COMMUNICATION EFFICIENT DECENTRALIZED COLLABORATIVE LEARNING OF HETEROGENEOUS DEEP LEARNING MODELS USING DISTILLATION AND INCENTIVE-BASED TECHNIQUES

ZAHID IQBAL

UNIVERSITI SAINS MALAYSIA

2023

COMMUNICATION EFFICIENT DECENTRALIZED COLLABORATIVE LEARNING OF HETEROGENEOUS DEEP LEARNING MODELS USING DISTILLATION AND INCENTIVE-BASED TECHNIQUES

by

ZAHID IQBAL

Thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

July 2023

ACKNOWLEDGEMENT

First, I would like to dedicate it to my mentor, my guru, syed-us-sadat, zeenat-e-sadat, fakhar-e-panjatan pak, mehboob-e-rabbani Hafiz Syed Muhammad Server Shah (SAW), my mother Umer Kalsoom Begum, my father Mirza Ghulam Nabi, my wife Faiza Zahid and to my sweet kids Muhammad Hasnaat, Muhammad Husnain and Aiza Batool.

I would like to express my deepest sincere gratitude to my supervisor, associate professor Dr Chan Huah Yong for providing me his support, guidance and motivation throughout this thesis.

Many thanks to all my colleagues and the staff in the School of Computer Sciences for supporting me during the whole period of my study. Specifically, Ms. Sheela, you are very cooperative and hardworking. May God bless you.

Finally, I would like to express my most sincere and deepest gratitude to my family, especially my grandfather, my father, my mother, my brothers and sisters, my wife and my kids. No doubt, without their love and support, it was not easy to complete this thesis. Thanks to all my relatives and friends in Pakistan and in Malaysia for their prayers and encouragement throughout my study.

zahid

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS

FL	Federate Learning
IID	Independent and Identical Distribution
DL	Decentralized Learning
SGD	Stochastic Gradient Descent
GKT	Group Knowledge Transfer
DL-SH	Decentralized Learning with Statistical Heterogeneity
DL-MH	Decentralized Learning with Model Heterogeneity

LIST OF APPENDICES

PEMBELAJARAN KOLABORATIF TIDAK BERPUSAT CEKAP KOMUNIKASI DARIPADA MODEL HETEROGEN PEMBELAJARAN MENDALAM MENGGUNAKAN TEKNIK PENYULINGAN DAN BERASASKAN INSENTIF

ABSTRAK

Peranti pintar, secara kolektif, mempunyai data masa nyata dan sangat berharga yang boleh digunakan untuk melatih model pembelajaran mendalam yang sangat cekap untuk aplikasi kecerdasan buatan. Walau bagaimanapun, disebabkan sifat sensitif data ini, penyelidik lebih mengambil berat tentang privasi data mereka dan tidak rela untuk berkongsinya. Oleh itu, terdapat keperluan untuk belajar daripada data berharga ini secara tak terpusat dengan penahanan data yang disetempatkan pada peranti yang digunakan ini dan melaksanakan pengiraan yang diperlukan secara cekap pada peranti ini dengan mengeksploitasi sumber perkomputan mereka. Keheterogenan statistik dan keheterogenan model sepenuhnya adalah antara cabaran utama dalam menggunakan pendekatan Pembelajaran Tidak Terpusat (PTT) dalam senario sebenar. Lazimnya, semua teknik PTT sedia ada mengandaikan bahawa semua peranti akan mempunyai seni bina model homogen. Walau bagaimanapun, dalam aplikasi sebenar PTT, disebabkan sumber perkomputan yang berbeza dan keperluan perniagaan yang berbeza bagi peranti, adalah intuitif bahawa mereka mungkin mempunyai seni bina model yang berbeza sama sekali. Kajian yang dilakukan untuk menangani masalah heterogeniti model sepenuhnya adalah sangat terhad. Dengan cara yang sama, beberapa kajian telah dilakukan untuk menangani keheterogenan statistik namun kebanyakannya sukar untuk digunakan dalam senario sebenar atau hanya untuk kes guna yang terhad. Sumbangan kajian ini adalah kepada dua pendekatan utama; yang pertama ialah memelihara privasi, kecekapan komunikasi dan pendekatan PTT yang teguh, DL-SH, untuk menangani keheterogenan statistik dengan cekap dan yang kedua ialah DL-MH untuk menangani cabaran model penuh heterogeniti dengan cekap di samping melengkapi keheterogenan statistik. di samping itu, kerja ini juga menambah baik pendekatan kedua dan mencadangkan DL-MH (I-DL-MH) berasaskan insentif pelanggan yang berkesan dan kos efektif untuk membolehkan pelanggan mendapat beberapa insentif terhadap latihan Federated Learning (FL). Eksperimen yang meluas telah dilakukan untuk menilai prestasi pendekatan yang dicadangkan menggunakan seni bina model yang berbeza ke atas taburan data yang pelbagai dan pada beberapa set data. Analisis empirikal dengan pelbagai keadaan seni pendekatan menunjukkan hasil yang menguntungkan untuk pendekatan yang dicadangkan. DL-SH memberikan kira-kira 153% peningkatan prestasi kepada model global berbanding dengan standard FL dan sekitar 150% peningkatan berbanding pendekatan Out of Distribution Detector in Neural Network (ODIN) di bawah taburan data paling kompleks (bukan IID) dengan FMNIST menggunakan ResNet18. Begitu juga, I-DL-MH memberikan sekitar 225% peningkatan prestasi dengan hanya satu pusingan penyulingan daripada model global di bawah taburan data bukan IID dengan Cifar10 menggunakan ResNet18.

