. At the same time, federated learning adds incremental learning for dynamic task training and learns at the same time in classification increments. When the sample importance is unbalanced, the matter is challenging to capture. This experiment is an ablation experiment. To use only the dual attention mechanism method of innovation 2, without using the pre-training module, it is compared with the Icarl-FedAvg learning algorithm to verify the effect of the dual attention mechanism on the accuracy improvement. Figures 9 and 10 show the accuracy curves of the dual attention mechanism module in test set Cifar10 and Cifar100.

The influence of the dual attention mechanism module in this paper is shown in Table 2. In the improved Incre-FL algorithm in this paper, the channel attention mechanism is added to the client; we can obtain the characteristics of the overall samples of each client, reduce the influence of noise, and add a feature space attention mechanism during federated aggregation, which can enhance the global model. The capture performance of the client's critical information will ultimately improve the accuracy of the image classification task. The dual attention mechanism in the experiment of this chapter includes the channel attention mechanism and the feature attention mechanism. Next, we verify the accuracy improvement effect of adding the channel attention mechanism and the feature attention mechanism separately.

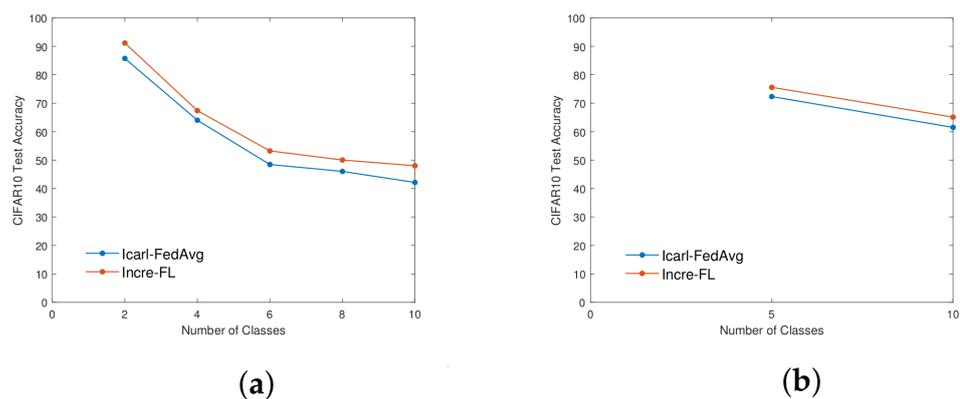


Figure 9. The accuracy curve of the dual attention mechanism module in the test set Cifar10. (a) Accuracy graph when the class increment is 2. (b) Accuracy graph when the class increment is 5.

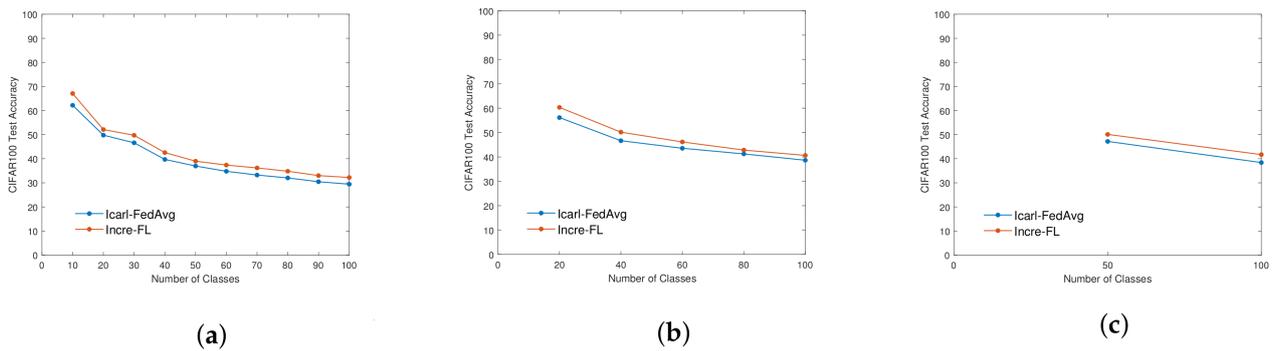


Figure 10. The accuracy curve of the dual attention mechanism module in the test set Cifar100. (a) Accuracy graph when the class increment is 10. (b) Accuracy graph when the class increment is 20. (c) Accuracy graph when the class increment is 50.

