Adaptive Expert Models for Federated Learning

Martin Isaksson[1,3](✉), Edvin Listo Zec[3,5], Rickard Cöster[2], Daniel Gillblad[4,7], and Sarunas Girdzijauskas[3,6]

[1] Ericsson Research, Stockholm, Sweden
martin.isaksson@ericsson.com
[2] Ericsson Global AI Accelerator, Stockholm, Sweden
[3] KTH Royal Institute of Technology, Stockholm, Sweden
[4] Chalmers AI Research Center, Chalmers University of Technology, Göteborg, Sweden
[5] RISE Research Institutes of Sweden, Göteborg, Sweden
[6] RISE Research Institutes of Sweden, Stockholm, Sweden
[7] AI Sweden, Stockholm, Sweden

Abstract. Federated Learning (FL) is a promising framework for distributed learning when data is private and sensitive. However, the state-of-the-art solutions in this framework are not optimal when data is heterogeneous and non-IID. We propose a practical and robust approach to personalization in FL that adjusts to heterogeneous and non-IID data by balancing exploration and exploitation of several global models. To achieve our aim of personalization, we use a Mixture of Experts (MoE) that learns to group clients that are similar to each other, while using the global models more efficiently. We show that our approach achieves an accuracy up to 29.78% better than the state-of-the-art and up to 4.38% better compared to a local model in a pathological non-IID setting, even though we tune our approach in the IID setting.

Keywords: Federated learning · Personalization · Privacy preserving

1 Introduction

In many real-world scenarios, data is distributed over organizations or devices and is difficult to centralize. Due to legal reasons, data might have to remain and be processed where it is generated, and in many cases may not be allowed to be transferred [10]. Furthermore, due to communication limitations it can be practically impossible to send data to a central point of processing. In many applications of Machine Learning (ML) these challenges are becoming increasingly important to address. For example, sensors, cars, radio base stations and mobile devices are capable of generating more relevant training data than can be practically communicated to the cloud [8] and datasets in healthcare and industry cannot legally be moved between hospitals or countries of origin.

R. Goebel et al. (Eds.): FL 2022, LNAI 13448, pp. 1–16, 2023.
https://doi.org/10.1007/978-3-031-28996-5_1

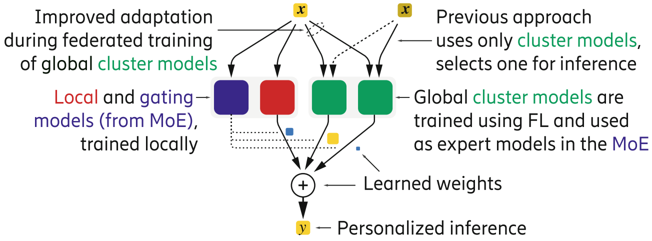

Improved adaptation during federated training of global cluster models
x
x
Previous approach uses only cluster models, selects one for inference
Local and gating models (from MoE), trained locally
Global cluster models are trained using FL and used as expert models in the MoE
Learned weights
+
y
Personalized inference

Fig. 1. Our approach adjusts to non-Independent and Identically Distributed (IID) data distributions by adaptively training a Mixture of Experts (MoE) for clients that share similar data distributions.

Federated Learning (FL) [1,27] shows promise to leveraging data that cannot easily be centralized. It has the potential to utilize compute and storage resources of clients to scale towards large, decentralized datasets while enhancing privacy. However, current approaches fall short when data is heterogeneous as well as non-Independent and Identically Distributed (non-IID), where stark differences between clients and groups of clients can be found. Therefore, personalization of collectively learned models will in practice often be critical to adapt to differences between regions, organizations and individuals to achieve the required performance [11,18]. This is the problem we address in this chapter.

Our approach adjusts to non-IID data distributions by adaptively training a Mixture of Experts (MoE) for clients that share similar data distributions.We explore a wide spectrum of data distribution settings: ranging from the same distribution for all clients, all the way to different distributions for each client. Our aim is an end-to-end framework that performs comparable or better than vanilla FL and is *robust* in all of these settings.

In order to achieve personalization, the authors of [11] introduce a method for training cluster models using FL. We show that their solution does not perform well in our settings, where only one or a few of the cluster models converge. To solve this, inspired by the Multi-Armed Bandit (MAB) field, we employ an efficient and effective way of balancing exploration and exploitation of these cluster models. As proposed by the authors of [24,29], we add a local model and use a MoE that learns to weigh, and make use of, all of the available models to produce a better personalized inference, see Fig. 1.

In summary, our main contributions are:

1. We devise an FL algorithm which improve upon [11] by balancing exploration and exploitation to produce better adapted cluster models, see Sect. 3.1;
2. We use said cluster models as expert models in an MoE to improve performance, described in Sect. 3.1;
3. An extensive analysis[1] of our approach with respect to different non-IID aspects that also considers the distribution of client performance, see Sect. 4.5.

[1] The source code for the experiments can be found at https://github.com/EricssonResearch/fl-moe.

2 Background

2.1 Problem Formulation

Consider a distributed and decentralized ML setting with clients $k \in \{1, 2, \ldots, K\}$. Each client k has access to a local data partition P^k that never leaves the client where $n_k = |P^k|$ is the number of local data samples. In this chapter we are considering a multi-class classification problem where we have $n = \sum_{k=1}^{K} n_k$ data samples $\boldsymbol{x}_i$, indexed $i \in \{1, 2, \ldots, n_k\}$, and output class label y_i is in a finite set. We further divide each client partition P^k into local training and test sets. We are interested in performance on the local test set in a non-IID setting, see Sect. 2.2.