COMMUNICATION EFFICIENT DECENTRALIZED COLLABORATIVE LEARNING OF HETEROGENEOUS DEEP LEARNING MODELS USING DISTILLATION AND INCENTIVE-BASED TECHNIQUES

ABSTRACT

Smart devices, collectively, have very valuable and real-time data which can be used to train very efficient deep learning models for AI applications. However, due to the sensitive nature of this data, people are more concerned about the privacy of their data and not willing to share it. Therefore, there is a need to learn from this valuable data in a decentralized fashion by withholding data localized on these intended devices and efficiently performing necessary computation on these devices by exploiting their computational resources. Statistical heterogeneity and fully model heterogeneity are among the key challenges in applying the Decentralized Learning (DL) approaches in real scenarios. Typically, all existing DL techniques assume that all devices would have homogeneous model architecture. However, in real applications of DL, due to different computational resources and distinct business needs of devices, it is intuitive that they may have completely different model architectures. Very limited work has been performed to address fully model heterogeneity problem. In the same way, some work has been performed to address the statistical heterogeneity however mostly is hard to apply in real scenarios or is only for limited use cases. The contribution of this work is mainly two approaches; the first is privacy-preserving, communication efficient and robust DL approach, DL-SH, to efficiently address the statistical heterogeneity and the second is DL-MH to efficiently handle the full model heterogeneity challenge while satisfying the statistical

heterogeneity. Additionally, this works also extends the second approach and proposes an effective and cost-effective, client incentive-based DL-MH (I-DL-MH) to enable the clients to get some incentives against FL training. Extensive experiments were performed to evaluate the performance of proposed approaches using different model architectures on various data distributions and on multiple datasets. Empirical analysis with various SOTA approaches shows auspicious results for proposed approaches. DL-SH gives around 153% performance gain to the global model as compared to standard FL and around 150% improvement as compared to the ODIN approach under the most complex data distribution (non-IID) with FMNIST using ResNet18. Similarly, I-DL-MH gives around 225% performance improvement with only one round distillation from the global model under non-IID data distribution with Cifar10 using ResNet18.

CHAPTER 1

INTRODUCTION

1.1 Overview

Recently, deep learning has achieved a very high peak of acceptance in the artificial intelligence and machine learning research community due to its ability to automatically extract and learn high-level complex features by the composition of low-level features. One of the most prominent features of deep learning which typically differs from traditional machine learning is its remarkable ability to extract and sufficiently learn these complex features automatically i.e. without the specific need for hard-coded rules or domain expert knowledge or more intermediate steps to solve a problem. Deep learning has already outperformed the numerous traditional approaches in many fields including image recognition, face detection, speech recognition, fraud detection and human action recognition (Simard, Steinkraus and Platt, 2003; Hinton *et al.*, 2012; Graves, Mohamed and Hinton, 2013; Hannun *et al.*, 2014; Taigman *et al.*, 2014; Krizhevsky, Sutskever and Hinton, 2017; Sezavar, Farsi and Mohamadzadeh, 2019; Gheitasi, Farsi and Mohamadzadeh, 2020; Salehi and Pouyan, 2020; Savadi Hosseini and Ghaderi, 2020; Shaeiri and Kazemitabar, 2020).

The optimal performance of deep learning models greatly depends upon the availability of a significantly enormous amount of valuable data and the availability of high computational resources. So, to get a robust and efficient deep learning model, we typically need a large amount of valuable data and ample computational resources.

As smart devices (including smart mobile phones, smart tablets and smart wearable devices) are being empowered with high computational resources with large memory storage and incredibly powerful sensors, people are rapidly switching from laptops and conventional desktop computers to these smart devices as primary computing sources (Pew Research Center, 2016). Particularly, the invention of AI-based smart chips (Neuromation, 2018) being incorporated in almost all the latest smartphones, has more significantly boosted this trend where companies' goal is to add neural network power to smart devices. These devices are generating very large and valuable data including their location history, pictures, typing patterns, medical history, life logging data, etc. So, there is a lot of valuable real-world data, however in a decentralized fashion, which can be used to train deep learning models to get more accurate and intelligent applications.

Though these devices, collectively, possess a large amount of data, however, usually, the nature of this data is highly sensitive, and people are not willing to compromise their privacy by sharing their personal data for model training. Furthermore, people have expressed more serious concerns about the privacy of their data, especially, after the recent breach of Facebook and other top companies' data (Hill and Swinhoe, 2021). Taking this into account, recently, European Union and many other countries have enforced different data protection laws (KPMG, 2017; GDPR, 2018; Act, 2020) and have made it almost impossible for companies to collect, transfer, use or integrate user's data without their consent for any purpose.

Traditionally in the distributed learning environment, to train a model, we typically collect all data at a central location and properly distribute it to separate parties for processing however, now, due to more privacy concerns of people and extremely strict privacy laws, it's almost impossible to collect updated real-time users' private data. However, on the other hand, such smart devices and many commercial organizations like Banks, Hospitals, etc. have significantly valuable user data that could be leveraged to build more accurate and state of art applications by training deep learning models on this valuable users' data. Therefore, such factors inspired researchers to tend to (pure) decentralized learning ensuring the privacy of data.

