Învățare Federată Ierarhică Bidirecțională

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 paradigmă de Învățare Automată Distribuită (referită ca ML din termenul englez "Machine Learning") care permite mai multor clienți să antreneze un model colaborativ comun fără a comunica date private. Aceasta a fost introdusa de McMahan et al. [30] 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. Aceste proprietăți au condus la aplicații FL utilizând grupuri mari de dispozitive de dimensiuni mici, cum ar fi predicția tastaturii mobile [11] pentru telefoanele Android și aplicații cu entități mai mari supuse cerințelor de confidențialitate, cum ar fi spitalele [34]. Aceste două tipuri de Învățare Federată sunt distinse de Kairouz et al. [16] ca FL cross-device și cross-silo. Pentru restul acestei lucrări un "client" se referă la o entitate deținătoare de date private ce efectuează antrenament federat (e.g., telefoane/spitale).

Creșterea preponderenței FL de la publicarea McMahan et al. [30] poate fi atribuită către două trenduri. În primul rând, o creștere a cerințelor de confidențialitate ale consumatorilor și ale 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 [30, 11, 6], Microsoft [37] și Meta [13, 31]. În al doilea rând, ML s-a extins către domenii cu cerințe stricte de confidențialitate precum sănătatea [34], Recunoașterea Activităților Umane [35, 32] sau colaborările între corporații [40]. 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 FL pentru Procesare a Limbajului Natural [26]. În mod similar, lansarea de ponderi (weights) open pre-antrenate [36] permite colaborarea între entități cu resurse computaționale reduse, utilizând framework-uri de FL [3].

Deși domeniul s-a bucurat de o atenție științifică și industrială sporită, beneficiile pe care le oferă 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 personalizarea 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, unde fiecare frunză este un client, iar fiecare nod intern este un server capabil de antrenament pe date proxy. În consecință, nodurile apropiate de frunze sunt personalizate pentru populația specifică de clienți a subarborelui, iar cele apropiate de rădăcină oferă modele generalizate. Această abordare este denumită Învățare Federată Ierarhică Bidirecțională (referită ca B-HFL din termenul englez "Bidirectional Hierarchical Federated Learning"). Mai mult, clienții pot executa antrenare asincronă cu modele persistente pentru a aborda shitful în distribuțiile lor de date.

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 Independenței și Identității Distribuției (IID). O astfel de *eterogenitate a datelor* [16, sec. 3.1] este împletită cu *eterogenitatea sistemelor* [16, sec. 7.2] deoarece clienții au abilități de calcul și viteze diferite de rețea. În plus, costurile de comunicare ale modelului între servere și clienți sunt

semnificative. Deoarece eterogenitatea datelor face construirea unui singur model global eficient pentru datele tuturor clienților să fie imposibilă, este propusă crearea unor niveluri arbitrare de personalizare sub forma Învățării Federate Ierarhice într-un mod ce îmbunătățește eficiența și permite evoluția sistemelor.

1.1.1 Eficiență

Eficiența și scalabilitatea au fost în centrul cercetării FL de la momentul în care Hard et al. [11] a aplicat FL pentru predicția tastaturii mobile de la Google. Pe baza lucrării Hard et al. [11], Bonawitz et al. [6] a demonstrat că FL poate fi folosit pentru a antrena modele în zeci de milioane de smartphone-uri. Cu toate acestea, în ciuda prognozelor optimiste de un miliard de dispozitive ale Bonawitz et al. [6], au apărut multiple limitări ale eficienței FL. Aceste limitări sunt de trei feluri: (a) FL sincron poate folosi eficient doar sute de dispozitive din miloane în fiecare rundă, (b) antrenarea federată este considerabil mai lentă decât antrenarea 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. [7].

Charles et al. [7] arată că performanța FL nu se îmbunătățește precum era de așteptat când numărul de clienți antrenați într-o rundă crește, în ciuda lucrărilor teoretice contrare [18]. Rezultatele lor experimentale arată că principala limitare a creșterii dimensiunii grupurilor 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 grupurilor și rezultă în incapacitatea de a învăța eficient din datele clienților. Astfel, având în vedere că algoritmii FL sunt intrinsec paraleli, scalabilitatea lor este limitată de capacitatea de a învăța eficient pe baza fiecărui exemplu de antrenament al clienților. În plus, în timp ce investigațiile originale ale Bonawitz et al. [6], Charles et al. [7] erau cross-device, problema învățării eficiente de la clienți se aplică și situațiilor 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 de 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 [16, sec. 3.1] și reprezintă eterogenitatea *în-client* mai degrabă decât eterogenitatea *între-clienți*, mai comună. Algoritmii sincroni de Învățare Federată [30, 33, 22] presupun că antrenarea clienților se realizează doar pe modelul federat primit la începutul unei runde. Chiar și sistemele ce mențin modele locale persistente [23], 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 [38, 31], precum PAPAYA de la Meta [13], permit clienților să fie utilizați în afara limitelor unei runde. Cu toate acestea, ele consideră antrenarea clienților doar pe cea mai recentă versiune accesibila a modelului federat.