Table 2. The influence of the dual attention mechanism module.

Dataset	Increments/Test Classes	Model	Accuracy (%)
CIFAR10	2/10	Icarl-FedAvg	42.14
		Incre-FL	48.01
	5/10	Icarl-FedAvg	61.49
		Incre-FL	65.11
CIFAR100	10/100	Icarl-FedAvg	29.44
		Incre-FL	32.22
	20/100	Icarl-FedAvg	38.65
		Incre-FL	40.62
	50/100	Icarl-FedAvg	38.42
		Incre-FL	41.72

Figures 11 and 12 show the test results of the Incre-FL algorithm, the Icarl-FedAvg algorithm, the SE-Icarl algorithm, and the Earl algorithm with the channel attention mechanism added separately on the CIFAR10 and CIFAR100 standard test sets. It can be seen from the figure that the accuracy of the algorithm in this paper is significantly better than the Icarl-FedAvg algorithm, and the SE-Icarl algorithm with the channel attention mechanism alone is better than the Icarl algorithm. The algorithm's accuracy in this paper on the CIFAR10 dataset is 47.53% and 64.47% when the classification increments are 2 and 5. The accuracies on the CIFAR100 dataset with classification increments of 10, 20, and 50 reach 32.09%, 40.20%, and 41.23%. The influence of the channel attention mechanism in this paper is shown in Table 3. The channel attention mechanism can model the dependencies between channels in the external network, help to focus on the importance of the overall features of the samples, and reduce the impact of noise on the final model training results. We add it to the Earl algorithm and the Icarl-FedAvg algorithm to improve the accuracy of image classification tasks.

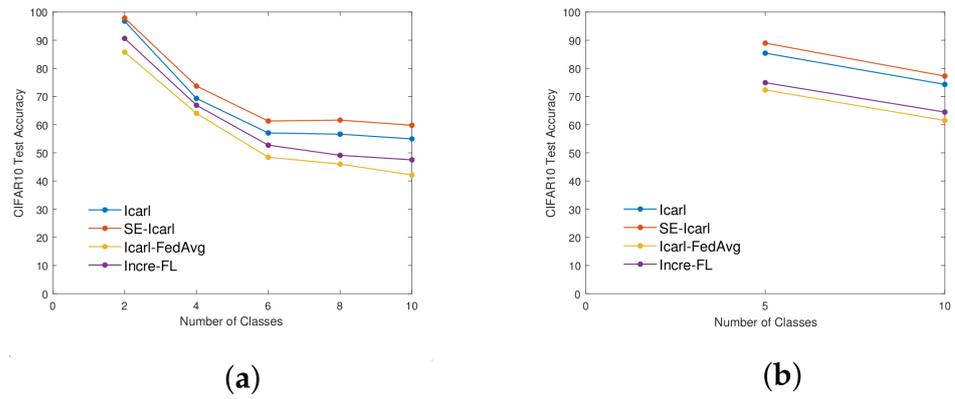


Figure 11. The accuracy curve of the channel attention mechanism module in the test set Cifar10. (a) Accuracy graph when the class increment is 2. (b) Accuracy graph when the class increment is 5.