In this scenario, it is intuitive to keep the private data of users stored locally and perform the necessary computation (model training) on these devices. Thus, ensuring the privacy guarantee of users' personal data and, on other hand, also utilizing the computational resources of client devices. Therefore, in this way, different devices will collaborate with each other to train a more efficient deep learning model(s). Different collaborative learning techniques (Shokri and Shmatikov, 2015; Konečný *et al.*, 2016; Brendan McMahan *et al.*, 2017; Jeong *et al.*, 2018; Corinzia and Buhmann, 2019) have been proposed to train deep learning models where different clients collaborate with each other to update their models by leveraging the learned knowledge of other clients rather than their private data. Specifically, recently, Google has coined a very promising decentralized learning technique called Federated learning (Konečný *et al.*, 2016; Brendan McMahan *et al.*, 2017) which has instantly attracted a large research community in Machine learning towards this research direction i.e., rather transferring data to code (computing), we move the code to data(model).

In federated learning, we typically assume some clients want to collaborate to train a global model for some specific tasks. All devices collaborate with each other through a centralized server (aggregation server). In the first place, the centralized server forwards the copy of the global model to all participants (active devices), these devices train their copy of the global model on their private data and send this updated model back to the server. After receiving all model updates from clients, the centralized server performs weighted aggregation on these local models (parameters/gradients) to update the primary global model. Subsequently, the centralized server again sends this updated model to all active clients, so active clients again train this model, and this process continues until the global model is converged. This thesis has discussed the fundamental federated learning concept thoroughly in section 2.3.

FL has many advantages as compared to traditional distributed machine learning approaches (Dean *et al.*, 2012; Shamir, Srebro and Zhang, 2014; Konečný *et al.*, 2016; Reddi *et al.*, 2016) like privacy, where devices do not need to share their private data with other devices including a centralized server; Low latency, as devices would have updated model locally, so they do not need to wait for inferencing from cloud-server. Huge computational resources, as usually hundreds of devices, could participate in FL so a lot of computational resources would be available to train the model. Similarly, FL can help to utilize the network bandwidth more efficiently as there is no need to transmit raw data to the cloud server rather just need to share the trained model parameters.

FL is a comparatively new research domain and recently, researchers have categorized FL into three main subcategories based on the distribution of data named 1) Horizontal FL where devices may have different samples of the dataset with the same feature space 2) Vertical FL where devices may have different samples but with different features, and 3) Federated transfer learning where devices may have a different sample with different features (Yang *et al.*, 2019). These subcategories are further discussed in chapter 2. This thesis focuses mainly on horizontal federated learning where usually small devices with less computational resources, but valuable data collaborate to train a single global model.

Though federated learning has been emerging as a very efficient decentralized learning framework to leverage the massively distributed, highly unbalanced, and non-independent and identically distributed (non-IID) private data of smart devices, however, it comes with many (unique) challenges related to data, model architecture, communication and privacy. Like, here, data is typically expected to be massively distributed, non-IID, unbalanced and inaccessible by other devices or centralized servers due to privacy constraints. Communication cost could be much higher as compared to computation cost and could experience challenges of limited and inconsistent bandwidth for different devices and of passive sampling. Participating devices may naturally require specialized or more personalized models based on their specific requirements. Similarly, Privacy is one of the primary focus of decentralized learning so local data of devices would not be accessible to any other party. Though the primary focus of this thesis is on model heterogeneity and statistical heterogeneity (non-IID) while reducing the communication and computation overhead however comprehensive overview of core challenges of federated learning along with recent research work has been covered in section 2.3.5.

The primary goal of federated learning is to train a single global model and it works with the assumption that all clients would have reasonable computational resources and all participating devices would be able to train the same complex model architecture. However, in the real scenario, this assumption is not logical as more complex deep learning models are being developed to get more accurate performance on real word complex tasks and, due to different constraints, it is not possible for all devices having valuable data to be capable to train the same complex model. More specifically, In model heterogeneity cases, due to different computational resources and different business needs (tradeoff between speed and accuracy), it is intuitive that devices may have different sizes of deep networks (different no. of layers) or may have entirely different network architectures like some devices may be using CNN, some device may be using ResNet, and some devices may be using Inception. This model heterogeneity problem becomes more challenging if we further relax the assumption that, even, heterogeneous clients should have the same number of target nodes in the output layer. Actually, this scenario becomes more intuitive in real applications of federated learning where clients would have non-IID data. So, all devices should have personalized models based on their categories of data. For instance, in healthcare applications, if a device can measure samples of only two diseases, then its model should have only two output nodes to classify those two diseases and if a device has more disease samples, then its' model should have more output nodes. In the existing literature, it is assumed that all collaborating models should have the same output layer which results in unnecessary computation and communication overhead.

Recently some research works have been performed to partially address the model heterogeneity however most of them work with some strong assumptions or are not feasible to be applied in practical FL scenarios. More specifically, these approaches are difficult to employ in our scenario, where entirely different model architectures, having different output layers based on available data, can also collaborate in FL settings. Some researchers (Wang, Kolar and Srebro, 2016; Smith *et al.*, 2017; Corinzia and Buhmann, 2019; Sattler, Muller and Samek, 2021) have leveraged distributed Multitask learning to address the model heterogeneity. However, some approaches can only be applied to convex problems (Smith *et al.*, 2017), or are very costly to scalable in large FL scenarios (Corinzia and Buhmann, 2019). Similarly, to make a global model personalized, most personalization techniques suggest retraining the (collaboratively trained) global model on the users' local private data (Sim, Zadrazil and Beaufays, 2019). Some researchers have proposed model personalization approaches using transfer learning (K. Wang *et al.*, 2019; Schneider and Vlachos, 2019; Mansour *et al.*, 2020). In transfer learning, usually, the last layers of a trained model are replaced with new layers to leverage the learned knowledge of the trained model on some new tasks. some researchers (Finn, Abbeel and Levine, 2017; Jiang *et al.*, 2019; Khodak, Balcan and Talwalkar, 2019; Fallah, Mokhtari and Ozdaglar, 2020) have also leveraged meta-learning to solve the personalization problem. Meta-learning is generally defined as "Learning to Learn" where a model is made adaptive by training it on multiple tasks in such a way that it can learn new tasks by providing very few examples of new tasks.

