DEIM2022 J34-4(day3 p38)

モデル構造の自動チューニングを用いたパーソナライズド連合学習手法

松田　光司†　　佐々木勇和†　　肖　　川†　　鬼塚　　真†

† 大阪大学大学院情報科学研究科
E-mail: †{matsuda.koji,sasaki,chuanx,onizuka}@ist.osaka-u.ac.jp

あらまし　連合学習は，サーバと複数のクライアントがデータセットを共有することなく，協同して機械学習モデルを構築する分散型の機械学習手法である．これまで，連合学習におけるデータ不均一性の問題に対処するために多くの手法が提案されてきた．既存手法では，サーバがモデル構造をチューニングする必要があるが，サーバはローカルデータを保持していないため，モデル構造のチューニングが困難である．本研究では，我々の先行研究である Federated Learning via Model exchange (FedMe) を拡張し，モデル構造のチューニングを可能とする．FedMe は，クライアントがモデルを互いに交換し合い，モデル同士を深層相互学習によって学習することで異種モデル構造間の学習を可能にする技術である．さらに，本拡張によって，モデルパラメータを学習しながらモデル構造を自動的にチューニングし，クライアント毎にモデルを構築する．ローカルデータに対してモデル構造を最適化するために，各クライアントは交換モデルと比較しながら自身のパーソナライズモデルのチューニングを行う．我々は 3 つの実データセットで実験を行い，FedMe がモデル構造を自動的にチューニングしながら，最新の連合学習手法よりも高精度であることを示す．

キーワード　連合学習，深層学習，エッジコンピューティング，IoT，深層相互学習

1　はじめに

スマートフォンやタブレット端末などのモバイル端末の普及に伴い，かつてないほど大量のデータが生成されている．これらのデータは，行動認識 [1]，次単語予測 [2]，ウェイクワード検出 [3] など様々なアプリケーションで機械学習モデルの構築に役立っている．しかしデータのプライバシーやネットワーク帯域の制限のため．クライアントからすべてのローカルデータを収集しサーバが集中的にモデルを学習することは現実的ではない．このようなプライバシーやネットワーク帯域の問題を解決するために，クライアントのローカルデータを共有せずにモデルを構築する分散型学習手法として連合学習が提案された [4]．

連合学習は，サーバと複数のクライアントが協同してモデルを構築する．連合学習の一般的な学習手順は（1）クライアントが自分のローカルデータに対してモデルを学習し学習後のモデルをサーバに送信するクライアント学習，(2）サーバがそれらのモデルを集約してグローバルモデルを構築するモデル集約という 2 つのステップから構成されている．この 2 つのステップを，グローバルモデルが収束するまで繰り返す．このように学習後のモデルを共有することで，クライアントのローカルデータをモデルの構築に有効的に活用することができる．

連合学習の課題　連合学習の課題の 1 つにデータの不均一性がある．データの不均一性とは，クライアント間でローカルデータの分布が異なる，すなわち独立同分布（IID）でないという課題である．データが不均一な環境では，各クライアントに対して最適な単一のグローバルモデルを学習することは困難である．実際，一般的な連合学習手法では，各クライアントが Non-IID のローカルデータを持つ場合，グローバルモデルのモデルパラメータが発散することが示されている [5]．データの不均一性に対処するために，パーソナライズド連合学習手法が提案されている [6–9]．これらの手法は，クライアント毎に最適化されたモデルであるパーソナライズドモデルを構築することを目的としている．

