Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/39/20230587

Model optimization method based on the MOON algorithm

Haogang Cao[1,4], Chenxin Dong[2] and Jingyi Zhou[3]

[1]School of Mechanical and Electronical Engineering,Xi'an University of Architecture and Technology Huaqing College,Shaanxi,710043,China
[2]School of Information Science and Engineering,Wuhan University of Science and Technology,Hubei,430081,China
[3]School of Data Engineering,Tianjin University of Finance and Economics Pearl River College,Tianjin,301811,China

[4]20070875@huanghuai.edu.cn

Abstract. Federated Learning is a revolutionary approach to machine learning. Its purpose is to enable multiple participants to collaboratively train machine learning models without the need to share local data. The main objective is to address issues related to data privacy and security. In traditional machine learning, data is typically centralized and stored in a single location or on cloud servers for training. However, this centralized training approach carries risks of potential data leakage, especially concerning sensitive and critical information. Industries such as healthcare and finance, which involve sensitive data, place a premium on safeguarding data privacy. Furthermore, in cases where data cannot be easily transferred or is subject to privacy regulations, centralized methods may face limitations. Federated Learning revolutionizes the conventional approach by introducing a decentralized model training process. It maintains data decentralization while achieving collaborative model optimization, greatly enhancing data privacy and security. The MOON algorithm, an integral part of federated learning, contributes to its novelty. As a significant component, the MOON algorithm facilitates new possibilities for federated learning. In this article, the research will elaborate on the MOON algorithm within the context of federated learning. And this article will delve into its description and optimization, elucidating how it enhances federated learning.

Keywords: Federated Learning, Contrast Learning, Data Privacy and Security, MOON Algorithm.

1. Introduction

With the rapid advancement of Internet of Things (IoT) technology and the social interconnectedness of the Internet, the number of issues at the edge of the Internet has increased exponentially [1]. Despite the emergence of distributed machine learning as an Introduction learning has been proposed by researchers to tackle large-scale data processing challenges. In comparison to distributed learning, federated learning doesn't substantially differ in modeling and training methods from traditional inference problems. While participants in federated learning can engage more deeply and flexibly in comparison, there are still different assumptions and requirements regarding ownership and privacy attributes of large datasets.

The primary objective of distributed learning is to delegate a task to multiple computing nodes, utilizing task parallelism to enhance modeling and training efficiency. This approach involves sampling

DOI: 10.54254/2755-2721/39/20230587

all node datasets into a central database, forming a uniform distribution range and scale. In contrast, federated learning is more focused on machine learning with heterogeneous datasets, where data conditions may vastly differ among different computing nodes. Federated learning, at its core, is a form of distributed machine learning primarily designed to address data privacy and security concerns. Its key idea is to establish federated machine learning AI models that guarantee privacy and security for different participants while utilizing techniques like homomorphic encryption, privacy-preserving intersections, and intermediate data exchanges to solve specific tasks.

The advantages of federated learning are evident, especially in safeguarding data privacy. However, since it allows multiple participants to collaboratively train a machine learning model, the heterogeneity in data distribution among participants can lead to decreased performance. To address this, the concept of training a global model over the entire dataset has been introduced to better represent the data, which also gives rise to contrastive training schemes like the MOON framework. MOON's core concept involves localized training for each party by exploiting differences in model representations, essentially employing contrastive training at the model level.

In the realm of machine learning, contrastive learning stands as a technique wherein algorithms develop the capacity to distinguish between data points exhibiting similarities and those displaying dissimilarities. The primary aim involves the acquisition of data representations that encapsulate the foundational structure and interconnections among discrete data points. Within the paradigm of contrastive learning, algorithms are instructed to amplify the semblance among akin data points while concurrently attenuating the resemblance among disparate data points. The attainment of this objective conventionally relies on training protocols oriented toward predicting the categorical affiliation of a pair of data points. Notably, contrastive learning has permeated diverse domains, including image recognition, natural language processing, and speech recognition, thereby demonstrating its broad utility.

MOON stands as a concise and efficient federated machine learning framework, employing innovative models and contrastive design to address non-independent synchronous statistical challenges. MOON outperforms other federated learning algorithms on various image distribution datasets. Moreover, with minimal adjustments to FedAvg, MOON can significantly enhance accuracy by a substantial percentage.