2.2 Regimes of Non-IID Data

In any decentralized setting it is common to have non-IID data that can be of non-identical client distributions [14,18], and which can be characterized as:

- *Feature distribution skew* (covariate-shift). The feature distributions vary between clients. Marginal distributions $\mathcal{P}(\boldsymbol{x})$ varies, but $\mathcal{P}(y \mid \boldsymbol{x})$ is shared;
- *Label distribution skew* (prior probability shift, or class imbalance). The distribution of class labels are different between clients, so that $\mathcal{P}(y)$ varies but $\mathcal{P}(\boldsymbol{x} \mid y)$ is shared;
- *Same label, different features.* The conditional distributions $\mathcal{P}(\boldsymbol{x} \mid y)$ varies between clients but $\mathcal{P}(y)$ is shared;
- *Same features, different label* (concept shift). The conditional distribution $\mathcal{P}(y \mid \boldsymbol{x})$ varies between clients, but $\mathcal{P}(\boldsymbol{x})$ is shared;
- *Quantity skew* (unbalancedness). Clients have different amounts of data.

Furthermore, the data independence between clients and between data samples within a client can also be violated.

2.3 Federated Learning

In a centralized ML solution data that may be potentially privacy-sensitive is collected to a central location. One way of improving privacy is to use a collaborative ML algorithm such as Federated Averaging (FedAvg) [27]. In FedAvg training of a global model $f_g(\boldsymbol{x}, \boldsymbol{w}_g)$ is distributed, decentralized and synchronous. A parameter server coordinates training on many clients over several communication rounds until convergence.

In communication round t, the parameter server selects a fraction C out of K clients as the set S_t. Each selected client $k \in S_t$ will train locally on n_k data samples $(\boldsymbol{x}_i, y_i), i \in P^k$, for E epochs before an update is sent to the parameter server. The parameter server performs aggregation of all received updates and updates the global model parameters $\boldsymbol{w}_g$. Finally, the new global model parameters are distributed to all clients.

We can now define our objective as

$$\min_{\boldsymbol{w}_g\in\mathbb{R}^d} \mathcal{L}(\boldsymbol{w}_g) \triangleq \min_{\boldsymbol{w}_g\in\mathbb{R}^d} \underbrace{\sum_{k=1}^{K} \frac{n_k}{n} \overbrace{\frac{1}{n_k}\sum_{i\in P^k} \underbrace{l\left(\boldsymbol{x}_i, y_i, \boldsymbol{w}_g\right)}_{\text{sample } i \text{ loss}}}^{\text{client } k \text{ average loss}}}_{\text{population average loss}}, \tag{1}$$

where $l\left(\boldsymbol{x}_i, y_i, \boldsymbol{w}_g\right)$ is the loss for $y_i, \hat{y}_g = f_g\left(\boldsymbol{x}_i, \boldsymbol{w}_g\right)$. In other words, we aim to minimize the average loss of the global model over all clients in the population.

2.4 Iterative Federated Clustering

In many real distributed use-cases, data is naturally non-IID and clients form clusters of *similar* clients. A possible improvement over FedAvg is to introduce cluster models that map to these clusters, but the problem of identifying clients that belong to these clusters remains. We aim to find clusters, subsets of the population of clients, that benefit more from training together within the subset, as opposed to training with the entire population.

Using Iterative Federated Clustering Algorithm (IFCA) [11] we set the expected largest number of clusters to be J and initialize one cluster model with weights $\boldsymbol{w}_g^j$ per cluster $j \in \{1, 2, \dots, J\}$. At communication round t each selected client k performs a cluster identity estimation, where it selects the cluster model $\hat{j}^k$ that has the lowest estimated loss on the local training set. This is similar to [26].

The cluster model parameters $\boldsymbol{w}_g^j$ at time $t+1$ are then updated by using only updates from clients the jth selected cluster model, so that (using model averaging [11,27])

$$n_j \leftarrow \sum\nolimits_{k\in\left\{S_t \,\middle|\, \hat{j}^k = j\right\}} n_k, \tag{2}$$

$$\boldsymbol{w}_g^j(t+1) \leftarrow \sum\nolimits_{k\in\left\{S_t \,\middle|\, \hat{j}^k = j\right\}} \frac{n_k}{n_j} \boldsymbol{w}^k(t+1). \tag{3}$$

2.5 Federated Learning Using a Mixture of Experts

In order to construct a personalized model for each client, [24] first add a local expert model $f_l^k(\boldsymbol{x}, \boldsymbol{w}_l^k)$ that is trained only on local data. Recall the global model $f_g(\boldsymbol{x}, \boldsymbol{w}_g)$ from Sect. 2.3. The authors of [24] then *learn to weigh* the local expert model and the global model using a gating function from MoE [12, 15, 29]. The gating function takes the same input $\boldsymbol{x}$ and outputs a weight for each of the expert models. It uses a Softmax in the output layer so that these weights sum to 1. We define $f_h^k\left(\boldsymbol{x}, \boldsymbol{w}_h^k\right)$ as the gating function for client k. The same model architectures are used for all local models, so $f_h^k(\boldsymbol{x}, \boldsymbol{w}) = f_h^{k'}(\boldsymbol{x}, \boldsymbol{w})$ and $f_l^k(\boldsymbol{x}, \boldsymbol{w}) = f_l^{k'}(\boldsymbol{x}, \boldsymbol{w})$ for all pairs of clients k, k'. For simplicity, we write $f_l\left(\boldsymbol{x}\right) = f_l^k\left(\boldsymbol{x}, \boldsymbol{w}_l^k\right)$ and $f_h\left(\boldsymbol{x}\right) = f_h^k\left(\boldsymbol{x}, \boldsymbol{w}_h^k\right)$ for each client k. Parameters $\boldsymbol{w}_l^k$ and $\boldsymbol{w}_h^k$ are local to client k and not shared. Finally, the personalized inference is

$$\hat{y}_h = f_h\left(\boldsymbol{x}\right) f_l\left(\boldsymbol{x}\right) + \left[1 - f_h\left(\boldsymbol{x}\right)\right] f_g\left(\boldsymbol{x}\right). \tag{4}$$

3 Adaptive Expert Models for Personalization

3.1 Framework Overview and Motivation

In IFCA, after the training phase, the cluster model with the lowest loss on the validation set is used for all future inferences. All other cluster models are discarded in the clients. A drawback of IFCA is therefore that it does not use all the information available in the clients in form of unused cluster models. Each client has access to the full set of cluster models, and our hypothesis is that if a client can make use of *all* of these models we can increase performance.