我々は最適なパーソナライズドモデルの構築のために，以下の研究課題を有している．

- **パーソナライズドモデルのモデル構造をどのように決定するか？**　既存のパーソナライズド連合学習手法では，サーバは事前にモデル構造をチューニングする必要がある．サーバはクライアントのローカルデータの分布を知らないため，構造の異なる複数のモデルを学習させ，モデル構造をチューニングする必要がある．しかし，この作業はサーバとクライアント間の通信コストが高く，実際に行うことが不可能である．近年，学習中にモデル構造を自動的にチューニングする自動モデル構造チューニング手法が提案された [10]．この手法は，サーバによって単一のグローバルモデルの構造をチューニングするものである．しかし，サーバによってチューニングされたモデル構造は，各クライアントにとって最適でない可能性が高い．また，サーバはチューニングしたモデルの精度をクライアントのローカルデータを用いて評価することができない．そのため，各クライアントは個別にモデル構造をチューニングする必要があり，データの不均一性によりクライアント間で最適なモデル構造が異なる可能性がある（実験研究の表 3 参照）．我々の知る限り，学習過程においてモデル構造を自動的にチューニングすることができるパーソナライズド連合学習手法は存在しない．各クライアントは他のクライアントのローカルデータを知らないため，モデル構造をチューニングするために他のクライアントのモデルを利用する学習手順が必要である．
- **各クライアントはどのようにして構造の異なる他のクライアントのモデルを利用し，モデルの精度を向上させるか？**　モ

デル構造はクライアント間で異なる可能性があるため，サーバは異なるパーソナライズドモデル間の集約を行うことができない場合がある．そのため，他のクライアントのモデルを利用するためにモデル集約に依存することは効果的ではない．そこで，ローカルデータと自身と異なる構造のモデルの両方を活用するための新たな学習手順が必要である．

貢献点　本稿では，我々の先行研究である Federated Learning via Model exchange (FedMe) [11] を拡張し，モデル構造のチューニングを可能とする．我々は，各クライアントが他のクライアントから受け取ったモデルである，交換モデルという概念を導入する．FedMe では，クライアントは交換モデルを利用してモデル構造のチューニングとモデルの学習を行うことができる．FedMe の学習手順は前述の研究課題を解決する．まず，クライアントは交換モデルの性能に基づいて自身のモデル構造をチューニングする．ローカルデータに対してモデル構造を最適化するために，各クライアントは自身のパーソナライズドモデルと交換モデルを比較し，最も良い性能であるモデル構造を選択する．このようにして，クライアントはモデル構造を自動的かつ匿名的に最適化することができる．次に，クライアントは自身のモデルと交換モデルの両方を学習する．そして，サーバは同じクライアントの学習後のモデルを集約する．モデル学習には，深層相互学習 [12] とモデルクラスタリングの 2 つの方法を用いる．深層相互学習は，お互いのモデルの出力を模倣するように学習することで，モデルの構造によらず 2 つのモデルを効果的に学習する．モデルクラスタリングでは，パーソナライズドモデルと類似している他のモデルを交換モデルとして選択することで，深層相互学習によってモデルがノイズにオーバーフィットするのを防ぐ．他のクライアントのローカルデータと交換モデルを用いて学習されるため，他のクライアントのローカルデータを集約後のモデルに反映させることができる．

我々は 3 つの実データセットを用いて，最先端手法と比較することで FedMe の性能を評価した．評価実験から，最先端手法において最適なモデル構造を手動でチューニングした場合でも，FedMe が最先端の手法より高い精度を達成することが示された．また，興味深いことに，従来の連合学習手法に fine-tuning を施すことで高精度なパーソナライズドモデルを構築できることも実験によって示された．我々は，今後の研究のためにソースコード（既存の手法を含む）を公開している [1]．

本稿の構成　本稿の構成は以下の通りである．2 章にて関連研究について説明し，3 章にて問題定義について説明する．4 章にて FedMe について説明し，5 章にて評価実験の結果を示す．6 章にて本稿をまとめ，今後の課題について論ずる．

2 関 連 研 究

連合学習に関する研究は McMahan らが連合学習を考案 [4] して以来活発に行われており，いくつかの論文では連合学習に関する研究がまとめられている [13,14]．

近年，多くの連合学習手法が提案されているが，本項ではページの都合上，代表的な手法のみを紹介する．ここでは，(1) データの不均一性，(2) パーソナライゼーション，(3) モデル構造のチューニングの 3 つの観点から連合学習手法を紹介する．データの不均一性のための手法は，クライアントが Non-IID なローカルデータを持つ環境において，適切にモデルを構築することを目的とする．パーソナライゼーションの手法は，クライアント毎に最適なパーソナライズドモデルを構築することを目的としている．パーソナライゼーションは，すべてのパーソナライズドモデルのモデル構造が同じである場合と異なる場合に分けることができる．モデル構造チューニングを用いた手法は，モデル構造を自動的にチューニングすることを目的としている．