2. Related work

2.1. Comparative Learning in Federated Learning

Federated learning is a decentralized learning methodology to enable multiple participants to collectively train models without the need to share raw data. Contrastive learning plays a pivotal role within the realm of federated learning. It is a technique that employs the comparison of sample similarities to facilitate learning, thereby enhancing the performance and privacy-preserving capabilities of federated learning. Over time, as technology has evolved, the study of contrastive learning within the context of federated learning has witnessed substantial progress. Exemplifications of this progress are observed in siamese networks and pseudo-siamese networks, both of which are neural network architectures utilized for contrastive learning tasks. These architectures contribute to the model's understanding of relationships among samples. For instance, algorithms like SimCLR and BARLOW TWINS are representative of contrastive learning based on siamese networks, while BYOL and MoCo represent contrastive learning based on pseudo-siamese networks [2]. Additionally, FedMatch [3] represents a progression in contrastive learning. It leverages shared knowledge among participants to enhance individual participants' model training capabilities.

Beyond advancing research, contrastive learning in federated learning finds extensive practical application. In federated learning scenarios, data is typically stored locally on devices without the capacity for direct data sharing. However, contrastive learning permits model training without sharing data, thereby affording heightened privacy protection [4]. Through contrastive learning, models can be imbued with improved robustness, elevating their ability to discern subtle differences among distinct samples [5].

DOI: 10.54254/2755-2721/39/20230587

2.2. Research and Applications of the MOON Algorithm

Given that the heterogeneity arising from diverse participants in federated learning can lead to divergent data distributions and potentially hamper performance, a solution is proposed in the form of the MOON-based contrastive learning method. This approach aims to enhance model centralization and generalization. Nevertheless, the increased complexity of the MOON algorithm may result in longer computation times. Furthermore, since the MOON algorithm involves certain hyperparameters, experimentation is required to validate the impact of these parameters on accuracy, ultimately guiding optimal hyperparameter selection. Comparative analysis indicates that when contrasted with algorithms like FedAvg, Fedprox, and SCAFFOLD, the MOON algorithm attains the highest accuracy, underscoring its superior precision compared to other algorithms [6].

Federated learning has evolved into a highly significant domain within the landscape of machine learning. Its applications span various fields, including but not limited to medical imaging [7] and object detection [8]. The MOON algorithm stands out in image-related applications compared to other advanced algorithms within the federated learning domain [6]. This observation underscores the substantial latent potential of the MOON algorithm for widespread application and the value of research.

3. Methodology

3.1. Federated Learning Framework Structure

Promising in the field of machine learning is Federated Learning (FL), which enables multiple clients to train together without sharing private data [9]. It is a distributed machine learning approach that aims to train models using data distributed across different devices while preserving data privacy. To enhance the performance of almost any machine learning algorithm, a very easy way is to train many different models on the same data and then average their predictions [10].

The general process of federated learning is as follows:

1. Participant Selection: Determine the participants in federated learning, which can include individual devices, edge devices, cloud servers, etc.

2. Model Initialization: Before starting federated learning, participants need to initialize a common model. Typically, this model is a pre-defined structure such as a neural network or decision tree.

3. Data Distribution: Each participant shares their local data with other participants, without sharing the raw data. Usually, data is protected using techniques such as encryption or differential privacy to ensure data privacy.

4. Local Training: Each participant trains the common model using their local data. This training process can involve multiple rounds, including steps such as forward propagation, backward propagation, and parameter updates.

5. Model Aggregation: After each round of local training, participants upload their locally trained model parameters to a central server. The central server aggregates the received parameters, typically through a simple averaging operation, to obtain an updated global model.

6.For the next round of local training, participants can employ the updated global model provided by the central server as their initial point. The central server forwards the modified global model to the participants for this purpose.

7. Iterative Process: The process of local training, model aggregation, and model update is repeated until a predefined stopping condition is met, such as reaching a certain number of training rounds or achieving model performance convergence.

DOI: 10.54254/2755-2721/39/20230587

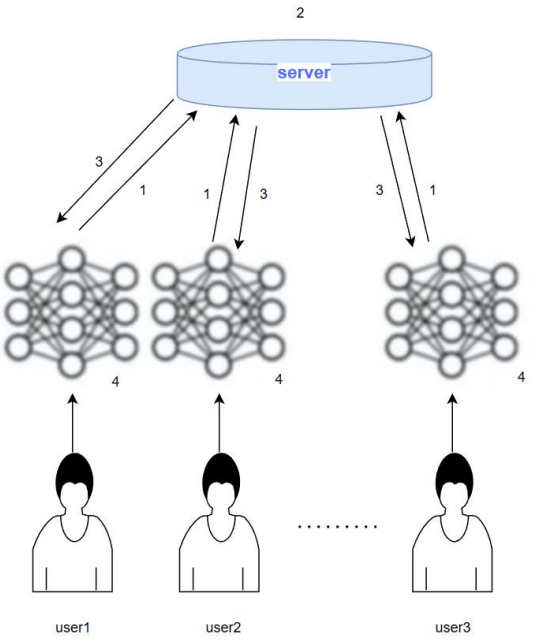

Figure 1. Federated Learning Framework Architecture (Photo/Picture credit :Original).

