Bidirectional Hierarchical Federated Optimisation

Alexandru-Andrei Iacob
Laboratorul de Informatică
Universitatea din Cambridge
Supervizat de Dr. Nicholas Lane
aai30@cam.ac.uk

1 Introducere

Învățarea Federată (referită ca FL din termenul englez "Federated Learning") este o paradigma de Învățare Automată (referită ca ML din termenul englez "Machine Learning") distribuită care permite mai multor clienți să instruiască un model colaborativ comun fără a comunica date private. Aceasta a fost introdusa de McMahan et al. [38] ca un mijloc de reducere a costurilor de comunicare și de diminuare a problemelor de confidențialitate legate de stocarea datelor sensibile într-o locație centralizată, urmând principiile colectării concentrate și a minimizării datelor descrise în raportul de confidențialitate White House [52]. Aceste proprietăți au condus la aplicații FL cu cohorte mari de dispozitive de mici, cum ar fi predicția tastaturii mobile [12] pentru telefoanele Android, și aplicații cu entități mai mari supuse cerințelor de confidențialitate, cum ar fi spitalele [44]. Aceste două tipuri de Învățarea Federată sunt distinse de Kairouz et al. [21] ca FL cross-device și cross-silo.

Creșterea preponderenței FL de la publicarea McMahan et al. [38] poate fi atribuită către două trenduri. În primul rând, o creștere a cerințelor de confidențialitate ale consumatorilor și a cadrului juridic a pus presiune pe companiile de tehnologie. Această presiune a condus la interesul pentru ML care protejează confidențialitatea în cadrul corporațiilor majore precum Google [38, 12, 50], Microsoft [49], Meta [17, 39], și Apple [41]. În al doilea rând, ML s-a extins către domenii cu cerințe stricte de confidențialitate cum ar fi sănătatea [44], Recunoașterea Activităților Umane [45, 40] sau colaborările între corporații concurente [55]. Mai mult, apariția Modelelor de Limbaj Mari (referită ca LLM din termenul englez "Large Language Model") [4] a făcut accesarea colecțiilor private de limbaj natural avantajoasă, conducând la dezvoltarea Învățării Federate de Procesare a Limbajului Natural [33]. În mod similar, lansarea de ponderi (weights) pre-antrenate open source [48] permite colaborarea între entități cu resurse computaționale reduse utilizând framework-uri de FL [3, 27, 13].

Deși domeniul s-a bucurat de o atenție științifică și industrială sporită, beneficiile pe care le oferă în ceea ce privește confidențialitatea și comunicarea cauzează provocări semnificative în ceea ce privește creșterea eficienței și evoluția sistemelor federate. În mod crucial, compromisul de a antrena un singur model global nu este potrivit atunci când clienții eterogeni necesită personalizare parțială sau completă a modelului pentru distribuția lor locală de date.

Această lucrare propune abordarea provocărilor menționate prin construirea de structuri de rețea federate ierarhice de tip arbore, care permit flux de date bidirecțional și potențial ciclic, unde fiecare frunză este un client, iar fiecare nod intern este un server capabil să se instruiască pe date publice proxy. În consecință, nivelurile din arbore mai apropiate de frunze sunt mai personalizate pentru populația specifică de clienți a unui subarbore, iar cele mai apropiate de rădăcină oferă modele mai generalizabile. Această abordare este denumită Învățare Federată Ierarhică Bidirecțională (referită ca B-HFL din termenul englez "Bidirectional Hierarchical Federated Learning"). Mai mult, clienților frunză din aceste structuri le este permis să execute antrenare asincronă folosind modele persistente pentru a ține cont de schimbările temporale în distribuțiile lor de date și a le facilita evoluția.

1.1 Motivație

În forma sa standard, FL operează direct pe clienți folosind un server centralizat pentru a distribui parametrii modelului și apoi pentru a-i agrega după antrenarea clientului; acest proces este repetat pentru mai multe runde. Cu toate acestea, datele în FL sunt supuse atributelor precum locația geografică a clientului, specificațiile senzorului și comportamentul clientului. Datorită acestor factori, distribuția federată încalcă ipoteza Independentei și Identității Distribuție (IID). O astfel de *eterogenitate a datelor* [21, sec. 3.1] este împletită cu *eterogenitatea sistemelor* [21, sec. 7.2] deoarece clienții au abilități de calcul și viteze diferite de rețea. În plus, costurile de comunicare ale transmiterii parametrilor modelului între servere și clienți sunt semnificative. Deoarece eterogenitatea datelor face ca obținerea unui singur model global eficient pe toate distribuțiile de date ale clienților să fie imposibilă, suntem preocupați de crearea unor niveluri arbitrare de personalizare sub forma Învățării Federate Ierarhice într-un mod care îmbunătățește eficiența învățării și permite acestor sisteme să evolueze.

1.1.1 Eficiență

Eficiența și scalabilitatea au fost în centrul cercetării FL de la momentul în care Hard et al. [12] a aplicat FL la predicția tastaturii mobile la Google. Pe baza lucrării Hard et al. [12], Wang et al. [50] s-a demonstrat că FL poate fi folosit pentru a instrui modele peste zeci de milioane de smartphone-uri. Cu toate acestea, în ciuda prognozelor optimiste de un miliard de dispozitive ale Wang et al. [50], au apărut multiple limitări ale eficienței FL. Aceste limitări sunt de trei feluri: (a) FL sincron poate folosi eficient doar sute de dispozitive în fiecare rundă, (b) instruirea federată este considerabil mai lentă decât instruirea centralizată, (c) dispozitivele utilizatorilor sunt nesigure, ceea ce duce la deconectarea acestora. Aceste limitări au primit o atenție suplimentară în evaluarea empirică a Charles et al. [6].

Charles et al. [6] arată că performanța FL nu se îmbunătățește așa cum era de așteptat atunci când numărul de clienți instruiți în fiecare rundă crește, în ciuda lucrărilor teoretice anterioare [23] care indicau contrariul. Rezultatele lor experimentale arată că principala limitare a creșterii dimensiunii cohortelor în setări Non-IID este diferența dintre actualizările de model ale clienților, indicată printr-un cosinus aproape zero între acestea. Această diferență limitează impactul fiecărei runde, provoacă randamente diminuate la creșterea dimensiunii cohortelor și rezultă în incapacitatea de a învăța eficient din datele clientului în paralel. Astfel, având în vedere că algoritmii FL sunt intrinsec paraleli, scalabilitatea în FL este limitată de capacitatea de a învăța eficient pe baza fiecărui exemplu de antrenament al clienților. În plus, în timp ce investigațiile originale ale Wang et al. [50], Charles et al. [6] erau cross-device, problema învățării eficiente de la clienți se aplică și setărilor cross-silo.

