Federated Learning Algorithm Based on Knowledge Distillation

Jiang, Donglin; Shan, Chen; Zhang, Zhihui

University of Nottingham Ningbo China, 199 Taikang East Road, Ningbo, 315100, Zhejiang, China.

First published 2021

Federated Learning Algorithm Based on Knowledge Distillation

1st Donglin Jiang
School of Computer Science
University of Nottingham
Ningbo, China
scydj2@nottingham.edu.cn

2nd Chen Shan
School of Computer Science
University of Nottingham
Ningbo, China
scycs2@nottingham.edu.cn

3rd Zhihui Zhang
School of Mathematical Sciences
University of Nottingham
Ningbo, China
smyzz6@nottingham.edu.cn

***Abstract*—Federated learning is a new scheme of distributed machine learning, which enables a large number of edge computing devices to jointly learn a shared model without private data sharing. Federated learning allows nodes to synchronize only the locally trained models instead of their own private data, which provides a guarantee for privacy and security. However, due to the challenges of heterogeneity in federated learning, which are: (1) heterogeneous model architecture among devices; (2) statistical heterogeneity in real federated dataset, which do not obey independent-identical-distribution, resulting in poor performance of traditional federated learning algorithms. To solve the problems above, this paper proposes FedDistill, a new distributed training method based on knowledge distillation. By introducing personalized model on each device, the personalized model aims to improve the local performance even in a situation that global model fails to adapt to the local dataset, thereby improving the ability and robustness of the global model. The improvement of the performance of local device benefits from the effect of knowledge distillation, which can guide the improvement of global model by knowledge transfer between heterogeneous networks. Experiments show that FedDistill can significantly improve the accuracy of classification tasks and meet the needs of heterogeneous users.**

Keywords—Federated learning, Knowledge distillation, Non-independent-identical-distribution, Heterogeneous network.

I. INTRODUCTION

Traditionally, deep learning model is trained on a single system or cluster by concentrating the data from distributed sources. However, the lack of effective development of data resources seriously limits the circulation of data, computing and capacity and richness of data sets in many applications. In order to better protect users' data and privacy, Google has proposed federated learning [1]: after the central server sends the initial model to distributed terminals, the terminal users can use their own data to train the model, and the local weight updating information is sent back to the central server, and then the updated model is broadcast to all the edge nodes after data aggregation by the central server. It is repeated in this way until the training termination condition is reached.

In the federated learning process, there is no exchange of original training data between edge nodes or between edge nodes and the central server. Therefore, this method can avoid the leakage of personal data to a large extent and protect data privacy of data side users [2]. In practice, federated learning faces the challenge of heterogeneity [3]: the heterogeneity of different nodes, including device heterogeneity, data heterogeneity and model heterogeneity. In the initial federated framework, users of all nodes must follow the model setting of the central parameter server, and they must be isomorphic models [4]. However, in the real environment, each participant tends to design his own personalized model according to the local data characteristics and his own preference.

Data heterogeneity [5] refers to that under the federated learning framework, the data of each client node does not satisfy independent-identical-distribution, and the local data of the device cannot be regarded as the random part of the sample data extracted from the overall distribution. Therefore, the data between any two devices may be two different data distributions. Sattler et al. [6] discuss that when the data set is non-independent-identical-distribution, the reduction of accuracy is inevitable.

In order to deal with the problem of heterogeneous models, one solution is to allow the global model to be updated locally and introduce personalized models. FedMD [4] points out that the personalized model can be trained on the public data set by means of transfer learning, and then transferred to the global model. FedProx [7] algorithm points out that the performance of the model is improved by limiting the range of model parameter dispersion to prevent the sharp reduction of accuracy caused by overdispersion of model parameters.

Similarly, for the problem of non-independent-identical-distribution data, Li et al. [8] prove that the convergence rate is closely related to the number of local iterations and that the learning rate must be attenuated when the data set of FedAvg algorithm is non-independent-identical-distribution. Zhao et al. [3] point out the phenomenon of precision attenuation of federated learning in the case of non-independent-identical-distribution is caused by the excessive dispersion of model parameters. However, in the case of independent-identical-distribution, the parameters of each model are always similar within the error range.