Figure 1 provides a simple illustration of the framework structure of federated learning, consisting of four components: 1, 2, 3, and 4. Among them, 1 refers to the "Transmit encryption gradient," which signifies the encryption of gradients during transmission as a privacy-preserving measure, ensuring the security of data during communication. 2 is denoted as "Security polymerization," referring to the secure aggregation strategy employed when consolidating model parameters from various participants to safeguard the data privacy of each participant. 3 represents "Back shared model updates," indicating the process of participants sharing updated information after updating their local models in federated learning, contributing collaboratively to enhance the global model. 4 stands for "Updating local model," signifying the continual improvement of the global model as each participant updates their local model based on the shared updates received from other participants. These concepts play vital roles in federated learning, ensuring both the security and performance enhancement of the model.

3.2. Algorithm Optimization

3.2.1. Enhancing Stability and Accuracy

In the MOON algorithm, we have introduced multi-threading to train multiple clients concurrently, leveraging multiple processing cores or resources to potentially improve overall efficiency.

Adjusting the batch size and learning rate are two important factors in improving the stability and accuracy of the algorithm. If the training set is relatively small, choosing a smaller batch size may be more appropriate to avoid overfitting. Conversely, if the training set is large, a larger batch size may be more efficient. Larger batch sizes consume more memory, so if memory is limited, a smaller batch size should be chosen based on available resources. Larger batch sizes allow for parallel processing of more samples, improving the utilization of computational resources. However, computational resource limitations should also be taken into account. Complex models may require smaller batch sizes to reduce the variance of model parameter updates, thus improving stability. We tested multiple batch sizes and ultimately chose a smaller batch size (32) that provides better visualization of the learning curve during training and is easier to debug and diagnose. Additionally, a smaller batch size can also improve the model's generalization ability. We started with smaller values for the learning rate, such as 0.1, 0.01,

DOI: 10.54254/2755-2721/39/20230587

and 0.001, and adjusted them based on experimental results. Further adjustments can be made as needed. A smaller learning rate may require longer training time but may converge to a local minimum more easily. A larger learning rate may accelerate training but can also lead to instability or divergence during training.

3.2.2. Reducing Runtime and Improving Efficiency

Model distillation, also known as model compression, is a technique used to compress complex models by transferring the knowledge from a large, high-performance model (referred to as the teacher model) to a smaller, lightweight model (referred to as the student model) while maintaining high performance. The goal of model distillation is to reduce model complexity and computational resource requirements while preserving high performance. The basic principle of model distillation is to use the teacher model's predictions as target labels and compare them with the student model's predictions. The student model is trained by minimizing the difference between the two using a loss function such as cross-entropy or mean squared error. Through this process, the student model learns the knowledge of the teacher model, including prediction capabilities, feature representations, and decision rules. In model distillation, the teacher model is typically a complex deep neural network with high accuracy and complexity. The student model, on the other hand, is a simplified model, often a shallow neural network or a linear model. Through distillation, the student model can achieve high accuracy while having a smaller model size and faster inference speed.

Model distillation has applications in various fields, including natural language processing, computer vision, and speech recognition. It can be used to deploy complex models on resource-constrained devices such as mobile devices and embedded systems. Additionally, model distillation can be used for transfer learning and model interpretability analysis, aiding in understanding and explaining the decision-making process of the model.

In this optimization, we used two models: Convolutional Neural Network(CNN) and Deep Neural Network(DNN), and two datasets: CIFAR-10 and MNIST. The CNN model was too complex, so we simplified it by reducing the number of convolutional layers and the dimensionality of the hidden layers. The distilled simplified models generated from this process have fewer parameters and computational complexity, enabling them to perform faster during the inference stage. This is beneficial for the MOON algorithm. Additionally, to further improve training speed, We distilled the dnn model [11], and we utilized GPUs for computation.

3.2.3. Enhancing Security

MOON falls under the domain of federated learning, where data security is a crucial consideration. Since the participants in federated learning may possess sensitive data, it is necessary to take appropriate security measures to protect the privacy and confidentiality of the data. Therefore, we implemented preliminary encryption of the data using the cryptography library provided by Python.