連合学習で最も基本的な手法は FedAvg [4] である．FedAvg ではクライアントの学習後のモデルをモデルパラメータの平均化により集約し，単一のグローバルモデルを構築する．データが不均一な環境では FedAvg の精度は低下するため，FedMA [15] や HierFAVG [16] など，FedAvg を発展させた手法がいくつか考案されている．これらの手法は学習後のモデルを集約して 1 つのグローバルモデルを構築するが，単一のモデルだけでは高い精度を達成することは困難である．

クライアントごとに異なるモデルを構築するパーソナライズド連合学習手法もいくつか提案されている [6,8,9]．パーソナライズド連合学習手法は，単一のグローバルモデルを構築する方法と比較して精度を向上させることができる．まず，クライアント毎に異なるモデルパラメータで同じモデル構造を持つモデルを構築するパーソナライズド連合学習手法について説明する．Mansour らは HypCluster と MAPPER を提案した [6]．HypCluster では，サーバは複数のグローバルモデルを作成し，クライアントは自身のローカルデータ上で最も精度の高いモデルのみを学習する．MAPPER では，クライアントはグローバルモデルとパーソナライズドモデルを学習し，それらのパラメータを加重平均する．T. Dinh らは，Moreau 包絡を用いた正規化によりグローバルモデルとパーソナライズドモデルを構築する pFedMe [8] を提案した．これらの手法では，すべてのパーソナライズドモデルが同じモデル構造である必要があるため，クライアント毎に異なるモデル構造を設計することができない．また，これらの手法は fine-tuning を行わず（すなわちモデルの構築後，クライアントはローカルデータ上でモデルを再学習しない），FedAvg のような単純な手法で fine-tuning を行う場合との比較を行っていないことに注意されたい．我々の実験では，ほとんどの手法が FedAvg で fine-tuning を行った場合より精度が低いことを示した．

クライアントごとに異なるパラメータと構造を持ったパーソナライズドモデルを構築するパーソナライズド連合学習手法も存在する [7,17]．クライアントはローカルデータのサイズや自身の計算資源に応じて任意のモデル構造を選択することができる．Shen らは Federated Mutual Learning (FML) [7] を提案した．FML では，サーバがグローバルモデルを配布し，クライアントはグローバルモデルとパーソナライズドモデルの両方を深層相互学習により学習する．我々はクライアントの学習に FML と同様の考え方を用いるが，FedMe はグローバルモデル

1：https://github.com/OnizukaLab/FedMe

を構築しない．Li らは Federated Learning に知識蒸留を取り入れた FedMD [17] を提案した．FedMD は，すべてのサーバとクライアントが利用することができるパブリックデータを必要とする．これらの手法では，クライアント毎に異なるモデル構造を設計することができるが．サーバとクライアントは学習前にモデルの構造を決定する必要がある．また我々の実験では，これらの方法が非パーソナライズド連合学習手法よりも低い精度であることが示されている．

モデル構造探索は深層学習分野で注目されており，あらかじめ定義された探索空間（例：層の種類や最大層数）の中から最適なモデル構造を探索する [18]．FedNAS [10] は，連合学習におけるモデル構造の探索を行う．FedNAS はモデル構造を自動的にチューニングするが，単一のグローバルモデルを構築することを目的としており，パーソナライズドモデルの構築は目的としていない．以上より，我々の手法である FedMe はデータの不均一性，クライアント毎にパラメータと構造が異なるモデルの構築，そしてモデル構造の自動チューニングの全てを満たすことができる初めての手法である．

