Appendix for "Learning Fast and Slow: Towards Inclusive Federated Learning"

Muhammad Tahir Munir, Muhammad Mustansar Saeed, Mahad Ali, Zafar Ayyub Qazi, Ihsan Ayyub Qazi, and Agha Ali Raza

Department of Computer Science, Lahore University of Management Sciences
{18030016,18030047,21100119,zafar.qazi,ihsan.qazi,agha.ali.raza}@lums.edu.pk

A.1 Ablation Study

Benefits of Activation-Based Model Pruning. Figure 1 illustrates benefits of using activation based sub-model selection with over random sub-model selection.

Fig. 1: Activation-based sub-model selection vs. Random sub-model selection.

Benefits of Model Generalization Module. We illustrate the performance of the model generalization module of Hasaas in Figure 2. We find that the model generalization module provided up to 6.7% improvement in test accuracy compared to the model without the generalization module.

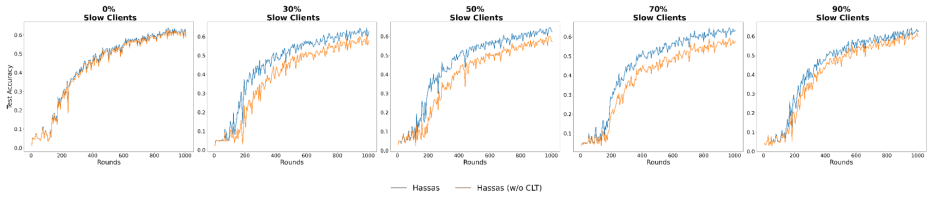

Fig. 2: Hasaas with and without CLT.

Choice of Normalizing σ by $1/\sqrt{t}$. Figure 3 shows that failing to normalize σ leads to large model parameters, which can cause the model to become unstable, distribution. Our empirical evaluation shows that $1/\sqrt{t}$ is the optimal choice for normalizing σ.

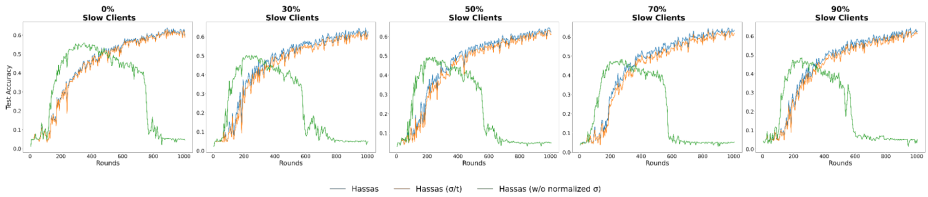

Fig. 3: With and Without normalized σ.

Using CLT with FedProx & FedAvg. We performed experiments by applying the model generalization module of Hasaas to FedAvg and FedProx on the FEMNIST skewed data. We observe that the differences between these schemes (i.e., FedAvg and FedProx) with and without CLT are small and not significant as illustrated in Figure 4.

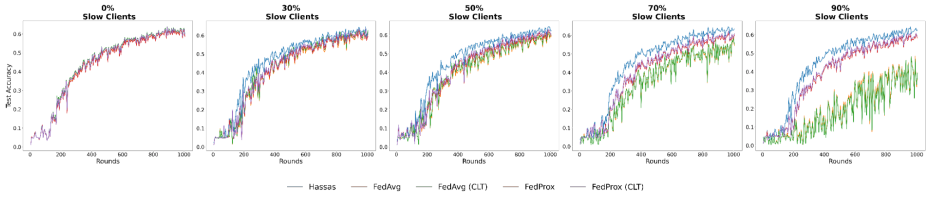

Fig. 4: FedAvg and FedProx with CLT.

A.2 Datasets and Models

Here we provide complete details of datasets used in our experiments. We experiment with diverse set of real world datasets.

FEMNIST. This is federated version of extended MNIST dataset which consists of images of 28 x 28. FEMNIST is an image classification dataset with 62 classes. This data is partitioned by writer of the digits/characters. We train a CNN model on this dataset. We have two versions of femnist dataset with different levels of statistical heterogeneity. One version is less heterogeneous and used by Ditto [2]. While we create the other version in LEAF by assigning 5 random classes to each client. We show that the Hasaas can be more beneficial in the face of statistical heterogeneity, i.e., skewed FEMNIST data. For FEMNIST, we use a model with two convolutional layers which are followed by pooling layers, and a final dense layer.