4. Experimental Design and Result Analysis

Within the scope of experimental, we have prepared two distinct datasets, denoted as MNIST and CIFAR-10, to serve as the focal subjects of investigation. To initiate this discourse, we shall embark upon an isolated exegesis pertaining to the aforementioned datasets. The MNIST dataset, an assemblage of images encapsulating handwritten numerical representations, is encompassed by four distinct archives, collectively enumerating 60,000 instances of training images, concomitant with an equivalent number of training labels, in addition to 10,000 instances of test images, juxtaposed with their corresponding test labels. This corpus is dichotomously stratified, bifurcating into two principal segments: a training dataset constituting 60,000 data points and a test training dataset comprising 10,000 data points. The former, embracing 60,000 data points, is further partitioned into a training subset, encompassing 55,000 data points, and a complementary validation subset, comprising 5,000 data points. Conversely, CIFAR-10 constitutes an expansive compendium tailored for the domain of computer vision, calibrated to address the ambit of generic object recognition. Within this domain, 60,000 images,

DOI: 10.54254/2755-2721/39/20230587

each possessing dimensions of 32x32 pixels and characterized by RGB color profiles, are hierarchically organized into a taxonomic framework encompassing ten distinct classes. This corpus is differentially apportioned, with a training subset accommodating 50,000 images, while the remaining 10,000 images compose the test subset. The methodologies harnessed for our analytical deliberations encompass the Convolutional Neural Network (CNN) and the Deep Neural Network (DNN). The model consists of three layers: a convolutional layer, a fully connected layer, and an output layer.

The hyperparameters are set as follows:

in_features=1: Number of input feature channels, default is 1.

num_classes=10: Number of classification categories, default is 10.

dim=256: Output dimension of the fully connected layer, default is 256.

Specifically, the model's structure is as follows:

Convolutional Layer (self.conv1): Input features undergo a convolution operation using a 5x5 convolution kernel with 32 output channels. No padding is applied (padding=0), and the stride is set to 1 (stride=1). The convolution involves a bias term (bias=True). The resulting feature maps are then processed through a ReLU activation function and a 2x2 max-pooling operation (nn.MaxPool2d(kernel_size=(2, 2))). Fully Connected Layer (self.fc1): The output feature maps from the convolutional layer are flattened into a one-dimensional vector and then passed through a fully connected layer. The input dimension is 32x12x12 (resulting from the convolution and pooling operations), and the output dimension is set to 'dim'.Output Layer (self.fc2): Connected to another fully connected layer is the output from the fully connected layer, where the input dimension is 'dim' and the output dimension is 'num_classes' (number of classification categories).

The accuracy of the original dual model is 0.658, with a runtime of 228.16 seconds. After our optimization, the accuracy of the dual model has improved to 0.669, and the runtime is reduced to 227.53 seconds

Experimental drawing:

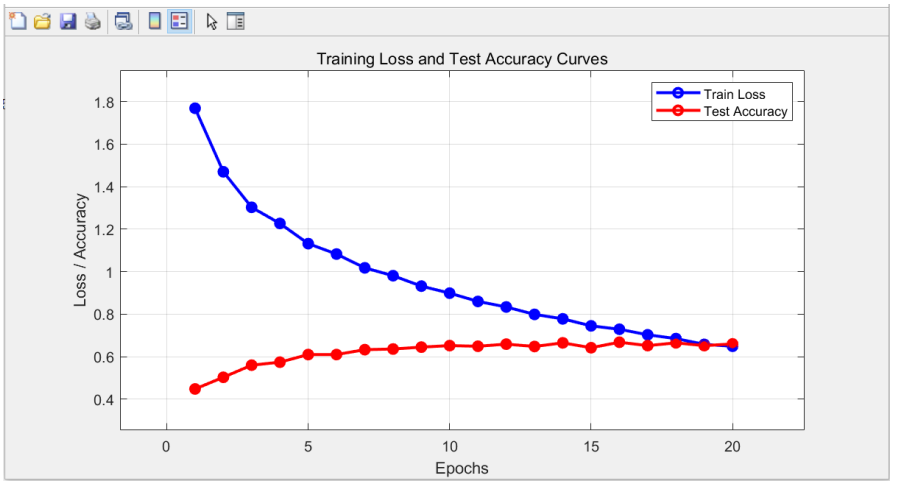

Figure 2. Accuracy and loss curve experimental diagram (Photo/Picture credit :Original).