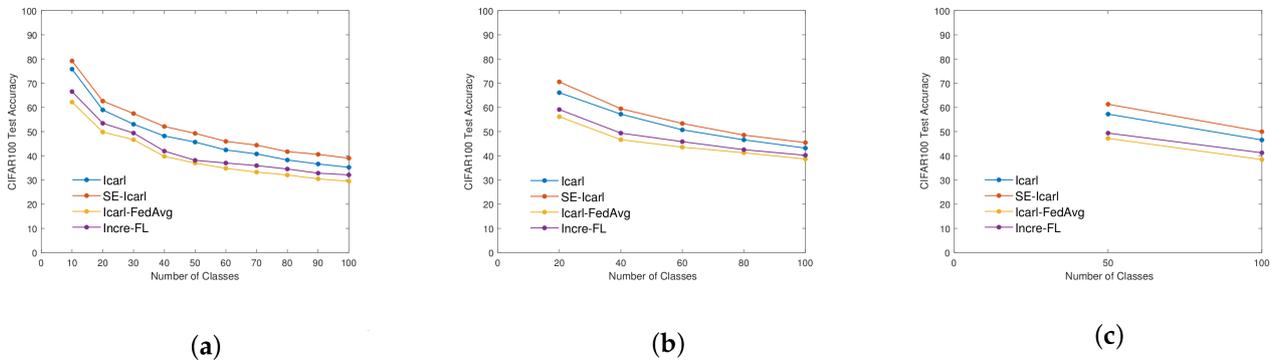


Figure 12. The accuracy curve of the channel attention mechanism module in the test set Cifar100. (a) Accuracy graph when the class increment is 10. (b) Accuracy graph when the class increment is 20. (c) Accuracy graph when the class increment is 50.

Table 3. The influence of the dual attention mechanism module.

Dataset	Increments/Test Classes	Model	Accuracy (%)
CIFAR10	2/10	Icarl-FedAvg	42.14
		Incre-FL	47.53
	5/10	Icarl-FedAvg	61.49
		Incre-FL	64.47
CIFAR100	10/100	Icarl-FedAvg	29.44
		Incre-FL	32.09
	20/100	Icarl-FedAvg	38.65
		Incre-FL	40.20
	50/100	Icarl-FedAvg	38.42
		Incre-FL	41.23

Figures 13 and 14 show the test results of the Incre-FL algorithm and the Icarl-FedAvg algorithm with the feature attention mechanism added separately on the CIFAR10 and CIFAR100 standard test sets. It can be seen from the figure that the accuracy of the algorithm in this paper is significantly better than that of the Earl-FedAvg algorithm. The algorithm’s accuracy in this paper on the CIFAR10 dataset when the classification increments are 2 and 5 reaches 45.40% and 63.73%. On the CIFAR100 dataset, the accuracy rates when the classification increments are 10, 20, and 50 reach 31.02%, 39.86%, and 40.32%.

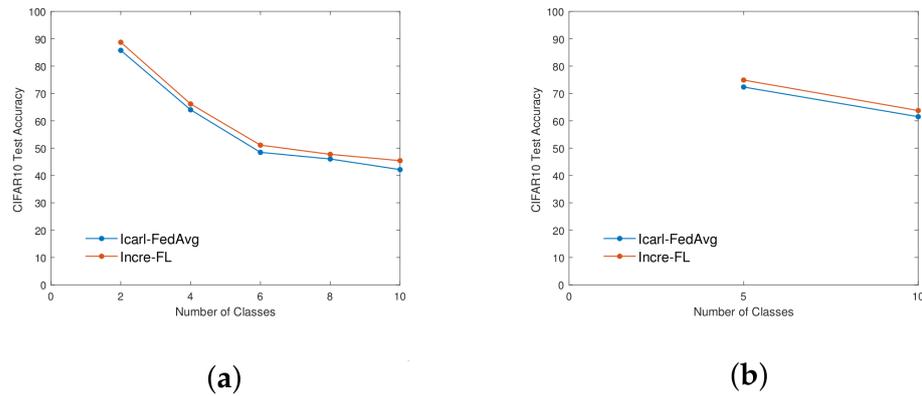


Figure 13. The accuracy curve of the feature attention mechanism module in the test set Cifar10. (a) Accuracy graph when the class increment is 2. (b) Accuracy graph when the class increment is 5.

