DIVIDE-AND-CONQUER FEDERATED LEARNING UNDER DATA HETEROGENEITY

Pravin Chandran, Raghavendra Bhat,
Avinash Chakravarthy and Srikanth Chandar

Intel Technology India Pvt. Ltd, Bengaluru, India

ABSTRACT

Federated Learning allows training of data stored in distributed devices without the need for centralizing training-data, thereby maintaining data-privacy. Addressing the ability to handle data heterogeneity (non-identical and independent distribution or non-IID) is a key enabler for the wider deployment of Federated Learning. In this paper, we propose a novel Divide-and-Conquer training methodology that enables the use of the popular FedAvg aggregation algorithm by over-coming the acknowledged FedAvg limitations in non-IID environments. We propose a novel use of Cosine-distance based Weight Divergence metric to determine the exact point where a Deep Learning network can be divided into class-agnostic initial layers and class-specific deep layers for performing a Divide and Conquer training. We show that the methodology achieves trained-model accuracy at-par with (and in certain cases exceeding) the numbers achieved by state-of-the-art algorithms like FedProx, FedMA, etc. Also, we show that this methodology leads to compute and/or bandwidth optimizations under certain documented conditions.

KEYWORDS

Federated Learning, Divide and Conquer, Weight divergence.

1. INTRODUCTION

Federated Learning has been proposed as a new learning paradigm to overcome the privacy regulations and communication overheads associated with central training [1,2]. In Federated Learning, a central server shares a global model with participating client devices and the model is trained on the local datasets available at the client device. The local dataset is never shared with the server, instead, local updates to the global model are shared with the server. The server combines the local updates from the participating clients using an Optimization (or Aggregation) Algorithm and creates a new version of the global model. This process is repeated for the required number of communication rounds until the desired convergence criteria are achieved.
Federated Learning differs significantly from traditional learning approaches in terms of optimization in a distributed setting, privacy preserving learning, and communication latency during the learning process [3]. Optimization in Distributed setting differs from the traditional learning approach due to statistical and systems heterogeneity [1]. The statistical heterogeneity manifests itself in the form of non-independent and identical distribution (non-IID) of training data across participating clients. The non-IID condition arises due to a host of reasons that is specific to the local environment and usage patterns at the client. Causes for the skewed data distribution have been surveyed extensively and it has been proven that any real-world scale deployment of Federated Learning should address the challenges around non-IID data. A good example specific to the medical domain can be found in [4]. Several approaches have been

David C. Wyld et al. (Eds): AIMLA, DBDM, CCNET - 2021
pp. 21-33, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.111302

studied to address the non-IID heterogeneity. Data Distillation which involves sharing of client data with central server [5, 6], Client specific local models or Personalization layers to customize the last few layers of the global model specific to the client data [7, 8, 9, 10, 11], Novel optimization algorithms are some of these most researched approaches.

Data Distillation techniques violate the strict privacy requirements. Client specific model approach results in multiple models, which does not cater to any specific requirement for a single model for deployment. In this paper, we focus on the Optimization Algorithm approach to address the non-IID challenge. While there are numerous state-of-the-art algorithms like FedProx [12], FedMA [13], FedMAX [14] etc., these approaches are not productized in a large scale to the best of knowledge of the authors. Hence, we focus on the most widely deployed FedAvg algorithm [1] and investigate improving its ability to handle non-IID data to the same level as state-of-the-art algorithms like FedMA, FedProx, FedMAX, etc.

The primary contribution of this paper is proposing a novel Divide-and-Conquer training methodology which in combination with FedAvg is able to meet state-of-the-art performance in simulated environment. Another contribution of this paper is the novel use of the Cosine Distance based Weight Divergence metric to partition the global model into class agnostic initial layers and class-specific deep layers. The two parts of the global model are trained in a mutually exclusive manner while freezing the other part. Under certain documented conditions, this approach also leads to better compute and bandwidth optimization.

The rest of the paper is organized as follows. Section 2 discusses the limitation with vanilla FedAvg algorithm while Section 3 explains the Divide-and-Conquer methodology. We document the simulation environment, experiments, and results in the simulated environment in Section 4 establishing the state-of-the-art credentials of the approach. Finally, we conclude the paper and discuss possible future work in Section 5.

