Ensemble distillation in federated learning

Oscar Eriksson

February 3, 2022

1 Background

The increased computing capabilities developed during the last decade have made it possible to deploy advanced machine learning models for different tasks, such as image classification and natural language processing. These models need a large amount of data, which might need to be collected at different locations and devices. However, with increased privacy concerns of user data and regulations such as GDPR, moving the data to a central server might not be desirable or even allowed. Federated learning (FL) is a technique proposed to overcome this privacy challenge in machine learning, which leverages computational resources and data at the local devices in a more secure way. This is done by only sharing information from a locally trained model, and not the data itself.

Among the open challenges in FL [1], one crucial step is to aggregate local models into a global model. The standard algorithm for this is FedAvg [2], which average over the local model weights into a global model. It has been shown that FedAvg converge to the ideal model[1] when sampled client data is assumed iid over the clients [2] [3], but the iid assumption does not always hold in practice. In the non-iid case, the performance of FedAvg degrades drastically [4]. This is because when sampling local data from a distribution that is not the same as the joint distribution of all clients data, the local objectives to minimize will be different for each client. Client models will therefore drift towards the minima of their own local objectives, and aggregating these models might not yield the optimum of the global objective.

FedAvg and recently modified versions ([5] [6] [7] [8]) all builds a global model based on the average of the client's local model parameters. This kind of algorithm can therefore only be applied when clients uses the same model size and architecture. However, combining multiple heterogeneous classifiers could be desirable when local devices have varying computational resources or amounts of data. This is possible with *ensemble learning methods*. These methods instead average over the output from an ensemble of models to make inference on new data. One problem with directly applying ensemble methods in FL is that the ensemble might become huge and infeasible to store at the central server. Several works ([9], [10], [11]) have approached this problem by using *knowledge distillation*, which is a way to transfer the combined knowledge of the ensemble into one single model based on predictions from the ensemble. In addition to being model agnostic, using *ensemble distillation* also has the potential of reducing communication cost and increasing privacy, since less information is sent from the clients to the central server.

With the mentioned benefits of ensemble methods and knowledge distillation in FL, the question arises whether ensemble distillation is also a good choice to address the challenge of having data non-iid over the clients. This is the subject of the project that will be outlined in this report. This planning report firstly covers the latest research in FL for non-iid scenarios and currently published ensemble distillation algorithms. Based upon this research, planned experiments are then presented to test ensemble distillation in non-iid scenarios, using existing algorithms with proposed modifications. Lastly, considerations about this project ethical and ecological aspects are discussed.

[1] A centrally trained model on all aggregated data.

1.1 Current research

This section presents some of the latest research in FL algorithms related to this project. All algorithms covered are summarized in Table 1.

Table 1: Differences between considered FL algorithms based on parameter sharing and ensemble distillation. The algorithms are compared according to which information is shared by client, if a public dataset is needed and if the approach is model agnostic.

Algorithm	Shared Information	Public Dataset	Model Agnostic
FedAvg [2]	Weights	No	No
FedProx [7]	Weights	No	No
PATE [12]	Predictions	Unlabeled	Yes
FedAD [11]	Predictions, Attention maps	Unlabeled	Yes
FedDF [9]	Weights	Unlabeled	No
FedMD [10]	Predictions	Labeled	Yes

Federated Learning for non-iid data. The standard FedAvg [2] algorithm performs a weighted average over the client models with weights proportional to local data size for each client. To address data heterogeneity, one line of work investigates more efficient weighting schemes based on the local losses [5][6]. Another approach is to introduce regularization at client-side to improve local training [7] [8]. In FedProx [7], a proximal term is added to the local loss function to counteract client drift.

Ensemble Distillation. Knowledge distillation is based on training a student model to approximate the output logits of a teacher model, and is first introduced for neural networks in [13] [14]. This can be extended to ensemble distillation, where the aim is to distill knowledge from an ensemble of teacher models to a student model. Papernot et al. [12] study the use of ensemble distillation in deep learning and particularly its privacy benefits. With their proposed algorithm PATE, they train a student model to mimic the behaviour of teacher models by training the student model on a public dataset, with the collective predictions from the teacher models on this dataset as targets. For a classification problem with classes $c \in C$ and clients $k \in \mathcal{K}$, the predictions used can for example be the soft labels $\mathbf{z}^k$ or hard labels $\mathbf{y}_k = \arg\max_c \mathbf{z}_c^k$. In FL, this public dataset needs to be non sensitive and allowed to be shared among the clients. However, with the procedure used in PATE, this dataset can be unlabeled. The algorithmic approach is explained in more detail in Algorithm 1.