3 事 前 知 識

この章では，問題定義について説明する．本項で使用する表記法を表 1 にまとめた．

分類タスクが与えられたとき，サーバと複数のクライアントが協同してクライアントのパーソナライズドモデルを構築する．S をクライアントの集合とし，クライアントの数を $|S|$ で表す．添え字 i を i 番目のクライアントのインデックスとする．例えば，D_i はクライアント i のローカルデータであり，n_i は D_i のサイズ（すなわちレコードの数）である．N は全クライアントの n_i の総和である．x と y はそれぞれローカルデータに含まれるレコードの特徴量とラベルである．ここでは分類タスクを想定しているので，y には M 個のクラスの中から 1 つのクラスが割り当てられる． T と E はそれぞれグローバル通信ラウンドとローカル学習エポックの総数である．グローバル通信とは，学習時のサーバとクライアント間の通信を意味する．ローカル学習とは，クライアントが自身のローカルデータ上でモデルを学習することを意味する．t はグローバル通信ラウンドのインデックスである．$w_{p_i}^t$ と $w_{ex_i}^t$ は，それぞれラウンド t におけるクライアント i のパーソナライズドモデルと交換モデルである．$Idx(w_{ex_i}^t)$ は $w_{ex_i}^t$ の元のクライアントのインデックスである． 例えば $Idx(w_{ex_i}^t)$ は，$w_{ex_i}^t$ がクライアント j のパーソナライズドモデルであれば，j を返す．

FedMe では単一のグローバルモデルではなく，各クライアントがパーソナライズドモデルを構築する．我々は最適化問題を以下のように定義する．

$$\{w_{p_1}, \ldots, w_{p_{|S|}}\} = \text{argmin} \sum_{i \in |S|} \mathcal{T}_i(w_{p_i}). \qquad (1)$$

$\mathcal{T}_i$ はクライアント i の目的関数であり，以下のように定義する．

$$\mathcal{T}_i = min\ \mathcal{L}(w_{p_i}, D_i), \qquad (2)$$

表 1: 本稿で使用する表記

記号	説明
S	全クライアントの集合
i	クライアントのインデックス
D_i	クライアント i のローカルデータ
n_i	D_i のサイズ
x_i, y_i	それぞれ D_i に含まれるサンプルの特徴量とラベル
M	ラベルのクラス数
T	グローバル通信ラウンド数
t	グローバル通信ラウンド数のインデックス
E	ローカル学習エポック数
$w_{p_i}^t$	ラウンド t におけるクライアント i のパーソナライズドモデル
$w_{ex_i}^t$	ラウンド t におけるクライアント i の交換モデル
$Idx(w_{ex_i}^t)$	$w_{ex_i}^t$ の元のクライアントのインデックス
C_k	クラスタ k に割り当てられた全モデル
K^t	ラウンド t のクラスタ数

$\mathcal{L}(w_{p_i}, D_i) : \Theta \longmapsto \mathbb{R}$ はクライアント i の損失関数を表し，D_i と w_{p_i} に対応する．Θ はモデル空間を表しており，固定ではない．これは [9] の最適化問題と類似している．[9] では，パーソナライズドモデルのモデル構造が固定されているため，w_{p_i} のサイズは事前に決定されており固定値である．これに対し，本稿の最適化問題では w_{p_i} のサイズも最適化するため，パーソナライズドモデルのモデル構造を最適化することを目的としている．この最適化問題を解くことで，クライアント毎に最適なパーソナライズドモデルを構築することができる．

4 FedMe

本章では拡張した FedMe について説明する．主として異なる点はモデルチューニングである．

4.1 アイデアとフレームワーク

我々の研究課題は，クライアント毎に最適なモデル構造を自動的にチューニングする方法と，クライアントが異なる構造のモデルを用いて自身のモデルを更新する方法である．我々の先行研究である FedMe [11] は，モデル交換によってクライアント毎に異なる構造のモデルを保持可能なパーソナライズド連合学習手法である．そこで，FedMe を拡張することで前述の研究課題を解決する．FedMe を拡張しモデルチューニングを行うためのアイデアは簡潔である．クライアントは他のクライアントから受け取った交換モデルを効果的な効果的なモデルの学習のためだけでなく，モデル構造のチューニングにも利用する．つまり，クライアントはモデルの学習とモデル構造のチューニングのためにモデルを交換する．

FedMe は拡張によって，次のような方法で交換モデルを効果的に利用する．第一に，クライアントは交換モデルの性能に基づいて，自身のパーソナライズドモデルをチューニングする．具体的には，クライアントは交換モデルの方がローカルデータに対する損失が小さい場合，自身のパーソナライズドモデルを交換モデルに置き換える．こうすることで，各クライアントはローカルデータに対する精度が向上するように，モデル構造を自動的にチューニングすることができる．第二に，クラ