In this paper, FedDistill algorithm is proposed. It is a new algorithm that uses knowledge distillation to improve the performance of the model in federated learning. It meets the needs of heterogeneous networks and data. In theory, these improvements provide convergent guarantee and consider the influence of heterogeneity.

II. Related Work

A. Problem Description

For machine learning tasks, we hope to minimize the cost function, that is, $\min_{\omega\in R^d} f(\omega)$, where $f(\omega)=\frac{1}{n}\sum_{i=1}^{n} f_i(\omega)$. For classical machine learning tasks, we define the cost function as $f_i(\omega)=l(x_i, y_i;\ \omega_i)$, which is the prediction error on the data set when the parameter of the model is ω_i [9][10][11]. Suppose that there are K nodes, P_k represents the data set on each node, $n_k = |P_k|$ shows the size of the data set of each node, the equation can be specified as follows:

$$\min_{\omega}\left\{F(\omega)\triangleq\sum_{k=1}^{N}p_kF_k(\omega)\right\}\qquad(1)$$

Suppose there are K nodes in a learning task, and each node has local data. At the beginning of each calculation, set a random factor C randomly, select some nodes, and then the central server sends the current global information to each node (such as the current global parameters). After receiving the initialization parameter ω_0, each selected node train its own model based on its local data set, updates local model and sends updated patameters to the central server. After aggregation of multiple nodes, central parameter server uses the updated information to train the global model again, and then sends the latest model parameters to each data node. The above process is an iterative process.

If the FedAvg algorithm communicates R times and updates E times locally, then, for the t-th communication, the central server broadcasts the latest model parameter ω_t to each device [12]. In the local update phase, make $\omega_t^k=\omega_t$, local training iterates E times, which is mathematically represented as follows:

$$\omega_{t+i+1}^{k}\leftarrow\omega_{t+i}^{k}-\eta_{t+i}\nabla F_k\left(\omega_{t+i}^{k},\xi_{t+i}^{k}\right)\qquad(2)$$

where η represents the learning rate, ξ is the way to extract data from device nodes.

After N nodes have been trained, their model parameters $\omega_t^1,\ \omega_t^2,\ \cdots,\ \omega_t^N$ will be transmitted to the central server, and the latest model parameters are aggregated by averaging the model parameters. Because non-independent-identical-distribution data and stochastic gradient descent method, the model after each aggregation may be different.

In particular, FedAvg algorithm needs to communicate twice in each round, which are the model parameters broadcast down from central server and the parameters are aggregated back up. Assuming that the total number of iterations is T, that is $T = 2R\times E$. Therefore, in this paper, the number of local distillations is increased to reduce the communication cost while keeping the same iteration times T.

B. Knowledge Distillation

Hinton et al. [9] propose that knowledge distillation uses soft targets of deep complex neural network as regularizer to constrain the loss of simpler neural network. Knowledge distillation supports the transfer of knowledge from a trained large model to a small model without changing the structure of the small model, so as to achieve the purpose of model compression. In the process of training small models, not only use traditional hard targets, but also define knowledge as soft targets (such as the output of large models softmax layer) and use large model to guide small model training. Taking the classification task as an example, soft targets enable student network obtain not only the class labels, but also the information related to the relationship between classes of data set to achieve better results.

In multi classification task, neural network usually uses "softmax" output layer to convert multi classification output into probability, while knowledge distillation improves "softmax" to soften it. Knowledge distillation is mathematically represented as:

$$q_i=\frac{\exp(Z^i/T)}{\sum_j\exp(Z^j/T)}\qquad(3)$$

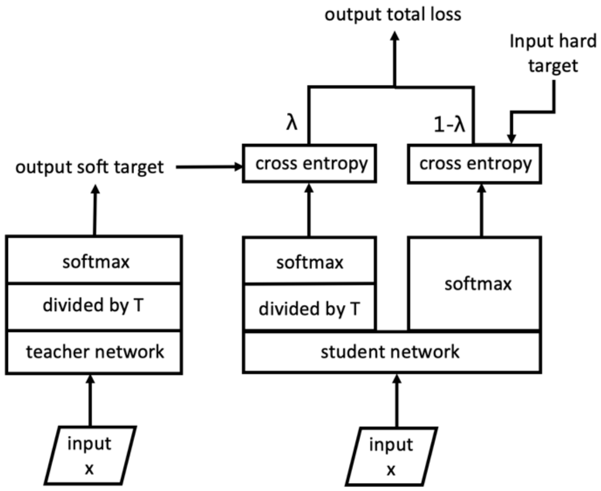

Fig. 1. Knowledge Distillation.

where Z^i is the prediction of each class, T is a hyper-parameter introduced by knowledge distillation. A higher value for T will produce a softer probability distribution.