FMNIST. We experiment on Fashion MNIST dataset used by Ditto [2]. It is also an image classification dataset which has 10 classes. We use the same model for FMNIST that we described above for FEMNIST.

CIFAR-10. CIFAR-10 is also an image classification dataset. To create a federated version of CIFAR-10, we randomly assign an image to a client. Each image has the same probability to belong to any client, which makes this dataset IID. CIFAR-10 model architecture is also same as FEMNIST.

SENT140. Sentiment140 [1] is a text dataset of tweets which is used for sentiment analysis task. We perform our experiments on sent140 used in FedProx [4]. We train a model with two LSTM layers and a fully connected layer. We use a pretrained 300D GloVe embedding [3]. Glove embeds each character of the input sequence into a 300-dimensional space and outputs one character per training sample after 2 LSTM layers and a fully-connected layer.

Table 1: Statistics of five real federated datasets.

Dataset	Devices	Samples	Mean	Stdev
FEMNIST	206	30774	149	59
FEMNIST (skewed)	1600	80000	50	0
FMNIST	500	72505	145	138
CIFAR-10	250	60000	240	0
Sent140	772	40,783	53	32

A.3 Real Device Experiments

Quantifying training time against different model sizes. We performed the experiments for different model sizes on synthetic and femnist dataset to quantify training time on actual devices.

Figure 5 shows the training time of linear model on slow and fast devices against different model sizes. Blue bar represents the training times of fast device, orange represents when a large model was served to the slow client, green bar represents when 30% pruned was served, and so on. Slow client takes $\sim 1.4-3.1\times$ more time as compared to fast client. We observe that the model training time is reduced by $1.4\times$ compared with fast client if we serve the 50% reduced model to the slow client. Figure 6 shows the training time of CNN model on FEMNIST dataset for different model drop rates.

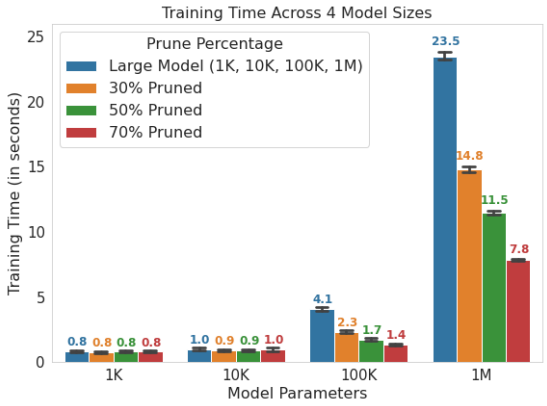

Fig. 5: Training time against different model sizes for synthetic dataset.

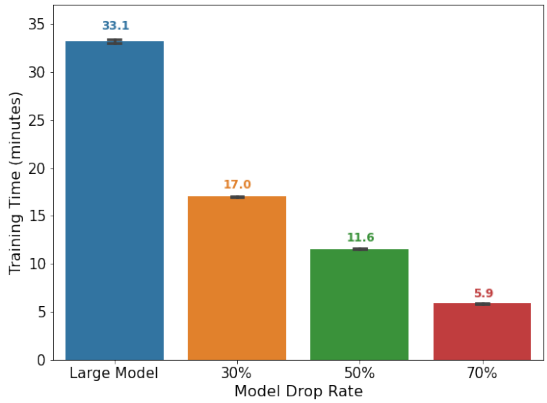

Fig. 6: Training times under Hasaas with different model sizes.

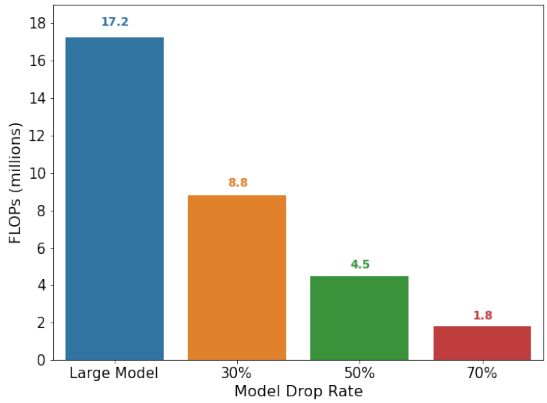

Fig. 7: FLOPS for different model drop rates.