2. FEDAVG AND ITS CHALLENGES

Federated Learning (FL) methods are designed to train over multiple devices, each holding their own data, with a central server driving the global learning objective across the entire network. The standard formulation of FL aims to find the minimizer of the overall population loss [12] shown in EQ1 below.

$$\min_{w} f(w) = \sum p_k \; F_k(w) = E_k \; [F_k(w)], \qquad (EQ1)$$
$$where\ N\ is\ number\ of\ devices, p_k \geq 0\ and \sum_k p_k = 1$$

In general, the local objectives measure the local empirical risk over possibly differing data distributions with samples available at each device. In a non-IID environment, the assumption of a global minimizer being representative of the overall population is not valid as every client has its own data distribution which differs from other clients and the overall population. Hence, on each client, a local objective function based on the client's data is used as a surrogate for the global objective function. At each outer iteration, a subset of devices is selected, and local solvers are used to optimize the local objective functions of the selected client. Each client then communicates its local model updates to the central server, which aggregates them and updates the global model accordingly. In addition to the usual hyper-parameters of traditional learning like batch size, optimizer, etc., Federated Learning has additional hyper-parameters like epochs per round (E_p), number of communication rounds, number of participants in each round, and optimization algorithm which can be tweaked for optimal performance.

In FedAvg, the local objective function at client k is $F_k(.)$, and the local solver is the stochastic gradient descent (SGD), with the same learning rate η and number of local epochs used on each client. At each round, a subset K<N of the total clients are selected and run SGD locally for E_p number of epochs, and then the resulting model updates are averaged. The details are summarized below.

Algorithm 1: Federated Averaging Algorithm

Input: *K, T, η, E_p, w^o, N, p_k, k=1,…,N*

for *t=1* to *T-1* do
Server selects a subset S_t of *K* clients at random.
Server sends w_t to all chosen clients
Each client $k \in S_t$ updates w_t for E_p epochs of SGD
On F_k, with step size η to obtain w_k^{t+1}
Each client $k \in S_t$, sends w_k^{t+1} back to server
Server aggregates the *w's* as $w_{t+1} = \frac{1}{K}\sum k \in S_t \ w_k^{t+1}$
end for

Tuning of the hyper-parameters is a critical requirement for optimal performance of FedAvg. The number of epochs plays a critical role in convergence as a greater number of epochs leads to faster convergence. This comes at the cost of higher compute on client devices but with the benefit of lower communication. However, the high number of epochs has diminishing returns on the speed of convergence in non-IID conditions. For FedAvg, there is a significant drop in reduction of accuracy due to weight divergence [5]. The trade-off between high number of epochs and convergence speed for FedAvg has been addressed in other optimization algorithms like FedProx, FedMA, FedMAX etc. FedProx is very similar to FedAvg but addresses the limitations of the latter by adding a proximal term to client cost functions to limit the impact of local updates within a particular range of global model. This approach allows the number of epochs to be tuned based on the non-IIDness of the client data. While it addresses the weight divergence issue with FedAvg, the convergence speed is slower at higher number of epochs when compared to other state-of-the-art algorithms [6,13,14,15]. FedMA offers the best accuracy and convergence speed in comparison to others but comes with significant compute cost on the client devices. The complexity of this algorithm is also high in comparison with FedAvg or FedProx leading to restrictions on its applicability on certain NN models.

An ideal optimization algorithm should come with the simplicity and elegance of FedAvg, allow for state-of-the-art accuracy in non-IID environments with comparable or better convergence speed. In this work, we present a novel Federated Training methodology that is well suited to handle non-IID challenges using the simple FedAvg algorithm. Our methodology eliminates performance overheads associated with methods like FedMA while achieving comparable accuracy. Since FedAvg is the de-facto standard in majority production deployments, the proposed method can be easily integrated to offer significant accuracy and convergence benefits with little performance overhead.

Note on the terminology: In the rest of the document, clients will be referred to as Collaborators and the server will be referred to as Aggregator, reflecting the role they play in the overall federation.

3. DIVIDE-AND-CONQUER TRAINING METHODOLOGY

The impact of non-IIDness of data in Federated Learning is well researched in literature. A non-IID data environment leads to over-fitting of local models to the skewed training data at individual collaborators resulting in distortion of previously aggregated feature detectors and descent of SGD optimizer to different local minima at different collaborators.