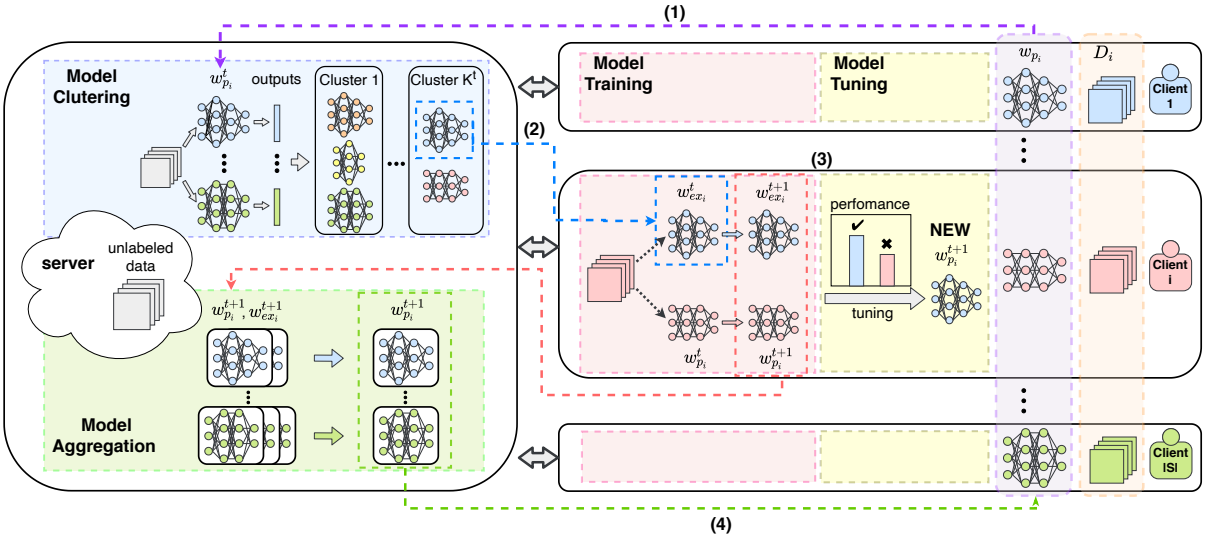


図 1: FedMe のフレームワーク [19]

イアントはパーソナライズドモデルと交換モデルの両方を学習し，サーバは同じクライアントの学習後のモデルを集約する．こ右することで，クライアント毎にパーソナライズドモデルの構造が異なっていても，モデルを学習することができる．第三に，クライアントは深層相互学習とモデルクラスタリングにより，パーソナライズドモデルと交換モデルの両方を効果的に学習する．深層相互学習によって，モデル構造に依存せず，互いのモデルの出力を模倣しながら 2 つのモデルを学習することができる．このとき，分布の異なるローカルデータ上でモデルが学習された場合，他のモデルの出力はノイズとなり，ノイズにオーバーフィットする可能性がある [20]．このようなノイズへのオーバーフィットを防ぐために，サーバはモデルクラスタリングを行い，類似した出力を持つモデルを交換モデルとして選択する．モデルクラスタリングは，Kmeans [21] を用いて，類似の出力を持つモデル同士でグループ化する．

図 1 に FedMe のフレームワークを示す．FedMe は，5 つの学習手順によってモデルを構築する．(0) 各クライアントは任意のモデル構造でパーソナライズドモデルを初期化し，(1) クライアントはパーソナライズドモデルをサーバに送信する．(2) サーバはモデルクラスタリングの結果に基づいて各クライアントの交換モデルを決定し，交換モデルをクライアントに送信する．(3) クライアントは，深層相互学習によってパーソナライズドモデルと交換モデルを学習し，交換モデルの性能に基づいてパーソナライズドモデルの構造をチューニングする．(4) クライアントが学習後の交換モデルとパーソナライズドモデルをサーバに送り返した後，サーバが全てのクライアントのパーソナライズドモデルと交換モデルを集約し，集約後のパーソナライズドモデルをクライアントへ送信する．(1)〜(4) の手順を，グローバル通信の回数が閾値 T に達するまで繰り返す．