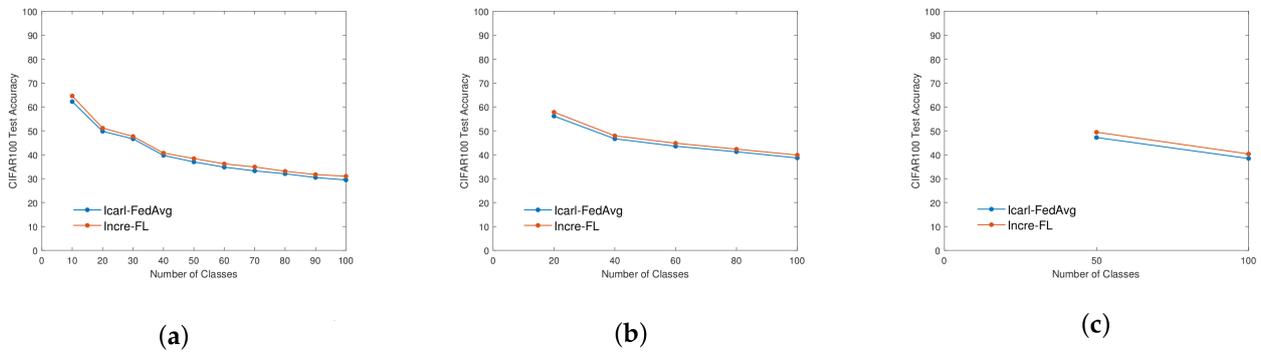


Figure 14. The accuracy curve of the feature attention mechanism module in the test set Cifar100. (a) Accuracy graph when the class increment is 10. (b) Accuracy graph when the class increment is 20. (c) Accuracy graph when the class increment is 50.

The influence of the feature attention mechanism in this paper is shown in Table 4. It can be seen from the table that adding the feature space attention mechanism can improve the accuracy of image classification tasks in incremental learning by enhancing the capture performance of crucial information.

Table 4. The influence of the dual attention mechanism module.

Dataset	Increments/Test Classes	Model	Accuracy (%)
CIFAR10	2/10	Icarl-FedAvg	42.14
		Incre-FL	45.40
	5/10	Icarl-FedAvg	61.49
		Incre-FL	63.73
CIFAR100	10/100	Icarl-FedAvg	29.44
		Incre-FL	31.02
	20/100	Icarl-FedAvg	38.65
		Incre-FL	39.86
	50/100	Icarl-FedAvg	38.42
		Incre-FL	40.32

4.3. Comparative Experiment—Overall Accuracy Analysis

Figures 15 and 16 show the test results of the algorithm in this paper, the pure Icarl strategy, and the federated averaging algorithm with the Icarl method on the CIFAR10 and CIFAR100 standard test sets. It can be seen from the figure that the average accuracy of the algorithm in this paper is significantly better than the federated averaging algorithm with the Icarl strategy and lower than the incremental learning algorithm Earl. The accuracy rate of Ince-FL is lower than that of the total learning algorithm Icarl because the gradual learning algorithm Icarl directly uses the neural network CNN to train the dataset. In federated learning, the participants will not expose the data to the server or other parties. Hence, the federated learning model performs slightly worse than the centrally trained model, and the additional security and privacy protection are undoubtedly more valuable than the loss of accuracy.

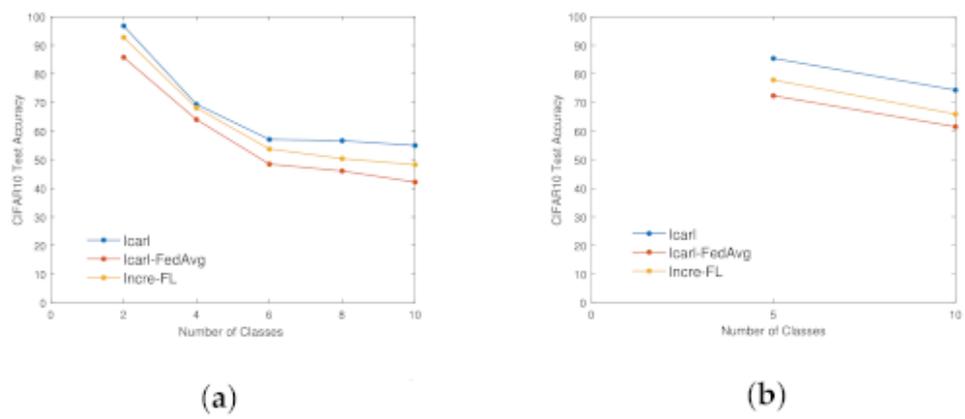


Figure 15. Cifar10 accuracy curve. (a) Accuracy graph when the class increment is 2. (b) Accuracy graph when the class increment is 5.