Ensemble Distillation in FL. One idea to extend PATE is to consider if some additional output from the ensemble can be beneficial for training the student model. This has been investigated in particular for FL by Gong et al. [11], which presents the algorithm FedAD where attention maps[2] on the public dataset are sent to the central server for training the student model, in addition to the soft labels. The authors also address the problem of having non-iid data by performing a weighted average when combining the soft labels, see equation (1).

[2] Heatmap over input image measuring which pixels have high effect on the classification.

Algorithm 1: The general ensemble distillation procedure, similar to PATE [12] and FedAD [11].

```
Input: Labeled private data $\{\mathcal{D}_k\}$, unlabeled public data $\mathcal{D}_0$, central model $\theta_\mathrm{s}$,
       local models $\{\theta_k\}$, $T$ distillation steps, batch size $S$, sample fraction $\gamma$, loss
       function $\mathcal{L}$.
Output: Trained central model $\theta_\mathrm{s}$.
Local Training: Train each local model $\theta_k$ with $\mathcal{D}_k$.
for each distillation step $t = 1, \dots, T$ do
    $\mathcal{K}_t \leftarrow$ random subset ($\gamma$ fraction) from $K$ local models
    $\mathbf{x}_0 \leftarrow$ a batch of public data from $\mathcal{D}_0$ with size $S$
    for each local $k \in \mathcal{K}_t$ do
        $\mathbf{z}^k \leftarrow f(\mathbf{x}_0; \theta_k)$
    end
    $\hat{\mathbf{z}} \leftarrow$ ensemble $\{\mathbf{z}^k | k \in \mathcal{K}_t\}$
    $\tilde{\mathbf{z}} \leftarrow f(\mathbf{x}_0; \theta_\mathrm{s})$
    Update: $\theta_\mathrm{s} \leftarrow \theta_\mathrm{s} - \frac{1}{S} \nabla_{\theta_\mathrm{s}} \mathcal{L}(\tilde{\mathbf{z}}, \hat{\mathbf{z}})$
end
```

$$\hat{\mathbf{z}}_c = \sum_{k \in \mathcal{K}_t} \omega_k^c \mathbf{z}_c^k, \quad \omega_c^k = \frac{N_c^k}{\sum_{k \in \mathcal{K}_t} N_c^k} \tag{1}$$

This is used in the ensemble step in Algorithm 1, where N_c^k is the number of samples of class c at client k. Equation (1) directly captures the distribution of the local data, which might cause privacy concerns. The authors presents results for non-iid scenarios, using the datasets CIFAR-10/100 [15] and CXR14 [16], where FedAD is shown to outperform other algorithms based on parameter sharing, but also other distillation based algorithms, such as FedDF [9] and FedMD [10].

PATE and FedAD are both aimed at distilling converged local models, i.e. there is only one round of local training at client. On the contrary, [9] [10] uses iterative updates of the student and teacher models. FedDF [9] first combines the local models with FedAvg and then trains a student model on a public dataset to approach the combined global model output, using Kullback-Leibler divergence as loss function. Since this approach is based on FedAvg, the potential benefits of better privacy, less communication and model agnosticism are lost. FedMD [10] trains the local models on both private and public data, see Algorithm 2. This approach needs labeled public data and it is model agnostic, but implemented to provide personalized models. This means that each client will have its own personal model in the end, which is more adapted to its private data.

2 Project plan

The basic procedure for ensemble distillation (Algorithm 1) has the benefit of being model agnostic and could, combined with weighting client contributions, improve performance with non-iid data. Handling the non-iid problem can also be approached by regulating

Algorithm 2: The FEDMD [10] algorithm.

```
Input: Labeled private data $\{\mathcal{D}_k\}$, labeled public data $\mathcal{D}_0$, local models $\{\theta_k\}$ with
       $k = 1, \ldots, m$, $T$ communication rounds, batch size $S$.
Output: Trained local models $\{\theta_k\}$.
Local Training: Train each local model $\theta_k$ to convergence on $\mathcal{D}_0$ and then on its
                private data $\mathcal{D}_k$.
for each round $t = 1, \ldots, T$ do
    $\mathbf{x}_0 \leftarrow$ a batch of public data from $\mathcal{D}_0$ with size $S$.
    Communicate: Each client computes soft labels $\mathbf{z}^k \leftarrow f(\mathbf{x}_0; \theta_k)$ and transmits
                 result to central server.
    Aggregate: Central server computes updated consensus by averaging:
               $\hat{\mathbf{z}} \leftarrow \frac{1}{m} \sum_k \mathbf{z}^k$.
    Distribute: Each client downloads the updated consensus $\hat{\mathbf{z}}$.
    Digest: Each client trains its model $\theta_k$ to approach $\hat{\mathbf{z}}$ on the public dataset $\mathcal{D}_0$.
    Revisit: Each client trains its model $\theta_k$ on its own private data for a few
             epochs.
end
```

local training, such as including training on the public dataset (Algorithm 2). Both of these algorithms are considered to be interesting candidates for this project, and are intended to be implemented as is for testing. To improve the performance in non-iid scenarios, the plan is also to extend or modify these algorithms with ideas that will now be discussed.