1.2 Rezumatul propunerii

Această propunere extinde lucrările realizate de Iacob et al. [14] și Iacob et al. [15] 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 modelelor 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 [33] sau cele bazate pe antrenare [23, 20, 42]. 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 grupurile de clienți sunt construite în mod semantic, această structură poate permite o creștere drastică a eficienței sistemului, deoarece fiecare cluster decide cum să optimizeze între generalizare și personalizare [2]. Contribuțiile potențiale ale propunerii 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 frunzelor (clienților) să mențină modele locale persistente care se antrenează asincron pentru a aborda shiftul temporal al setului de date, (b) capabilitatea ca orice nod din arbore 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 [12] pentru a permite un flux de date modificabil fără a schimba infrastructura de comunicare.

3. Evaluări empirice extinse care iau în considerare scenarii cu sau fără clustere semnificative de clienți în sarcini de recunoaștere a limbajului sau, dacă timpul permite, a vorbirii umane.
4. O lucrare științifică care este menită publicării la conferința ICLR sau MLSys. Această publicație va fi urmată de o lucrare destinată pentru conferința MobiCom ce 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 Publicații anterioare

Propunerea din acest document a apărut drept consecință naturală a cercetărilor privind Învățarea Federată Personalizată și Învățarea Federată Ierarhică efectuate în timpul Masterului meu în Informatică Avansată și în primul an al doctoratului meu în laboratorul Cambridge ML Systems, condus de Dr. Nicholas Lane.

Iacob et al. [14] a investigat compromisul dintre generalizare și personalizare, care este în centrul acestui studiu, din perspectiva "Fair" FL (FFL) și a interacțiunilor sale cu adaptarea locală (fine-tuning) a modelului federat post-antrenament. Deoarece FFL încearcă să construiască o distribuție mai uniformă a acurateții pentru modelul federat pe seturile de date de testare locale ale clienților, așteptarea era fie reducerea necesității personalizării, fie oferirea unui punct de plecare mai avantajos din care să fie efectuată adaptarea locală. Rezultatele experimentale au arătat că FFL nu aduce beneficii. În schimb, are potențiale dezavantaje pentru aplicarea ulterioară a adaptării. Aceste dezavantaje au dus la propunerea unui algoritm FL conștient de personalizarea ulterioară ("Personalisation-aware Federated Learning") care încearcă să anticipeze funcțiile de cost comune, utilizate în timpul fine-tuning-ului pe parcursul procesului FL.

Iacob et al. [15] a evaluat performanța Recunoașterii Activităților Umane Federate [35] folosind date multimodale adunate de la diferite tipuri de senzori. Scopul era evaluarea efectului menținerii datelor în cadrul stocării private, cu un nivel crescător de confidențialitate. Studiul a demonstrat că gruparea clienților în funcție de tipul de senzor care a produs setul lor de antrenament atenuează eficient impactul necesității de confidențialitate la nivel de subiect uman, de mediu și de senzor. Această lucrare a fost un precursor direct al B-HFL, deoarece se baza pe o structură de model în două niveluri în care fiecare client antrenează atât un model la nivel de grup, cât și modelul federat global, folosind o abordare de învățare mutuală [42]. Această lucrare a fost ulterior extinsă pentru a lua în considerare adaptabilitatea unor astfel de sisteme la adăugarea unui nou tip de senzor (grup) în federație; extinderea a fost trimisă simpozionului MobiUK. Învățarea mutuală a fost aleasă pentru a relaționa modelele la nivel de grup cu cel global, deoarece permite utilizarea de arhitecturi divergente ce împart doar stratul final. In ciuda succesului său, această metodă de antrenament necesită ca potențialii clienți să aibă o cantitate mare de date și resurse pentru a antrena modele. Natura costisitoare a procedurii a impus o mișcare spre o abordare bazată pe agregarea modelelor.

Ambele lucrări anterioare au fost implementate în cadrul framework-ului Flower [3]. Cu toate acestea, scara experimentării necesare pentru validarea completă a B-HFL ar fi nerealizabilă pe motorul de simulare public. Prin urmare, am contribuit la construirea unui nou motor ce dublează debitul simulărilor FL prin plasarea bazată pe ML a clienților pe GPU-uri. Lucrarea ce prezintă tehnicile noastre, pentru care împărtășesc un credit egal de contribuție ca autor principal, "High-throughput Simulation of Federated Learning via Resource-Aware Client Placement" a fost trimisă la conferința Mobicom și așteaptă răspunsul.