1.1.2 Evoluție

Seturile de date ale clienților care formează o rețea federată nu sunt în general statice. Clienții pot șterge datele imediat după generare, periodic sau ad-hoc, în funcție de necesitățile de memorie sau cererile proprietarului. În plus, caracteristicile datelor nou adăugate se pot modifica în timp într-un mod gradat sau imediat. De exemplu, în sarcinile de recunoaștere a imaginilor, tranzițiile sezoniere pot modifica încet imaginile capturate, în timp ce schimbarea locațiilor sau actualizarea hardware-ului camerei poate duce la schimbări discrete. Această problemă este cunoscută sub numele de “shift” al setului de date [21, sec. 3.1] și reprezintă eterogenitatea *în-client* mai degrabă decât eterogenitatea *între-clienți* mai comună. Algoritmii sincroni de Învățare Federată [38, 42, 28] presupun că antrenarea clienților se realizează doar pe modelul federat primit la începutul unei runde. Chiar și sistemele care mențin modele locale persistente, cum ar fi (Ditto) [29], presupun că acest model persistent este folosit doar în timpul rundelor FL. Prin urmare, abordările actuale nu pot capta schimbările în distribuția datelor unui client. Sistemele asincrone de FL [53, 39], cum ar fi PAPAYA de la Meta [17], permit clienților să fie utilizați în afara limitelor rundei. Cu toate acestea, ele presupun în mod similar antrenarea clienților doar pe cea mai recentă copie a modelului federat pe care o pot accesa.

1.2 Rezumatul Propunerii

Această propunere extinde lucrările realizate de Iacob et al. [18] și Iacob et al. [19] pe subiectele de Învățare Federată personalizată, respectiv ierarhică. Sistemul propus comunică datele într-o structură de tip arbore, așa cum este ilustrat în Fig. 1. În mod crucial, parametrii modelului pot circula în ambele sensuri, iar nodurile pot aplica actualizări parțiale de la părinții lor prin agregare. În plus, fiecare nod poate asocia o pondere diferită parametrilor copiilor și părinților în timp ce folosește metode precum optimizatori adaptivi de server [42] sau metodele bazate pe antrenare [29, 25, 57]. Algoritmii adaptivi sunt relevanți deoarece permit fiecărui nod din arbore să se distingă în funcție de starea sa anterioară fără a necesita ajustarea suplimentară a parametrilor. În final, în cazul în care cohortele de clienți sunt grupate în mod semantic, această structură poate permite o creștere drastică a eficienței sistemului, deoarece fiecare cluster decide cum să optimizeze compromisul generalizare-personalizare [2]. Contribuțiile potențiale ale propunerii către acest domeniu includ:

1. O familie de algoritmi FL ierarhici și scalabili care permit un control fin asupra personalizării și generalizării de la rădăcina globală până la frunzele complet personalizate.
2. Investigarea a trei tehnici complementare permise de aceste structuri ierarhice: (a) permiterea clienților de la nivelul frunzelor să mențină modele locale persistente care se antrenează asincron pentru a aborda shiftul setului de date, (b) făcând ca orice nod din arbore să fie capabil să se antreneze cu un set de date proxy pentru a injecta o perspectivă generală modelului, (c) construirea de conexiuni verticale suplimentare în arbore similare cu conexiunile reziduale [14] pentru a permite un flux de date modificabil fără a schimba infrastructura de comunicare de bază.
3. Evaluări empirice extinse care iau în considerare scenarii cu sau fără clustere semnificative de clienți în sarcini de recunoaștere a limbajului menite publicării la conferințele ICLR sau MLSys. Această publicație va fi urmată de o lucrare destinată pentru MobiCom care investighează antrenamentul asincron pe dispozitive cu resurse limitate cu shift de set de date folosind clusterul Raspberry Pi FL din laboratorul Cambridge ML Systems.

2 Cercetare

The proposal in this document emerged as a natural consequence of research on Personalised Federated Learning and Hierarchical Federated Learning I began during my MPhil in Advanced Computer Science and the first year of my PhD in the Cambridge ML Systems lab led by my supervisor Dr. Nicholas Lane.

Iacob et al. [18] investigated the trade-off between generalisation and personalisation, which is at the heart of this work, from the perspectives of Fair Federated Learning and its interactions with local adaptation (fine-tuning) of the federated model post-training. Since Fair Federated Learning attempts to construct a more uniform accuracy distribution for the federated model over the local test sets of clients, the expectation was to either reduce the need for personalization or to provide a better starting point from which to carry it out. The experimental results showed that Fair FL brings no benefits and potential downsides towards later personalization and led to the proposal of a Personalisation-aware FL algorithm that attempts to anticipate the common regularises used during fine-tuning throughout the FL process.

Iacob et al. [19] evaluated the performance of Federated Human Activity Recognition [45] when trained using multimodal data gathered from different sensor types at increasing levels of privacy. It showed that grouping clients based on the type of sensor that produced their training set effectively mitigated the impacts of privacy being required at a human subject, environment, and sensor level simultaneously. It was a direct precursor to Bidirectional Hierarchical Federated Learning as it relied on a two-tiered model structure where each client trained both a group-level model and the global federated model using a mutual learning approach [57]. This work was later extended to consider the adaptability of such two-tiered systems to the addition of a new sensor type (group) into the federated; the extension was submitted to the MobiUK symposium. Mutual learning was chosen to relate the group-level and global models since it allowed divergent architectures that only shared the output layer. However, despite its success, this training method requires clients to have a high amount of data and local epochs to train both models. The expensive nature of the procedure prompted a move towards a model-averaging approach.

Both of the previous works were implemented in the Flower [3] FL framework; however, the scale of experimentation required for fully validating B-HFL would be unfeasible on the publicly available simulation engine. As such, I have contributed to constructing a new engine that doubles Flower simulations' throughput by intelligent ML-based client placement on GPUs. The paper presenting our techniques,for which I share an equal-contribution credit as a primary author, "High-throughput Simulation of Federated Learning via Resource-Aware Client Placement" has been submitted to Mobicom and is pending review.