Recently some researchers (Jeong *et al.*, 2018; He, Annavaram and Avestimehr, 2020; Ma, Yonetani and Iqbal, 2021) have employed a communication-efficient and very effective approach, called distillation (Hinton, Vinyals and Dean, 2015; Anil *et al.*, 2018) to partially leverage the heterogeneous models in FL settings. Distillation has been proven to be a very efficient approach to effectively transfer knowledge among independent models, more specifically, by efficiently distilling the knowledge from a trained model to an untrained model. As FL, also, typically require all the models to transfer their knowledge (trained model) to the centralized server for aggregation so here distillation seems to be a potential approach which could be leveraged in FL settings to also address the model heterogeneity challenges. However, distillation also has its specific limitations in that it only works on IID data distribution whilst in FL, data distribution is naturally expected as non-IID so it could be challenging to properly apply distillation in the FL setting. Like, recently (Jeong *et al.*, 2018) proposed a Federated distillation algorithm which could be employed as a variation scheme of FL to learn in a decentralized environment. Though the author just adopted this approach to demonstrate that codistillation (a variant of distillation) is a more communication-efficient technique as compared to standard FL, however, the author applies the codistillation approach after making all data distributions as IID using Generative Adversarial Network (GAN) approach. Distillation and Codistillation have been explained thoroughly in section 2.4 and section 2.5 respectively.

1.2 Problem Statement

In decentralized settings where we assume that there is no centralized server to properly manage the distribution of data so it is very likely that clients would have highly unbalanced and non-IID data as each device or user may have their specific preferences. For instance, in healthcare applications, some devices might have samples of only two diseases whilst others may have samples of 10 diseases. So, all devices would have a different number of samples (unbalance) and samples of different diseases (non-IID). Many techniques have been proposed to sufficiently address the statistical challenges however most of them work with some unrealistic assumptions or are difficult to implement in a real FL scenario. Like the initial work in FL, FedAvg (Brendan McMahan *et al.*, 2017); a state of art algorithm based on SGD, shows that it can handle a certain amount of non-IID data however (Zhao *et al.*, 2018) show empirically that for high skewed non-IID data, the performance of the convolutional neural network, trained using FedAvg can drop reasonably by 51 % on CIFAR10.

Some researchers (Smith *et al.*, 2017; Corinzia and Buhmann, 2019; Sattler, Muller and Samek, 2021) leveraged Multitask Learning to efficiently handle the non-IID distribution by personalizing each task (device model) on their private data. (Jeong *et al.*, 2018; Zhao *et al.*, 2018) tackle the statistical heterogeneity by properly distributing some public or private data among local clients. Although it could be reasonable solutions to address the statistical challenge, however, it could be difficult to carefully arrange labelled public data or to convince local clients to breach their privacy by distributing their private data. The more feasible and realistic approach could be if unlabelled data could be properly utilized to efficiently handle the statistical challenge as getting some unlabelled data is much more convenient rather than collecting correctly annotated labelled public data or sharing private data.

Likewise, the model heterogeneity (having different model architectures) brings some new challenges like naïve federated learning can't be applied directly to this setting when clients have a different number of parameters (simple averaging of parameters or gradients is not possible) so tackling this scenario, there is a need to investigate some other collaborative learning techniques which might be leveraged in this specific real-world scenario of decentralized learning.

To address the model heterogeneity challenge, some researchers (Smith *et al.*, 2017) have leveraged distributed Multitask learning to handle the model heterogeneity challenge however in a significantly limited way as their work only focuses on convex problems and could not be applied to deep learning (non-convex) problems. In a similar fashion (Corinzia and Buhmann, 2019) has used the Bayesian network with multitask learning for model heterogeneity which is applicable to non-convex problems however is very costly to be scalable in large, federated learning scenario as they refine the models sequentially. One more recent work (Sattler, Muller and Samek, 2021) proposed a secondary method to improve the performance of client models. It uses the clustering approach with multitask learning to further improve the performance of models by personalizing the separate models of different devices once the federated learning has been performed. Therefore, significantly limited work has been performed to properly address the model heterogeneity challenges as mentioned earlier.

Furthermore, in FL, when a centralized server performs aggregation on the client's update then the server assigns some weights to these updates before aggregation. These weights are assigned based on the number of training samples on which a model was trained. However, it might not be an efficient way to assign weightage like a device might have more samples, however, its model is not well trained (due to diff. factors as discussed later). Similarly, a model trained on a few training examples might be more efficient. Therefore, there is a need to devise a more efficient way to assign weightage to different models' output. For instance, it could be a more promising solution if weights are based on the client model's learning so if a client is more confident about its output, then the server may give it more weightage.

Typically, the main objective of federated learning is to train the global model where all participating devices collaborate to share their knowledge with the global model. In federated learning settings, typically, it is assumed that clients would be voluntarily convinced to participate in federated learning training however, in practical scenarios, it could be very difficult to convince the clients to allow someone to consume their computational and communicational resources along with valuable data with no incentives. In pioneer work (Konečný *et al.*, 2016; Brendan McMahan *et al.*, 2017) of federated learning by Google, they use the complete model sharing approach where in each round, participating clients can also get the updated trained model from the server (as incentives) so they can also use that updated model on their devices. Moreover, Google has access to (android) operating systems (OS) of millions of devices, so they can return the incentives in the form of OS updates. However, for small third-party developers/organizations, it might not be possible to give incentives to participating clients in the same manner. Particularly, under fully model heterogeneous settings, this could be more difficult where all clients may have entirely different model architectures including different targets. Thus, the client cannot distil the knowledge from the global model straightforwardly. Thus, cost-efficient incentive schemes are required to motivate the client devices to participate in federated learning training.