3 Revizuirea literaturii

Obiectivul standard FL poate fi modelat așa cum se vede în Eq. (1)

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

unde F este obiectivul federat, C este setul de clienți, θ este modelul și F_c este funcția de cost a clientului c ponderată de fracția lui p_c din numărul total de exemple. Această formulare presupune că este antrenat un singur model global fără a ține cont de distribuția performanței sale pe seturile de date ale clienților. Federated Averaging (FedAvg) [30] antrenează modelul global pe clienți, pentru fiecare rundă t sumează actualizarea $\theta_t^c - \theta_t$ de la clientul c ponderată de p_c cu modelul rundei anterioare θ_t folosind rata de învățare η, prezentat în Eq. (2)

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

Incapacitatea de a aduna informațiile clienților în același rezervor de date și necesitatea de a construi combinații aproximative de parametri ai modelului, reprezintă principalele cauze ale provocărilor din FL.

3.1 Eterogenitate

S-a demonstrat că datele Non-IID au un impact atât asupra performanței practice [43], cât și asupra limitelor teoretice de convergență [24]. Prin urmare, merită detaliate unele forme de eterogenitate pe care Kairouz et al. [16] le identifică. Cea mai frecvent abordată formă este quantity skew cauzată de faptul că clienții au cantități diferite de date disponibile. Algoritmii standard de FL tratează eficient quantity skew prin intermediul unei reponderări simple (Eq. (2)). Celălalt tip de eterogenitate frecvent considerat este asimetria distribuției etichetelor. Deși aceste forme de eterogenitate au fost cele mai investigate, situațiile în care caracteristicile și etichetele nu sunt relaționate în același mod între clienți sunt mai nefavorabile și pot necesita o formă de clustering sau personalizare pentru a fi abordate. În cel mai rău caz, fiecare client poate reprezenta un task complet diferit, similar cu Învățarea Automată Multi-task (referită ca MTL din termenul englez "Multi-task Learning"), cu posibil nicio suprapunere în spațiul de soluții al clienților.

Eterogenitatea sistemului (hardware) Dispozitivele din cadrul rețelei federate pot diferi în ceea ce privește capacitatea de calcul, stocarea, viteza rețelei și fiabilitatea. Ele se pot diferenția, de asemenea, de ele însele într-un alt moment de timp în funcție de puterea baterie sau conexiunea la rețea. Este important de menționat că variațiile în hardware-ul care generează date, cum ar fi senzorii, sunt legate de eterogenitatea lor. Cu toate acestea, eterogenitatea sistemului afectează procesul FL independent de date. De exemplu, hardware-ul mai lent poate duce la clienți care întârzie, care prelungesc rundele în FL sincron sau operează pe parametri învechiți în FL asincron. În plus, nesiguranța rețelei creează abandon, ceea ce necesită supraeșantionarea clienților [6] și afectează eficacitatea menținerii stării clienților între runde.

Shiftul temporal al setului de date Antrenarea eficientă a modelelor ML pe parcursul întregii vieți este obiectivul învățării continue [9]. Cu toate acestea, aplicarea învățării continue în contextul FL este problematică din două motive principale. În primul rând, obiectivul de optimizare (Eq. (1)) intenționează să găsească un model de compromis pentru toți clienții și nu se poate potrivi exact cu toate datele lor. Prin urmare, dacă setul de date al unui client se schimbă independent de întreaga rețea, modelul federat va avea dificultăți în a se adapta. În al doilea rând, tehnicile de învățare continuă, precum Elastic-weight Consolidation [21], sunt concepute pentru tipuri incrementale de MTL. In aceste tipuri de MTL, etichetele de task sunt cunoscute, cantități mici de date anterioare pot fi încă disponibile pentru cazuri specializate [21] și pot exista diferite straturi finale ale modelului pentru fiecare task. Cerințele de confidențialitate ale FL fac ca astfel de soluții să fie dificile fără adăugarea unei memorii locale persistente.

3.2 Eficiența Învățării Federate

Trendurile pe care Charles et al. [7] le-au descoperit au implicații majore asupra FL. Cele care limitează eficiența FL în setările Non-IID sunt de un interes deosebit. Se pot observa trei efecte semnificative. În primul rând, clienții foarte eterogeni pot provoca reduceri bruște ale acurateții când modelele lor sunt agregate. În al doilea rând, grupurile mai mari de clienți aduc îmbunătățiri diminuate pentru acuratețea finală și viteza de convergență a modelelor. În al treilea rând, grupurile mai mari scad eficiența sistemului, deoarece sunt necesare mai multe exemple pentru fiecare îmbunătățire a acurateții.

Aceste comportamente sunt analoge cu limitările bine-cunoscute de eficiență și generalizare a antrenamentului cu batch-size mari în ML [17]. Charles et al. [7] constată că problemele de eficiență sunt cauzate de scăderea mărimii pseudo-gradientului în mod proporțional cu creșterea dimensiunii grupului și de ortogonalitatea actualizărilor clienților. Autorii constată, de asemenea, că optimizatorii adaptivi sunt preferabili pe măsură ce dimensiunile grupurilor cresc datorită invarianței lor față de mărimea pseudo-gradientului.