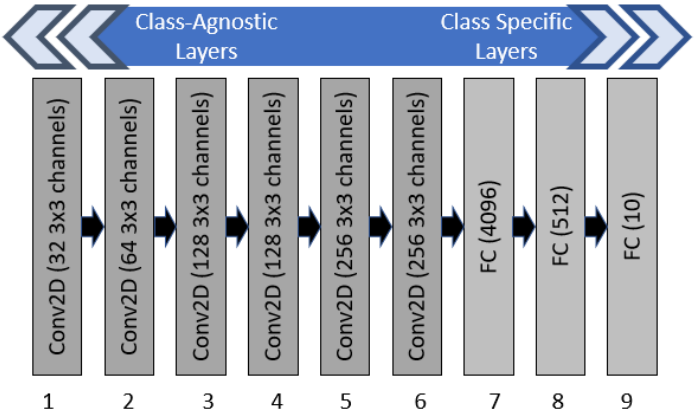

Figure 1. DNN Layer Significance - VGG9 Image Classification Topology

Typically, the initial layers of a Deep Neural Network (DNN) learn low level or class agnostic features and deeper layers are responsible for learning high level or class-specific features as illustrated for a vision architecture, VGG9 [16], in Figure 1. For training paradigms like Transfer Learning \cite{tlearn1}, data scarcity mandates the use of special training methods that learn class agnostic features from generic datasets and learn class specific features for any new tasks by freezing the initial layers. This process of decoupling feature-learning and task-learning has been successfully applied to multiple training tasks including recent advances like Few Shot Learning [17]. This work extends the idea to Federated Learning to address the challenges with non-IID. Our methodology involves splitting the given DNN into two parts, namely (a) Class Agnostic Layers and (b) Class Specific Layers.

The two parts are trained separately. Federated Learning is typically performed using several Communication Rounds (CR), where trained weights from individual collaborators are aggregated together in a central Aggregator. Our proposed method configures collaborators to perform feature-learning and task-learning or fine-tuning in alternate rounds as shown in Figure 2. Weights corresponding to relevant trained layers alone are transferred over to the Aggregator, which results in communication bandwidth reduction. Communication saving is realized during model transfers in both directions (i) Transfer of global models to Collaborators and (ii) Transfer of local trained models from Collaborator to the Aggregator.

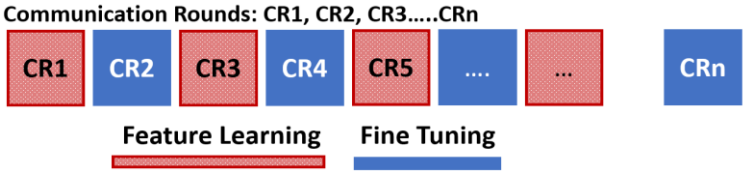

Figure 2. Divide and Conquer Training Methodology using alternate Feature-Learning and Fine-Tuning rounds. CR1, C2, represent the communication rounds.

Class Agnostic layers, comprised of initial layers of the DNN architecture, are trained more aggressively as compared to Class-Specific layers. Class Agnostic layer training is treated similar to feature-learning. Class Specific layers, consisting of deep layers are trained similar to fine-tuning. This ensures that weight divergence across different collaborators, due to non-IIDness of constituent data is minimal and features are insulated from distortion that would otherwise occur due to combined learning of all layers.

While methods like FedProx limit weight divergence, they penalize all layers of the network and hinder learning in Class Agnostic layers. Our approach addresses this by allowing different layers of a network to train differently after grouping initial layers separately from deep layers. Training rounds are configured to alternate between feature-learning and fine-tuning to facilitate learning under non-IID conditions by freezing relevant layers of DNN architecture. At the beginning of a communication round, the aggregator broadcasts the desired hyper-parameter configurations to collaborators, together with specifications for layers to be frozen. The exact point at which a DNN architecture must be broken into two parts is decided based on 'weight divergence' observed from the pre-pass round of training. The key contributions of our paper can be summarized to the following two key points:

- Novel methodology, called Divide-and-Conquer (D&C), to train topology in pairs of feature-learning and fine-tuning steps to handle non-IID conditions.
- Novel use of weight-divergence metric, observed from the pre-pass round of training, to split the given DNN topology into Class Agnostic and Class Specific layers. This metric provides a measure of non-IIDness across participating collaborators as a mapping of the layers of DNN architecture they impact the most.

Choice of layers that are chosen for base class feature-learning as against novel class fine-tuning is a hyper-parameter in Divide-and-Conquer training methodology. Few options for splitting the VGG9 topology is shown in Figure 3. For instance, Divide3, divide at layer3, assigns layers 1 to 3 for learning class agnostic features and remaining layers for learning class or task specific features. This hyper-parameter is dependent on the weight-divergence metric which in turn reflects the non-IIDness of data.