A.4 Network Experiments

Figure 8 shows the impact of network heterogeneity on convergence time. To induce heterogeneity, we fixed the fast client's network speed to 20 Mbps and varied the network speed of the slow client from 1 Mbps to 10 Mbps. To tackle the network heterogeneity, we experimented with serving the same (small/large) and differential model based on devices' capabilities. We also evaluated the both schemes using the varying model drop rates (MDR), i.e., 0% - 90%. We measured the round completion time and test accuracy for each configuration until 100 rounds. The results show that network heterogeneity slowdowns the convergence, e.g., for MDR (0%), a slow client takes 19.1 hours on a 1 Mbps which is 1.2× more time than a 5 Mbps (i.e., 15.9 hours). Moreover, we also observed that this heterogeneity relatively impacts more on the large models compared to small models, i.e., relatively more improvements in convergence time when increasing bandwidth to 5 Mbps for MDR (0% and 30%) than MDR ($>50\%$).

In contrast to same model serving, differential serving the model (i.e., serving the large model to the fast client and small model to the slow client) achieves better accuracy over same model serving and comparable time reduction for MDR (30% and 50%). However, for MDR (70% and 90%), we see a relatively more time in differential model serving than the same model serving. In the differential model serving, the fast client is now becoming the bottleneck in the computation phase that's why we see a relatively more time, however, accuracy is improved across all MDRs.

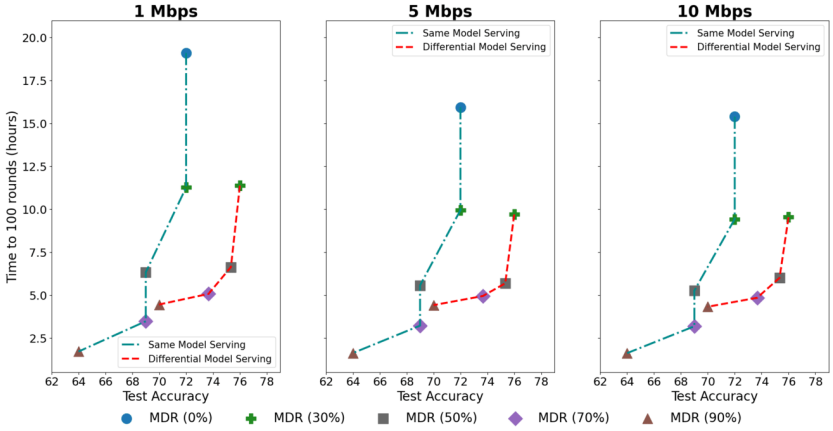

Fig. 8: Impact of network heterogeneity to model test accuracy and time to 100 rounds.

Figure 9 shows the comparison of time spent during different stages (i.e., computation and communication) in the federated learning process. The results show that communication cost reduces as we increase the bandwidth, i.e., cost is reduced by 4.9× on increasing the bandwidth to 5 Mbps for MDR (30%). We also observe that communication cost is not improved beyond certain bandwidth (e.g., 5 Mbps in our experiments). As MDR increases, communication cost is reduced significantly on a lower Mbps (i.e., 1 Mbps), however, bandwidth (> 5 Mbps), increasing the MDR does not improve communication cost significantly. We observed that a client spends more time in the computation phase than in communication, i.e., for MDR (30%), computation (9.1 hours) and communication (2.1 hours). As MDR increases, computation cost reduces significantly, i.e., cost reduced from 9.1 hours to 1.6 hours.

The Table 2 shows the benefits of serving the differential model over the same large model in terms of FLOPs and time. The results show that, for 50% MDR, time to obtain 70% accuracy is improved by 7.6× compared to a large model which is as a result of 3.68× reduction in the FLOPs.

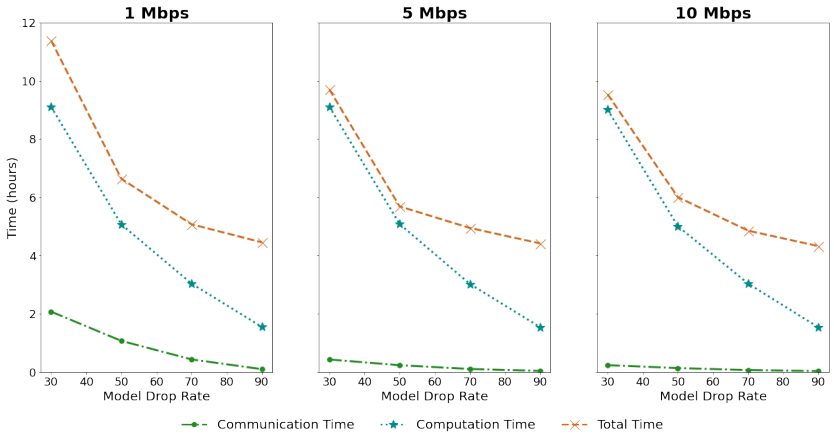

Fig. 9: Time spent during different stages in FL using different model drop rates.