4.2 アルゴリズム

本節では，FedMe のアルゴリズムについて説明する．FedMe の擬似コードを Algorithm1 に示す．クライアントがモデルを初期化した後（1 行目），FedMe は学習を開始する．まずクライアントはパーソナライズドモデルをサーバーに送信する（4 行目）．サーバーはラベルのなしデータを使用してモデルをクラスタリングし（6 行目），各クライアントは自身のモデルと同じクラスタに属する交換モデルをサーバーから受け取る（8-9 行目）．その後，各クライアントはパーソナライズドモデルと交換モデルを学習し（11 行目），新しいパーソナライズドモデルのインデックスを決定する（13 行目）．各クライアントは 2 つの学習後のモデルと a_i^t をサーバに送信する（14 行目）．サーバは各モデルのパラメータを平均化することで集約し（16 行目），a_i^t に基づいて各クライアントにモデルを送信する（18 行目）．これらのステップをグローバル通信のラウンド数が T になるまで繰り返す．

続いて初期化，モデル学習，モデルチューニング，モデルクラスタリング，そしてモデル集約の詳細について説明する．

初期化　まず FedMe では，クライアントが自身のモデル構造を初期化する．FedMe ではクライアントは任意のモデル構造を使用することができるので，ローカルデータに応じてモデル構造を決定することができる．例えば，クライアントは自身のローカルデータに対して最適なモデル構造を構築する．もちろん，サーバーが任意のモデル構造を決定し，クライアントに配布することも可能である．

モデルトレーニング　FedMe では，各クライアントがパーソナライズドモデルを交換することで，複数のクライアントのローカルデータ上でモデルを学習させることができる．これにより，各クライアントが異なる構造のモデルを持っていてもモデルを学習させることが可能である．

各クライアントは，パーソナライズドモデルと交換モデルを深層相互学習によりローカルデータ上で学習させる．パーソナライズドモデルと交換モデルの間で深層相互学習を行うことで，それぞれを個別に学習させるよりも精度を向上させることができる．実際，深層相互学習によって複数のモデルを学習させた場合，モデルの推論性能が向上することが知られている [12]．したがって，深層相互学習は FedMe での学習において大きな

Algorithm 1 Algorithm of FedMe.

Input: number of global communication rounds T, number of local training epochs E, set of clients S and their local data $\{D_i\}_1^{|S|}$, unlabeled data U, numbers of cluster $\{K^1, K^2, \cdots K^T\}$, learning rate η
Output: personalized models $\{w_{p_i}^T\}_1^{|S|}$

Initialize($w_{p_i}^0$) **on all client** i
for $t = 1, \cdots, T$ **do**
  for $i \in S$ **do**
    Client i **sends** $w_{p_i}^{t-1}$ **to server**
  end for
  $\{C_1, \ldots, C_{K^t}\} \leftarrow$ **Model Clustering**($\{w_{p_i}^{t-1}\}_{i \in S_t}$, U, K^t)
  for $i \in S$ **do**
    $w_{ex_i}^{t-1} \leftarrow w \in C_k$ *that includes* $w_{p_i}^{t-1}$
    Server sends $w_{ex_i}^{t-1}$ **to client** i
    for $e = 1, \cdots, E$ **do**
      Model Training($w_{p_i}^{t-1}$, $w_{ex_i}^{t-1}$, D_i)
    end for
    $a_i^t \leftarrow$ **Model Tuning**($w_{p_i}^t$, $w_{ex_i}^t$, D_i)
    Client i **sends** $w_{p_i}^t, w_{ex_i}^t, a_i^t$ **to server**
  end for
  Model aggregation($\{w_{p_i}^t, w_{ex_i}^t\}_{i \in S}$)
  for $i \in S$ **do**
    Server sends aggregated $w_{p_{a_i^t}}^t$ **to client** i
  end for
end for

利点を持つ．

ここで，パーソナライズドモデルと交換モデルの損失関数 $\mathcal{L}_p$ と $\mathcal{L}_{ex}$ をそれぞれ以下のように定義する．

$$\mathcal{L}_p = -\sum_{(x,y)\in D_i}\sum_{m=1}^{M} I(y,m) log(p_p^m(x)) + \sum_{(x,y)\in D_i}\sum_{m=1}^{M} p_{ex}^m(x) log\frac{p_{ex}^m(x)}{p_p^m(x)}, \tag{3}$$

$$\mathcal{L}_{ex} = -\sum_{(x,y)\in D_i}\sum_{m=1}^{M} I(y,m) log(p_{ex}^m(x)) + \sum_{(x,y)\in D_i}\sum_{m=1}^{M} p_p^m(x) log\frac{p_p^m(x)}{p_{ex}^m(x)}, \tag{4}$$