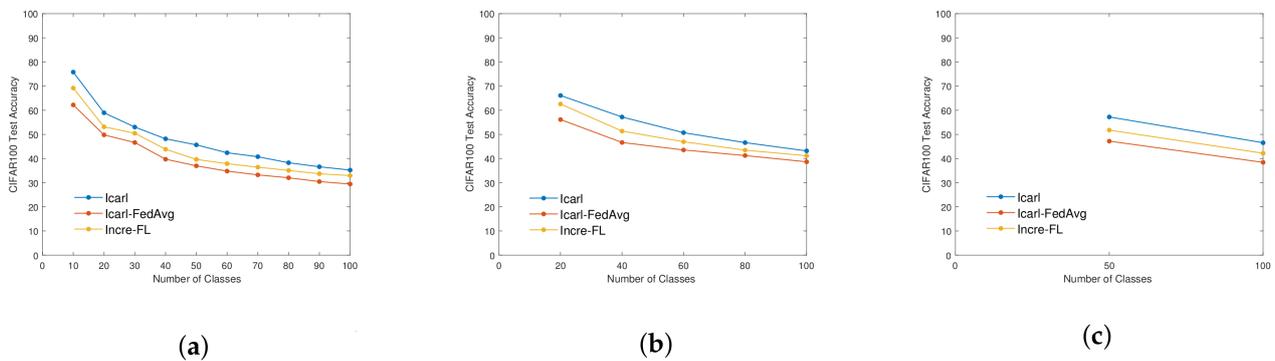


Figure 16. Cifar100 accuracy curve. (a) Accuracy graph when the class increment is 10. (b) Accuracy graph when the class increment is 20. (c) Accuracy graph when the class increment is 50.

Table 5 shows the accuracy of our algorithm on the two datasets. Its test accuracy is higher than the separate addition of the two modules and lower than the test accuracy of CNN training with the Icarl strategy after all client data sets are collected in one place. At the same time, a comparison is made with the Icarl-FedAvg algorithm, the number of clients is set to 2, the client data are all independent and identically distributed, and the comparison is made on two real data sets. Overall, Ince-FL maintains good performance in almost all scenarios. Compared with the Icarl-FedAvg algorithm, the performance improvement of Ince-FL mainly comes from the addition of the pre-training module, which ensures the balance of the pre-training samples and reduces the impact of the large sample client on the global model. We designed a dual attention mechanism module and added the SE module based on the classic graph convolutional neural network. This

module can help the model to obtain the characteristics of the overall samples of each client during model training and effectively reduce the impact of noise. The federated aggregation algorithm based on the feature attention mechanism is added to the global model to enhance the capture performance of the global model for critical feature information from three perspectives. Since a blockchain-oriented online federated incremental learning algorithm proposed by Luo Changyin et al. [16] is not a class incremental method, it is not used as comparative data.

Table 5. The influence of the dual attention mechanism module.

Dataset	Increments/Test Classes	Model	Accuracy (%)
CIFAR10	2/10	Icarl-FedAvg	42.14
		Incre-FL	48.24
	5/10	Icarl-FedAvg	61.49
		Incre-FL	65.90
CIFAR100	10/100	Icarl-FedAvg	29.44
		Incre-FL	32.94
	20/100	Icarl-FedAvg	38.65
		Incre-FL	41.15
	50/100	Icarl-FedAvg	38.42
		Incre-FL	42.19

