Paper ID: 968

Heterogeneous Federated Learning with Scalable Server Mixture-of-Experts

Jingang Jiang*, Yanzhao Chen*, Xiangyang Liu, Haiqi Jiang, Chenyou Fan

South China Normal University

1.Motivation

- Traditional symmetric federated architectures are difficult to deploy large models on resource-constrained devices.
- Existing federated MoE methods (e.g., FedMix/FedJETs) suffer from low aggregation efficiency and performance under Non-IID data due to static aggregation strategies.

2.Method

Fed-MoE: An asymmetric federated framework in which the server level is a large MoE (composed of main experts and routted experts), and the client side is a single expert model.

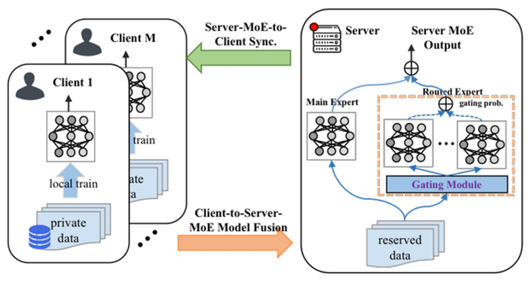

Figure 1: Overview of Fed-MoE. Compact client models federate into a large unified server Mixture-of-Experts.

The training of Fed-MoE is divided into three stages (Stage-A to Stage-C), completing a round of federated learning iterations, as shown in Figure 2.

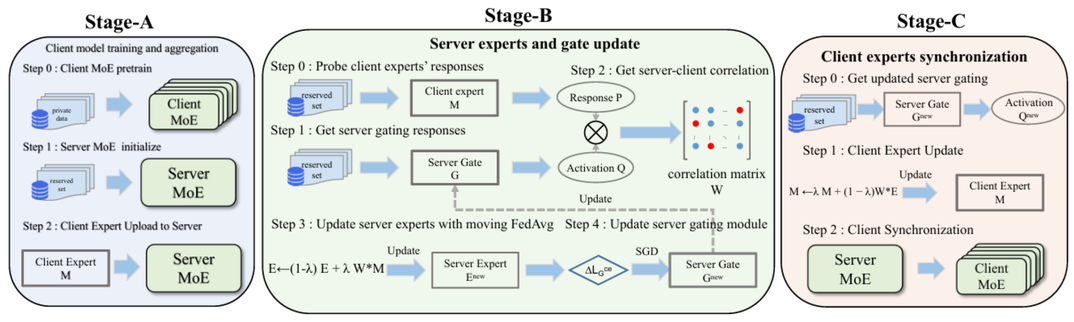

Figure 2: Overview of our Fed-MoE pipeline. Stage A-C completes one FL round. Stage-A trains client experts and sends to server. Stage-B iteratively updates server experts and gate. Stage-C synchronizes updated client experts back to clients.

➢ *Stage-A: Local client training and uploading*

The server aggregates models from m randomly selected client as a federation denoted as $M = \{M_1, M_2, \dots, M_{m-1}, M_m.\}$.

➢ *Stage-B: Server experts and gate update*

Stage-B iteratively updates server experts E and gate G for T iterations, which we decompose as the following steps.

- ***Step-0: Probe client experts' responses***

For all client expert models, extract the confidence P_y corresponding to the true class.

- ***Step-1: Get server gating responses***

Repeat *Step-1* to *Step-4* for T inner - loop iterations. At t-th iteration, we get the activation prob from the gating module. distribution as:

$$Q \leftarrow G(X) \in \mathbb{R}^{k\times 1}$$

- ***Step-2: Get server-client correlation***

The outer product of Q and P_y is a correlation matrix:

$$W \leftarrow Q \times P_y^T \in \mathbb{R}^{k\times m}$$

We subsequently apply a row-wise softmax operation to normalize the correlation matrix to get W^r.

- ***Step-3: Update server experts with moving FedAvg***

Update server experts using the moving average strategy:

$$E_0^{t+1} \leftarrow (1-\lambda)\cdot E_0^{t+1} + \lambda \cdot \bar{M},$$

$$E_i^{t+1} \leftarrow (1-\lambda)\cdot E_i^{t+1} + \lambda \cdot W^r \cdot M, \forall i \geq 1,$$

The $\lambda \in (0,1)$ controls the moving-average rate. We use simple averaging for $\bar{M}$ with $\bar{M} = \frac{1}{m}\sum_{i=1}^{m} M_i$. The term $\bar{M}$ assigns relevant client parameters weighted by correlation W^r and adds up to server weights.

- ***Step-4: Update server gating module***

We design the cross-entropy task loss with gating entropy regularization, outlined as follows:

$$L^{gate}(G,a) = L_G^{ce} + \beta \cdot L_G^{\text{ent}}$$

$$L_G^{ent} = E_X[H(Q_X)]$$

$$= -1/|D| \sum_{X\in D}\sum_{k=1}^{K} Q_X[k]\cdot log Q_X[k]$$

➢ *Stage-C: Client experts synchronization*

Using the updated gating module G and the client response P_y, build the extended correlation matrix. Normalize it column-wise to derive the updated server-client correlation matrix W^c. Then, we use moving average to update the client model.