3 Background and Related Work

The standard FL objective can be modelled as seen in Eq. (1)

$$\min_{\theta} F(\theta) = \sum_{c \in C} p_c F_c(\theta) \ , \tag{1}$$

where F is the federated objective, C is the client set, θ is the model, and F_c is the loss of client c weighted by their fraction of the total number of examples p_c. This formulation assumes that a single global model is being trained without regard for the distribution of its performance across client datasets. Federated Averaging (FedAvg) [38] trains the global model locally on clients, for each round t it sums the update

$\theta_t^c - \theta_t$ from client c weighted by p_c with the previous model θ_t using learning rate η, as seen in Eq. (2)

$$\theta_{t+1} = \theta_t + \eta \left(\sum_{c \in C} p_c \left(\theta_t^c - \theta_t \right) \right) . \tag{2}$$

The inability to colocate client data and the need to construct rough mixtures of model parameters as a compromise represent the leading causes of FL-specific challenges.

3.1 Heterogeneity

Non-IID data has been shown to impact both practical accuracies [58, 15] and theoretical convergence bounds [30]. It is thus worth detailing some forms of heterogeneity that Kairouz et al. [21] identify. The most commonly addressed form is quantity skew caused by clients having different amounts of data available. Standard FL algorithms effectively address Quantity skew via a simple reweighing (Eq. (2)). The other frequently-considered type of heterogeneity is label-distribution skew which is quantity skew per class. While these forms of heterogeneity have been most investigated, situations where features and labels are not related in the same manner across clients are far more pathological and may require some form of clustering or personalisation to tackle. In the worst-case scenario, each client may represent an entirely different task, as in Multi-Task Learning, with potentially no overlap in their solution space.

System (hardware) heterogeneity Devices within the federated network may differ regarding computational ability, storage, network speed, and reliability. They may also differ from themselves at a different point in time as their battery power, network connection, or operational mode vary. Importantly, variations in data-generating hardware, such as sensors, are linked to data heterogeneity. However, system heterogeneity and device unreliability harm the FL process independently of data. For example, slower hardware may result in straggling clients which elongate rounds in synchronous FL or operate on stale parameters in asynchronous FL. In addition, network or device unreliability creates dropout, which requires oversampling clients [50] and harms the effectiveness of maintaining client state across rounds.

Dataset Shift and Continual Learning Allowing ML models to participate in lifelong learning effectively is the goal of continual learning [8]; however, applying continual learning to the FL context is problematic for two primary reasons. First, the optimisation objective (Eq. (1)) intends to find a compromise model across all clients and cannot precisely fit all their data. Consequently, if the dataset of one client shifts independently of the whole network, the federated model will find it hard to adapt. Second, continual learning techniques such as Elastic-weight Consolidation [26], PackNet [36], and Learning without Forgetting [31] are designed for task-incremental settings where class labels are known, small amounts of previous data may still be available for specialised use cases [26], and there may even be different output heads for each task. The privacy requirements of FL make such solutions difficult at the level of the federated network without the addition of persistent local storage.

3.2 Federated Learning Efficiency

It is now worth expanding on the trends that Charles et al. [6] discovered. Those that limit the efficiency of FL in Non-IID settings where clients perform multiple SGD steps are of particular interest. Three significant effects can be observed. First, highly heterogeneous clients may cause sudden reductions in accuracy when their models are aggregated. Second, larger cohorts bring diminishing improvements in final accuracy and speed of convergence. Third, larger cohorts decrease data efficiency as more examples are needed for every accuracy gain.

These behaviours are approximately analogous to the well-known efficiency and generalisation limitations of large-batch training in centralised ML [22]. Charles et al. [6] find that data efficiency issues are caused by decreasing pseudo-gradient norms with increased cohort sizes and by the near-orthogonality of client updates following multiple steps of local training. The authors also find that adaptive optimisers fare better as cohort sizes grow due to scale invariance, making them particularly attractive aggregation algorithms.

3.2.1 Adaptive Federated Optimisation

Of particular relevance to this proposal are Federated Averaging with Server Momentum (FedAvgM) [16] and the more general Federated Adaptive Optimisation (FedOPT) [42]. They extend the concepts of momentum and adaptive optimisation [9, 24, 43] to Federated Learning on the *server-side* by treating client updates as pseudo-gradients and maintaining information across rounds on server-side accumulators. This structure allows such strategies to minimise the impact of individual rounds by averaging their pseudo-gradients and derived quantities with those of previous rounds. Since the outcome of individual rounds is highly variable based on the combination of clients selected and the model's current state, such techniques offer a more consistent optimisation trajectory.

Specifically, following the account provided by Reddi et al. [42] as shown in Eq. (3)

$$\Delta_t = \frac{1}{|C|}\sum_{c \in C} (\theta_t^c - \theta_t) \tag{3a}$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\Delta_t \tag{3b}$$

$$v_t = \beta_2 v_t + (1-\beta_2)\Delta_t^2 \tag{3c}$$

$$\theta_{t+1} = \theta_t + \eta \frac{m_t}{\sqrt{v_t} + \tau} \tag{3d}$$

for a given round t and federated model θ_t each client c in the selected set C trains the model locally to construct a personalised version θ_t^c. The pseudo-gradient Δ_t is then computed by averaging the differences between these personalised and federated models as shown in Eq. (3a). All operations on tensors are element-wise including division between tensors.

The first-moment accumulator m_t can then be constructed as the weighted average of the previous accumulator m_t and Δ_t using weight β_1 as shown in Eq. (3b). Thus, the pseudo-gradient of the current round is smoothed by those of the previous rounds decayed using β_1. Similarly, for the version of FedOpt based on Adam [24] the second-moment accumulator v_t keeps track of the element-wise second power of the pseudo-gradient denoted by Δ_t^2 as shown in Eq. (3c). These two accumulators are then used to compute the updated model for the next round θ_{t+1} using the server learning rate η as shown in Eq. (3d). Notably, the term $\sqrt{v_t}$ refers to the element-wise square root; it is used to normalise model parameters and make the algorithm scale-invariant to the pseudo-gradient. Finally, τ controls the adaptivity of FedOPT.

