Paper number: 1659

FEDERATED TRACE: A NODE SELECTION METHOD FOR MORE EFFICIENT FEDERATED LEARNING

Zirui Zhu, Lifeng Sun

INTRODUCTION

Federated Learning (FL) is a learning paradigm, which allows the model to directly use a large amount of data in edge devices for training without heavy communication costs and privacy leakage. An important problem that FL faced is the heterogeneity of data at different edge nodes, resulting in a lack of convergence efficiency. Now, one way to solve this problem is to speed up the convergence of FL through delicate node selection instead of random node selection. In this paper, we propose a node selection method called FedTrace. In FedTrace, we define the training trace and use it to guide the selection of nodes in each round of training. Experiments on various settings demonstrate that our method significantly reduces the number of communication rounds required in FL.

Training Trace

We think that the data distribution of edge nodes can guide the selection of nodes, but it is inaccessible in FL because this information is closely related to the privacy of users. Therefore, we hope there are some values that can implicitly reflect the data distribution of the node. Some performance metrics (such as accuracy, entropy, etc.) of the global model on the edge nodes are a good choice. However, there is still a problem that they contain too little information. So we extend these metrics in the time domain to include more information. For the global model generated in each round of training, we record the values of its metrics at each node to obtain several time series. The time series will form a trace in the space formed by these metrics, so we call it training trace (Fig.1).

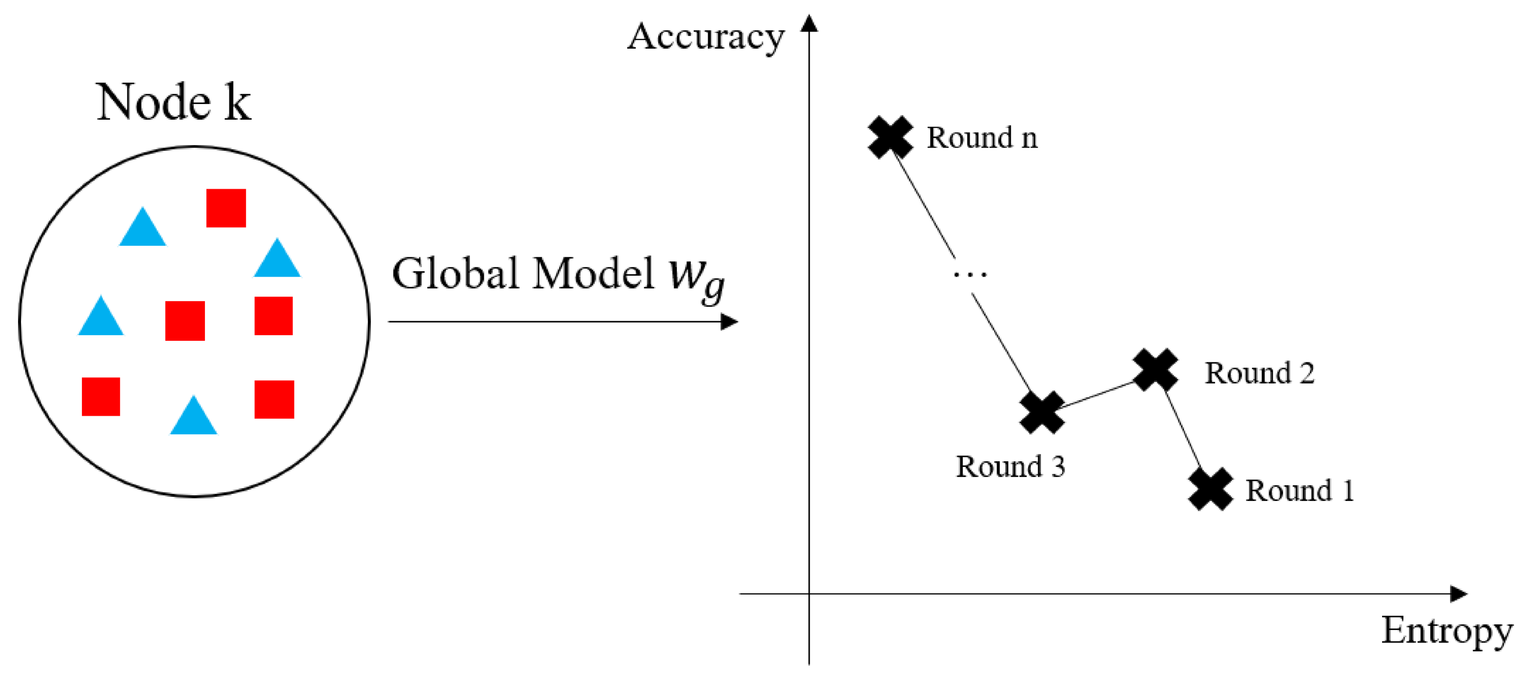

Figure 1: The training trace formed by accuracy and entropy.

Federated Trace

Algorithm 1 Federated Trace

FedTrace:
initialize w_g^1
for each round t = 1, 2, 3, ... **do**
  $m \leftarrow \max(C \cdot K, 1)$
  for each node k **do** in parallel
    calculate A_k^t and E_k^t with w_g^t
    update training trace of node k
  end for
  cluster training traces into N_c clusters
  $S_t \leftarrow$ **NodeSelection**(m)
  for each node $k \in S_t$ **do** in parallel
    $w_k^t \leftarrow w_g^t$
    update w_k^t by local training
  end for
  $w_g^{t+1} \leftarrow \frac{1}{N_{S_t}} \sum_{k \in S_t} N_k \cdot w_k^t$
end for

NodeSelection(m)**:**
$S \leftarrow$ random select $\lfloor \frac{m}{N_c} \rfloor$ nodes from each cluster
if $m \% N_c \neq 0$ **then**
  $d \leftarrow m - |S|$
  for i = 1, 2, ... , d **do**
    $C \leftarrow$ i-th largest cluster
    $v \leftarrow$ randomly select one node from $C (v \notin S)$
    $S \leftarrow S + v$
  end for
end if
return S

EXPERIMENTS

Evaluation Method
- The number of communication rounds required for the global model to reach a specific accuracy.
- Baseline: FedAvg and FedActive

Dataset
- MNIST and FashionMNIST
- Artificial Partition (AP) and Dirichlet Partition (DP)
- Bigger α or smaller σ indicates more uniform data distribution.