$$M \leftarrow \lambda \cdot M + (1-\lambda)\cdot (W^c)^T \cdot E$$

3.Experiment

➢ *Fed-MoE Results*

Dateset	FEMNIST			CIFAR10			SENT140			YELP			AVG
Model	CNN			ResNet			BERT			GPT-2			Acc
Client Num.	10	50	100	10	50	100	10	50	100	10	50	100	
FedAvg 2017	91.89	75.84	74.02	62.30	28.37	24.63	75.90	75.38	73.98	51.44	52.53	50.50	61.39
FedProx 2020	91.66	77.88	76.01	61.88	35.04	32.13	76.06	76.89	75.38	52.88	52.68	52.58	63.42
CentMoE 2017	57.27	57.27	57.27	51.08	51.08	51.08	74.64	74.64	74.64	51.15	51.15	51.15	58.54
FedMix 2021	88.97	83.30	80.83	61.72	59.67	57.20	76.18	76.25	76.01	52.73	51.54	51.23	67.96
FedJETs 2023	89.54	76.96	79.71	66.65	57.81	55.84	71.90	69.83	69.55	50.12	47.97	48.97	65.40
Fed-MoE	**92.11**	**86.03**	**82.58**	**67.62**	**65.52**	**60.73**	**77.56**	**77.96**	**78.10**	**54.11**	**54.12**	**53.46**	**70.83**

Table 2: Classification accuracy (%) on FEMNIST, CIFAR-10, SENT-140 and Yelp datasets with Non-IID settings. Vision tasks are shaded in yellow, and language tasks are in green.

In all datasets, Fed-MoE demonstrates significant advantages. In extreme Non-IID (e.g. only 375 samples per client side in CIFAR-10), Fed-MoE can still maintain stable performance, while other methods experience significant fluctuations due to the failure of the parameter averaging strategy.

➢ *Ablation*

- ***Multi-task training procedures.***

Fed-MoE variants	FEMNIST	CIFAR
w/o GEnt & Sync	78.04	61.27
+GEnt	78.57 (+0.5)	62.07 (+0.8)
+Sync	81.48 (+3.4)	63.94 (+2.6)
+GEnt+Sync (Fed-MoE)	**86.03** (+8.0)	**65.52** (+4.2)

Table 4: Ablation of multi-task training.

The Table 4 evaluates the proposed gating entropy (GEnt) loss and client synchronization (Sync) in multi-task training, finding that combining both Gent and Sync yields significant gains (8.0% and 4.2%). GEnt encourages server experts to specialize in tasks, while Sync unifies the data space across clients, enhancing effectiveness, especially for Non-IID data.

- ***Weight of Gating Entropy.***

GEnt weight	w/o	10^{-4}	10^{-3}	10^{-2}	10^{-1}
Fed-MoE	63.94	64.25	**65.52**	63.14	62.60

Table 5: Ablation of gating entropy weight β in Eq.(9).

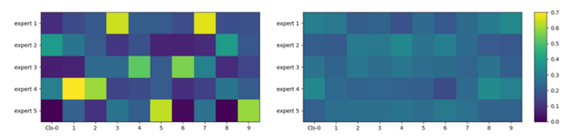

Figure 3: Gating heat-maps reveals each expert (row) specifies certain classes (col.), with left $\beta = 10^{-1}$ and right $\beta = 10^{-3}$.

As shown in Table 5 and Fig. 3, $\beta = 10^{-3}$ achieves a balance between specificity and versatility, distributing the gating more evenly across multiple experts while still maintaining strong accuracy.

- ***Communication costs, training and inference efficiency***

Dataset	FEMNIST (ResNet)	Yelp (GPT-2)
MoE Params.	6.5 / 26 / 52 (M)	0.36 / 0.93 / 1.59 (B)
Comm. Cost	33 / 130 / 33 (M)	1.02 / 2.79 / 1.02 (B)

Table 1: Server MoE parameters and FL communication costs for FedAvg, FedMix, and our Fed-MoE.

Server MoE	5-Exp	10-Exp	20-Exp	30-Exp
Avg-MoE	82.78	82.83	**83.01**	82.92
Fed-MoE	**86.03**	84.77	85.46	85.07

Table 3: Ablation of the number of server experts.

Top-L	1	2	3	5
FEMNIST	**86.03**	84.49	82.76	84.73
CIFAR	**65.52**	65.50	65.16	64.98

Table 6: Ablation of Top-L routed experts in inference.

	0-Main	1-Main	2-Main	3-Main
FEMNIST	81.05	**86.03**	80.61	83.29
SENT140	75.11	**77.96**	77.11	76.88

Table 8: Ablation of the number of server experts.

Table 1 shows Fed-MoE's communication cost; Table 3 and 8 respectively demonstrate the effects of the number of routed and main experts on model performance; Table 6 reveals the impact of the number of activated experts on model inference. Results indicate Fed-MoE excels in communication cost, balancing training and inference efficiency with performance.

- ***Server reserved data***

Fig. 4 shows an accuracy gap of about 2-3% between IID and Non-IID scenarios for both Fed-MoE and FedMix. However, Fed-MoE showed a slight advantage than FedMix in AUC metrics on both datasets.

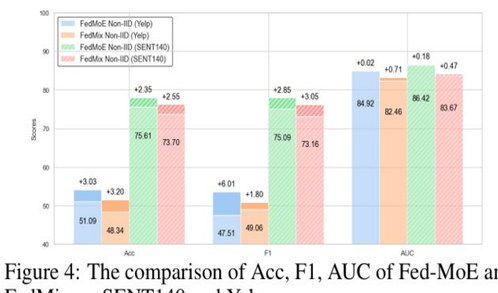

Figure 4: The comparison of Acc, F1, AUC of Fed-MoE and FedMix on SENT140 and Yelp.

4.Conclusion

- We propose Fed-MoE, an efficient asymmetric FL scheme to build a large server-side MoE from client experts.
- We introduce dynamic expert scheduling and collaborative optimization (main + routed experts), with gating entropy regularization to enhance expert differentiation efficiency.
- Ablation studies demonstrate improved convergence and communication performance, highlighting the scheme's effectiveness.