Therefore, there is a need to propose a more robust and communication-efficient decentralized learning framework by appropriately addressing their specific limitations to learn from fully heterogeneous models (having different model architectures with different target nodes) while reasonably satisfying the statistical heterogeneity.

1.3 Research Questions

- How to efficiently handle statistical heterogeneity with unlabeled public data in FL settings.
- How fully heterogeneous models (having different architectures along with different target labels) could be leveraged in the current federated learning framework.
- How to encourage clients to participate in current federated learning training

1.4 Research Objectives

- To propose an efficient decentralized learning approach to address the statistical heterogeneity using unlabeled data.
- To propose a model agnostic approach to enable the fully heterogeneous model architectures to efficiently perform decentralized learning whilst satisfying the statistical heterogeneity.

- To propose a cost-effective approach to encourage clients to participate in federated learning settings.

1.5 Key contributions

The key contributions of this thesis are;

1. A confidence matrix is computed for each client by training a binary classifier on unlabeled data to address the statistical heterogeneity problem. It gives around 153% performance gain to the global model as compared to standard FL under the most complex data distribution (non-IID) with FMNIST using ResNet18.
2. Cost-effective mapping and masking schemes are applied to clients' outputs to enable fully heterogeneous model architectures to participate in federated learning training under non-IID settings. It reduces the communication cost by around 99% as compared to standard FL under the most complex data distribution (non-IID) with CIFAR100 using ResNet18.
3. Very appealing incentives are provided to clients in the form of updated knowledge from the global model with negligible additional communication costs. Client models get almost similar performance to the global model by following this incentive approach.

1.6 Scope of Thesis

Figure 1.1 illustrates the scope of this research work. It shows the target areas of this research work. For instance, it addresses the challenges in horizontal FL under cross-device FL settings. Similarly, it selects partially and fully heterogeneous models with various data distribution settings including IID, non-IID and hybrid IID data distributions. Furthermore, it considers various image classification models[1] for the evaluation of proposed techniques on benchmark datasets. it considers two very popular deep learning models (ResNet (He *et al.*, 2016), and DenseNet (Huang *et al.*, 2017)) as deep and shallow model architecture respectively to check the impact of our proposed approach under different model architecture settings.

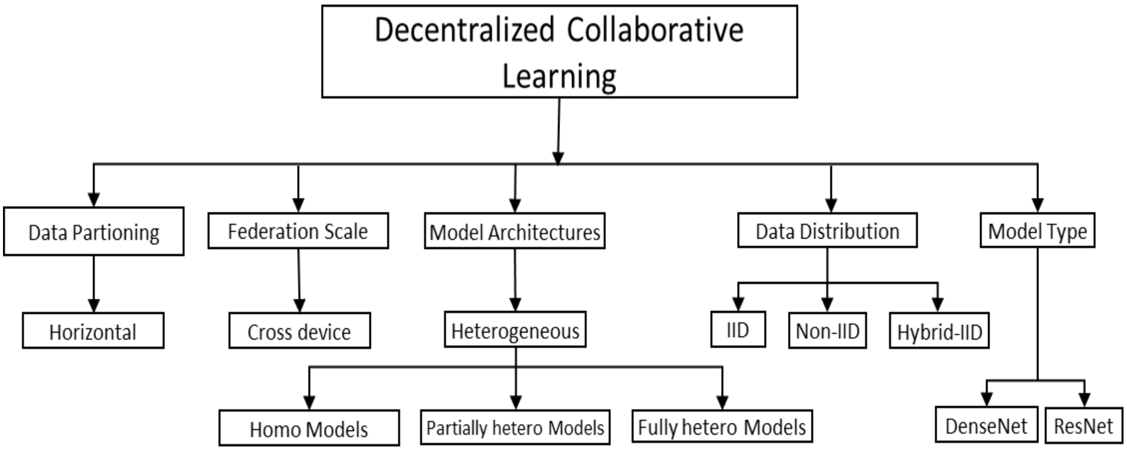

Figure 1.1 Scope of thesis

1.7 Thesis Organization

This thesis has been divided into multiple chapters for a better understanding of the viewers.

Chapter 2 presents the basic concepts of various model learning approaches including data parallelism and model parallelism. It further discussed the overview of decentralized/federated learning including its definition, types, applications, and key challenges of FL. It extensively presents the critical overview of existing approaches to address these challenges following the overview of knowledge-sharing approaches like distillation and codistillation.

[1] Though, here only two image classification model architectures are selected (as they are more commonly used in literature), however this work is not specific to just these algorithms rather it is almost architecture independent and can be extended to other deep learning model architectures.

Chapter 3 first presents an overview of the proposed research methodology followed in this thesis. This clearly illustrates the general steps followed to achieve the research objectives stated in section 1.4 along with the relationship between research questions and research objectives. Subsequent sections comprehensively explain the proposed approaches including their technical details. Finally, it explains the implementation details and evaluation metrics used to implement and evaluate the proposed approaches in this thesis. It explains in detail all the datasets, data distributions, model architectures and benchmarks used in this thesis.