It is sometimes advantageous to incorporate a local model, as in Sect. 2.5, especially when the local data distribution is very different from other clients. We therefore modify the MoE [24] to incorporate *all* the cluster models from IFCA [11] *and* the local model as expert models in the mixture, see Fig. 2. We revise (4) to

$$\hat{y}_h = g_l f_l^k(\boldsymbol{x}) + \sum_{j=0}^{J-1} g_j^k f_g^j(\boldsymbol{x}), \tag{5}$$

where g_l is the local model expert weight, and g_j^k is the cluster model expert weight for cluster j from $f_h^k(\boldsymbol{x})$, see Fig. 2.

However, importantly, we note that setting J in [11] to a large value produces few cluster models that actually converge, which lowers performance when used in a MoE. The authors of [34] note that this method is difficult to train in practice and that the performance is worse than FedAvg together with fine-tuning. We differ from [11] in the cluster estimation step in that we select the same number of clients $K_s = \lceil CK \rceil$ in every communication round, regardless of J. This spreads out more evenly over the global cluster models. Since cluster models are randomly initialized we can end up updating one cluster model more than the others by chance. In following communication rounds, a client is more likely to select this cluster model, purely because it has been updated more. This also has the effect that as J increases, the quality of the updates are reduced as they are averaged from a smaller set of clients. In turn, this means that we needed more iterations to converge. Therefore, we make use of the ε-greedy algorithm [31] in order to allow each client to prioritize gathering information (*exploration*) of the cluster models or use the estimated best cluster model (*exploitation*). In each iteration a client selects a random cluster model with probability ε and the currently best otherwise, see Algorithm 3.

By using the ε-greedy algorithm we make more expert models converge and avoid a mode collapse. We can then use the gating function f_h^k from MoE to adapt to the underlying data distributions and weigh the different expert models. We outline our setup in Fig. 1 and provide details in Fig. 2 and Algorithms 1 to 4.

When a cluster model has converged it is not cost-effective to transmit this cluster model to every client, so by using per-model early stopping we can reduce communication in both uplink and downlink. Specifically, before training we initialize $\mathcal{J} = \{1, 2, \ldots, J\}$. When early stopping is triggered for a cluster model we remove that cluster model from the set $\mathcal{J}$. The early-stopping algorithm is described in Algorithm 1.

Algorithm 1. Adaptive Expert Models for FL — server

```
procedure SERVER($C, K$)
    initialize $\mathcal{J} \leftarrow \{1, 2, \ldots, J\}$, $\left\{ \boldsymbol{w}_g^j(0) \;\middle|\; j \in \mathcal{J} \right\}$    ▷ Initialize $J$ global cluster models
    $K_s \leftarrow \lceil CK \rceil$    ▷ Number of clients to select per communication round
    for $t \in \{1, 2, \ldots\}$ do    ▷ Until convergence
        $S_t \subseteq \{1, 2, \ldots, K\}, |S_t| = K_s$    ▷ Random sampling of $K_s$ clients
        for all $k \in S_t$ do    ▷ For all clients, in parallel
            $\boldsymbol{w}^k(t+1), n_k, \hat{j}^k \leftarrow k.$CLIENT$\left( \left\{ \boldsymbol{w}_g^j \;\middle|\; j \in \mathcal{J} \right\} \right)$    ▷ Local training (Alg. 2)
        for all $j \in \{1, 2, \ldots, J\}$ do    ▷ For all cluster models
            $n_j \leftarrow \sum_{k \in \{S_t \,|\, \hat{j}^k = j\}} n_k$    ▷ Total number of samples for cluster model $j$ from clients where $j = \hat{j}$
            $\boldsymbol{w}_g^j(t+1) \leftarrow \sum_{k \in \{S_t \,|\, \hat{j}^k = j\}} \frac{n_k}{n_j} \boldsymbol{w}^k(t+1)$    ▷ Update cluster model $j$ with clients where $j = \hat{j}$
            if early stopping triggered for model $j$ then
                $\mathcal{J} \leftarrow \mathcal{J} \setminus j$    ▷ Optional: Remove $j$ from the set of selectable cluster models $\mathcal{J}$
```

Algorithm 2. Adaptive Expert Models for FL — client

```
procedure CLIENT($\left\{ \boldsymbol{w}_g^j \;\middle|\; j \in \mathcal{J} \right\}$)
    $\hat{j} \leftarrow$ CL.-EST.$\left( \varepsilon, \left\{ \boldsymbol{w}_g^j \;\middle|\; j \in \mathcal{J} \right\} \right)$, $n_k \leftarrow |P^k|$    ▷ Estimate best cluster (Alg. 3)
    $\boldsymbol{w}^k(t+1) \leftarrow$ UPDATE$\left( \boldsymbol{w}_g^{\hat{j}}(t), n_k \right)$    ▷ Perform local training using cluster model $\hat{j}$ (Alg. 4)
    return $(\boldsymbol{w}^k(t+1), n_k, \hat{j})$
```

4 Experiments

4.1 Datasets

We use three different datasets, with different non-IID characteristics, in which the task is an image multi-class classification task with varying number of classes. We summarize these datasets in Table 1.

- **CIFAR-10** [20], where we use a technique from [24] to create client partitions with a controlled *Label distribution skew*, see Sect. 4.2;
- **Rotated CIFAR-10** [11], where the client feature distributions are controlled by rotating CIFAR-10 images—an example of *same label, different features*;
- **Federated Extended MNIST (FEMNIST)** [4,5] with handwritten characters written by many writers, exhibiting many of the non-IID characteristics outlined in Sect. 2.2.