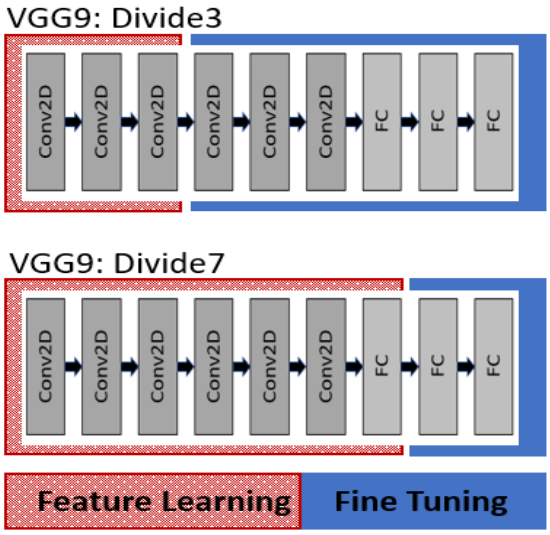

Figure 3. VGG9: Topology Division and assignment of layers to Feature-Learning and Fine-Tuning Groups

After determining an optimal split, feature-learning} and fine-tuning is achieved by control of other hyper-parameters like number of Epochs E_p and Learning rate η. Fine-Tuning round of learning is scheduled using lower E_p and η, which is aligned with the conditions under which FedAvg performs the best in non-IID conditions. Federated Learning at a faster pace is achieved by alternating low-level feature-learning and high-level fine-tuning along with appropriate hyper-parameters as described in the next section.

4. DIVIDE-AND-CONQUER: EXPERIMENTS, RESULTS, AND DISCUSSION

This section describes the simulation environment, experiments done, and results. The comparison with other state-of-the-art approaches is also captured in the results section to establish the state-of-the-art credentials of our proposed approach.

4.1. Experimental Setup

We present observations from Divide-and-Conquer on VGG9 topology using 3 different non-IID conditions as in [13], which includes coverage for convolutional layers and LSTMs. Classification and NLP models used were also same as [13].

- Classification using Color Skewed CIFAR10 Dataset [19]: CIFAR10 dataset is split into two groups of 5 classes each, with each class assigned uniquely to the two collaborators. To skew the data further using a 95-5% skew pattern, 95% of images in the first group are converted to gray-scale and 5% of images in the second group are converted to gray-scale. This results in the first collaborator holding gray-scale dominant data and the second collaborator holding color dominant data.
- Classification using Class Imbalanced CIFAR10 Data: Data is distributed non-uniformly across different collaborators to create non-IID conditions from the perspective of total training data per collaborator as well as the number of records per class.
- Next Character prediction model on Shakespeare dataset [18] leveraging non-IIDness in speaking-roles: Data corresponding to each speaking-role in the play is grouped to create unique collaborators, to simulate natural non-IID condition. For the trial, we selected only clients with a minimum of 10k data points and sampled a random subset of 66 clients.

4.2. Hyper Parameter Tuning

4.2.1. Fine Tuning Epoch and Learning Rate

Divide-and-Conquer allows the use of variable hyper-parameters for different parts of the network. As discussed earlier, we train feature-learning group more aggressively than fine-tuning group by control of parameters like E_p and η. Use of lower E_p for fine-tuning rounds result in slightly better accuracy compared to higher epochs. This is because the local models are skewed by over-fitting to non-IID data at the individual collaborators. By using lower values for epoch and learning rate for fine-tuning rounds, we achieve better accuracy while simultaneously reducing compute requirements needed for fine-tuning rounds. Data from Color Skewed distribution is presented in Figure 4. This observation is in alignment with the behaviour of FedAvg where a high number of E_p leads to lower training accuracy due to weight divergence.

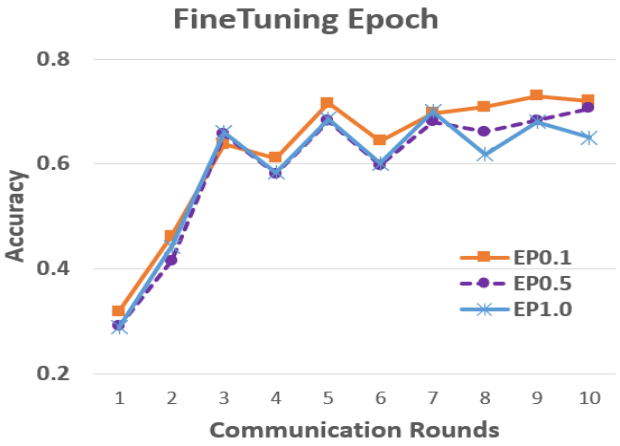

Figure 4. Effect of Training Epochs: Ep0.1 corresponds to fine-tuning epoch that is 10% of the value used for feature-learning

4.2.2. Fine Tuning Epoch and Learning Rate

Depending on the nature and magnitude of non-IIDness, the Class Agnostic and Class Specific layers in a given model will diverge across different collaborators. We explored, weight divergence in the learned model, to guide D&C methodology. The metric given below (EQ2) was explored in [5].