III. FedDistill: The Proposed Solution

A. *System model*

Under the premise of the basic process of FedAvg, this paper proposes FedDistill algorithm that introduces personalized model. In this algorithm, each node has two models: (1) local node i copies the global model, which is recorded as θ_i; (2) local node i designs a personalized model Γ_i independently. Based on the knowledge distillation introduced above, the personalized model $\Gamma_i$is used as a teacher network to guide the student network θ_i, and then the collaborative training model θ_i is sent back to the central server for aggregation.

B. *Proposed scheme*

In distributed machine learning, averaging model parameters is a simple and efficient method. In the case of ideal data distribution, the model does not infringe the privacy data of other devices and can meet the needs of collaborative training model.

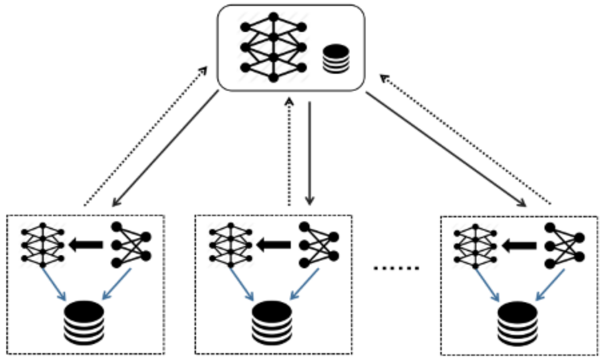

Fig. 2. FedDistill Algorithm.

Zhao et al. point out that [3] in the non-independent-identical-distribution federated learning scenario, each device node can learn its own data set D_i, but the performance of global model on test data set $D_0 = \{D_1, D_2, \cdots, D_N\}$ is poor, which has a sharp reduction of precision, and the convergence rate of federated learning model also affected. We suppose that we artificially construct a strongly convex and smooth distributed optimization problem.

Assumption [8]: for functions $F_1, F_2, \cdots, F_N$ are all μ-strongly convex, that for all v and ω, can be mathematically represented as:

$$F_i(v) \geq F_i(\omega) + (v-\omega)^T \nabla F_i(\omega) + \frac{\mu}{2} ||v-\omega||_2^2 \quad (4)$$

For a node, it is expected that the aggregated model parameter ω^t is as close to the optimal solution ω^*as possible. That is:

$$||\omega^t - \omega^*||_2 = \Omega \bullet ||\omega^*||_2 \quad (5)$$

where Ω is the boundary function, which shows the relation of the number of local iterations E and learning rate η.

Through knowledge distillation, knowledge is transferred from the local personalized model to the global model. It is shown in experiments that this method can get a solution closer to ω^*. Algorithm 1 summarizes all this.

Algorithm 1. FedDistill

```
Function FedDistill (N,E,α)
for t=1 to N do:
    m = max(C*K,1)
    $S_t$ = the set of m nodes
    for each k ∈ $S_t$ do:
        $\omega_{t+1}^{k}$ = Distill(k, $\omega_t$, $\omega_t'$)
        $\omega_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} \omega_{t+1}^{k}$
Function Distill(k,ω,q):
  Γ ⟵ divide $P_k$ into data set which batchsize is B
  for i = 1 to E do:
    for each b ∈ Γ do:
        ω = ω − α ∇l(ω, b, q)
  return ω
```

IV. Experiment and Result Analysis

A. *Simulation Design*

In order to verify the effectiveness of the algorithm, FedDistill is evaluated on different tasks, models and data sets in this paper. In order to better describe data heterogeneity and its impact on convergence, a combined data is evaluated to deal with statistical heterogeneity more accurately. By assigning data to different devices, the data characteristics of federated learning that are non-independent-identical-distribution are fully considered. 10 classification tasks are taken as an example, five edge device nodes are selected, so each node has only two kinds of data samples. The real public data set and synthetic data set are introduced as follows.