3.2.1 Optimizarea Federată Adaptivă

De o relevanță specială pentru această propunere este Optimizarea Federată Adaptivă (FedOPT) [33]. FedOPT extind conceptul de optimizare adaptivă [19] la FL pe *latura serverului*, tratând actualizările clienților ca pseudo-gradienți și păstrând informații în acumulatori între runde. Această structură permite ca FedOpt să minimizeze impactul rundelor individuale, agregând pseudo-gradienții lor cu cei ai rundelor anterioare. Deoarece rezultatul rundelor individuale este variabil în funcție de combinația de clienți selectați și de starea curentă a modelului, asemenea tehnici oferă o traiectorie de optimizare mai consecventă.

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

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

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

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

Urmând formularea din Reddi et al. [33], precum în Eq. (3),pentru o anumită runda t și model federat θ_t, fiecare client c în setul selectat C antrenează local modelul pentru a construi o versiune personalizată θ_t^c. Pseudo-gradientul Δ_t este apoi calculat drept mediei diferențelor dintre modelele personalizate și cel federat, precum în Eq. (3a). Operațiile pe tensori sunt element cu element, inclusiv împărțirea între tensori.

Acumulatorul mediei m_t poate fi construit drept media ponderată a acumulatorului anterior m_{t-1} și Δ_t folosind ponderea β_1, așa cum se arată în Eq. (3b). În mod similar, pentru versiunea FedOpt bazată pe Adam [19], acumulatorul v_t urmărește puterea a doua a fiecărui element a pseudo-gradientului, denotată prin Δ_t^2, așa cum se arată în Eq. (3c). Aceste două acumulatoare sunt apoi folosite pentru a calcula modelul actualizat pentru runda următoare θ_{t+1}, folosind rata de învățare a serverului η, precum în Eq. (3d). De remarcat, termenul $\sqrt{v_t}$ se referă la rădăcina pătrată a fiecărui element. Termenul este utilizat pentru a face algoritmul invariant față de mărimea pseudo-gradientului. În final, τ controlează adaptivitatea FedOPT.

FedOPT prezintă mai multe proprietăți promițătoare în contextul FL ierarhic. În primul rând, Reddi et al. [33] arată că este foarte robust la alegerea exactă a hiperparametrilor, inclusiv a ratei de învățare, în comparație cu FedAvg. În al doilea rând, invarianța algoritmului abordează parțial problemele observate de Charles et al. [7] cauzate de pseudo-gradienții apropiați de zero. În al treilea rând, ei oferă un mijloc de a diferenția automat ratele de învățare ale mai multor servere în funcție de starea acumulatoarelor lor.

3.3 Lucrări corelate

Pentru a aborda compromisul dintre optimizarea pentru performanța globală și pentru performanța pe datele unui client specific, ce poate fi observat în Eq. (1), două direcții generale au apărut în literatura științifică. Prima, exemplificată de "Fair" FL [22], încearcă să modifice importanța unui client în funcția obiectiv federată pentru a schimba eficiența modelului final pentru acel client. A doua relaxează cerința unui singur model global prin: personalizarea modelului federat [41, 43], menținerea unui model local persistent [23], gruparea clienților pe baza similitudinii [29] și construirea ierarhiilor [28, 1]. Deoarece familia propusă de algoritmi B-HFL se încadrează în a doua categorie, această secțiune va detalia lucrările strâns legate și limitările acestora. În final, proprietățile dorite ale B-HFL sunt rezumate în Tabelul 1.

3.3.1 Învățare Federată Personalizată

Învățarea Federată complet personalizată creează un model adițional pentru fiecare client. Cea mai comună metodă este personalizarea prin adaptare locală (fine-tuning), a modelului federat după antrenament [41], cu posibile tehnici precum Învățare Mutuală [42] sau Elastic-weight Consolidation [21]. Cu toate acestea, optimizarea în două etape este dificil de implementat într-un ciclu de viață FL unde modelul federat poate necesita un antrenament suplimentar ulterior adaptării. Mai mult, nu oferă un punct de mijloc între modelele globale și cele locale, ceea ce afectează capacitatea unor astfel de sisteme de a integra noi clienți.

O abordare mai recentă este reprezentată de Ditto [23] pentru setări în care clienții sunt vizitați frecvent și pot menține starea între runde. Ditto permite clienților să mențină un model local persistent și să-l antreneze alături de cel federat în timpul rundelor FL. Cele două modele sunt conectate prin includerea distanței l_2 dintre parametrii lor în funcția de cost a modelului local. Cu toate acestea, în ciuda beneficiilor dovedite de "fairness" și "robustness", Ditto nu abordează modificările setului de date în cadrul clientului, deoarece modelele funcționează numai în timpul rundelor de antrenament.