FedOPT presents several promising properties in the context of hierarchical FL. First, Reddi et al. [42] show it is highly resilient to the exact choice of hyperparameters, including learning rate, compared to standard FedAvg and FedAvgM. Second, their scale-invariance partially addresses the issues observed by Charles et al. [6] regarding the near-zero pseudo-gradients caused by the near-orthogonality of client updates. Third, they provide a means of automatically differentiating the learning rates of multiple servers based on the state of their accumulators without having to carry out hyperparameter tuning.

3.3 Related Work

To tackle the inherent trade-off between optimising for the average global performance versus the performance on the data of a specific client which can be seen in Eq. (1), two overall directions emerged in the literature. The first, exemplified by Fair Federated Learning [28], attempts to modify the importance of a client in the federated objective function to change the final model's effectiveness for that client. The second relaxes the single global model requirement by personalising the federated model [56, 46, 58], maintaining persistent fully-local models alongside it [29], clustering clients based on similarity [37, 10], or building hierarchies [35, 1]. Since the proposed B-HFL family of algorithms falls in the second camp, this section shall detail the most closely related work and present its limitations. Finally, the desired properties of the federated system and their relation to previous work are summarised in Table 1.

3.3.1 Personalised Federated Learning

Fully personalised FL refers to creating one model per client in addition to the global one. The most common means of achieving this is by local adaptation, or fine-tuning, of the federated model after training [56] with the potential additions of techniques such as Knowledge Distillation [57] or Elastic-weight Consolidation [26]. However, this two-stage optimisation is challenging to implement in an FL lifecycle where the federated model may need additional training after the adaptation phase has already been carried out. Furthermore, it provides no middle ground between the global and local models, which hurts the ability of such systems to integrate new clients, which may be incapable of fine-tuning.

A more recent approach is represented by Ditto [29] for settings where clients are visited frequently and can maintain state across rounds. Ditto allows clients to maintain a persistent local model and train it alongside the federated one during FL rounds. The two models are connected by incorporating the l_2 distance between their weights within the loss function of the local one. However, despite its proven benefits of fairness and robustness, persistent local models still face the challenges of traditional personalised models. Finally, they do not address dataset shifts within the client, as they only operate during training rounds.

3.3.2 Hierarchical Federated Learning and Clustering

The most relevant subfield of FL to our proposal is Hierarchical Federated Learning (HFL) introduced by Liu et al. [35]. Their proposed HierFAVG algorithm was developed primarily to handle the communication challenges of traditional cloud-based FL. In order to obtain scales of millions of participating clients [12, 50], FL systems relied on cloud infrastructure to connect devices over a wide geographic area and thus incurred additional latency. This trade-off was considered worthwhile since the larger populations were necessary for convergence, and edge servers, while capable of fast client communication, could not draw on a sufficient data pool. Liu et al. [35] argue that a two-level structure resolves the tensions between edge servers close to the clients and cloud servers. Abad et al. [1] propose an identical algorithm for heterogeneous celluar networks where edge servers are small cell base stations, and a central macro base station replaces the cloud server. Similarly to Liu et al. [35], Abad et al. [1] focus on reducing communication costs and go further in this direction by utilising update sparsification techniques [34, 47].

Clustering clients is an orthogonal synergistic technique that attempts to group participants based on a similarity metric. These clusters are constructed using various approaches, from clustering the model parameters directly as done in Ouyang et al. [40] do or using the loss of clients when assigned to a specific cluster as Mansour et al. [37] and Ghosh et al. [10] do. Clusters may also exist naturally based on characteristics like geographic location or language.

Previous works in HFL show a series of limitations. The HierFAVG algorithm directly extends FedAvg [38] by allowing the cloud server to treat edge servers as clients. However, because Liu et al. [35] and Abad et al. [1] only consider communication efficiency, they do not allow the edge servers to maintain greater personalisation and instead replace their model entirely during cloud-aggregation. Furthermore, their system does not consider asynchronicity, proxy training, or multi-level hierarchies. Regarding clustering, the available algorithms fail to obtain the desired trade-off between generalisation and personation. Standard clustering algorithms in FL assume data-sharing between clusters is unnecessary and do not directly map onto a hierarchical communication structure. Finally, they are not meant to provide a single global model besides the cluster models for applications where it would be beneficial.

Table 1: Gap analysis table showing proposed system's properties and overlap with closely related work.

Related Work	Hierarchical Structure	Personalisation	Allows Persistent Models	General Group Models	Meaningful Group Models	Asynchronous Work
Local Adaptation		✓				
Ditto		✓	✓			
Clustering						
HieFAVG	✓			✓	✓	
Asynchronous FL						✓
Bidirectional Hierarchical FL	✓	✓	✓	✓	✓	✓

4 Proposal

Given the shortcomings of traditional hierarchical FL systems, this work proposes Bidirectional Hierarchical Federated Learning (B-HFL), an alternative family of methods that optimises data and communication efficiency. This is achieved by using the hierarchical structure to organize communication between servers and control the dissemination of training parameters through the following design choices:

1. While previous methods such as HierFAVG [35, 1] entirely replace the edge-server and client models after global aggregation takes place, B-HFL performs partial aggregation between a children node and their parent, which allows children to maintain their local weights while incorporating global information. We propose modeling this in two phases:
   (a) Leaf-to-root aggregation: clients finish training, and their information is propagated up the tree. Each internal node has an internal parameter T_n, which determines after how many rounds it sends its updates to the parent. This value is equivalent to local client epochs and may be the same for all nodes at a given tree level or independently set per node.
   (b) Root-to-leaf aggregation: After a node has received and aggregated the training result from some or all of its children, it propagates its parameters down their subtree. The cost of this propagation is proportional to the depth of the subtree; however, the connection speed between internal nodes can be assumed to be higher than that of the clients to edge servers.
2. Internal nodes within the hierarchical structure can train on proxy datasets to regularise training as done by Guha et al. [11], Zhao et al. [58]. Proxy training is especially relevant for language modelling as large public corpora are available. In order to avoid operating on stale parameters, the natural point to add such training is after leaf-to-root aggregation reaches the node and before

root-to-leaf aggregation takes place. However, the latency incurred from such training may be too large. In that case, it can operate on stale parameters asynchronously while its subtrees execute.

3. All nodes may be allowed to operate synchronously or asynchronously concerning other nodes on the same level if necessary during leaf-to-root aggregation. For leaves (clients) under the control of an edge-server, this is equivalent to traditional asynchronous FL [53]. For an internal node, the same federated asynchronous strategies [39, 17] can be applied when receiving models from the child nodes, with client execution being replaced by the execution of the entire subtree.