Chapter 4 presents the empirical analysis and results of the proposed approaches DL-SH and DL-MH. Moreover, it also presents the empirical analysis of the I-DL-MH approach.

Chapter 5 presents the conclusion and future directions related to this research work.

CHAPTER 2

LITERATURE REVIEW

This chapter provides an in-depth review of basic concepts, essential knowledge and recent works related to this research work. This chapter first illustrates a brief overview of different categories of model learning approaches. Subsequently, an overview of centralized learning approaches including data parallelism and model parallelism is presented in section 2.1. An overview of different decentralized learning approaches is discussed in section 2.2. Section 2.3 explains federated learning, and its types followed by its core challenges. The concept of distillation is explained in section 2.4 while Codistillation approaches including distillation are illustrated in section 2.5. Finally, section 2.6 present a comprehensive summary of this chapter.

Traditional machine learning models, usually, have few parameters and aren't data hungry such that even a small but reasonable amount of input samples are sufficient to learn a model. This traditional machine learning requires human intervention for hand-crafted features. This works well for such applications where relevant features can be constructed by domain experts easily. However, traditional machine learning doesn't perform well for complex machine learning problems where it is required to extract complex features from unstructured data like images, video, speech, etc.

Deep learning comes into action to properly address these limitations by automatically extracting and learning these complex features from raw data. Deep learning has been enjoying unprecedented success in many fields including image recognition, face detection, speech recognition, fraud detection and human action recognition (Simard, Steinkraus and Platt, 2003; Hinton *et al.*, 2012; Graves, Mohamed and Hinton, 2013; Hannun *et al.*, 2014; Taigman *et al.*, 2014; Krizhevsky, Sutskever and Hinton, 2017; Sezavar, Farsi and Mohamadzadeh, 2019; Gheitasi, Farsi and

Mohamadzadeh, 2020; Salehi and Pouyan, 2020; Savadi Hosseini and Ghaderi, 2020; Shaeiri and Kazemitabar, 2020). However, to learn these complex features and to achieve superior accuracy, deep learning requires very deep and complex models (containing a significantly ample number of parameters) and an enormous amount of training data. It means that a lot of computational resources are required to train these very deep and complex models and to keep a huge amount of data.

Standalone machines having some GPUs might be able to train some small deep models with a limited amount of data, however, it limits the size of the model and data on a single machine. These limitations motivated the researchers to develop the distributed system to train very large deep and complex networks (Dean *et al.*, 2012; Chilimbi *et al.*, 2014). Where multiple machines collaborate to train a more efficient and accurate model by sharing their resources.

We can further classify distributed learning (aka collaborative learning) into two key categories; Centralized learning and Decentralized learning based on the distribution of data and model training. More Precisely, the primary difference between Centralized and Decentralized learning (DL) approaches is the existence of a centralized server to properly manage the distribution of data or training. Though, decentralized learning approaches, usually, also include a centralized server however that is just used for parameter averaging and assert no control over the distribution of data.

2.1 Centralized distributed learning approaches

Centralized distributed learning is a kind of client-server architecture where all devices are linked to a centralized computation node (server). This centralized server takes the responsibility to properly distribute the model (training) or data among connected devices in a balanced and IID manner. These devices typically perform the necessary computation on input data/model and then return it to the server. Then server, further, processes these computed results as per specific requirements/objectives.

Centralized distributed learning could be further categorized, mainly, in two subcategories called data parallelism (for very huge data) and model parallelism (for very deep models).

2.1.1 Data Parallelism

Data parallelism becomes effective, usually, in a scenario when there is an enormous amount of data that can't be processed on a single machine due to limited memory resources or it could take a very long time to process it on a single machine. In data parallelism, as illustrated in Figure 2.1, a centralized server splits the training data in a balanced way and properly distributes it to multiple computing nodes. Where each worker performs the training on its sub-dataset and calculates the updates (model parameters). After performing training on their sub-datasets, all clients send these updates to the centralized computing node (server) which synchronized these updates until the objective is complete.

(Dean *et al.*, 2012) proposed a very large-scale distributed framework called DistBelief to train a very large deep network (with 1 billion connections) using clusters of thousands of machines to get very high accuracy on the ImageNet 22K category classification task. They proposed Downpour SGD (asynchronous SGD) and Sandblaster to increase the training speed of deep networks. Firstly, they properly distribute the model parameters on multiple servers, called parametric servers. Furthermore, there are many workers, linked to these parametric servers, to independently process a minibatch of data. These workers download the updated parameters of the model from these parametric servers, efficiently compute gradients of the loss to these parameters after training on their minibatch of data, and finally, send these updated gradients to parametric servers which further updates the model by aggregating these updates asynchronously. Though this method scales well up to hundreds of workers however faces the problem of stalled updates due to its asynchronous nature. Furthermore, in this method, parameters may often diverge because these parameter servers don't communicate with each other.

Similarly, DANE (Shamir, Srebro and Zhang, 2014) and its variants (Reddi *et al.*, 2016) have also shown communication efficient performance on convex problems with the assumption that data is properly distributed in IID fashion with few clients.

Apparently, it seems that federated learning also works in the data parallelism fashion, however, data parallelism approaches in a centralized learning environment work with the key assumptions of IID data distributions and there is a centralized server to properly control the distribution of data while these assumptions strongly contradict with the federated learning environment.