5. Conclusions

In business, people not only need one to ensure data security but also need to be able to cope with the dynamic changes in training tasks. We need an algorithm that can solve the cost loss caused by retraining all old and new data in the traditional training model in the face of new tasks and data. Therefore, whether the user data can be dynamically updated has become the key to measuring the pros and cons of an algorithm. To study the federated learning algorithm that can cope with the dynamic changes of training tasks and keep the data confidential, this paper proposes a federated incremental learning framework, adding a pre-training module to it, which can improve the convergence speed of the model, and proposes a fusion based on a dual attention mechanism. The strategy can help reduce the impact of considerable sample noise on the final model training results. At the same time, it can strengthen the capture of the importance of features and alleviate the feature imbalance problem when adding incremental learning for dynamic task training to classify incremental learning. The experimental data implemented on the standard data set show that the algorithm can improve the accuracy of the common dataset.

Author Contributions: Conceptualization, K.H. and S.J.; methodology, K.H., M.L. and Y.Y.; software, K.H. and M.L.; validation, K.H. and J.W.; formal analysis, K.H. and Y.L.; investigation, K.H. and S.J.; resources, K.H., F.Z., S.J. and Y.Y.; data curation, K.H.; writing—original draft preparation, K.H., M.L., S.G., S.J. and F.Z.; writing—review and editing, M.L. and Y.Y.; visualization, K.H. and F.Z.; validation, S.G.; supervision, K.H.; project administration, K.H. and Y.Y.; funding acquisition, K.H.; All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of PR China (42075130).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data and code used to support the findings of this study are available from the first author upon request (001600@nuist.edu.cn).

Acknowledgments: The financial support of Nanjing Ta Liang Technology Co., Ltd, Nanjing Fortune Technology Development Co. Ltd is deeply appreciated. The authors would like to express heartfelt thanks to the reviewers and editors who submitted valuable revisions to this article.