	FedAvg	FedActive	FedTrace
AP ($\sigma = 0.55$)	27	26	**20**
AP ($\sigma = 0.73$)	49	42	**33**
AP ($\sigma = 0.82$)	88	75	**39**
AP ($\sigma = 0.91$)	106	94	**43**
DP ($\alpha = 0.1$)	238	204	**152**
DP ($\alpha = 10$)	80	68	**56**

Table 1: Experimental result on MNIST.

	FedAvg	FedActive	FedTrace
AP ($\sigma = 0.55$)	37	34	**32**
AP ($\sigma = 0.73$)	61	58	**50**
AP ($\sigma = 0.82$)	136	123	**112**
AP ($\sigma = 0.91$)	283	263	**236**
DP ($\alpha = 0.1$)	465	384	**295**
DP ($\alpha = 10$)	67	62	**55**

Table 2: Experimental result on FashionMNIST.

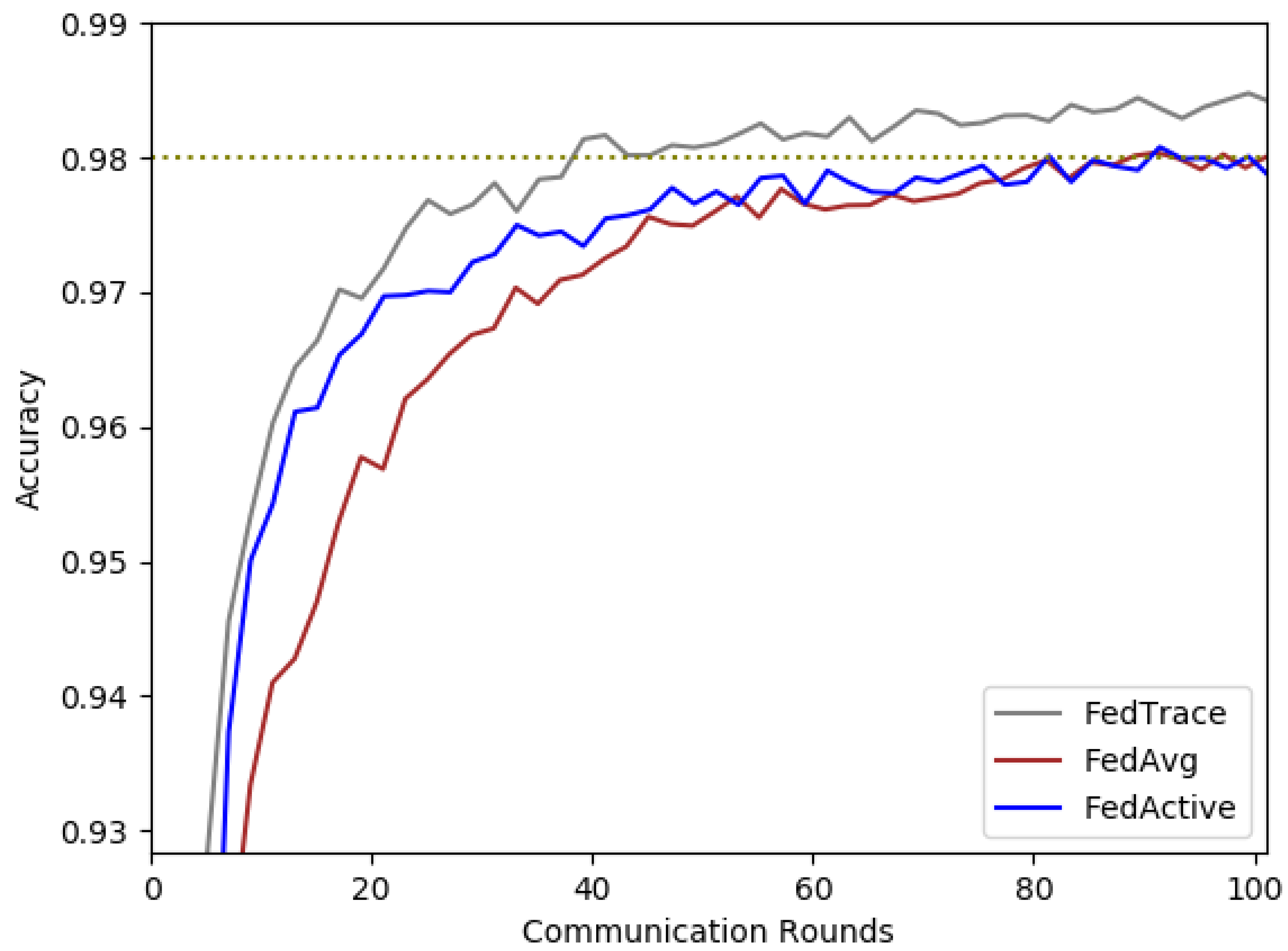

Figure 2:Accuracy v.s. Communication rounds (MNIST dataset, Artificial Partition with σ = 0.82)