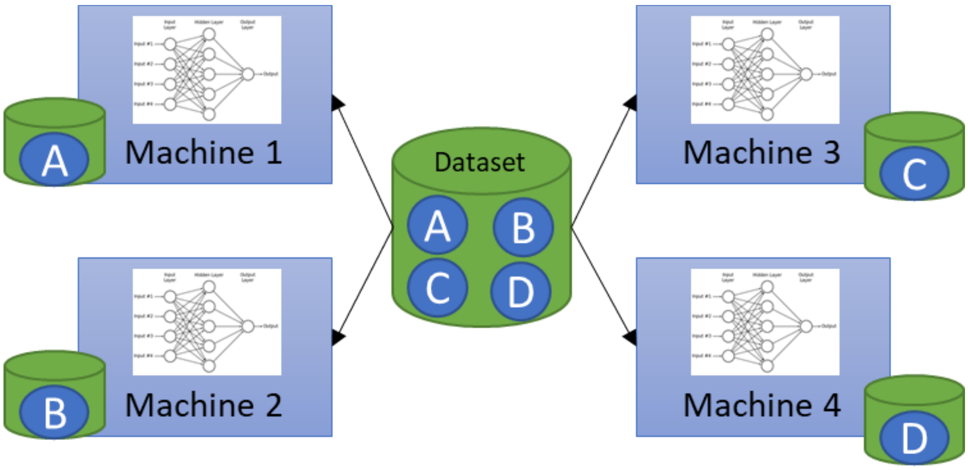

Figure 2.1 Data Parallelism example

2.1.2 Model Parallelism

Model parallelism is more effective in cases when there is a very large model having billions of parameters which can't be trained on a single machine due to limited memory and computational resources. In model parallelism, a centralized server splits the large model and distributes the sub-parts like different layers of a large network to multiple computing nodes as illustrated in Figure 2.2. These computing nodes train their assigned sub-part of a network (model) and then returned it to a centralized server which combines all these computations by aggregation or using some other methods. As compared to data parallelism, model parallelism is more complex and as there are so many different machine-learning model architectures therefore there is no standard to distribute the model.

Recently, different techniques have been proposed for the parallelization of deep models. For instance, (Teerapittayanon, McDanel and Kung, 2017; Ko *et al.*, 2019) argued that we can train the first few layers of a neural network on the local device(s) and the last layers can be trained on a server or cloud. Similarly, we can also use this model parallelism technique for more efficient data scoring.

Another very efficiently scalable large-scale deep learning framework named Adam (Chilimbi *et al.*, 2014) has shown very good classification performance on ImageNet 22K category tasks using only a cluster of 120 machines. Though in a naïve federated learning setting, we don’t assume model parallelism, however recently another potential approach (Split Learning aka SL) based on model parallelism is also being used to perform privacy-preserving decentralized learning (Gupta and Raskar, 2018). FL has been discussed in detail in section 2.3.

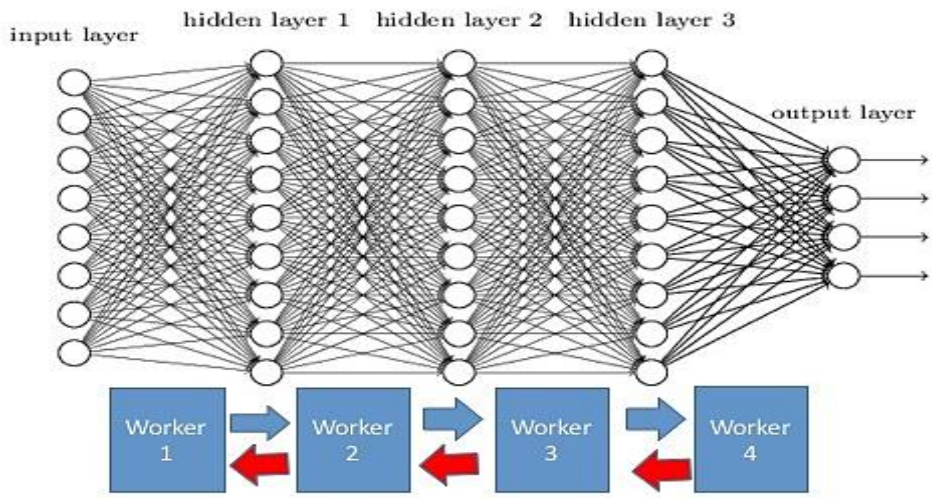

Figure 2.2 Model Parallelism example

2.2 Decentralized learning approaches

In decentralized collaborative learning, usually, there is no centralized server to properly distribute the data or model training among computing nodes. We can, loosely, explain it as a kind of peer-to-peer network where there is no single entity to claim the ownership of data and model training. Here we are considering the more practical scenario of decentralized computing where there are hundreds of devices having valuable but private data. These devices are not willing to compromise their privacy so they are not willing to share their data however still, they show consensus to participate in model training by performing some model training on their local devices and then these updated models could be shared with other devices. So, in this way, participating devices would collaborate with each other without compromising the privacy of their private data.

There are many decentralized learning approaches (Shokri and Shmatikov, 2015; Konečný *et al.*, 2016; Brendan McMahan *et al.*, 2017; Lalitha *et al.*, 2018; Corinzia and Buhmann, 2019) to learn from decentralized data however current state of art decentralized learning approach, called federated learning, pioneered and coined by

Google has been extensively used to learn from a highly unbalanced, massively distributed and private decentralized data. federated learning has been explained in detail in section 2.3.

Contrary to traditional distributed learning, in decentralized learning (federated learning)[2] there are a lot of unique challenges related to data, model architecture, communication, and privacy. Like, here, data is expected to be massively distributed, non-IID, unbalanced and inaccessible by other devices or centralized servers due to privacy constraints. Communication cost could be much higher as compared to computation cost and could face challenges of low and inconsistent bandwidth for different devices and, stagnant data. Participating devices may require specialized or more personalized models based on their requirements. Similarly, Privacy is one of the primary focuses of decentralized learning so local data of devices would not be accessible to any other party.