ここで，p_p^m と p_{ex}^m は，それぞれクラス m に対するパーソナライズドモデルと交換モデルの予測値である．これらの式の第 1 項と第 2 項は，それぞれクロスエントロピー誤差とカルバック・ライブラー（KL）ダイバージェンスである．関数 $I(y,x)$ は $y = m$ のとき 1 を返し，それ以外のときは 0 を返す．

クライアント i は上記の損失関数を最小化するように，2 つのモデルを更新する．

$$w_{p_i}^t \leftarrow w_{p_i}^{t-1} - \eta \nabla \mathcal{L}_p, \tag{5}$$

$$w_{ex_i}^t \leftarrow w_{ex_i}^{t-1} - \eta \nabla \mathcal{L}_{ex}, \tag{6}$$

ここで，η は学習率，$\nabla\mathcal{L}_p$ と $\nabla\mathcal{L}_{ex}$ はそれぞれパーソナライズドモデルと交換モデルの勾配である．

モデルチューニング　クライアントは任意のモデル構造のモデルを使用可能なため，クライアントはモデル構造を自由に変更することができる．FedMe の学習過程では，グローバル通信のたびに他のクライアントのモデルを交換モデルとして受け取るため，各クライアントは自身のモデル構造を最適化する機会を多く持つ．交換モデルが自身の現在のモデルよりも高い性能を示した場合，クライアントは交換モデルを基に自身のモデル構造をチューニングする．

ここで，FedMe ではモデル構造のチューニング方法に制限がない．本稿では，FedMe の性能を検証するために，パーソナライズドモデルを交換モデルに置き換えるという簡単なチューニング方法を用いている．具体的には，各クライアントは深層相互学習によりモデルを学習した後，ラウンド t で自身のパーソナライズドモデルと交換モデルのどちらかを選択する．FedMe は，クライアント i が選択するパーソナライズドモデルのインデックスを表す a_i^t を，以下のように計算する．

$$a_i^t = \begin{cases} i & \text{if } w_{p_i}^t \in \underset{w=w_{p_i}^t, w_{ex_i}^t}{argmin} \mathcal{L}(w, D_i), \\ Idx(w_{ex_i}^t) & \text{otherwise.} \end{cases} \tag{7}$$

この式では，各クライアントはパーソナライズドモデルと交換モデルの損失を比較し，交換モデルの損失がパーソナライズドモデルの損失より小さければ，パーソナライズドモデルを交換モデルに置き換える．

もちろん，置き換える代わりに，例えばモデルの層数を増やすなど，他のチューニング方法を用いることも可能である．また，ここでは損失を考慮してモデル構造をチューニングしているが，各クライアントがモデルサイズや推論時間など独自の基準を持つことも可能である．我々は FedMe での最適なモデルチューニング方法を今後の課題として残す．

モデルクラスタリング　クライアント間のデータが不均一であるため，パーソナライズドモデルの出力はクライアントごとに異なる．出力が大きく異なるモデル間で深層相互学習を行うと，モデルがノイズにオーバーフィットする可能性がある．FedMe では，モデルを出力に基づいてクラスタリングし，各クライアントは自身のパーソナライズドモデルと類似した出力を持つモデルをサーバから交換モデルとして受け取る．

モデルクラスタリングにより，パーソナライズドモデルの出力と交換モデルの出力の差が小さくなり，ノイズへのオーバーフィットを防ぐことができる [22]．一方，類似した出力を持つモデルのみを用いて学習すると，汎化性能が失われてしまう．そこで，学習初期にはモデルクラスタリングを行わず，モデルの汎化性能を高めるようにする．そして学習が進むにつれて，モデルクラスタリングのクラスタ数を増やす．こうすることで，汎化性能を維持したまま，オーバーフィットすることなくモデルをクライアント毎に最適化するすることができる．