From the research presented in previous section, there are two ways of addressing data heterogeneity with ensemble learning: 1) weighting the predictions from teacher models or 2) regulate the local training. The idea with this project is to test ensemble distillation algorithms in both of these aspects, using only model agnostic approaches. Algorithmic ideas for this project based on these two approaches are presented below.

1) Weighting the soft labels as in equation (1) can cause privacy concerns since this approach completely reveals the local data distributions. Therefore, other weighting schemes where ensemble predictions can be weighted directly at client based on local data information will be considered. The general idea is that a client should not contribute as much with predictions in a class where it has few data samples or shows bad performance. Based on this, preliminary weighting schemes to test in this project are presented in Table 2.

	Scheme 1	Scheme 2
ω_k^c	N_k^c	$\sum_{\mathbf{x} \in \mathbf{x}_0} I(f(\mathbf{x}; \theta_k) = y_0^c(\mathbf{x}))/(N_c^k + \epsilon)$

Table 2: Proposed weighting schemes for averaging ensembles soft labels on public data $\mathcal{D}_0$. $y_0^c(\mathbf{x})$ are the true labels of the public data in class c and ϵ is a small number to avoid division by zero. Scheme 2 can be seen as a class specific accuracy and needs the public data to be labeled.

Since soft labels should be in the range $[0, 1]$, the combined soft labels should also be normalized by $\bar{\mathbf{z}}^c = \hat{\mathbf{z}}^c / \sum_c \hat{\mathbf{z}}^c$ after having calculated the weighted average $\hat{\mathbf{z}}^c$ using any of the schemes in Table 2.

2) Regulating local training can be done by including training on the public dataset, as done in FedMD (Algorithm 2) by using a labeled public dataset. This algorithm will be included for testing in this project. Having a labeled public dataset yields further training setups that need to be tested. Can a better global model be achieved by solely training on public data? How does a distilled model perform when simply including public data in local training? There will clearly be a trade off between the size of the public dataset and the performance of the algorithms. Different sizes of the public data will therefore also be tested in this project. However, since labeled public data might not be available in real scenarios, modifications to make FedMD only depend on unlabeled public data are also aimed to be investigated during the project.

3 Aim

This project aims to investigate the use of ensemble distillation algorithms in FL for the purpose of improving performance in non-iid scenarios. New ideas to extend existing algorithms will also be investigated, as described in the previous section.

4 Objective

The objective is to implement a selected number of FL algorithms, based on parameter sharing and ensemble distillation, to compare their performance when data is non-iid over the clients. Furthermore, the objective is to introduce novel ideas to improve current ensemble distillation techniques. A secondary objective is also to implement a selected ensemble distillation algorithm in FEDn [17], if compatible with the current FEDn framework. The research questions for this project are the following:

- What is the current research on using ensemble distillation in FL?
- How can existing ensemble distillation algorithms be improved for non-iid scenarios?

5 Limitations

For the purpose of improving existing algorithms mentioned in section 1.1, this project will be limited to only consider model agnostic algorithms. The problem task considered will be image classification and experiments will be limited to a few selected dataset commonly used for testing machine learning algorithms. The FL settings will be limited to a fixed number of clients and a few different non-iid scenarios will be used for testing, corresponding to different strengths of class imbalances over the clients private data. This is further explained in the next section.

6 Method

A test framework will be implemented in Python for pseudo-distributed testing of selected algorithms. Machine learning models will be implemented with Pytorch [18], alternatively Tensorflow [19]. In addition to the ensemble distillation algorithms that will be implemented, algorithms based on parameter sharing such as FedAvg and FedProx will also be implemented as a baseline for comparison. The datasets and the non-iid distributions of data will be inspired by [11]. Hence, the datasets intended for testing are CIFAR-10/100 [15] and CXR8 [16], and non-iid data will be generated by Dirichlet sampling with a few different values for the concentration parameter α. Evaluation will be done on a balanced subset of available data, using accuracy as metric. The computationally intensive experiments will by carried out using cloud computing provided by AI Sweden.