Algorithm 3. Adaptive Expert Models for FL — cluster assignment

```
procedure CL.-EST.($\varepsilon, \left\{ \boldsymbol{w}_g^j \;\middle|\; j \in \mathcal{J} \right\}$)
    return $\begin{cases} \operatorname{argmin}_{j \in \mathcal{J}} \sum_{i \in P^k} l\left( \boldsymbol{x}_i, y_i, \boldsymbol{w}_g^j \right) & \text{prob. } 1 - \varepsilon \quad ▷ \text{ Lowest loss model} \\ \mathcal{U}\{1, J\} & \text{prob. } \varepsilon \quad ▷ \text{ Random assignment} \end{cases}$
```

Algorithm 4. Adaptive Expert Models for FL—local update

```
procedure UPDATE($w^k(t+1), n_k$)                                  ▷ Mini-batch gradient descent
    for $e \in \{1, 2, \ldots, E\}$ do                              ▷ For a few epochs
        for all batches of size $B$ do                              ▷ Batch update
            $w^k(t+1) \leftarrow w^k(t+1) - \frac{\eta}{B} \nabla_{w^k(t+1)} \sum_{i=1}^{B} l_i\left(x_i, y_i, w^k(t+1)\right)$    ▷ Local parameter update
    return $w^k(t+1)$
```

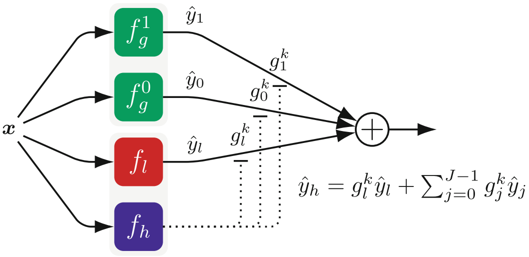


Fig. 2. Our solution with 2 global cluster models. Each client k has one local expert model $f_l(x, w_l^k)$ and share $J = 2$ expert cluster models $f_g^j(x, w_g^j)$ with all other clients. A gating model $f_h(x, w_h^k)$ is used to weigh the expert cluster models and produce a personalized inference $\hat{y}_h$ from the input x.

Table 1. Dataset summary statistics — number of samples per client.

				Samples, training set			**Samples, test set**		
Dataset	**Classes**	**K**	**C**	Mean	Min	Max	Mean	Min	Max
CIFAR-10	10	50	0.1	500.0	500	500	100.0	100	100
Rot. CIFAR-10	10	200	0.1	500.0	500	500	100.0	100	100
FEMNIST	62	182	0.07	199.7	98	387	22.7	11	44

4.2 Non-IID Sampling

In order to construct a non-IID dataset from the CIFAR-10 dataset [20] with the properties of class imbalance that we are interested in we first look at [27]. A *pathological non-IID* dataset is constructed by sorting the dataset by label, dividing it into shards of 300 data samples and giving each client 2 shards.

However, as in [24], we are interested in varying the degree of *non-IIDness* and therefore we assign two majority classes to each client which make up a fraction p of the data samples of the client. The remainder fraction $(1 - p)$ is sampled uniformly from the other 8 classes. When $p = 0.2$ each class has an equal probability of being sampled. A similar case to the *pathological non-IID* above is represented by $p = 1$. In reality, p is unknown.

4.3 Model Architecture

We start with the benchmark model defined in [4] which is a Convolutional Neural Network (CNN) model with two convolutional layers and one fully connected layer with fixed hyperparameters. However, in our case where n_k is small, the local model is prone to over-fitting, so it is desirable to have a model with lower capacity. Similarly, the gating model is also prone to overfitting due to both a small local dataset and the fact that it aims to solve a multi-label classification problem with fewer classes (expert models), than in the original multi-class classification problem. The local model, gating model and cluster models share the same underlying architecture, but therefore have hyperparameter such as number of filters in a hidden layer individually tuned, see Sect. 4.4. The `AdamW` [25] optimizer is used to train the local model and the gating model, while Stochastic Gradient Descent (SGD) [2] is used to train the cluster models to avoid issues related to momentum parameters when averaging. We use negative log-likelihood loss in (1).

4.4 Hyperparameter Tuning

Hyperparameters are tuned using [23] in four stages and used for all clients. For each model we tune the learning rate η, the number of filters in two convolutional layers, the number of hidden units in the fully connected layer, dropout, and weight decay. For the ε-greedy exploration method we also tune ε.

First, we tune the hyperparameters for a local model and for a single global model. Thereafter, we tune the hyperparameters for the gating model using the best hyperparameters found in the earlier steps. Lastly, we tune ε with two cluster models $J = 2$. For the no exploration experiments we set $\varepsilon = 0$.

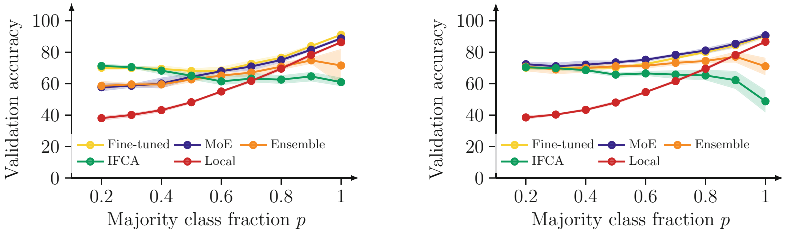
Hyperparameters depend on p and J but we tune the hyperparameters for a fixed majority class fraction $p = 0.2$, which corresponds to the IID case. The tuned hyperparameters are then used for all experiments. We show that our method is still robust in the fully non-IID case when $p = 1$. See Table 2 for tuned hyperparameters in the CIFAR-10 experiment.

