Supplementary Material for FedLTN: Federated Learning for Sparse and Personalized Lottery Ticket Networks

Anonymous ECCV submission

Paper ID 6634

Appendices

A Architecture of Custom-CNN

In order to assess the performance of FedLTN and other FL algorithms on a small neural network (feasible to be deployed on resource-constrained edge devices) we use a custom-built CNN with the architecture shown in Figure A.1. It has two convolutional layers with max-pooling followed by three fully-connected layers. Consequently, the model size is a fraction of ResNet18, with the custom CNN's 62,006 parameters only taking up 240KB of memory.

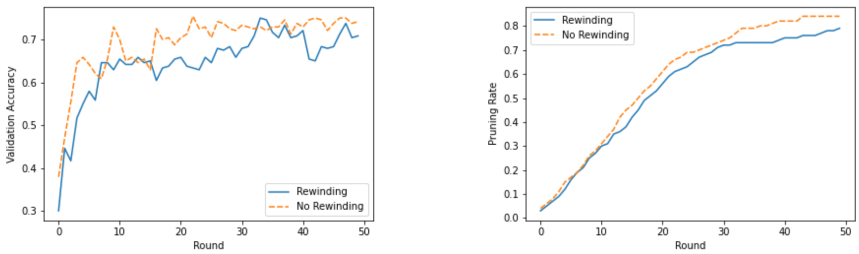
Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 6, 28, 28]	456
MaxPool2d-2	[-1, 6, 14, 14]	0
Conv2d-3	[-1, 16, 10, 10]	2,416
MaxPool2d-4	[-1, 16, 5, 5]	0
Linear-5	[-1, 120]	48,120
Linear-6	[-1, 84]	10,164
Linear-7	[-1, 10]	850

Fig. A.1: Architecture of custom CNN.

Notably, no batch normalization is present for any of the layers. As such, in limiting the network size, we also sacrifice the BN-preservation component of our proposed FedLTN framework. Furthermore, the performance of FedBN is the same as vanilla Federated Averaging.

B Effect of Rewinding

We conduct experiments with rewinding to iteration 0 after pruning. We find that rewinding performs poorer than our method with no rewinding in terms of pruning rate, convergence speed, and communication cost.



(a) (b)

Fig. B.2: Left (a): Comparision of validation accuracies at each round. We observe that our method converges faster without rewinding to round 0. Right (b): Comparison of pruning rate at each round. Skipping rewinding increases the pruning speed and thus overall reducing the communication cost.

C Custom CNN Results on CIFAR-10

Dataset	Algorithm	Avg Test Accuracy%	Min Test Accuracy%
	FedAvg	50.0	50.0
	LotteryFL(0.1)	72.6	51.8
	LotteryFL(0.5)	71.6	52.3
	LotteryFL(0.9)	70.6	57.5
CIFAR-10	FedLTN (0.1)	**79.4**	**62.0**
Custom CNN	FedLTN(0.5)	<u>78.0</u>	<u>61.0</u>
No BN Layers	FedLTN(0.9)	72.98	57.5
	FedLTN(0.1; rewind)	72.95	57.3
	FedLTN(0.5; rewind)	64.6	48.8
	FedLTN(0.9; rewind)	64.3	50.0

Table C.1: Performance of FedLTN against baselines for a custom CNN model with no BN layers.

D Jump-Start for TinyImageNet

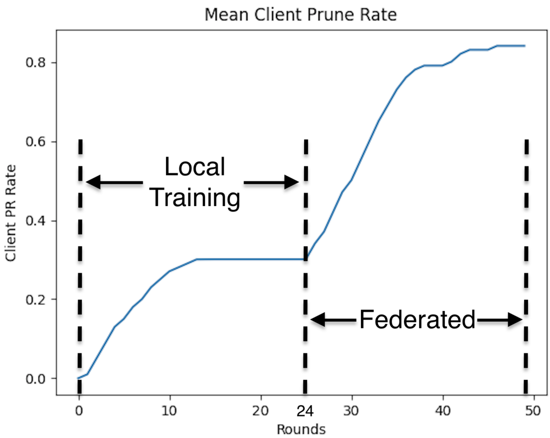


Fig. D.3: Mean prune rate for ResNet18 on TinyImageNet using 25 Jump-Start and 25 FedLTN rounds. 50% participation rate and 10% prune step was used for all rounds. We set a max prune of 30% for Jump-Start and 90% for FedLTN.

E Performance on FEMNIST (LEAF benchmark)

Table E.2 compares FedLTN's performance with other baselines on the FEMNIST dataset. We use the same hyperparameters as used for CIFAR-10 and TinyImageNet.

Algorithm	Avg Test
FedAvg	60.95
LotteryFL(0.5)	61.45
FedLTN(0.9)	66.95
FedLTN(0.9; jumpstart)	65.55

Table E.2: Performance on the FEMNIST (LEAF) dataset