7 Time plan

Weeks	Activity
1-4	Literature study: research of current FL algorithms and ensemble distillation.
1-3	Write planning report
3-6	Develop test framework. Implement FedAvg and FedProx.
6-9	Implement selected ensemble distillation algorithms in the framework and perform tests with non-iid data.
9-12	Try to improve performance with new ideas.
12-14	Final testing and generating of results.
13-20	Write final report

8 Ethical and ecological considerations

This project will consist of implementation and analysis of different approaches to combine multiple machine learning models trained on private data. The physical actions performed during the project will therefore only involve the project executioner, and hence the project have no direct impact on other individuals. Ethical concerns are therefore not important in this manner.

Advancements in FL in general leads to more powerful machine learning models and increased privacy of sensitive data. While it can be argued that more automation with machine learning models could decrease job opportunities, studies in UK show that automation with AI will yield a positive net employment [20]. Increased privacy of user data would yield better control of one's digital footprint, which is positive in the ethical perspective.

The energy usage during this project is not of major concern, since the work will mainly be performed using a laptop computer. Computationally intensive tasks will be carried out using computers at AI Sweden EdgeLab, which will be a more energy consuming phase in project. However, the ecological footprint of using cloud computing is low since these computers are shared by many users.

References

[1] P. Kairouz, H. B. McMahan, B. Avent, *et al.*, *Advances and open problems in federated learning*, 2021. arXiv: `1912.04977 [cs.LG]`.

[2] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, *Communication-efficient learning of deep networks from decentralized data*, 2017. arXiv: `1602.05629 [cs.LG]`.

[3] F. Zhou and G. Cong, "On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization," *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, Jul. 2018. DOI: `10.24963/ijcai.2018/447`. [Online]. Available: `http://dx.doi.org/10.24963/ijcai.2018/447`.

[4] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, *Federated learning with non-iid data*, 2018. arXiv: `1806.00582 [cs.LG]`.

[5] M. Mohri, G. Sivek, and A. T. Suresh, *Agnostic federated learning*, 2019. arXiv: `1902.00146 [cs.LG]`.

[6] T. Li, M. Sanjabi, A. Beirami, and V. Smith, *Fair resource allocation in federated learning*, 2020. arXiv: `1905.10497 [cs.LG]`.

[7] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, *Federated optimization in heterogeneous networks*, 2020. arXiv: `1812.06127 [cs.LG]`.

[8] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, *Scaffold: Stochastic controlled averaging for federated learning*, 2021. arXiv: `1910.06378 [cs.LG]`.

[9] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, *Ensemble distillation for robust model fusion in federated learning*, 2021. arXiv: `2006.07242 [cs.LG]`.

[10] D. Li and J. Wang, *Fedmd: Heterogenous federated learning via model distillation*, 2019. arXiv: `1910.03581 [cs.LG]`.

[11] X. Gong, A. Sharma, S. Karanam, *et al.*, "Ensemble attention distillation for privacy-preserving federated learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 15 076–15 086.

[12] N. Papernot, M. Abadi, Ú. Erlingsson, I. Goodfellow, and K. Talwar, *Semi-supervised knowledge transfer for deep learning from private training data*, 2017. arXiv: `1610.05755 [stat.ML]`.

[13] C. Buciluundefined, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '06, Philadelphia, PA, USA: Association for Computing Machinery, 2006, pp. 535–541, ISBN: 1595933395. DOI: `10.1145/1150402.1150464`. [Online]. Available: `https://doi.org/10.1145/1150402.1150464`.

[14] G. Hinton, O. Vinyals, and J. Dean, *Distilling the knowledge in a neural network*, 2015. arXiv: `1503.02531 [stat.ML]`.

[15] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[16] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," *CoRR*, vol. abs/1705.02315,

2017. arXiv: `1705.02315`. [Online]. Available: `http://arxiv.org/abs/1705.02315`.

[17] M. Ekmefjord, A. Ait-Mlouk, S. Alawadi, *et al.*, *Scalable federated machine learning with fedn*, 2021. arXiv: `2103.00148 [cs.LG]`.

[18] A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[19] Martín Abadi, Ashish Agarwal, Paul Barham, *et al.*, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015. [Online]. Available: `https://www.tensorflow.org/`.

[20] "The potential impact of artificial intelligence on uk employment and the demand for skills," H. D. III and A. Singh, Eds., ser. BEIS Research Report, vol. 2021/042, PricewaterhouseCoopers LLP, Aug. 2021.