4.5 Results

We summarize our results for the class imbalance case exemplified with the CIFAR-10 dataset in Table 3. In Fig. 3, we see an example of how the performance varies when we increase the non-IID-ness factor p for the case when $J = 3$. In Fig. 3a we see the performance of IFCA [11] compared to our solution in Fig. 3b. We also compare to: a local model fine-tuned from the best cluster model, an entirely local model, and an ensemble model where we include *all* cluster models as well as the local model with equal weights. In Fig. 4 we vary the number of cluster models J for different values of the majority class fraction p.

Table 2. Tuned hyper-parameters in the CIFAR-10 experiment for the global cluster models, the local models and the gating model.

Model	η	**Conv1**	**Conv2**	**FC**	**Dropout**	**Weight Dec.**	**E**	ε
Global	5.86×10^{-3}	128	32	1024	0.80	1.10×10^{-3}	3	0.33
Local	2.69×10^{-4}	32	256	256	0.76	9.89×10^{-3}		
Gate	3×10^{-6}	12	12	8	0.78	6.88×10^{-4}		



(a) No exploration. **(b)** ε-greedy exploration.

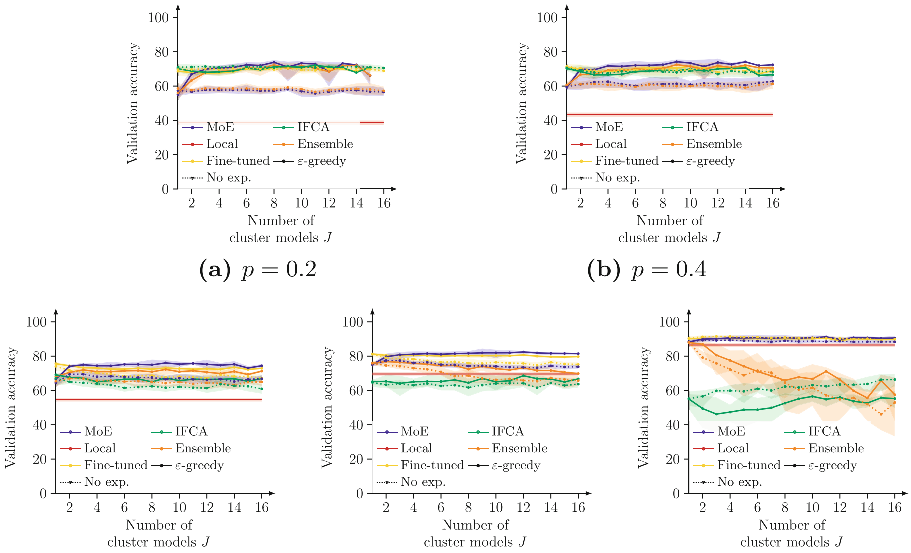
Fig. 3. Results for CIFAR-10. Comparison between no exploration and our ε-greedy exploration method for $J = 6$. Our proposed MoE solution with ε-greedy exploration is superior in all cases from IID to pathological non-IID class distributions, here shown by varying the majority class fraction p.

Table 3. Results for CIFAR-10 and $p \in \{0.2, 0.4, \ldots, 1\}$ when $J = 6$. Mean μ and standard deviation σ for our exploration method ε-greedy and without exploration. We compare our proposed MoE solution to the baseline from IFCA [11]. Our proposed solution is superior in all but one case, indicated by **bold** numbers.

			MoE		**IFCA**		**Ensemble**		**Fine-tuned**		**Local**	
p	**Exp. strategy**	# trials	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
0.2	ε-greedy (ours)	7	**72.39**	1.26	70.38	0.74	70.82	1.87	70.16	0.61	38.52	0.75
	No exploration	6	57.73	1.95	71.25	0.86	58.58	1.96	70.13	1.04	38.06	0.87
0.4	ε-greedy (ours)	6	**72.05**	1.79	68.59	1.00	69.96	2.33	70.28	1.05	43.36	0.47
	No exploration	9	60.12	2.37	68.29	1.56	59.54	1.31	69.42	1.54	43.16	0.71
0.6	ε-greedy (ours)	8	**75.22**	0.75	66.53	0.98	71.44	1.17	72.50	0.54	54.63	0.33
	No exploration	9	67.94	0.75	61.47	2.27	65.15	0.87	68.27	1.71	55.04	0.45
0.8	ε-greedy (ours)	14	**81.09**	1.18	65.23	2.40	74.44	0.76	80.13	1.02	69.49	0.58
	No exploration	15	75.04	0.93	62.59	1.95	70.82	1.56	76.49	1.27	69.55	0.70
1.0	ε-greedy (ours)	14	90.76	0.82	48.79	5.35	71.06	4.03	90.26	1.02	86.65	0.39
	No exploration	6	88.79	0.52	60.97	2.07	71.56	9.12	**91.11**	0.33	86.37	0.31

An often overlooked aspect of performance in FL is the inter-client variance. We achieve a smaller inter-client variance, shown for CIFAR-10 in Fig. 6a and Table 3.

We see that for CIFAR-10 our ε-greedy exploration method achieves better results for lower values of p by allowing more of the cluster models to converge—thereby more cluster models are useful as experts in the MoE, even though the models are similar, see Fig. 5a. For higher values of p we see that the cluster models are adapting to existing clusters in the data, see Fig. 5c. The most interesting result is seen in between these extremes, see Fig. 5b. We note that the same number of clients pick each cluster model as in IFCA, but we manage to make a better selection and achieve higher performance.

