Understanding the Main Challenges of Federated Learning

KRISTI GJERKO 301101
MILAD BEIGI HARCHEGANI 289813
CHRISTIAN DAL POZZOLO 303406

Abstract

Federated Learning (FL) is a novel machine learning approach introduced by Google in 2016, designed to enable the exploitation of sensitive and privacy-protected data. FL is based on a client-server architecture where multiple devices collaboratively train a shared model while keeping their individual data on device. In this project we become familiar with the standard federated scenario, analyze the gradient inversion attack, and replicate some well-known experiments.

1. Introduction

Nowadays, a lot of data cannot be exploited by traditional Deep Learning (DL) methods due to its sensitive and privacy-protected nature such as the data collected by the cameras or GPS sensors in our mobile phones, or produced by IoT devices.

Federated Learning (FL) is a machine learning scenario aiming to use privacy-protected data without violating the regulations in force. The goal is to learn a global model in privacy-constrained scenarios leveraging a client-server architecture. The central model is kept on the server-side and, unlike standard DL settings, has no direct access to the data, which is stored on multiple edge devices, i.e. the clients. Thanks to a paradigm based on the exchange of the model parameters between clients and server through multiple rounds, the global model is able to extract knowledge from data without breaking the users' privacy.

The focus of this project is to understand the main challenges of the FL setting: i) statistical heterogeneity, i.e. the non-i.i.d. distribution of the clients' data which leads to degraded performances and unstable learning; ii) systems heterogeneity, i.e. the presence of devices having different computational capabilities (e.g. smartphones VS servers); iii) privacy concerns, deriving from the possibility of obtaining information on clients' data from the updated model exchanged on the network by using the gradient inversion attack.

2. Methods

2.1. Centralized training

The performance of any model trained in a federated setting is upper bounded by the results obtained in the centralized setting. Therefore, as a first step, we define the upper bound by training a ResNet model in a centralized way.

ResNet (Residual Network) is a deep learning neural network architecture that was introduced in 2015. It was developed to address the vanishing gradients problem in very deep neural networks by utilizing residual connections that skip one or more layers. This allows the network to learn and preserve information in deeper layers, which results in improved accuracy and convergence.

More specifically we use a ResNet20 architecture which typically follows this pattern:

1. Initial convolutional layer: This layer performs spatial feature extraction from the input image and reduces the size of the feature map.

2. Stack of residual blocks: A stack of several residual blocks are stacked one after the other to form the core of the ResNet20 architecture. Each block contains two or three convolutional layers with batch or group normalization and ReLU activation functions.

3. Global average pooling layer: This layer aggregates the features from the entire feature map and reduces the size of the feature map to 1x1.

4. Final dense layer: This layer performs the final classification and outputs the predicted class probabilities for the input image.

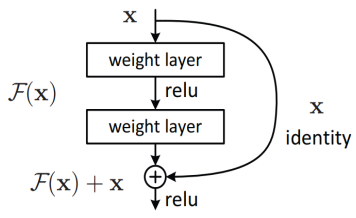

Figure 1. ResNet building block

2.2. Statistical heterogeneity

In federated learning, each participant has its own data, which may have different distributions and characteristics compared to the data of other participants. This statistical heterogeneity can arise due to various factors, such as differences in the data collection methods, differences in the populations being sampled, and differences in the distribution of features or labels. This can have a significant impact on the performance of a machine learning model trained on the federated data, as the model may be more adapted to some participants' data and less adapted to others.

Moreover, this heterogeneity can also introduce fairness and privacy concerns. For example, if one participant's data is heavily biased towards a particular group, the model trained on the federated data may produce biased results for that group.

To observe the impact of statistical heterogeneity on performance, we insert our ResNet20 architecture in the federated setting of FedSAM and perform experiments using different data distributions.

2.3. System heterogeneity

In federated learning, system heterogeneity refers to the differences in computational capabilities and resources among the participating devices. This heterogeneity can pose challenges in training a machine learning model that can be efficiently executed on all participating devices and deliver accurate results.

For example, some devices may have limited memory or processing power, while others may have high-end GPUs. This can make it difficult to find a common set of model parameters that can be used by all devices, as some may not be able to execute the model due to computational limitations. Additionally, the communication bandwidth between devices can also vary, making it challenging to exchange model updates in a timely and efficient manner.

We simulate a heterogeneous system where the central model in the server is the ResNet20 and the local models on the clients are CNNs with varying number of layers and filters. In this case, training a global model that is a simple average of the local models may not be effective, as it may not capture the best features of the different models. Transfer learning and model distillation can be utilized to overcome this.

2.4. Gradient inversion attack

Federated learning has the potential to provide privacy benefits over traditional centralized learning methods, however, it is not inherently private and still poses privacy risks. This decentralized approach can help protect the privacy of the data, as the raw data never leaves the participants' devices and is never combined in a single location. Despite these benefits, the model updates sent from each participant to the server may still contain information about the data used to train the model, and an adversary who can observe the updates may be able to infer information about the data.

We attempt this attack on our ResNet20 model and demonstrate that it is possible to reconstruct the original images.