Approach	Time (minutes)	FLOPS (millions)
FedAvg	136.9	4.49
Hasaas (30%)	17.7	2.44
Hasaas (50%)	17.9	1.22
Hasaas (70%)	53.8	0.51
Hasaas (90%)	63.9	0.12

Table 2: Time and FLOPs to reach 70% accuracy, here, Hasaas is without CLT.

A.5 Simulation Experiments

Figure 11 shows the training loss on multiple real datasets under various levels of system heterogeneity. Figure 12 shows the test loss variance for less system heterogeneity environment.

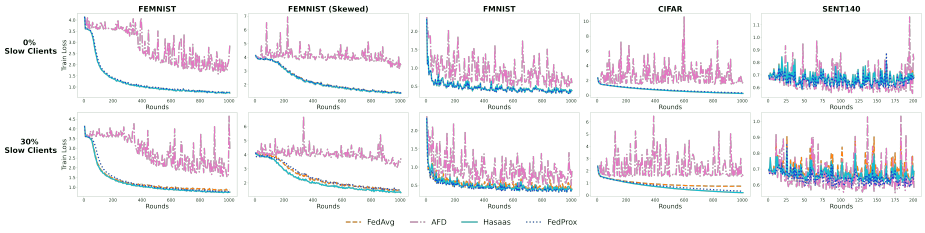

Fig. 10: Train loss as a function of round number for 0% and 30% system heterogeneity on various datasets.

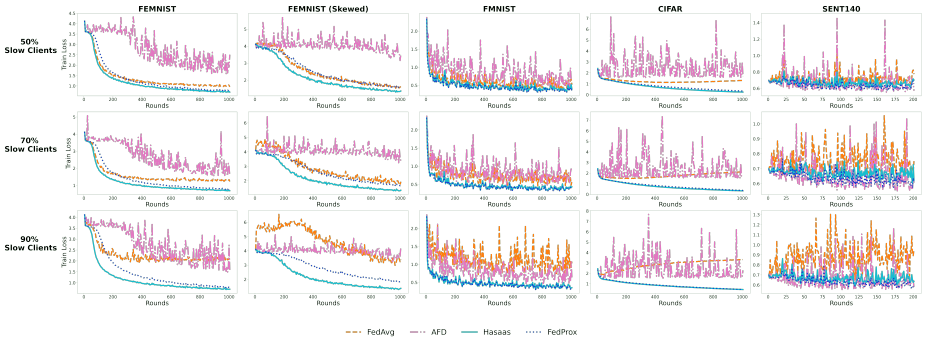

Fig. 11: Train loss as a function of round number for 50%, 70% and 90% system heterogeneity on various datasets indicates that Hasaas results in significant convergence improvements.

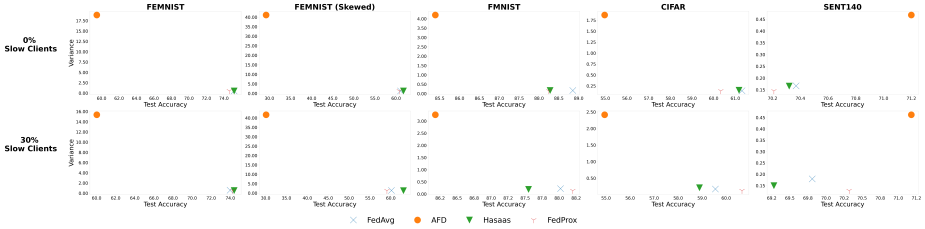

Fig. 12: Variance of Test Loss vs Test Accuracy in less system heterogeneity.

References

1. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. Processing pp. 1–6 (2009), http://www.stanford.edu/~alecmgo/papers/TwitterDistantSupervision09.pdf
2. Li, T., Hu, S., Beirami, A., Smith, V.: Ditto: Fair and robust federated learning through personalization. CoRR **abs/2012.04221** (2020), https://arxiv.org/abs/2012.04221
3. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. vol. 14, pp. 1532–1543 (01 2014). https://doi.org/10.3115/v1/D14-1162
4. Sahu, A.K., Li, T., Sanjabi, M., Zaheer, M., Talwalkar, A.S., Smith, V.: Federated optimization in heterogeneous networks. arXiv: Learning (2020)