Throughout this experiment, we made progress in the following areas:

1. Enhanced Stability and Accuracy: We improved model stability and accuracy by adjusting batch sizes and learning rates.

2. Increased Security: We discussed encryption techniques to enhance the data security of the MOON algorithm.

3. Reduced Runtime and Enhanced Efficiency: We researched simplifying the model architecture to decrease runtime and improve efficiency.

DOI: 10.54254/2755-2721/39/20230587

5. Conclusion and Future Work

The MOON algorithm represents a highly significant innovation in the field of federated learning. Through our enhancements and optimizations, the MOON algorithm has achieved improvements in both accuracy and operational speed within the context of contrastive learning frameworks. This advancement has introduced new perspectives and possibilities to the domain of federated learning, with profound impacts in two critical aspects.

Firstly, the integration of the MOON algorithm into the contrastive learning paradigm enables models to better capture the diversity and similarities among data samples. The enhanced accuracy of the MOON algorithm not only improves predictive performance but also contributes to optimizing data utilization, enhancing generalization capabilities, and reducing the risk of privacy leakage. These achievements aptly showcase the immense potential of contrastive learning within the realm of federated learning, offering an innovative avenue for elevating model performance and privacy protection to new heights.

Secondly, by optimizing algorithms and utilizing hardware acceleration, the operational speed of the MOON algorithm has been boosted. This holds great significance for the field of federated learning, expediting model training and deployment processes, enhancing the efficiency of federated learning, and thereby expanding its application scope across diverse domains. Such acceleration attracts greater participation and promotes technological development and dissemination.

Furthermore, the MOON algorithm offers substantial room for further improvements and research directions. For instance, exploring more efficient algorithmic strategies and refining network architectures to increase operational speed could be pursued. Moreover, coupling MOON-based federated learning with techniques such as incremental learning and online learning could achieve real-time and adaptive model updates. In the context of heterogeneous data and multimodal learning, the MOON algorithm may contribute to optimizing feature extraction and fusion techniques. Looking ahead, the MOON algorithm could also extend its reach into broader fields like natural language processing and the Internet of Things, providing robust tools for tackling complex problems.

In summary, the heightened accuracy and operational speed of the MOON algorithm have made pioneering contributions to the field of federated learning. Its value in enhancing model performance, driving research progress, expediting training processes, and other aspects cannot be underestimated. The prospects for future enhancements and applications of the MOON algorithm are promising, underscoring its role as a driving force and a forward-looking direction in the practical implementation and research advancement of federated learning.

Authors Contribution

All the authors contributed equally and there names were listed in alphabetical order.

References

[1] M. Chiang and T. Zhang.Fog and IoT: An overview of research opportunities[J].IEEE Internet of Things Journal,2016,3(6):854-864

[2] Chen Qingyu, Ji Fanfan, Yuan Xiaotong.Contrastive Learning with Dual-Layer Optimization based on Pseudo-Siamese Networks[J].Pattern Recognition and Artificial

[3] Chen J, Zhang R, Guo J, et al. FedMatch: federated learning over heterogeneous question answering data[C]//Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2021: 181-190.

[4] Li Weijia.Research on Personalized Federated Learning Algorithm Based on Contrastive Learning[D].Xidian University,2022.DOI:10.27389/d.cnki.gxadu.2022.000667.

[5] Li Q, Wen Z, Wu Z, et al. A survey on federated learning systems: Vision, hype, and reality for data privacy and protection[J]. IEEE Transactions on Knowledge and Data Engineering, 2021.

[6] Li Q, He B, Song D. Model-contrastive federated learning[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 10713-10722.

DOI: 10.54254/2755-2721/39/20230587

[7] Kaissis G A, Makowski M R, Rückert D, et al. Secure, privacy-preserving, and federated machine learning in medical imaging[J]. Nature Machine Intelligence, 2020, 2(6): 305-311.
[8] Chen J, Zhang R, Guo J, et al. FedMatch: federated learning over heterogeneous question answering data[C]//Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2021: 181-190.
[9] H Brendan McMahan, Eider Moore, Daniel Ramage, et al. Communication-efficient learning of deep networks from decentralized data. In AISTATS, 2017.
[10] T. G. Dietterich. Ensemble methods in machine learning. In Multiple classifier systems, pages 1–15. Springer, 2000.
[11] Hinton, Geoffrey, Vinyals, Oriol, Dean, Jeff.Distilling the Knowledge in a Neural Network.Oriol Vinyals [view email][v1] Mon, 9 Mar 2015 15:44:49 UTC