Expressly, parameters aggregated from the leaf nodes (clients) up through the tree are fine-tuned to relevant local data. In contrast, parameters transmitted from parents to children are averaged over more numerous populations. When servers cover meaningfully clustered clients, these populations may be less related (e.g., covering multiple languages). Furthermore, if internal nodes are allowed to train on proxy datasets, they inject additional training into the federated models and provide regularisation for the entire tree. In traditional FL approaches, training on the server directly controlling the clients can impose overly strong regularisation; however, in B-HFL, higher nodes in the tree already represent a global picture and have limited impact at the leaves as their influence gets diluted through multiple intermediary nodes. Finally, allowing each client to maintain a persistent model across rounds and aggregate with their parents rather than entirely replacing their model makes them identical to any other node except for not having children.

Since not all nodes in the tree are required to be capable of training, it is worth distinguishing models which have been optimised via additional learning rather than mere aggregation. Specifically, training data being available may enable more efficient learning-based aggregation methods such as mutual learning [57] or l_2-based regularisation [29]. Additionally, updates constructed via training directly may offer a better optimisation signal. Thus, this work proposes adding dataflows directly between training nodes (e.g., clients and the root) while using the underlying hierarchical communication structure, like residual connection in ResNet [14]. For example, the system could allow the K client updates of each server with the highest absolute value to pass all the way to the root, where they may be merged via either training or adaptive optimisation with independent accumulator states. This sort of vertical connection provides highly dynamic and potentially cyclic dataflow. Another avenue worth exploring is allowing nodes, especially clients, to train asynchronously using their persistent model. This would permit clients to account for local dataset shift using well-known techniques from the Continual Learning literature [8, 31, 26].

The system may bring several potential benefits:

1. Can accommodate nodes having different aggregation methods, learning rates, dynamic optimiser states for leaf-to-root and root-to-leaf aggregation. Similarly to the number of rounds T, parameters related to aggregation may be independent or set on a per-tree or per-level basis.
2. Smaller cohorts for each edge-server avoids the issue of decreasing pseudo-gradients norms noticed by Charles et al. [6], as does cluster clients prior to edge-server assignment.
3. While persistent local models are known to work well in cross-silo FL, this hierarchical structure makes them relevant in cross-device settings by potentially allowing a much larger number of clients to be sampled every thus permitting them to be visited more than once.
4. Can naturally integrate Secure Aggregation [5, 20] at the level of each edge-server. As first noted by Wang et al. [50], this reduces additional communication cost of training C clients with Secure Aggregation from $\mathcal{O}(C^2)$ to $\mathcal{O}(C^2/M)$ where M is the number of edge-servers. Secure Aggregation and Differential Privacy [51] only need to be applied at the lowest level of the tree.

4.1 Example System

An example of a B-HFL system, which would be the primary deliverable of this proposal, may be seen in Fig. 1. The central server controls a proxy dataset used to train after it performs aggregation. Intermediary servers perform only aggregation. All servers send their updates to the parent after every round.

Each node, including the clients, runs at-least two stateful FedOPT server optimizers with separate learning rates, one for the leaf-to-root aggregation with the averaged pseudo-gradient Δ_t and one for parent aggregation. Even if the same leaf-to-root learning rate $\eta^{\uparrow}$ and root-to-leaf learning rate $\eta^{\downarrow}$ were to be used for all nodes in the tree or at a given level, the independent server optimiser states would distinguish the aggregation procedure of their node based on historical trends.

The residual connections serve different functions between the leaf-to-root and root-to-leaf stages. For the upward stage, they collect the client update with the highest absolute value from all edge servers, thus sending one additional model to the central server for each edge-server. The central server may then maintain independent optimiser states for each incoming "residual" connection. For the downward stage,

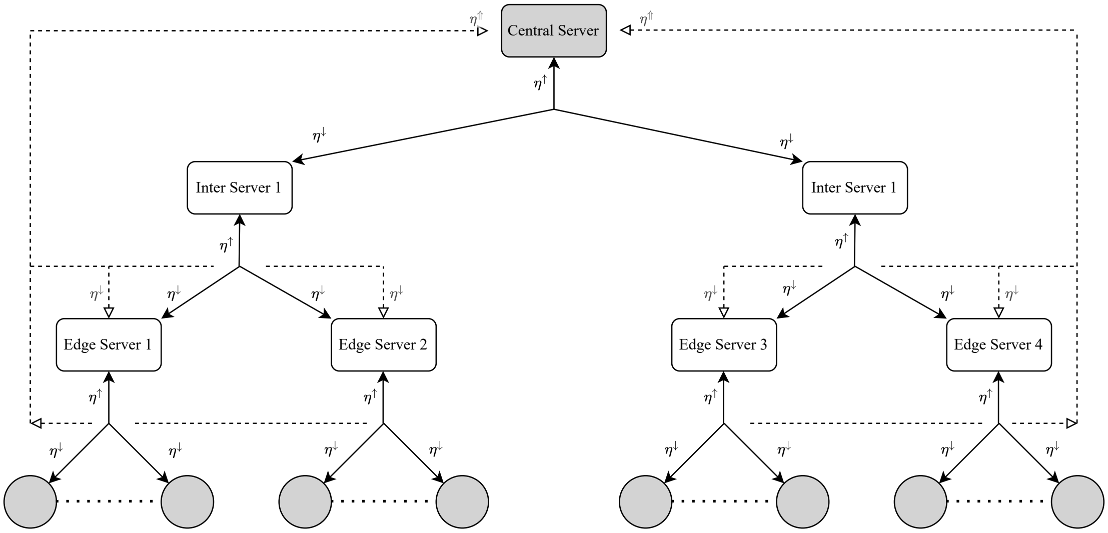

Figure 1: Diagram of an example B-HFL system. Solid lines represent communication links, while dashed lines represent conceptual "residual" connections using the underlying links. Nodes capable of training, such as clients or the central server with a proxy dataset, are in grey. When model parameters propagate up, nodes merge the incoming pseudo-gradients and update theirs model using the leaf-to-root learning rate $\eta^{\uparrow}$. The same happens when parameters flow from parents to children nodes with learning rate $\eta^{\downarrow}$. Since the dashed lines communicate 0 to K models, $\eta^{\Uparrow}$ may represent 0 to K aggrgations using a $\eta^{\downarrow}$ learning rate.

they provide the edge servers with a chance to directly benefit from the training of the central server without having to rely on the models of the intermediary servers. While this last component is somewhat superfluous in the small hierarchy shown by Fig. 1, it would prove highly relevant for profound structures. For example, for deep hierarchies, parameters that receive extra training at the central server might get averaged several times before reaching the edge servers and thus influencing the leaves.