$$W_d = ||W_1 - W_2||/||W_1|| \quad \text{(EQ2)}$$

$$W_d = CosineDist(W_1, W_2)/||W_1|| \quad \text{(EQ3)}$$

We modified the divergence computation as shown in (EQ3), to capture direction aware divergence to guide our D&C methodology. Weight divergence from VGG9 model for Color Skewed non-IID simulation described in Section 4.1 is shown in Figure 5. A pre-pass training is initially performed for 5 rounds using entire model and layer-wise divergence strategy is devised using observations from the pre-pass round as reference. For notation, model at end of pre-pass comprising 5 rounds is denoted by M4. L1, L2 denotes different layers of VGG9 and M5, M6, etc., corresponding to models from future communication rounds post pre-pass. Observing pre-pass model M4, we find that the layer-wise divergence is low for the initial set of layers and starts to increase around Layer5. D&C can be applied around layer 5 to split the topology for creating feature-training and fine-tuning groups.

To validate the efficacy of W_d, Accuracy and convergence behaviour for VGG9 under different layer division schemes were checked using a brute force sweep across different splits. Accuracy for different division schemes is presented in Figure 6. As discussed in Section 2, Divide5 corresponds to division after layer5. From the figure, Divide5 offers the best accuracy and convergence speed under the given non-IID condition. All runs used 20 epochs for feature-learning and 4 epochs for fine-tuning. Likewise, learning rate for fine-tuning round was half that of feature-learning. Learning Rate Decay was also applied across the communication rounds starting from 0.001 and reducing by 10% for every round.

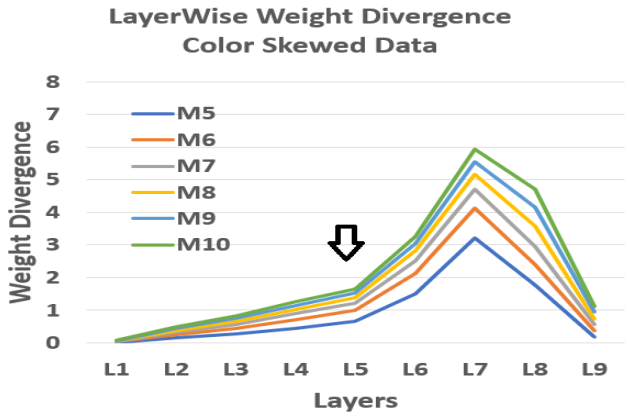

Figure 5. LayerWise Weight Divergence for Color Skewed distribution at different communication rounds. L1, L2 corresponds to layers and M5, M6 corresponds to model at end of successive communication rounds. Divergence is low for initial layers suggesting opportunities for Divide-and-Conquer

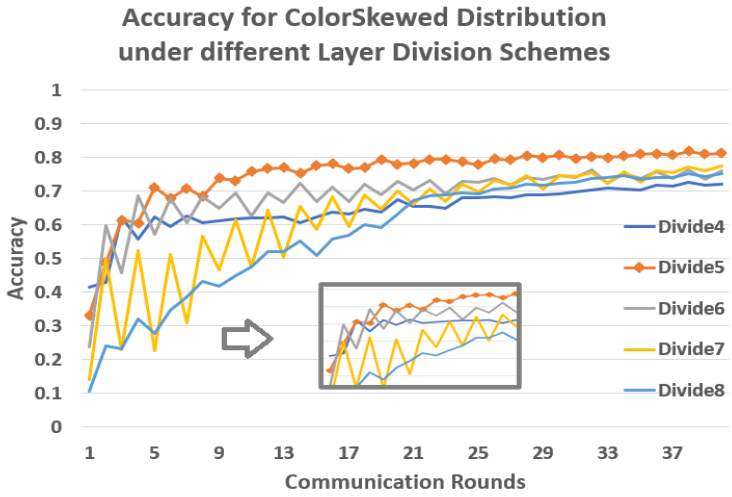

Figure 6. Training Accuracy under different Layer Division Schemes. Divide5 offers optimal results in-line with weight divergence

For certain division schemes (ex: Divide7), large spread is seen in accuracy between feature-learning and fine-tuning rounds suggesting that the layer assignment strategy for the two groups is sub-optimal. For Divide7, Divide6, etc., we find that accuracy is higher for fine-tuning rounds (Communication Round CR=2, 4, 6,…) and drops for feature-learning rounds (CR=1,3,5,…). The trend however reverses for Divide5 where the accuracy is higher for feature-learning rounds and marginally drops for fine-tuning group. The divergence between feature-learning and fine-tuning is also minimal in this split. When fewer layers are present in feature-learning group as in Divide4, we find that the rate of learning starts to fall, and accuracy spread between the two learning groups increases again. This suggests that Divide5 is an optimal split for this topology for this non-IID dataset thereby validating the usage of weight divergence metric to determine the point in a model where the layer split can be performed.