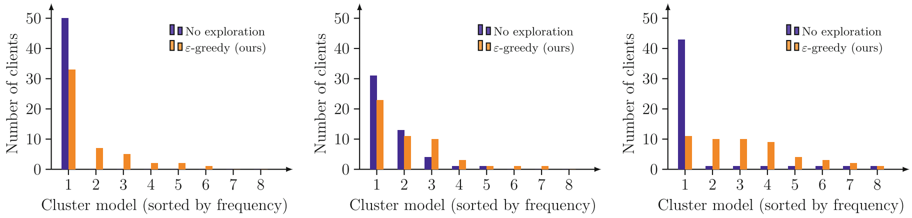


(a) $p = 0.2$ **(b)** $p = 0.4$

(c) $p = 0.6$ **(d)** $p = 0.8$ **(e)** $p = 1$

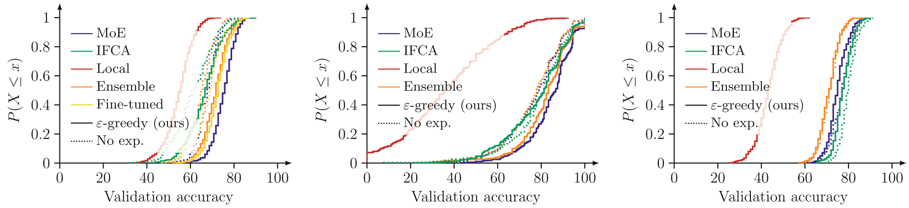
Fig. 4. Results for CIFAR-10. Comparison between no exploration (colored dashed lines) and the ε-greedy exploration method (colored solid lines). Our proposed MoE solution with the ε-greedy exploration outperforms all other solutions, including the baseline from IFCA [11]. It performs better the greater the non-IIDness, here seen by varying the majority class fraction p. Furthermore, our solution is robust to changes in the number of cluster models J.

For the rotated CIFAR-10 case in Table 4 and 7b we see that IFCA manages to assign each client to the correct clusters at $J = 2$, and in this *Same label, different features* case our exploration method requires a larger J to achieve the same performance. We also note the very high $\varepsilon = 0.82$. More work is needed on better exploration methods for this case.



(a) $p = 0.2$ **(b)** $p = 0.6$ **(c)** $p = 1$

Fig. 5. Results for CIFAR-10. The number of clients in each cluster for the different exploration methods. Clusters are sorted—the lowest index corresponds to the most picked cluster. Our ε-greedy exploration method picks the cluster models more evenly.



(a) CIFAR-10, J=6, p=0.6. **(b)** FEMNIST, J=9. **(c)** Rot. CIFAR-10, J=2.

Fig. 6. CDF of client accuracy. Comparison between no exploration (colored dashed lines) and the ε-greedy exploration method (colored solid lines). Our proposed MoE solution with ε-greedy exploration improves accuracy and fairness for two of the datasets.

The FEMNIST dataset represents a more difficult scenario since there are many non-IID aspects in this dataset. We find from Table 5 and Fig. 7a that for FEMNIST the best performance is achieved when $J = 9$ and in Fig. 6b we show the distribution of accuracy for the clients for the different models.

5 Related Work

The FedAvg algorithm [27] is the most prevalent algorithm for learning a global model in FL. This algorithm has demonstrated that an average over model parameters is an efficient way to aggregate local models into a global model. However, when data is non-IID, FedAvg converges slowly or not at all. This has given rise to personalization methods for FL [14,18]. Research on how to handle non-IID data among clients is ample and expanding. Solutions include fine-tuning locally [33], meta-learning [9,17], MAB [30], multi-task learning [22], model heterogeneous methods [7,13], data extension [32], distillation-based methods [16,21] and Prototypical Contrastive FL [28].

Mixing local and global models has been explored by [6], where a scalar α is optimized to combine global and local models. In [29] the authors propose to use MoE [15] and learn a gating function that weighs a local and global expert to

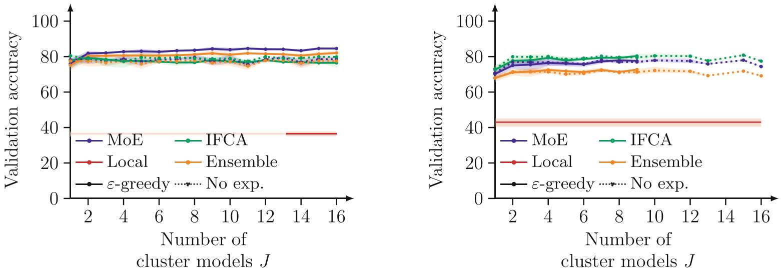
Table 4. Results for Rotated CIFAR-10. Mean μ and standard deviation σ with varying number of cluster models J, for our exploration method ε-greedy and for the baseline exploration method from IFCA. At J = 2 all clients have picked the correct cluster model.

			MoE		**IFCA**		**Ensemble**		**Local**	
J	**Exp. strategy**	# trials	μ	σ	μ	σ	μ	σ	μ	σ
1	ε-greedy (ours)	7	70.20	1.35	72.63	2.01	67.89	1.23	42.79	1.10
	No exploration	10	70.70	0.95	**73.08**	0.87	68.14	1.15	43.27	0.92
3	ε-greedy (ours)	7	75.45	2.44	77.91	1.90	71.45	1.97	42.76	1.15
	No exploration	10	76.94	1.14	**79.95**	0.92	71.51	1.13	42.94	1.65
6	ε-greedy (ours)	3	75.59	1.20	78.59	0.92	71.26	1.55	43.70	0.84
	No exploration	4	76.49	0.86	**79.30**	0.51	70.78	0.82	43.09	1.12
9	ε-greedy (ours)	1	77.58		**80.33**		72.65		40.55	
	No exploration	6	76.96	0.96	79.6	1.27	71.38	1.46	42.53	1.90

Table 5. Results for FEMNIST. Mean μ and standard deviation σ with varying number of cluster models J for our exploration method ε-greedy and for the baseline exploration method from IFCA.