5 Plan and Timeline

The presented family of Bidirectional Hierarchical Federated Learning algorithms will be developed during the PhD period and will form part of the final PhD thesis. In addition, before the final thesis, it offers opportunities for conference publications that significantly contribute to Federated Learning. Given the novelty of FL in general and hierarchical FL in particular, there is ample room for further developments in the structure of B-HFL as the fields mature.

The summer period of the end of my first year of the PhD shall be dedicated to implementing the example version of B-HFL in the Flower [3] FL framework affiliated with our research group. The framework is currently tuned to standard FL settings and would require heavy API modifications to execute and simulate hierarchical FL effectively. However, the previous work on group-level models for Federated Human Activity Recognition of Iacob et al. [19] and the effective FL simulation engine I contributed to can be the basis for implementing and streamlining the process.

The autumn Michaelmas Term of my second year will have as a main objective the publication of a conference paper based on the example system proposed in Section 4.1. I have already discussed this with my supervisor Dr. Nicholas Lane, and we have agreed that both ICLR and MLSys would be appropriate venues. Given the growing importance of LLMs, and the trade-offs recently discovered by Agarwal et al. [2] in terms of their generalization and personalization abilities with or without pre-trained weights, they represent a natural application for the proposed hierarchical system. Moreover, multi-language text prediction provides a naturally clustered FL application corresponding to real-world scenarios where countries have independent edge servers for FL and must collaborate at a continental and global level. The study would use a large multi-lingual BERT model [7] together with two multi-language datasets [e.g., 32, 54] for training. One dataset will be partitioned by language, and the other will be kept as a proxy dataset at the central server in Fig. 1. The study's goals would be to compare the final accuracy of each model at every level of the hierarchy on the client test sets and the centralised test set created from the proxy dataset. The expectation would be for the model performance on the data of a specific client to be proportional to their proximity to that client in the tree. Alternatively, for the proxy test set and the union of all client test sets, accuracy should be proportional to the proximity to the central server. In addition, ablation studies on the "residual" connections, adaptive optimization, or persistent local models will also be performed with efficiency comparisons between node-execution asynchronicity at different levels of the tree. Finally, if time allows, the paper could include other naturally-clustered tasks, such as speech recognition for multilingual data or algorithmic clustering of a standard dataset.

Following the publication of this work, a natural extension during Lent and Easter terms would be to tackle a setting where clients continuously generate and delete data with limited local storage. The example system would be extended to allow asynchronous training on all nodes, including the leaves, which run parallel to the actual FL component. Each client would generate a data stream while having a fixed internal memory to operate on during training. Real resource constraints and asynchronicity can be modelled using the Raspberry Pi FL cluster at Cambridge ML Systems. This work would likely be intended for MobiCom, the same venue we submitted the Flower simulation engine to, or another systems-oriented conference.

If successful, the second year of my PhD would bring a valuable contribution to the field of Federated Learning, result in one or more conference publication with a potential workshop work along the way, and allow me to proceed through the rest of the PhD program with a substantial amount of progress towards my final thesis. It would also result in a major extensenion to the Flower [3] FL framework with potential for future colaborations or employment with the Flower Labs startup funded by Y Combinator. Following the completion of my PhD, I intend to pursue a career in either private or academic research.

References

[1] Mehdi Salehi Heydar Abad, Emre Ozfatura, Deniz Gündüz, and Özgür Erçetin. Hierarchical federated learning ACROSS heterogeneous cellular networks. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pages 8866–8870. IEEE, 2020. doi: 10.1109/ICASSP40776.2020.9054634. URL `https://doi.org/10.1109/ICASSP40776.2020.9054634`. Cited on page 5, Cited on page 6

[2] Ankur Agarwal, Mehdi Rezagholizadeh, and Prasanna Parthasarathi. Practical takes on federated learning with pretrained language models. In Andreas Vlachos and Isabelle Augenstein, editors, *Findings of the Association for Computational Linguistics: EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 454–471. Association for Computational Linguistics, 2023. URL `https://aclanthology.org/2023.findings-eacl.34`. Cited on page 2, Cited on page 8

[3] Daniel J. Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Titouan Parcollet, and Nicholas D. Lane. Flower: A friendly federated learning research framework. *CoRR*, abs/2007.14390, 2020. URL `https://arxiv.org/abs/2007.14390`. Cited on page 1, Cited on page 3, Cited on page 8, Cited on page 9

[4] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ B. Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258, 2021. URL `https://arxiv.org/abs/2108.07258`. Cited on page 1

[5] Kallista A. Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for federated learning on user-held data. *CoRR*, abs/1611.04482, 2016. URL `http://arxiv.org/abs/1611.04482`. Cited on page 7

[6] Zachary Charles, Zachary Garrett, Zhouyuan Huo, Sergei Shmulyian, and Virginia Smith. On large-cohort training for federated learning. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 20461–20475, 2021. URL `https://proceedings.neurips.cc/paper/2021/hash/ab9ebd57177b5106ad7879f0896685d4-Abstract.html`. Cited on page 2, Cited on page 4, Cited on page 5, Cited on page 7

[7] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.747. URL `https://doi.org/10.18653/v1/2020.acl-main.747`. Cited on page 8

[8] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2022. doi: 10.1109/TPAMI.2021.3057446. Cited on page 4, Cited on page 7

[9] John C. Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, 2011. doi: 10.5555/1953048.2021068. URL `https://dl.acm.org/doi/10.5555/1953048.2021068`. Cited on page 4

[10] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *IEEE Trans. Inf. Theory*, 68(12):8076–8091, 2022. doi: 10.1109/TIT.2022.3192506. URL `https://doi.org/10.1109/TIT.2022.3192506`. Cited on page 5, Cited on page 6

[11] Neel Guha, Ameet Talwalkar, and Virginia Smith. One-shot federated learning. *CoRR*, abs/1902.11175, 2019. URL `http://arxiv.org/abs/1902.11175`. Cited on page 6