2.5. SCAFFOLD

Stochastic Controlled Averaging is a method for federated learning that aims to improve the robustness and stability of the federated learning process. In traditional federated learning, each participant trains a model on its own data locally and sends the model updates to a central server, which aggregates the updates to obtain a global model. This process can be sensitive to the quality of the data used by each participant and the stability of the updates.

SCAFFOLD addresses these issues by introducing random sampling and weighting of the model updates from each participant. In SCAFFOLD, the central server randomly selects a subset of the participants to contribute to the aggregation of the model updates. The updates from each participant are then weighted based on the quality of their data and the stability of their updates. This random sampling and weighting helps to reduce the impact of noisy updates and ensures that the aggregated model is robust and stable.

3. Results

3.1. Centralized training

The ResNet20 architecture with batch normalization has achieved a 63.28% accuracy on the CIFAR100 test dataset, whereas its variant using group normalization has achieved 60.16% accuracy. Although the results are similar, batch normalization outperforms group normalization in this centralized scenario.

Group normalization is a normalization technique that splits the channel dimension of the activations into groups and normalizes each group separately. This allows for normalization to be performed locally on each participant's data, without the need for aggregation across samples. This makes group normalization well-suited for use in federated learning scenarios, where each participant's data distribution may be different and the batch size may be small.

In contrast, batch normalization requires a large batch size to estimate the mean and variance statistics accurately, and may result in sub-optimal performance in federated learning scenarios where the batch size is small.

3.2. Statistical heterogeneity

In FL, when the data distribution is heterogeneous across participating devices, it can lead to issues that can degrade the performance of the trained global model.

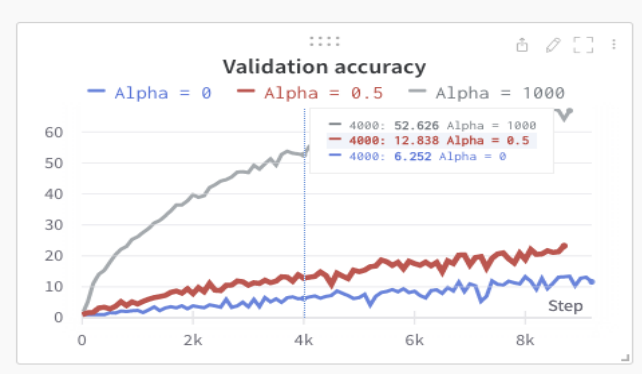

Figure 2. Validation accuracy

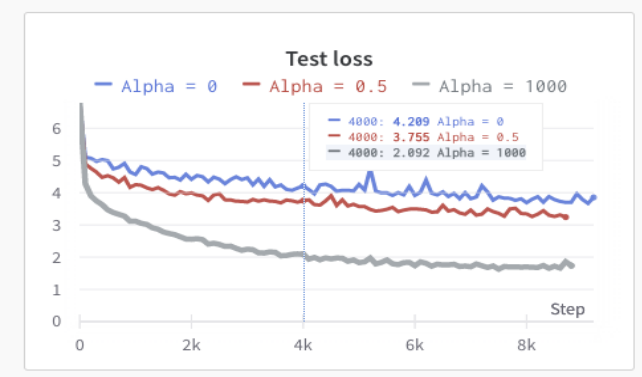

Figure 3. Test loss

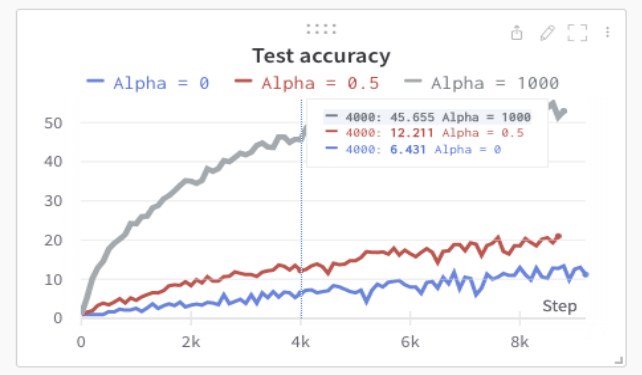

Figure 4. Test accuracy

When the data distribution is non-identically and independently distributed (non-i.i.d.) across devices, it can result in a biased global model that does not accurately capture the underlying patterns in the data. This can lead to overfitting or underfitting on certain devices, which can affect the overall performance of the model and generalization on unseen data.

Heterogeneous data distribution can make it more difficult for the global model to converge, as it may require more rounds of communication between devices to find a common set of model parameters that can be used by all devices. This can lead to longer training times and increase the computational requirements of the model.

3.3. System heterogeneity

A pre-trained global model is used as a starting point and fine-tuned on the local models of participating devices. This approach leverages the knowledge gained from the pre-trained model and adapts it to the local models, ensuring that the global model is able to capture the best features of all local models.

The local models are used to teach the global model by distilling their knowledge into a common set of model parameters. This approach involves training the global model to predict the outputs of the local models, ensuring that the global model is able to capture the best features of all local models.

The experiment was performed on the CIFAR10 dataset which was shuffled and equally distributed among the clients. The accuracy of the participating models on the test set improves gradually but its final performance is lower than that of the same models if the datasets were declassified and made available to every client.