The FL works in such a way that its primary goal is to train a single global model with the collaboration of several devices whilst in practical scenarios, there are many situations where this assumption might not be true like for devices with different computational resources, it is intuitive that they might not have the capacity to train the same model architecture rather different devices might have different model architectures. Similarly, in the case of large organizations like hospitals and bank sectors, which also have private and sensitive information, it is intuitive that due to different business needs, they may have different model architectures so in such scenarios, sticking with the assumption of naïve federated learning to train the same deep learning model is not practical.

[2] As current privacy preserving decentralized learning domain has been first coined by Google as Federated Learning so almost all research work, being done in such decentralized learning environment, refer this domain as Federated Learning.

The model heterogeneity (having different model architectures) brings some new challenges like naïve federated learning can't be applied directly to this setting when clients have a different number of parameters (simple averaging of parameters or gradients is not possible) so to tackle this scenario, there is a need to investigate some other collaborative learning techniques which might be leveraged in this specific real-world scenario of decentralized learning.

Recently researchers have proposed some approaches to overcome the different challenges of federated learning like (Lalitha *et al.*, 2018) propose a fully decentralized learning approach to omit the need for a centralized server and clients only collaborate with just one hop device using a Bayesian-like approach however it's not scalable to real federated learning settings as it does not address the system heterogeneity and statistical heterogeneity issues of FL.

To address the statistical heterogeneity challenge of FL, some research work (Jeong *et al.*, 2018; Zhao *et al.*, 2018; Guha, Talwalkar and Smith, 2019) suggest sharing some local private data of clients or using some proxy labelled data on the server however these approaches could be difficult to implement like it could be difficult to annotate the data to make it labelled data, it could also create communication overhead and, in addition, it may also violate the key assumption (privacy) in FL.

There are some approaches (Smith *et al.*, 2017; Corinzia and Buhmann, 2019; Sattler, Muller and Samek, 2021) that reveal that Multi-Task Learning (MTL) could be a natural way to handle statistical heterogeneity by training separate but related models (tasks) on the client. However, these approaches also have different limitations. For instance (Smith *et al.*, 2017) only work for convex optimization problems. (Corinzia and Buhmann, 2019) is not scalable for large FL settings as it refines the models sequentially which could be much more expensive. Some researchers (Jeong *et al.*, 2018; Duan *et al.*, 2019) also suggest using data augmentation to make the local data distributions of clients as IID using GAN or some other approaches however these approaches could be computationally expensive and may risk the data privacy of participating clients.

To address the model heterogeneity challenge, Some researchers (Smith *et al.*, 2017) have leveraged distributed Multitask learning to handle the model heterogeneity challenge however in a significantly limited way as their work only focuses on convex problems and could not be applied to deep learning (non-convex) problems. In a similar fashion (Corinzia and Buhmann, 2019) has used the Bayesian network with multitask learning for model heterogeneity which is applicable to non-convex problems however is very costly to be scalable in large, federated learning scenario as they refine the models sequentially. One more recent work (Sattler, Muller and Samek, 2021) proposed a secondary method to improve the performance of client models. It uses the clustering approach with multitask learning to further improve the performance of models by personalizing the separate models of different devices once the federated learning has been performed. Therefore, significantly limited work has been performed to properly address the model heterogeneity challenges as mentioned earlier.

Some researchers (Jeong *et al.*, 2018; Lalitha *et al.*, 2018) have also proposed alternatives or modified schemes of federated learning like (Lalitha *et al.*, 2018) proposed one-hop federated learning where they omit the centralized server and devices only communicate with only their one hop devices. However, they work with the assumption of IID distribution and homogeneous model architectures. Similarly, some researchers (Gupta and Raskar, 2018) have proposed split learning (SL) as an alternative approach. In SL, usually, all devices try to train a single copy of the model by training different parts/layers of a single model. It mainly adopts the model parallelism approach and has two variants 1) with weight sharing: in this variant, devices are assumed to share the training weights among each other for synchronization, however it comes with a high risk of information leakage and high communication overhead. 2) without weight sharing: in this variant, devices are assumed to take an alternating turn of epochs to work with the server which is also a not realistic assumption in real DL settings. As in real DL settings, there is also a big challenge of the dropout of participating devices.

Another potential alternative approach called Distillation (Hinton, Vinyals and Dean, 2015; Anil *et al.*, 2018) has been proven to be a very efficient approach to efficiently transfer knowledge among independent models, more specifically, by efficiently distilling the knowledge from a trained model to an untrained model. As FL, also, typically require all the models to transfer their knowledge (trained model) to the centralized server for aggregation, Thus, apparently, distillation seems to be a potential approach which could be leveraged in FL settings to also address the model heterogeneity challenges. However, distillation also has its limitations in that it only works on IID data distribution whilst in FL, data distribution is naturally expected as non-IID so it could be challenging to properly apply distillation in the FL setting. Like, recently (Jeong *et al.*, 2018) proposed a Federated distillation algorithm which could be employed as an alternative scheme of FL to learn in a decentralized environment. Though, the author just adopted this approach to demonstrate that codistillation (a variant of distillation) is a much more communication-efficient technique as compared to standard FL however the author applies the codistillation approach after making all data distributions as IID using Generative Adversarial Network (GAN) approach. Distillation and Codistillation have been explained thoroughly in section 2.4 and section 2.5 respectively.