[12] Andrew Hard, Kanishka Rao, Rajiv Mathews, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *CoRR*, abs/1811.03604, 2018. URL `http://arxiv.org/abs/1811.03604`. Cited on page 1, Cited on page 2, Cited on page 6

[13] Chaoyang He, Songze Li, Jinhyun So, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, Li Shen, Peilin Zhao, Yan Kang, Yang Liu, Ramesh Raskar, Qiang Yang, Murali Annavaram, and Salman Avestimehr. Fedml: A research library and benchmark for federated machine learning. *CoRR*, abs/2007.13518, 2020. URL `https://arxiv.org/abs/2007.13518`. Cited on page 1

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90. Cited on page 3, Cited on page 7

[15] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip B. Gibbons. The non-iid data quagmire of decentralized machine learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4387–4398. PMLR, 2020. URL http://proceedings.mlr.press/v119/hsieh20a.html. Cited on page 4

[16] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *CoRR*, abs/1909.06335, 2019. URL http://arxiv.org/abs/1909.06335. Cited on page 4

[17] Dzmitry Huba, John Nguyen, Kshitiz Malik, Ruiyu Zhu, Mike Rabbat, Ashkan Yousefpour, Carole-Jean Wu, Hongyuan Zhan, Pavel Ustinov, Harish Srinivas, Kaikai Wang, Anthony Shoumikhin, Jesik Min, and Mani Malek. PAPAYA: practical, private, and scalable federated learning. In Diana Marculescu, Yuejie Chi, and Carole-Jean Wu, editors, *Proceedings of Machine Learning and Systems 2022, MLSys 2022, Santa Clara, CA, USA, August 29 - September 1, 2022*. mlsys.org, 2022. URL https://proceedings.mlsys.org/paper/2022/hash/f340f1b1f65b6df5b5e3f94d95b11daf-Abstract.html. Cited on page 1, Cited on page 2, Cited on page 7

[18] Alex Iacob, Pedro Porto Buarque Gusmão, and Nicholas Lane. Can fair federated learning reduce the need for personalisation? In *Proceedings of the 3rd Workshop on Machine Learning and Systems*, EuroMLSys '23, page 131–139, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400700842. doi: 10.1145/3578356.3592592. URL https://doi.org/10.1145/3578356.3592592. Cited on page 2, Cited on page 3

[19] Alex Iacob, Pedro Porto Buarque Gusmão, Nicholas Lane, Armand Koupai, Mohammud Bocus, Raul Santos-Rodriguez, Robert Piechocki, and Ryan McConville. Privacy in multimodal federated human activity recognition. In *To be Published in Proceedings of the 3rd On-Device Intelligence Workshop*, MLSys '23, 2023. URL https://sites.google.com/g.harvard.edu/on-device-workshop-23/home?authuser=0. Cited on page 2, Cited on page 3, Cited on page 8

[20] Swanand Kadhe, Nived Rajaraman, Onur Ozan Koyluoglu, and Kannan Ramchandran. Fastsecagg: Scalable secure aggregation for privacy-preserving federated learning. *CoRR*, abs/2009.11248, 2020. URL https://arxiv.org/abs/2009.11248. Cited on page 7

[21] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista A. Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, 14(1-2):1–210, 2021. doi: 10.1561/2200000083. URL https://doi.org/10.1561/2200000083. Cited on page 1, Cited on page 2, Cited on page 4

[22] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=H1oyRlYgg. Cited on page 4

[23] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local SGD on identical and heterogeneous data. In Silvia Chiappa and Roberto Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 4519–4529. PMLR, 2020. URL http://proceedings.mlr.press/v108/bayoumi20a.html. Cited on page 2

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980. Cited on page 4, Cited on page 5

[25] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114. URL https://www.pnas.org/doi/abs/10.1073/pnas.1611835114. Cited on page 2

[26] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. Cited on page 4, Cited on page 5, Cited on page 7

[27] Fan Lai, Yinwei Dai, Sanjay Sri Vallabh Singapuram, Jiachen Liu, Xiangfeng Zhu, Harsha V. Madhyastha, and Mosharaf Chowdhury. Fedscale: Benchmarking model and system performance of federated learning at scale. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 11814–11827. PMLR, 2022. URL https://proceedings.mlr.press/v162/lai22a.html. Cited on page 1

[28] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=ByexElSYDr. Cited on page 2, Cited on page 5

[29] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 6357–6368. PMLR, 2021. URL http://proceedings.mlr.press/v139/li21h.html. Cited on page 2, Cited on page 5, Cited on page 7

[30] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=HJxNAnVtDS. Cited on page 4

[31] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. Cited on page 4, Cited on page 7

[32] Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Bruce Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Rangan Majumder, and Ming Zhou. XGLUE: A new benchmark dataset for cross-lingual pre-training, understanding and generation. *CoRR*, abs/2004.01401, 2020. URL https://arxiv.org/abs/2004.01401. Cited on page 8

[33] Bill Yuchen Lin, Chaoyang He, Zihang Ze, Hulin Wang, Yufen Hua, Christophe Dupuy, Rahul Gupta, Mahdi Soltanolkotabi, Xiang Ren, and Salman Avestimehr. Fednlp: Benchmarking federated learning methods for natural language processing tasks. In Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz, editors, *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 157–175. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-naacl.13. URL https://doi.org/10.18653/v1/2022.findings-naacl.13. Cited on page 1

[34] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and Bill Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=SkhQHMW0W. Cited on page 6

[35] Lumin Liu, Jun Zhang, Shenghui Song, and Khaled B. Letaief. Client-edge-cloud hierarchical federated learning. In *2020 IEEE International Conference on Communications, ICC 2020, Dublin, Ireland, June 7-11, 2020*, pages 1–6. IEEE, 2020. doi: 10.1109/ICC40277.2020.9148862. URL https://doi.org/10.1109/ICC40277.2020.9148862. Cited on page 5, Cited on page 6

[36] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2017. Cited on page 4

[37] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *CoRR*, abs/2002.10619, 2020. URL https://arxiv.org/abs/2002.10619. Cited on page 5, Cited on page 6

[38] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Aarti Singh and Xiaojin (Jerry) Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 2017. URL http://proceedings.mlr.press/v54/mcmahan17a.html. Cited on page 1, Cited on page 2, Cited on page 3, Cited on page 6