			MoE		**IFCA**		**Ensemble**		**Local**	
J	**Exp. strategy**	# trials	μ	σ	μ	σ	μ	σ	μ	σ
1	ε-greedy (ours)	3	75.99	3.03	77.80	3.88	75.14	3.20	36.27	0.76
	No exploration	3	78.15	0.79	**80.48**	0.77	77.53	0.57	36.35	0.39
3	ε-greedy (ours)	3	**82.07**	0.22	78.35	0.74	80.49	1.28	36.12	0.78
	No exploration	3	77.59	0.60	78.21	3.63	76.52	1.95	36.15	0.43
6	ε-greedy (ours)	3	**82.72**	0.81	77.11	1.03	80.48	1.57	36.90	0.88
	No exploration	3	78.26	0.71	79.04	1.58	77.42	0.89	36.13	1.04
9	ε-greedy (ours)	3	**84.37**	1.01	78.05	0.84	81.82	1.62	36.72	0.85
	No exploration	3	77.06	1.88	78.73	1.35	76.49	1.45	36.52	0.76

enhance user privacy. This work is developed further in [24], where the authors use a gating function with larger capacity to learn a personalized model when client data is non-IID. We differ in using cluster models as expert models, and by evaluating our method on datasets with different non-IID characteristics. Recent work has studied clustering in FL settings for non-IID data [3,11,19,26]. In [11] the authors implement a clustering algorithm for handling non-IID data in form of covariate shift. Their proposed algorithm learns one global model per cluster with a central parameter server, using the training loss of global models on local data of clients to perform cluster assignment. In their work, they only perform clustering in the last layer and aggregate the rest into a single model. If a global model cluster is unused for some communication rounds, the global cluster model



(a) FEMNIST.

(b) Rotated CIFAR-10.

Fig. 7. Results for FEMNIST and rotated CIFAR-10. Comparison between no exploration (colored dashed lines) and the ε-greedy exploration method (colored solid lines). Our proposed MoE solution is superior in the FEMNIST case, but need more cluster models to achieve similar performance to the baseline in the rotated CIFAR-10 case.

is removed from the list to reduce communication overhead. However, this means that a client cannot use other global cluster models to increase performance.

6 Discussion

We adapted the inspiring work by [11] to work better in our setting and efficiently learned expert models for non-IID client data. Sending all cluster models in each iteration introduces more communication overhead. We addressed this by removing converged cluster models from the set of selectable cluster models in Algorithm 1, although this is not used in our main results. This only affects the result to a minor degree, but has a larger effect on training time due to wasting client updates on already converged models. Another improvement is the reduces complexity in the cluster assignment step. A notable difference between our work and IFCA is that we share *all* weights, as opposed to only the last layer in [11]. These differences increase the communication overhead further, but this has not been our priority and we leave this for future work.

7 Conclusion

In this chapter, we have investigated personalization in a distributed and decentralized ML setting where the data generated on the clients is heterogeneous with non-IID characteristics. We noted that neither FedAvg nor state-of-the-art solutions achieve high performance in this setting. To address this problem, we proposed a practical framework of MoE using cluster models and local models as expert models and improved the adaptiveness of the expert models by balancing exploration and exploitation. Specifically, we used a MoE [24] to make better use of the cluster models available in the clients and added a local model. We showed that IFCA [11] does not work well in our setting, and inspired by the MAB field, added an ε-greedy exploration [31] method to improve the adaptiveness of the cluster models which increased their usefulness in the MoE. We evaluated our method on three datasets representing different non-IID settings, and found that our approach achieve superior performance in two of the datasets, and is robust in the third. Even though we tune our algorithm and hyperparameters in the IID setting, it generalizes well in non-IID settings or with varying number of cluster models—a testament to its robustness. For example, for CIFAR-10 we see an average accuracy improvement of 29.78% compared to IFCA and 4.38% compared to a local model in the pathological non-IID setting. Furthermore, our approach improved the inter-client accuracy variance with 60.39% compared to IFCA, which indicates improved fairness, but 60.98% worse than a local model.

In real-world scenarios data is distributed and often displays non-IID characteristics, and we consider personalization to be a very important direction of research. Finding clusters of similar clients to make learning more efficient is still an open problem. We believe there is potential to improve the convergence of the cluster models further, and that privacy, security and system aspects provide interesting directions for future work.

Acknowledgment. This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

The computations were enabled by the supercomputing resource Berzelius provided by National Supercomputer Centre at Linköping University and the Knut and Alice Wallenberg foundation.

We thank all reviewers who made suggestions to help improve and clarify this manuscript, especially Dr. A. Alam, F. Cornell, Dr. R. Gaigalas, T. Kvernvik, C. Svahn, F. Vannella, Dr. H. Shokri Ghadikolaei, D. Sandberg and Prof. S. Haridi.

References

1. Bonawitz, K., Eichner, H., et al.: Towards federated learning at scale: system design. In: Proceedings of Machine Learning and Systems (MLSys), Stanford, CA, USA (2019). https://proceedings.mlsys.org/book/271.pdf
2. Bottou, L., Curtis, F.E., Nocedal, J.: Optimization methods for large-scale machine learning. SIAM Rev. **60**(2), 223–311 (2018). https://doi.org/10.1137/16M1080173, https://epubs.siam.org/doi/10.1137/16M1080173
3. Briggs, C., Fan, Z., Andras, P.: Federated learning with hierarchical clustering of local updates to improve training on non-IID data. In: International Joint Conference on Neural Networks (IJCNN), Glasgow, United Kingdom. IEEE (2020). https://doi.org/10.1109/IJCNN48605.2020.9207469
4. Caldas, S., et al.: LEAF: a benchmark for federated settings. CoRR abs/1812.01097 (2018). http://arxiv.org/abs/1812.01097
5. Cohen, G., Afshar, S., Jonathan, T., van Schaik, A.: EMNIST: extending MNIST to handwritten letters. In: International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA. IEEE (2017). https://doi.org/10.1109/IJCNN.2017.7966217
6. Deng, Y., Kamani, M.M., Mahdavi, M.: Adaptive personalized federated learning. CoRR abs/2003.13461 (2020). https://arxiv.org/abs/2003.13461