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

References

1. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. *arXiv* **2017**, arXiv:1602.05629.
2. McMahan, H.B.; Moore, E.; Ramage, D.; y Arcas, B.A. Federated Learning of Deep Networks using Model Averaging. *Electr. Power Syst. Res.* **2017**, arXiv:1602.05629v3.
3. Qi, M. Light GBM: A Highly Efficient Gradient Boosting Decision Tree. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3146–3154.
4. Chen, X.; Huang, L.; Xie, D.; Zhao, Q. EGBMMDA: Extreme Gradient Boosting Machine for MiRNA-Disease Association prediction. *Cell Death Dis.* **2018**, *9*, 3. [[CrossRef](#)] [[PubMed](#)]
5. Saunders, C.; Stitson, M.O.; Weston, J.; Holloway, R.; Bottou, L.; Scholkopf, B.; Smola, A. Support Vector Machine. *Comput. Sci.* **2002**, *1*, 1–28.
6. Schmidhuber. Deep Learning in Neural Networks: An Overview. *Neural Netw.* **2015**, *61*, 85–117. [[CrossRef](#)] [[PubMed](#)]
7. Haykin, S. *Neural Networks: A Comprehensive Foundation*, 3rd ed.; Macmillan: New York, NY, USA, 1998.
8. Graves, A.; Mohamed, A.R.; Hinton, G. Speech Recognition with Deep Recurrent Neural Networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013.
9. Krizhevsky, A.; Sutskever, I.; Hinton, G. *ImageNet Classification with Deep Convolutional Neural Networks*; Curran Associates Inc.: New York, NY, USA, 2012.
10. Liu, Y.; Chen, T.; Yang, Q. Secure Federated Transfer Learning. *arXiv* **2018**, arXiv:1812.03337.
11. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated Machine Learning: Concept and applications. *arXiv* **2019**, arXiv:1902.04885.
12. Zhuo, H.H.; Feng, W.; Lin, Y.; Xu, Q.; Yang, Q. Federated deep Reinforcement Learning. *arXiv* **2020**, arXiv:1901.08277.
13. Peng, Y.; He, M.; Wang, Y. A federated semi-supervised learning approach for network traffic classification. *arXiv* **2021**, arXiv:2107.03933
14. Li, Z.; Hoiem, D. Learning without Forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 2935–2947. [[CrossRef](#)]
15. Wu, C.; Herranz, L.; Liu, X.; van de Weijer, J.; Raducanu, B. Memory Replay GANs: Learning to generate images from new categories without forgetting. In Proceedings of the The 32nd International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018.
16. Luo, C.; Chen, X.; Ma, C.; Wang, J. An online federated incremental learning algorithm for blockchain. *J. Comput. Appl.* **2021**, *41*, 363.
17. Rebuffi, S.A.; Kolesnikov, A.; Sperl, G.; Lampert, C.H. iCaRL: Incremental Classifier and Representation Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
18. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
19. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
20. Yu, H.; Yang, S.; Zhu, S. Parallel Restarted SGD with Faster Convergence and Less Communication: Demystifying Why Model Averaging Works for Deep Learning. *AAAI Tech. Track Mach. Learn.* **2018**, *33*, AAAI-19. [[CrossRef](#)]
21. Hu, K.; Wu, J.; Li, Y.; Lu, M.; Weng, L.; Xia, M. FedGCN: Federated Learning-Based Graph Convolutional Networks for Non-Euclidean Spatial Data. *Mathematics* **2022**, *10*, 1000. [[CrossRef](#)]
22. Hu, K.; Wu, J.; Weng, L.; Zhang, Y.; Zheng, F.; Pang, Z.; Xia, M. A novel federated learning approach based on the confidence of federated Kalman filters. *Int. J. Mach. Learn. Cybern.* **2021**, *12*, 3607–3627. [[CrossRef](#)]
23. Lin, Z.; Feng, M.; Santos, C.N.D.; Yu, M.; Xiang, B.; Zhou, B.; Bengio, Y. A Structured Self-attentive Sentence Embedding. *arXiv* **2017**, arXiv:1703.03130.
24. Zhao, H.; Jian, J.; Koltun, V. Exploring Self-attention for Image Recognition. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
25. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Gomez, A.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
26. Liu, Y.; Yang, Q.; Chen, T. Tutorial on Federated Learning and Transfer Learning for Privacy, Security and Confidentiality. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019.
27. Douillard, A.; Cord, M.; Ollion, C.; Robert, T.; Valle, E. PODNet: Pooled Outputs Distillation for Small-Tasks Incremental Learning. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020.
28. Zhang, D.; Chen, X.; Xu, S.; Xu, B. Knowledge Aware Emotion Recognition in Textual Conversations via Multi-Task Incremental Transformer. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020.