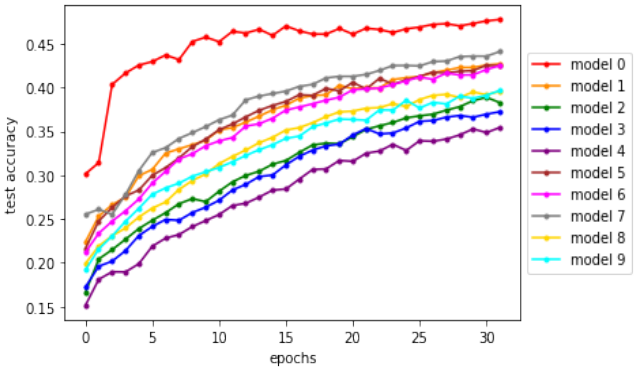

Figure 5. ResNet building block

3.4. SCAFFOLD

SCAFFOLD has been shown to improve the performance of federated learning in comparison to traditional federated learning methods, especially when there is significant variability in the quality of the data and updates from the participants.

In our experiment, SCAFFOLD achieved 11% accuracy after 1000 rounds with an alpha of 0.5, demonstrating an improvement over FedAvg which only reached 4% accuracy under the same conditions.

4. Discussion and conclusion

Federated Learning (FL) was born to address the privacy concerns of training machine learning models on sensitive, decentralized data.

We studied the standard federated framework which consists of a central server and multiple clients, where each client holds its own local data and the central server coordinates the training of a global model using this decentralized data. The experiments on FedSAM show that the performance of our ResNet model in a federated setting is upperbound by its performance in a centralized setting.

Unbalanced and heterogeneous data distribution across clients can also impact performance and result in training issues such as skewed models and reduced accuracy. To address the problem of statistical heterogeneity, solutions include weighting the loss function according to the data distribution, using transfer learning to adapt the global model to the local distributions, and re-sampling or augmenting the data.

We then saw through FedMD that systems heterogeneity can affect FL training. The devices participating in FL have different capacities and consequently different models. This heterogeneity should be taken into account in order to be able to extract information from the local models and translate it into value for the global model.

Even though FL was created with privacy in mind there still exist attacks that can compromise it. We attempted the gradient inversion attack on our ResNet20 model by trying to capture model updates and go back to the training data it came from and were able to reconstruct pretty clear images.

In conclusion, we can say that Federated Learning is an approach with a lot of potential when its main issues are addressed correctly. Some real-world applications of FL include personalized mobile services such as keyboard prediction and mobile health or secure machine learning on sensitive data such as credit card transactions and healthcare records.

References

[1] Google AI Blog, Federated Learning: Collaborative Machine Learning without Centralized Training Data.

[2] McMahan, Brendan et al. Communication-Efficient Learning of Deep Networks from Decentralized Data.

[3] Reddi, Sashank, et al. Adaptive federated optimization.

[4] Li, Tian, et al. Federated Learning: Challenges, Methods, and Future Directions.

[5] Kairouz, Peter, et al. Advances and Open Problems in Federated Learning.

[6] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification.

[7] Hsu TM.H. et al. Federated Visual Classification with Real-World Data Distribution.

[8] Sai Praneeth Karimireddy, et al. Scaffold: Stochastic controlled averaging for federated learning.

[9] Acar, D. A. E., Zhao, Y., Navarro, R. M., Mattina, M., Whatmough, P. N., Saligrama, V. Federated learning based on dynamic regularization.

[10] Tian Li, et al. Federated optimization in heterogeneous networks. In I. Dhillon, D. Papailiopoulos, and V. Sze, editors, Proceedings of Machine Learning and Systems, volume 2.

[11] He, Chaoyang, Murali Annavaram, and Salman Avestimehr. Group knowledge transfer: Federated learning of large CNNs at the edge.

[12] Geiping, Jonas, et al. Inverting gradients-how easy is it to break privacy in federated learning?

[13] Huang, Yangsibo, et al. Evaluating gradient inversion attacks and defenses in federated learning.

[14] Wei, Kang, et al. Federated learning with differential privacy: Algorithms and performance analysis.

[15] He, Kaiming, et al. Deep residual learning for image recognition.

[16] Hsieh, Kevin, et al. The non-iid data quagmire of decentralized machine learning. International Conference on Machine Learning.

[17] Wu, Yuxin, and Kaiming He. Group normalization.

[18] Luo, Mi, et al. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data.

[19] Park, Jung Wuk, et al. Sageflow: Robust Federated Learning against Both Stragglers and Adversaries.

[20] Li, Xingyu, et al. Stragglers are not disaster: A hybrid federated learning algorithm with delayed gradients.

[21] Zhang, Hongyi, et al. mixup: Beyond empirical risk minimization.

[22] Huang, Yangsibo, et al. Instahide: Instance-hiding schemes for private distributed learning.

[23] He, Kaiming, et al. Deep residual learning for image recognition.

[24] Li, Daliang, and Junpu Wang. FedMD: Heterogenous federated learning via model distillation.