3.3.2 Învățare Federată Ierarhică și Clustering

Subdomeniul FL cel mai relevant pentru propunerea aceasta este Învățarea Federată Ierarhică (referită ca HFL din termenul englez "Hierarchical Federated Learning") introdusă de Liu et al. [28]. Algoritmul propus (HierFAVG) a fost dezvoltat în principal pentru a gestiona provocările de comunicare ale sistemelor FL anterioare. Pentru a gestiona milioane de clienți participanți [11, 6], sistemele FL se bazau pe infrastructura

cloud, având viteze de comunicare reduse, pentru a conecta dispozitivele pe o zonă geografică largă. Acest compromis a fost dorit, deoarece populațiile mari erau necesare pentru convergență, iar serverele edge, deși capabile de comunicare rapidă cu clientul, nu puteau să obțină suficienți clienți. Liu et al. [28] susțin că o structură pe două niveluri rezolvă tensiunile între serverele edge și serverele cloud. Abad et al. [1] propun un algoritm identic pentru rețelele celulare eterogene. Similar cu HierFAVG [28], Abad et al. [1] se axează pe reducerea costurilor de comunicare utilizând tehnici de compresare a pseudo-gradienților [27].

Clusterizarea clienților este o tehnică sinergică ce încearcă să grupeze participanții pe baza unei metrice de similaritate. Aceste clustere sunt construite folosind abordări precum clusterizarea directă a parametrilor modelului [32] sau utilizarea valorii funcției de cost a clienților atunci când sunt atribuiți unui anumit cluster [29]. Clusterele pot exista și natural pe baza unor caracteristici precum locația geografică sau limba.

Lucrările anterioare în HFL arată o serie de limitări. Algoritmul HierFAVG extinde direct FedAvg [30] permițând serverului cloud să trateze serverele edge ca pe clienți. Cu toate acestea, deoarece Liu et al. [28] și Abad et al. [1] iau în considerare doar eficiența comunicației, ei nu permit serverelor edge să mențină o personalizare mai mare și, în schimb, le înlocuiesc complet modelul în timpul agregării în cloud. În plus, sistemul lor nu ia în considerare asincronia, antrenamentul pe seturi de date proxy sau ierarhiile multi-nivel. În ceea ce privește clusterizarea, algoritmii disponibili nu reușesc să obțină compromisul dorit între generalizare și personalizare. Algoritmii de clusterizare standard în FL presupun comunicarea parametrilor între clustere a fi inutilă și nu se mapează direct pe o structură de comunicare ierarhică. Mai mult de atât, acești algoritmi nu sunt meniți să ofere un singur model global în afara modelelor de clustere.

Table 1: Tabel de analiză ce arată proprietățile sistemului propus și suprapunerea cu lucrările corelate.

Lucrări Corelate	Structură Ierarhică	Personalizare	Permite Modele Persistente	Modele Generale de Grup	Modele Semnificative de Grup	Antrenament Asincron
Adaptare Locală		✓				
Ditto		✓	✓			
Clustering						
HieFAVG	✓			✓	✓	
FL Asincron						✓
Învățare Federată Ierarhică Bidirecțională	✓	✓	✓	✓	✓	✓

4 Propunere

Având în vedere limitările tradiționale ale sistemelor HFL, acest studiu propune Învățare Federată Ierarhică Bidirecțională (B-HFL), o familie alternativă de metode care optimizează eficiența datelor și a comunicațiilor. Acest lucru se realizează prin utilizarea structurii ierarhice pentru a organiza comunicarea între servere și pentru a controla diseminarea parametrilor prin următoarele alegeri de design:

1. În timp ce metodele anterioare, cum ar fi HierFAVG [28, 1], înlocuiesc complet modelele edge-server și ale clienților după ce are loc agregarea globală, B-HFL realizează agregarea parțială între un nod copil și părintele său, ceea ce permite copiilor să-și mențină parametrii locali în timp ce încorporează informații globale. Propun modelarea acestei proceduri în două faze:
   (a) Agregare de la frunze către rădăcină: clienții finalizează antrenamentul, iar informațiile lor sunt propagate înspre rădăcina arborelui. Fiecare nod intern are un parametru T_n, care determină după câte runde trimite actualizările către părinte. Această valoare este echivalentă cu epocile locale ale clientului și poate fi: aceeași pentru toate nodurile, aceeași pentru toate nodurile de la un anumit nivel al arborelui sau setată independent pentru fiecare nod.
   (b) Agregare de la rădăcină către frunze: După ce un nod a primit și a agregat rezultatul antrenamentului de la unii sau toți copiii săi, își propagă parametrii în josul subarborelui său. Costul acestei propagări este proporțional cu adâncimea subarborelui. Cu toate acestea, conexiunea dintre nodurile interne este mai rapidă decât cea a clienților către serverele edge.
2. Nodurile interne din cadrul structurii ierarhice pot fi antrenate pe seturi de date proxy pentru regularizarea antrenamentului [10, 43]. Antrenamentul proxy este în mod special relevant pentru modelarea limbajului, deoarece sunt disponibile corpuri textuale publice de dimensiuni considerabile. Pentru a evita operarea pe parametrii învechiți, momentul natural pentru a adăuga un astfel de antrenament este imediat după ce agregarea de la frunză către rădăcină ajunge la nod. Cu toate acestea, latența rezultată dintr-un astfel de antrenament poate fi prea mare. În acest caz, antrenamentul poate utiliza asincron parametrii învechiți în timp ce subarborii nodului se execută.
3. Toate nodurile pot fi lăsate să funcționeze sincron sau asincron în ceea ce privește alte noduri de pe același nivel, dacă este necesar, în timpul agregării de la frunze către rădăcină. Pentru frunzele (clienții) sub controlul unui server edge, acest lucru este echivalent cu FL asincron tradițional [38]. Pentru un nod intern, aceleași strategii federate asincrone [31, 13] pot fi aplicate atunci când primesc modele de la nodurile copil, execuția clientului fiind înlocuită de execuția subarborelui.