7. Diao, E., Ding, J., Tarokh, V.: HeteroFL: Computation and Communication Efficient Federated Learning for Heterogeneous Clients. In: 9th International Conference on Learning Representations (ICLR), Austria (2021). https://openreview.net/forum?id=TNkPBBYFkXg
8. Berggren, V., Inam, R., Mokrushin, L., Hata, A., Jeong, J., Mohalik, S.K., Forgeat, J., Sorrentino, S.: Artificial intelligence and machine learning in next-generation systems. Technical report, Ericsson Research, Ericsson AB (2018). https://www.ericsson.com/en/white-papers/machine-intelligence
9. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, NSW, Australia. PMLR (2017). https://doi.org/10.5555/3305381.3305498, http://proceedings.mlr.press/v70/finn17a.html
10. GDPR: Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data and the free movement of such data (2016). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:02016R0679-20160504
11. Ghosh, A., Chung, J., Yin, D., Ramchandran, K.: An efficient framework for clustered federated learning. In: Advances in Neural Information Processing Systems (NeurIPS) (2020). https://proceedings.neurips.cc/paper/2020/file/e32cc80bf07915058ce90722ee17bb71-Paper.pdf
12. Hanzely, F., Richtárik, P.: Federated learning of a mixture of global and local models. CoRR abs/2002.05516 (2020). https://arxiv.org/abs/2002.05516
13. He, C., Annavaram, M., Avestimehr, S.: FedNAS: federated deep learning via neural architecture search. CoRR abs/2004.08546 (2020). https://arxiv.org/abs/2004.08546
14. Hsieh, K., Phanishayee, A., Mutlu, O., Gibbons, P.B.: The non-IID data quagmire of decentralized machine learning. In: Proceedings of the 37th International Conference on Machine Learning (ICML). PMLR (2020). http://proceedings.mlr.press/v119/hsieh20a.html
15. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Comput. **3**(1), 79–87 (1991). https://doi.org/10.1162/neco.1991.3.1.79, https://ieeexplore.ieee.org/document/6797059
16. Jeong, E., Oh, S., Kim, H., Park, J., Bennis, M., Kim, S.L.: Communication-efficient on-device machine learning: federated distillation and augmentation under non-IID private data. CoRR abs/1811.11479 (2018). http://arxiv.org/abs/1811.11479
17. Jiang, Y., Konečný, J., Rush, K., Kannan, S.: Improving federated learning personalization via model agnostic meta learning. CoRR abs/1909.12488 (2019). http://arxiv.org/abs/1909.12488
18. Kairouz, P., McMahan, H.B., et al., B.A.: Advances and open problems in federated learning, vol. 14, pp. 1–210 (2021). https://doi.org/10.1561/2200000083
19. Kim, Y., Hakim, E.A., Haraldson, J., Eriksson, H., da Silva Jr., J.M.B., Fischione, C.: Dynamic clustering in federated learning. In: International Conference on Communications (ICC), Montreal, QC, Canada, pp. 1–6. IEEE (2021). https://doi.org/10.1109/ICC42927.2021.9500877
20. Krizhevsky, A.: Learning multiple layers of features from tiny images (2009). https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf
21. Li, D., Wang, J.: FedMD: heterogenous federated learning via model distillation. CoRR abs/1910.03581 (2019). http://arxiv.org/abs/1910.03581

22. Li, T., Hu, S., Beirami, A., Smith, V.: Ditto: fair and robust federated learning through personalization. In: Proceedings of the 38th International Conference on Machine Learning (ICML). PMLR (2021). http://proceedings.mlr.press/v139/li21h.html
23. Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J.E., Stoica, I.: Tune: a research platform for distributed model selection and training. CoRR abs/1807.05118 (2018). http://arxiv.org/abs/1807.05118
24. Listo Zec, E., Mogren, O., Martinsson, J., Sütfeld, L.R., Gillblad, D.: Federated learning using a mixture of experts. CoRR abs/2010.02056 (2020). https://arxiv.org/abs/2010.02056
25. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: 7th International Conference on Learning Representations (ICLR), New Orleans, LA, USA (2019). https://openreview.net/forum?id=Bkg6RiCqY7
26. Mansour, Y., Mohri, M., Ro, J., Suresh, A.T.: Three approaches for personalization with applications to federated learning. CoRR abs/2002.10619 (2020). https://arxiv.org/abs/2002.10619
27. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA. PMLR (2017). http://proceedings.mlr.press/v54/mcmahan17a.html
28. Mu, X., et al.: FedProc: prototypical contrastive federated learning on non-IID data. CoRR abs/2109.12273 (2021). https://arxiv.org/abs/2109.12273
29. Peterson, D.W., Kanani, P., Marathe, V.J.: Private federated learning with domain adaptation. CoRR abs/1912.06733 (2019). http://arxiv.org/abs/1912.06733
30. Shi, C., Shen, C., Yang, J.: Federated multi-armed bandits with personalization. In: The 24th International Conference on Artificial Intelligence and Statistics (AISTATS) (2021). http://proceedings.mlr.press/v130/shi21c.html
31. Sutton, R.S.: Generalization in reinforcement learning: successful examples using sparse coarse coding. In: Advances in Neural Information Processing Systems (NeurIPS). MIT Press (1995). https://proceedings.neurips.cc/paper/1995/file/8f1d43620bc6bb580df6e80b0dc05c48-Paper.pdf
32. Tijani, S.A., Ma, X., Zhang, R., Jiang, F., Doss, R.: Federated learning with extreme label skew: a data extension approach. In: International Joint Conference on Neural Networks (IJCNN), Shenzhen, China. IEEE (2021). https://doi.org/10.1109/IJCNN52387.2021.9533879
33. Wang, K., Mathews, R., Kiddon, C., Eichner, H., Beaufays, F., Ramage, D.: Federated evaluation of on-device personalization. CoRR abs/1910.10252 (2019). http://arxiv.org/abs/1910.10252
34. Wu, S., et al.: Motley: benchmarking heterogeneity and personalization in federated learning. CoRR abs/2206.09262 (2022). https://doi.org/10.48550/arXiv.2206.09262