1) Public data set

The task of image classification, which is common in machine learning, is used to do the experiment. MNIST and CIFAR10 are selected as the basic data sets, and both are 10 classification tasks: MNIST contains 60000 training dataset and 10000 test dataset (image size: 28 × 28). CIFAR10 data set contains 50000 training dataset and 10000 test dataset, and each image is 32 × 32 coloured image. Because the data set can quickly judge the performance of different algorithms, it is widely used. In the experiment, we reinforce the CIFAR10 dataset by rotating and cropping.

2) Synthetic dataset

In order to generate data set, the synthetic (α, β) function is constructed to generate different data distribution. For each node i, the data (X_i, Y_i) is constructed according to the model $\mathrm{Y} = \mathrm{argmax}(\mathrm{softmax}(\omega \mathrm{X} + \mathrm{b}))$. The parameter α controls the distribution of ω and b, and the parameter β controls the data distribution of X. Actually, α and β are the value of variance. The larger the values of α and β, the more difficult it is to train a good model for federated learning.

B. Results and Performance Evaluation

In order to avoid randomness of the experiment, the research includes three models: (1) CNN1: a simple convolutional neural network with three convolution kernels (size 3x3): 32 channels in the first layer and 64 channels in the second and the third layers, with a 2x2 pooling layer in the first two layers and two fully connected layers (1024 and 64 units respectively) after the third layer; (2) CNN2: a convolutional neural network, with three convolution kernels (size 3x3). All three layers have 128 channels, with a 2x2 pooling layer in the first two layers and a fully connected layer (2048 units) following the third layer. All layers use ReLU as the activation function. CNN2 has more parameters, complex architecture and stronger performance; (3) Logistic regression model, which is used to synthesize data sets.

1) Validity of knowledge distillation

In order to verify the effectiveness of knowledge distillation in federated learning, CNN1 network is selected to verify in MNIST data set, compared with FedAvg algorithm as baseline. Set batchsize = 128, learning rate α = 0.01. The accuracy of central server node is selected as evaluation index, the number of global iterations N and the number of local iterations E are adjusted. The results of experiment are shown in Table I.

The results of this experiment show that knowledge distillation can improve the performance of local nodes and global node at the same time. This promotion is due to a better local performance which leads to the corresponding improvement of the global average accuracy after aggregation.

TABLE I. MNIST Experiment Table

Algorithm	N	E	Local Node Acc	Central Node Acc
FedAvg	10	1	97.30%	
FedDistilll			98.41%	97.48%
FedAvg	20	1	98.49%	
FedDistilll			98.89%	98.69%
FedAvg	10	5	97.60%	
FedDistilll			98.52%	98.26%
FedAvg	50	1	98.82%	
FedDistilll			99.09%	98.70%

2) Synthetic data experiment

Due to the improvement caused by distillation, experiments are continued on the synthetic data. Logistic regression model is used to do classification task on synthetic data, and the values of parameters α and β are controlled as 1. The results of the experiment are shown in Fig. 3. It can be found that FedDistill can gain higher accuracy and converge more quickly in the early stage.

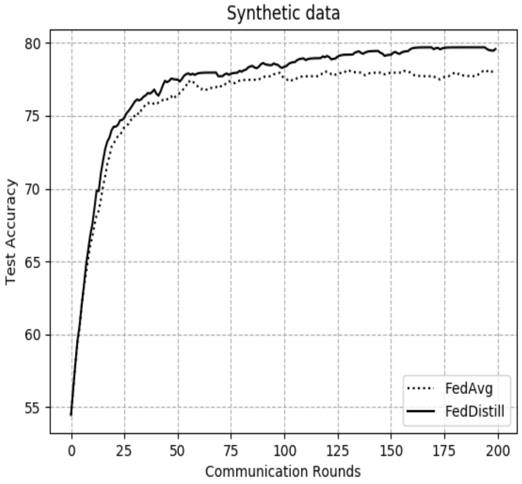

Fig. 3. Synthetic Data Experiment.