Parametrii agregați de la nodurile frunze (clienți) prin arbore sunt antrenați fin la datele locale relevante. În contrast, parametrii transmiși de la părinți la copii sunt agregați peste populații mai numeroase. Când serverele cuprind clienți grupați semnificativ, aceste populații numeroase pot fi mai puțin legate (de exemplu, conținând mai multe limbi). În plus, dacă nodurile interne sunt lăsate să se antreneze pe seturi de date proxy, ele injectează antrenament suplimentar în modelele federate și oferă regularizare pentru întregul arbore. În abordările FL tradiționale, antrenamentul pe serverul care controlează direct clienții poate impune o regularizare prea puternică. Totuși, în B-HFL, nodurile superioare din arbore reprezintă deja o imagine globală și au un impact limitat asupra frunzelor, deoarece influența lor se diluează prin intermediul mai multor noduri intermediare. În final, clienții sunt tratați omogen cu orice alt nod, deoarece mențin un model persistent local pe parcursul rundelor și fiecare client este agregat doar parțial cu părintele.

Deoarece nu toate nodurile din arbore sunt obligate să fie capabile de antrenament, merită distinse modelele care au fost optimizate prin antrenare suplimentară, în loc de simplă agregare. În mod specific, disponibilitatea datelor de antrenament poate permite metode de agregare mai eficiente, cum ar fi învățarea mutuală [42] sau regularizarea bazată pe l_2 [23]. În plus, actualizările construite prin antrenament direct pot oferi un semnal de optimizare mai bun. Astfel, această lucrare propune adăugarea de fluxuri de date directe între nodurile capabile de antrenament (de exemplu, clienți și rădăcină) în timp ce se folosește structura de comunicare ierarhică subiacentă, ca o conexiune reziduală în ResNet [12]. De exemplu, sistemul ar putea permite ca actualizările cu cea mai mare valoare absolută a K clienți de la fiecare server să fie transmise către rădacină, unde ele pot fi unite fie prin antrenament, fie prin optimizare adaptivă. Acest tip de conexiune verticală oferă un flux de date foarte dinamic și potențial ciclic. O altă cale ce merită explorată este permiterea nodurilor, în special a clienților, de a se antrena în mod asincron folosindu-și modelul persistent. Acest lucru ar permite clienților să țină cont de schimbarea setului de date local, folosind tehnici bine-cunoscute din literatura privind învățarea continuă [9, 21]. Sistemul poate aduce multiple beneficii potențiale:

1. Poate găzdui noduri care au metode diferite de agregare, rate de învățare, stări dinamice ale optimizatorului pentru agregarea de la frunză către rădăcină și de la rădăcină către frunză. În mod similar numărului de runde T, parametrii legați de agregare pot fi independenți sau omogeni.
2. Grupurile mai mici pentru fiecare server edge evită problema scăderii mărimii pseudo-gradientului [7]. Un efect similar este generat de clusterizarea clienților pentru serverele edge.
3. În timp ce modelele locale persistente sunt cunoscute pentru funcționarea lor eficientă în FL cross-silo, această structură ierarhică le face relevante în setările cross-device prin posibilitatea de a selecta un număr mai mare de clienți, permițându-le să fie vizitați în mod repetat.
4. Poate integra în mod natural Secure Aggregation (SecAgg) [5] la nivelul fiecărui server edge. Precum menționat de Bonawitz et al. [6], ierarhizarea reduce costul de comunicare suplimentar al antrenării a C clienți cu SecAgg de la $\mathcal{O}(C^2)$ la $\mathcal{O}(C^2/M)$ unde M este numărul de servere edge.