For the Class Imbalanced non-IID condition, the weight-divergence is high across all the layers of the topology Figure 7, suggesting that Divide-and-Conquer might not offer significant accuracy benefits. We chose to divide after layer8, in-line with traditional transfer learning

strategies, where the last layer is used for fine-tuning for realizing bandwidth savings. The Next Character prediction model has 3 layers and we again use the last layer for fine-tuning.

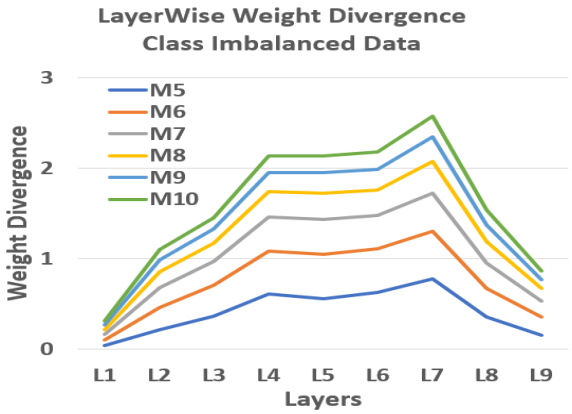

Figure 7. LayerWise Weight Divergence for class-imbalanced distribution at different communication rounds. L1, L2 corresponds to layers and M5, M6 corresponds to model at end of successive communication rounds. Divergence is high for all layers.

In the current work, layer division is determined using a pre-pass run and the division scheme is fixed for the entire duration of training. Future work will extend this to explore a dynamic scheme assignment where layers from a group can be reassigned to other group based on observed trend in feature-learning accuracy vs fine-tuning accuracy over few communication rounds.

5. RESULTS

Results from the Divide-and-Conquer methodology under different non-IID scenario is presented in this section. For training we use 20 epochs for feature-learning and 4 epochs for fine-tuning. Learning rate was initialized to 0.001 and allowed to decay by 10% for every communication round. Learning rate for fine-tuning was 50% of learning-rate for feature-learning round.

Divide-and-Conquer uses half the network bandwidth for data transfers compared to FedAvg, as the full model is transferred for every two communication rounds. For FedMA, results from equivalent matched averaged round is presented based on equivalency established in [13]. Though FedMA uses much lower communication bandwidth, compute overhead for layer matching increases with model depth as well as width, making it less desirable for practical deployments. We show that our methodology yields similar accuracy levels as more complex algorithms like FedMA in acceptable rounds of communication.

Note: In the tables providing the comparison across different approaches, Divide-and-Conquer is captured under D&C.

5.1. Image Classification: Color Skewed Distribution

Training accuracy and convergence profile for different aggregation algorithms using Color Skewed 95-5% CIFAR10 data are shown in Figure 8. For this category of non-IIDness, the model reaches high accuracy with much smaller communication rounds compared to FedAvg. Divide5 was used for this analysis as described in the earlier section along with the same values for E_p and

η. Results for additional levels of Color Skew is presented in Table 1. We chose 18 rounds of communication for the comparison to align with FedMA.

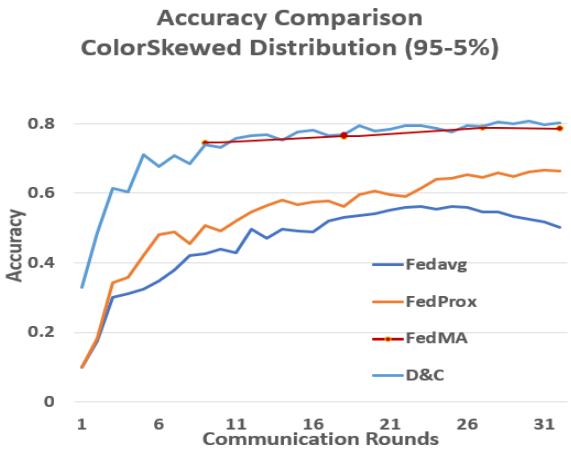

Figure 8. Accuracy Comparison for Color Skewed Distribution with 95-5% skew. Accuracy and Convergence rate for Divide-and-Conquer (using FedAvg) is higher than FedAvg