3) Reduce communication cost

In federated learning, the communication bottleneck is related to both communication times and communication contents. Here we reduce the communication cost relying on the increase of local update times in the same configuration. In the case of $\mathrm{T} = 2\mathrm{R} \times \mathrm{E}$, keeping T total traffic fixed, increasing the local update times E so as to reduce communication cost R. CNN1 and CNN2 were selected to test on CIFAR10 dataset. According to the experimental results in Table II, it can be found that increasing the value of local times E can not only improve the accuracy of model, but also reduce the communication cost. However, it should be noted that if the value of E is too large, the value of R will be too small, and at this time the accuracy of the model will reduce, which is due to the lack of communication round. It can be seen that there is a trade-off phenomenon in the value of E and a reasonable E value can lead to considerable improvement.

TABLE II. Communication Cost Table

N=50, batchsize=128, K=4, CIFAR10						
Model	E	Edge 1	Edge 2	Edge 3	Edge 4	Central Device
CNN1	2	59.62%	59.17%	58.99%	59.86%	52.94%
CNN2	2	62.60%	62.63%	63.15%	62.73%	63.59%
CNN2	5	69.34%	69.22%	69.06%	69.18%	70.49%
CNN2	**20**	**60.18%**	**60.14%**	**60.25**	**60.59%**	**61.49%**

V. Conclusion

The research on distributed machine learning has great significance. In recent years, federated learning is an important development of distributed machine learning. Compared with the traditional distributed machine learning algorithm, it pays more attention to data privacy protection. However, the existing federated learning framework does not support the heterogeneous distributed model, which has some disadvantages. The paper proposes FedDistill algorithm, testing its accuracy and communication cost to compare with FedAvg algorithm. The experimental results show that the proposed algorithm can be used to assist the central node to train other client nodes and can improve the whole model learning ability.

This paper has not considered how to eliminate the disadvantages of average aggregation. In the future, we will further consider how to propose a more efficient and real-world distributed machine learning method.

References

[1] McMahan, H.B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.Y. (2016) Communication-Efficient learning of deep networks from decentralized data. In: International Conference on Artificial Intelligence and Statistics. Fort Lauderdale.

[2] Geyer, R.C., Klein, T., Nabi, M. (2017) Differentially private federated learning: a client level perspective. In: Conference on Neural Information Processing Systems. Long Beach.

[3] Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V. (2018) Federated learning with Non-IID data. arXiv: 1806.00582.

[4] Li, D., Wang, J. (2019) FedMD: heterogenous federated learning via model distillation. In: Conference on Neural Information Processing Systems. Vancouver. arXiv: 1910. 03581.

[5] Li, T., Sahu.A.K., Talwalkar, A., Smith, V. (2020) Federated learning: challenges, methods, and future directions. IEEE Signal Processing Magazine, 37(3): 50-60.

[6] Sattler, F., Wiedemann, S., Müller, K., Samek, W. (2019) Robust and communication-efficient federated learning from Non-IID Data. IEEE transactions on neural networks and learning systems. IEEE, Piscataway. pp. 1-14.

[7] Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V. (2018) Federated optimization in heterogeneous networks. In: Conference on Machine Learning and Systems. Austin.

[8] Li, X., Huang, K., Yang, W., Wang, S., Zhang, Z. (2020). On the convergence of FedAvg on Non-IID data. In: International Conference on Learning Representations. Addis Ababa.

[9] Hinton, G., Vinyals, O., Dean, J. (2015) Distilling the knowledge in a neural network. In: NIPS Deep Learning and Representation Learning Workshop.

[10] Jimmy, B., Rich, C. (2014) Do deep nets really need to be deep? In: Advances in Neural Information Processing Systems. Montreal. pp. 2654-2662.

[11] Buciluǎ, C., Caruana, R., Niculescu-Mizil, A. (2006) Model compression. In: International conference on Knowledge discovery and data mining. Philadelphia. pp. 535-541.

[12] Yang, Q., Liu, Y., Chen, T., Tong, Y. (2019) Federated machine learning: concept and applications. J. ACM Transactions on Intelligent Systems and Technology. 10(2): 1-19.