4.1 Exemplu de sistem

Un exemplu de sistem B-HFL, care ar fi principalul rezultat al acestei propuneri, poate fi văzut în Fig. 1. Serverul central controlează un set de date proxy folosit pentru antrenament post-agregare. Toate serverele își trimit actualizările către părinte după fiecare runda. Fiecare nod, inclusiv clienții, rulează cel puțin două optimizatoare FedOPT cu stări separate și rate de învățare, una pentru agregarea de la frunză către rădăcină cu pseudo-gradientul mediu Δ_t și una pentru agregarea părinților. Chiar dacă aceeași rată de învățare de la frunză la rădăcină $\eta^{\uparrow}$ și rata de învățare de la rădăcină la frunză $\eta^{\downarrow}$ ar fi utilizate pentru toate nodurile, stările independente ale optimizatorului serverului ar distinge procedura de agregare a nodului.

Conexiunile reziduale îndeplinesc funcții diferite între etapele de la frunză la rădăcină și de la rădăcină la frunză. Pentru etapa ascendentă, ele colectează actualizarea clientului cu valoarea absolută cea mai mare de la toate serverele edge, trimițând astfel un model suplimentar la serverul central pentru fiecare server edge. Serverul central poate apoi menține stări independente ale optimizatorului pentru fiecare conexiune "reziduală" de intrare. Pentru etapa descendentă, ele oferă serverelor edge șansa de a beneficia direct de antrenamentul serverului central fără a fi nevoie să se bazeze pe modelele diluate ale celor intermediare.

5 Plan

Familia de algoritmi prezentată pentru Învățarea Federată Ierarhică Bidirecțională va fi dezvoltată în perioada doctoratului și va face parte din teza finală de doctorat. În plus, înainte de teza finală, oferă oportunități pentru publicații la conferințe ce contribuie semnificativ la FL. Având în vedere noutatea FL

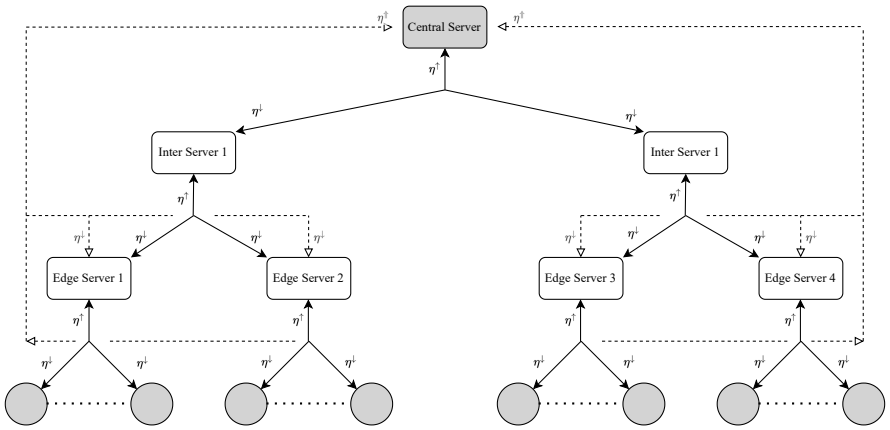

Figure 1: Diagramă a unui exemplu de sistem B-HFL. Liniile solide reprezintă legături de comunicare, în timp ce liniile punctate reprezintă conexiuni "reziduale" conceptuale, folosind legăturile de bază. Nodurile capabile de antrenament, cum ar fi clienții sau serverul central cu un set de date proxy, sunt în gri. Când parametrii modelului se propagă în sus, nodurile combină pseudo-gradienții de intrare și își actualizează modelul, folosind rata de învățare de la frunză către rădăcină $\eta^{\uparrow}$. La fel se întâmplă când parametrii curg de la nodurile părinte către nodurile copil cu rata de învățare $\eta^{\downarrow}$. Deoarece liniile punctate comunică de la 0 la K modele, $\eta^{\Uparrow}$ poate reprezenta de la 0 la K agregări, folosind o rată de învățare $\eta^{\uparrow}$.

în general și a HFL în particular, există un spațiu larg pentru dezvoltări ulterioare în structura B-HFL pe măsură ce domeniile se maturizează. Perioada de vară de la sfârșitul primului meu an de doctorat va fi dedicată implementării versiunii exemplu a B-HFL în cadrul framework-ului FL Flower [3] afiliat grupului nostru de cercetare. Acest framework este în prezent reglat pentru setările standard FL și ar necesita modificări importante ale API-ului pentru a executa și simula eficient HFL. Lucrările anterioare privind modelele la nivel de grup pentru Recunoașterea Activității Umane Federate ale Iacob et al. [15] și motorul eficient de simulare FL la care am contribuit pot fi baza pentru implementarea sistemului.