29. Ahn, H.; Kwak, J.; Lim, S.; Bang, H.; Kim, H.; Moon, T. SS-IL: Separated Softmax for Incremental Learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021.
30. Kim, J.Y.; Choi, D.W. Split-and-Bridge: Adaptable Class Incremental Learning within a Single Neural Network. *arXiv* **2021**, arXiv:2107.01349.
31. Shmelkov, K.; Schmid, C.; Alahari, K. Incremental Learning of Object Detectors without Catastrophic Forgetting. In Proceedings of the IEEE international Conference on Computer vision, Venice, Italy, 22–29 October 2017.
32. Shoham, N.; Avidor, T.; Keren, A.; Israel, N.; Benditkis, D.; Mor-Yosef, L.; Zeitak, I. Overcoming Forgetting in Federated Learning on Non-IID Data. *arXiv* **2019**, arXiv:1910.07796.
33. Hu, G.; Zhang, W.; Ding, H.; Zhu, W. Gradient Episodic Memory with a Soft Constraint for Continual Learning. *arXiv* **2020**, arXiv:2011.07801.
34. Masana, M.; Liu, X.; Twardowski, B.; Menta, M.; Weijer, J.V.D. Class-incremental learning: Survey and performance evaluation. *arXiv* **2020**, arXiv:2010.15277.
35. Fallah, M.K.; Fazlali, M.; Daneshmand, M. A symbiosis between population based incremental learning and LP-relaxation based parallel genetic algorithm for solving integer linear programming models. *Computing* **2021**, 1–19. [[CrossRef](#)]
36. Bielak, P.; Tagowski, K.; Falkiewicz, M.; Kajdanowicz, T.; Chawla, N.V. FILDNE: A Framework for Incremental Learning of Dynamic Networks Embeddings. *Knowl.-Based Syst.* **2021**, *4*, 107453. [[CrossRef](#)]
37. Hu, K.; Jin, J.; Zheng, F.; Weng, L.; Ding, Y. Overview of behavior recognition based on deep learning. *Artif. Intell. Rev.* **2022**, 1–33. [[CrossRef](#)]
38. Cho, K.; Merriënboer, B.V.; Bahdanau, D.; Bengio, Y. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *arXiv* **2014**, arXiv:1409.1259.
39. Krizhevsky, A.; Hinton, G. Learning Multiple Layers of Features from Tiny Images. *Handb. Syst. Autoimmune Dis.* **2009**, *1*
40. Chen, B.Y.; Xia, M.; Huang, J.Q. MFANet: A Multi-Level Feature Aggregation Network for Semantic Segmentation of Land Cover. *Remote Sens.* **2021**, *13*, 731. [[CrossRef](#)]
41. Xia, M.; Wang, T.; Zhang, Y.H.; Liu, J.; Xu, Y.Q. Cloud/shadow Segmentation based on Global Attention Feature Fusion Residual Network for Remote Sensing Imagery. *Int. J. Remote Sens.* **2021**, *42*, 2022–2045. [[CrossRef](#)]
42. Xia, M.; Cui, Y.C.; Zhang, Y.H.; Xu, Y.M.; Liu, J.; Xu, Y.Q. DAU-Net: A Novel Water Areas Segmentation Structure for Remote Sensing Image. *Int. J. Remote Sens.* **2021**, *42*, 2594–2621. [[CrossRef](#)]
43. Xia, M.; Liu, W.A.; Wang, K.; Song, W.Z.; Chen, C.L.; Li, Y.P. Non-intrusive Load Disaggregation based on Composite Deep Long Short-term Memory Network. *Expert Syst. Appl.* **2020**, *160*, 113669. [[CrossRef](#)]
44. Xia, M.; Zhang, X.; Liu, W.A.; Weng, L.G.; Xu, Y.Q. Multi-stage Feature Constraints Learning for Age Estimation. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 2417–2428. [[CrossRef](#)]
45. Hu, K.; Ding, Y.; Jin, J.; Weng, L.; Xia, M. Skeleton Motion Recognition Based on Multi-Scale Deep Spatio-Temporal Features. *Appl. Sci.* **2022**, *12*, 1028. [[CrossRef](#)]
46. Hu, K.; Weng, C.; Zhang, Y.; Jin, J.; Xia, Q. An Overview of Underwater Vision Enhancement: From Traditional Methods to Recent Deep Learning. *J. Mar. Sci. Eng.* **2022**, *10*, 241. [[CrossRef](#)]
47. Hu, K.; Chen, X.; Xia, Q. A Control Algorithm for Sea–Air Cooperative Observation Tasks Based on a Data-Driven Algorithm. *J. Mar. Sci. Eng.* **2021**, *9*, 1189. [[CrossRef](#)]
48. Lu, E.; Hu, X. Image super-resolution via channel attention and spatial attention. *Appl. Intell.* **2021**, *10*, 2260–2268. [[CrossRef](#)]
49. Chen, J.; Yang, L.; Tan, L.; Xu, R. Orthogonal channel attention-based multi-task learning for multi-view facial expression recognition. *Pattern Recognit.* **2022**, *129*, 108753. [[CrossRef](#)]
50. Yao, L.; Ding, W.; He, T.; Liu, S.; Nie, L. A multiobjective prediction model with incremental learning ability by developing a multi-source filter neural network for the electrolytic aluminium process. *Appl. Intell.* **2022**, *1*, 1–23. [[CrossRef](#)]
51. Yuan, K.; Xu, W.; Li, W. An incremental learning mechanism for object classification based on progressive fuzzy three-way concept. *Inform. Sci.* **2022**, *584*, 127–147. [[CrossRef](#)]