Table 1. Accuracy for Color Skewed Distribution for 18 communication rounds under different levels of skew for 2 collaborators. D&C (using FedAvg) delivers high classification accuracy under this non-iidness

#Col	SKEW	FedAvg	FedProx	FedMA	D&C
2	95-5%	53.1%	56.2%	81.0%	80.1%
2	75-25%	52.8%	74.6%	78.8%	79.2%
2	50-50%	49.1%	67.2%	79.9%	81.8%

5.2. Image Classification: Class Imbalance Distribution

For Class Imbalanced data, the observed weight divergence from the pre-pass run was high for most layers. This indicates that D&C does not offer opportunities for Layer Splitting for accuracy improvements, though it might still offer opportunity for bandwidth saving. As an experiment, we divided the topology at layer8, similar to traditional transfer learning. D&C yields slightly lower accuracy compared to FedAvg and FedMA for half the bandwidth requirement, as documented in Table 2. However, if bandwidth saving is not sacrificed and D&C is run for additional rounds to get a similar amount of model transfer as FedAvg, the performance of Divide-and-Conquer is marginally better. This is captured in the table under the column D&C. Though FedMA achieves its accuracy levels using much lower communication bandwidth, compute overhead for layer matching increases with model depth as well as width, as discussed earlier, making it less desirable for practical deployments.

Given the results, in cases where weight divergence suggests no clear split layer, it is recommended adopt D&C solely for bandwidth saving. As collaborator count increases in our experimental setup, training data per collaborator decreases, (as same data is divided across the collaborators). This could also lead to increased divergence, when feature-learning is done aggressively on sparse data. In a truly federated set up with a large training corpus across collaborators, we expect our methodology to offer better accuracy improvements.

Table 2. Accuracy for Class-Imbalanced Distribution for 18 communication rounds using 5 & 10 collaborators. Accuracy from D&C (using FedAvg) is in-line with other algorithms at half the bandwidth requirement.

#Col	FedAvg	FedProx	FedMA	D&C	D&C'
5	88.5%	87.5%	87.5%	87.1%	89.3%
10	83.5%	80.0%	82.5%	76.8%	82.2%

An extreme case of Class Imbalance based heterogeneity is when each collaborator exclusively holds data from one unique class. All the tested algorithms performed poorly (accuracy less than 15%) under this scenario, suggesting a need for more research in this area.

5.3. Next-Character Prediction: Speaker-Role based non-IID Distribution

Results from application of D&C to a character prediction model is shown in Table 3. At end of 9 communication rounds, the accuracy from D&C is comparable to other algorithms while only requiring half the amount of data transfer as FedAvg. 9 rounds of communication were chosen to align with FedMA.

Table 3. Accuracy for next-character prediction using lstm-model for 9 communication rounds using 66 collaborators. Accuracy from D&C (using FedAvg) is in-line with other algorithms at half the bandwidth requirement.

#Col	FedAvg	FedProx	FedMA	D&C
66	50.8%	44.6%	47.4%	49.6%

6. CONCLUSIONS

In this work, we presented a weight-divergence based, Divide-and-Conquer algorithm which builds on popular FedAvg algorithm to achieve state-of-the-art accuracy under non-IIDness. By training network in parts, our novel methodology is shown to a) Achieve faster convergence when low-level features are well-represented b) Reduce communication by half, because of training and weight exchange in parts, and c) Require less compute compared to state-of-the-art techniques like FedMA, which has performance overheads from weight matching. A static topology splitting strategy is adapted in this work, where the topology is divided at the beginning of training using a pre-pass run. Future work can explore a dynamic Divide-and-Conquer strategy where layers are moved between feature-learning and fine-tuning groups based on accuracy observations during training. Future work can also explore the application of Divide-and-Conquer methodology to learning paradigms like Few Shot Learning to identify Class Agnostic layers for the backbone network.

REFERENCES

[1] McMahan, H.B., Moore, E., Ramage, D., Hampson, S., & Arcas, B.A. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. AISTATS.

[2] Li, T., Sahu, A.K., Talwalkar, A.S., & Smith, V. (2020). Federated Learning: Challenges, Methods, and Future Directions. IEEE Signal Processing Magazine, 37, 50-60.

[3] Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konecný, J., Mazzocchi, S., McMahan, H.B., Overveldt, T.V., Petrou, D., Ramage, D., & Roselander, J. (2019). Towards Federated Learning at Scale: System Design. ArXiv, abs/1902.01046.

[4] Xu, J., & Wang, F. (2021). Federated Learning for Healthcare Informatics. Journal of Healthcare Informatics Research, 1 - 19.