Semestrul de toamnă, Michaelmas, al celui de-al doilea an va avea ca obiectiv principal publicarea unui articol de conferință bazat pe sistemul exemplu propus în Secțiunea 4.1. Am discutat deja acest lucru cu supervizorul meu, Dr. Nicholas Lane, și am convenit că atât ICLR cât și MLSys ar fi conferințe potrivite. Având în vedere importanța crescândă a LLM-urilor și compromisurile recent descoperite de Agarwal et al. [2] în ceea ce privește abilitățile lor de generalizare și personalizare, ele reprezintă o aplicație naturală pentru sistemul ierarhic propus. Mai mult, predicția textului în mai multe limbi oferă o aplicație FL, grupată în mod natural, corespunzătoare scenariilor din lumea reală în care țările au servere independente edge pentru FL și trebuie să colaboreze la un nivel continental și global. Studiul ar folosi un model BERT multilingv mare [8] împreună cu două seturi de date multilingve [e.g., 25, 39] pentru antrenament. Un set de date va fi împărțit după limbă, iar celălalt va fi păstrat ca un set de date proxy la serverul central din Fig. 1. Obiectivele studiului ar fi să compare acuratețea finală a fiecărui model la fiecare nivel al ierarhiei pe seturile de test ale clienților și setul de test centralizat partiționat din setul de date proxy inițial. Așteptarea ar fi ca performanța modelului pe datele unui anumit client să fie proporțională cu proximitatea acestuia față de acel client în arbore. Alternativ, pentru setul de test proxy și uniunea tuturor seturilor de test ale clienților, acuratețea ar trebui să fie proporțională cu proximitatea față de serverul central. În plus, studii de ablație privind conexiunile "reziduale" sau optimizarea adaptivă vor fi efectuate. În cele din urmă, dacă timpul permite, lucrarea poate include alte sarcini grupate natural, cum ar fi recunoașterea vorbirii.

După publicarea acestei lucrări, o extensie naturală în semestrele Lent și Easter ar fi abordarea unui mediu în care clienții generează și șterg continuu date având memorie limitată. Sistemul exemplu ar fi extins pentru a permite antrenament asincron pe toate nodurile, inclusiv frunzele, care rulează în paralel cu componenta FL reală. Fiecare client ar genera un flux de date având o memorie internă fixă pe care să opereze în timpul antrenamentului. Constrângerile reale de resurse și asincronia pot fi modelate folosind clusterul Raspberry Pi FL de la Cambridge ML Systems. Această lucrare ar fi destinată pentru MobiCom, același loc unde am prezentat motorul de simulare Flower.

Dacă propunerea este reușită, cel de-al doilea an al doctoratului meu ar aduce o contribuție valoroasă în domeniul FL și s-ar concretiza în una sau mai multe publicații la conferințe împreuna cu o parte din teza finală. De asemenea, ar reprezenta o extensie majoră a framework-ului Flower [3] cu potențial pentru viitoare colaborări sau angajări în startup-ul Flower Labs finanțat de Y Combinator. După finalizarea doctoratului, intenționez să urmez o carieră în cercetarea privată sau academică.

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. p.5, p.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. p.2, p.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. p.1, p.3, p.8

[4] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ B. Altman, Simran Arora, and et al. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258, 2021. URL https://arxiv.org/abs/2108.07258. p.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. p.7

[6] Kallista A. Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloé Kiddon, Jakub Konečný, Stefano Mazzocchi, Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. Towards federated learning at scale: System design. In Ameet Talwalkar, Virginia Smith, and Matei Zaharia, editors, *Proceedings of Machine Learning and Systems 2019, MLSys 2019, Stanford, CA, USA, March 31 - April 2, 2019*. mlsys.org, 2019. URL https://proceedings.mlsys.org/book/271.pdf. p.1, p.2, p.4, p.5, p.7

[7] 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. p.2, p.4, p.5, p.7

[8] 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. p.8

[9] 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. p.4, p.7

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

[11] 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. p.1, p.2, p.5

[12] 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. p.2, p.7

[13] 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. p.1, p.2, p.6

[14] 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. p.2, p.3

[15] 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. p.2, p.3, p.8

[16] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, and et al. 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. p.1, p.2, p.4

[17] 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. p.4

[18] 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. p.2

[19] 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. p.4, p.5

[20] 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. p.2

[21] 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. p.4, p.5, p.7

[22] 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. p.2, p.5

[23] 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. p.2, p.5, p.7

[24] 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. p.4

[25] 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. p.8

[26] 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. p.1

[27] 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. p.6

[28] 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. p.5, p.6

[29] 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. p.5, p.6

[30] 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. p.1, p.2, p.3, p.6

[31] 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. p.1, p.2, p.6

[32] 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. p.1, p.6

[33] 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. p.2, p.4, p.5

[34] 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. p.1

[35] 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. p.1, p.3

[36] 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. p.1

[37] 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. p.1

[38] 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. p.2, p.6

[39] 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. p.8

[40] 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. p.1

[41] 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. p.5

[42] 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. p.2, p.3, p.5, p.7

[43] 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. p.4, p.5, p.6