[39] John Nguyen, Kshitiz Malik, Hongyuan Zhan, Ashkan Yousefpour, Mike Rabbat, Mani Malek, and Dzmitry Huba. Federated learning with buffered asynchronous aggregation. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event*, volume 151 of *Proceedings of Machine Learning Research*, pages 3581–3607. PMLR, 2022. URL https://proceedings.mlr.press/v151/nguyen22b.html. Cited on page 1, Cited on page 2, Cited on page 7

[40] Xiaomin Ouyang, Zhiyuan Xie, Jiayu Zhou, Jianwei Huang, and Guoliang Xing. Clusterfl: a similarity-aware federated learning system for human activity recognition. In Suman Banerjee, Luca Mottola, and Xia Zhou, editors, *MobiSys '21: The 19th Annual International Conference on Mobile Systems, Applications, and Services, Virtual Event, Wisconsin, USA, 24 June - 2 July, 2021*, pages 54–66. ACM, 2021. doi: 10.1145/3458864.3467681. URL https://doi.org/10.1145/3458864.3467681. Cited on page 1, Cited on page 6

[41] Matthias Paulik, Matt Seigel, Henry Mason, Dominic Telaar, Joris Kluivers, Rogier C. van Dalen, Chi Wai Lau, Luke Carlson, Filip Granqvist, Chris Vandevelde, Sudeep Agarwal, Julien Freudiger, Andrew Byde, Abhishek Bhowmick, Gaurav Kapoor, Si Beaumont, Áine Cahill, Dominic Hughes, Omid Javidbakht, Fei Dong, Rehan Rishi, and Stanley Hung. Federated evaluation and tuning for on-device personalization: System design & applications. *CoRR*, abs/2102.08503, 2021. URL https://arxiv.org/abs/2102.08503. Cited on page 1

[42] Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=LkFG3lB13U5. Cited on page 2, Cited on page 4, Cited on page 5

[43] Sebastian Ruder. An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.04747, 2016. URL http://arxiv.org/abs/1609.04747. Cited on page 4

[44] Micah J. Sheller, Brandon Edwards, G. Anthony Reina, Jason Martin, Sarthak Pati, Aikaterini Kotrotsou, Mikhail Milchenko, Weilin Xu, Daniel Marcus, Rivka R. Colen, and Spyridon Bakas. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific Reports*, 10(1):12598, 2020. doi: 10.1038/s41598-020-69250-1. URL https://doi.org/10.1038/s41598-020-69250-1. Cited on page 1

[45] Konstantin Sozinov, Vladimir Vlassov, and Sarunas Girdzijauskas. Human activity recognition using federated learning. In Jinjun Chen and Laurence T. Yang, editors, *IEEE International Conference on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications, ISPA/IUCC/BDCloud/SocialCom/SustainCom 2018, Melbourne, Australia, December 11-13, 2018*, pages 1103–1111. IEEE, 2018. doi: 10.1109/BDCloud.2018.00164. URL https://doi.org/10.1109/BDCloud.2018.00164. Cited on page 1, Cited on page 3

[46] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *CoRR*, abs/2103.00710, 2021. URL https://arxiv.org/abs/2103.00710. Cited on page 5

[47] Hanlin Tang, Shaoduo Gan, Ce Zhang, Tong Zhang, and Ji Liu. Communication compression for decentralized training. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7663–7673, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/44feb0096faa8326192570788b38c1d1-Abstract.html. Cited on page 6

[48] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023. doi: 10.48550/arXiv.2302.13971. URL https://doi.org/10.48550/arXiv.2302.13971. Cited on page 1

[49] Ewen Wang, Ajay Kannan, Yuefeng Liang, Boyi Chen, and Mosharaf Chowdhury. FLINT: A platform for federated learning integration. *CoRR*, abs/2302.12862, 2023. doi: 10.48550/arXiv.2302.12862. URL https://doi.org/10.48550/arXiv.2302.12862. Cited on page 1

[50] Ewen Wang, Ajay Kannan, Yuefeng Liang, Boyi Chen, and Mosharaf Chowdhury. FLINT: A platform for federated learning integration. *CoRR*, abs/2302.12862, 2023. doi: 10.48550/arXiv.2302.12862. URL https://doi.org/10.48550/arXiv.2302.12862. Cited on page 1, Cited on page 2, Cited on page 4, Cited on page 6, Cited on page 7

[51] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farhad Farokhi, Shi Jin, Tony Q. S. Quek, and H. Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Trans. Inf. Forensics Secur.*, 15:3454–3469, 2020. doi: 10.1109/TIFS.2020.2988575. URL https://doi.org/10.1109/TIFS.2020.2988575. Cited on page 7

[52] White House. Consumer data privacy in a networked world: A framework for protecting privacy and promoting innovation in the global digital economy. *Journal of Privacy and Confidentiality*, 4(2), Mar. 2013. doi: 10.29012/jpc.v4i2.623. URL https://journalprivacyconfidentiality.org/index.php/jpc/article/view/623. Cited on page 1

[53] Chenhao Xu, Youyang Qu, Yong Xiang, and Longxiang Gao. Asynchronous federated learning on heterogeneous devices: A survey. *CoRR*, abs/2109.04269, 2021. URL https://arxiv.org/abs/2109.04269. Cited on page 2, Cited on page 7

[54] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. *CoRR*, abs/2010.11934, 2020. URL https://arxiv.org/abs/2010.11934. Cited on page 8

[55] Han Yu, Zelei Liu, Yang Liu, Tianjian Chen, Mingshu Cong, Xi Weng, Dusit Niyato, and Qiang Yang. A sustainable incentive scheme for federated learning. *IEEE Intell. Syst.*, 35(4):58–69, 2020. doi: 10.1109/MIS.2020.2987774. URL https://doi.org/10.1109/MIS.2020.2987774. Cited on page 1

[56] Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov. Salvaging federated learning by local adaptation. *CoRR*, abs/2002.04758, 2020. URL https://arxiv.org/abs/2002.04758. Cited on page 5

[57] Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. Deep mutual learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4320–4328. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00454. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_Deep_Mutual_Learning_CVPR_2018_paper.html. Cited on page 2, Cited on page 3, Cited on page 5, Cited on page 7

[58] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *CoRR*, abs/1806.00582, 2018. URL http://arxiv.org/abs/1806.00582. Cited on page 4, Cited on page 5, Cited on page 6