[5] Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., & Chandra, V. (2018). Federated Learning with Non-IID Data. ArXiv, abs/1806.00582.
[6] Lin, T., Kong, L., Stich, S.U., & Jaggi, M. (2020). Ensemble Distillation for Robust Model Fusion in Federated Learning. ArXiv, abs/2006.07242.
[7] Fallah, A., Mokhtari, A., & Ozdaglar, A. (2020). Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach. NeurIPS.
[8] Ghosh, A., Chung, J., Yin, D., & Ramchandran, K. (2020). An Efficient Framework for Clustered Federated Learning. ArXiv, abs/2006.04088.
[9] Hanzely, F., & Richtárik, P. (2020). Federated Learning of a Mixture of Global and Local Models. ArXiv, abs/2002.05516.
[10] Dinh, C.T., Tran, N.H., & Nguyen, T.D. (2020). Personalized Federated Learning with Moreau Envelopes. ArXiv, abs/2006.08848.
[11] Hanzely, F., Hanzely, S., Horvath, S., & Richtárik, P. (2020). Lower Bounds and Optimal Algorithms for Personalized Federated Learning. ArXiv, abs/2010.02372.
[12] Sahu, A.K., Li, T., Sanjabi, M., Zaheer, M., Talwalkar, A.S., & Smith, V. (2020). Federated Optimization in Heterogeneous Networks. arXiv: Learning.
[13] Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., & Khazaeni, Y. (2020). Federated Learning with Matched Averaging. ArXiv, abs/2002.06440.
[14] Chen, W., Bhardwaj, K., & Marculescu, R. (2020). FedMAX: Mitigating Activation Divergence for Accurate and Communication-Efficient Federated Learning. ECML/PKDD
[15] Shoham, N., Avidor, T., Keren, A., Israel, N., Benditkis, D., Mor-Yosef, L., & Zeitak, I. (2019). Overcoming Forgetting in Federated Learning on Non-IID Data. ArXiv, abs/1910.07796.
[16] Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR, abs/1409.1556.
[17] Yan, X., Chen, Z., Xu, A., Wang, X., Liang, X., & Lin, L. (2019). Meta R-CNN: Towards General Solver for Instance-Level Low-Shot Learning. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 9576-9585.
[18] Caldas, S., Wu, P., Li, T., Konecný, J., McMahan, H.B., Smith, V., & Talwalkar, A.S. (2018). LEAF: A Benchmark for Federated Settings. ArXiv, abs/1812.01097.
[19] Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images.

AUTHORS

Raghavendra Bhat (Raghu) is a Principal Engineer at Intel Technology India Pvt. Ltd. He has a BTech in Computer Science from NIT Warangal and PGDBA (Finance) from Symbiosis Pune. He has over 22 years of industry experience spread across domains like Network Management, Embedded platform development for Mobile, IoT and Biometrics solutions. In his current role at Intel, he leads exploration in Healthcare space as part of Vertical Solutions and Services Group. In his role, the primary focus is on design, development, and ecosystem adoption of optimized AI solutions on Intel AI portfolio for Speech, Language, Image and Video analytics use cases. Currently, a significant focus is on AI Training at the Edge where training methodologies like Federated Learning, Incremental Learning etc. He is on the ISO and BIS standardization panels of Blockchain and AI. His interests extend to SW architecture methodologies, Blockchain and Aadhaar ecosystem where he has contributed towards Iris biometric device development and specifying device security requirements. As part of IEEE standards organization, he is leading the pre-standardization study for low resourced Indian languages. He has several patents and publications to his credit.

Pravin Chandran works as Deep Learning R&D Engineer at Intel Technology India Pvt, Ltd. He holds a M.S in EE from Clemson University, USA and B.E from University of Madras, India. He has 13 years of professional experience in wide range of areas including ML/DL, Software Development, Statistical Design Analysis, Yield Estimation, VLSI EDA Methodology and SoC Design.

Avinash Chakravarthi (Avi) works as Deep learning scientist at Intel, he joined Intel as graduate intern after completing his bachelors from VIT University in Electronics, 2016. His interests and area of work include Federated learning, Bio inspired computing and Software development.

Srikanth Chandar graduated from PES University, and currently works an AI Engineer at Intel Corporation. Anything ML excites him, and he likes taking up a challenge that could shape a new paradigm in the same space, be it application-based or research. Voice cloning using GANs, and Communication optimization in FL are his recent research pursuits. Another thing that interests him other than talking about himself in the third person is Animal Welfare. He feels very strongly about animal abuse and runs an NGO (Dystopia-Animal Welfare) to fight the same through campaigns, volunteering activities, and also technical projects.