連合学習ではデータを共有しないため，モデルクラスタリングにローカルデータを利用することができない．そこで FedMe はワンショット連合学習 [23] のように，サーバがラベルなしデータにアクセス可能と仮定し，ラベルなしデータ U を入力として用いる．

モデルクラスタリングには Kmeans [21] を用いる．サーバはまず，ラベルなしデータを用いて各モデルの出力を計算する．次に，サーバはこれらの出力を用いて Kmeans を行い，モデルを K^t 個のクラスタに分割する．

モデル交換では，クライアントは自身のパーソナライドモデルと同じクラスタのモデルを交換モデルとしてサーバからランダムに 1 つ受け取る．クラスタ内にモデルが 1 つしかない場合，クライアントは他のクラスタからランダムにモデルを受け取る．

モデル集約　クライアント i は w_{p_i} と w_{ex_i} を同時に学習する．そのため，クライアントごとに学習後のモデルを新しいモデルに集約する必要がある．FedMe は FedAvg のようにモデルパラメータを平均化することでモデルを集約する．

$$w_{p_i}^t \leftarrow \frac{1}{(s_i+1)}(w_{p_i}^t + \sum_{j=1}^{|S|} u_{i,j}^t w_{ex_j}^t), \tag{8}$$

s_i は，交換モデルとして $w_{p_i}^t$ を受け取ったクライアントの総数である．$u_{i,j}$ は，どのクライアントがモデル $w_{p_i}^t$ を交換モデルとして受け取ったかを表し，以下の式で定義する．

$$u_{i,j}^t = \begin{cases} 1 & \text{if } i = Idx(w_{ex_j}^t), \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

モデル構造の違いに依存しないように，モデルパラメータは各クライアントのパーソナライズドモデル毎に平均化される．

5 評価実験

この章では，データの不均一性が高い 3 つのデータセットを用いて FedMe の精度を検証する．実験では，**Q1.**「FedMe は最先端の手法と比較してどの程度精度が高いか？」，**Q2.**「モデルチューニングはうまく動作しているか？」，そして **Q3.**「FedMe は最先端の手法と比較してどの程度学習に時間がかかるか？」という 3 つの問に答えることを目的としている．

簡単化のために，Pytorch を用いて単一の GPU マシンで仮想的にクライアントとサーバを作成し実験を行う．

5.1 実験設定

データセット　実験では，FEMNIST，CIFAR-10，Shakespeare の 3 つのデータセットを使用した．これらのデータセットは，既存研究 [4–6, 15, 17, 20] で頻繁に利用されているものである．FEMNIST（Federated EMNIST-62）[24] は，62 種類のラベルを持つ手書き文字の画像データである．このデータセットは書き手によって 3,400 のサブデータに分割されている．CIFAR-10 [25] は，10 種類のラベルを持つ写真画像である．このデータセットを [15] と同様にディリクレ分布を用いて 20 個のサブデータに分割する．ラベルとローカルデータサイズの不均一性の度合いを決定するために，α_{label} と α_{size} という 2 つのパラメータを用いる．本実験では，α_{label} を 0.5，α_{size} を 10 としている．Shakespeare [5] は，「The Complete Works of William Shakespeare」に含まれる台詞で構成されるデータセットである．このデータセットは役者によって 143 のサブデータに分割されている．表 2 は，データセットに含まれるクライアントのデータ数の統計である．なお，CIFAR-10 は各テストでランダムに分割しているため，統計量は一例である．

タスク　FEMNIST および CIFAR-10 データセットに対しては画像分類を，Shakespeare データセットに対しては与えられた文の次の文字を推論する次文字予測を行う．

表 2: データセットの統計量

Datasets	Total	Mean	SD	Max	Min
FEMNIST	671585	197.53	76.69	418	16
CIFAR-10	49000	2450	1024.66	5018	1131
Shakespeare	413629	2892.51	5445.89	33044	2

モデル　既存研究 [15, 26] に従い，各データセットに異なるモデルを使用する．FEMNIST と Shakespeare では，それぞれ CNN と LSTM を用いる [26]．CIFAR-10 では，[15] と同様の修正を加えた VGG を使用する．各データセットに対して，層数を変化させた 4 つのモデルを用いる．CNN と LSTM に対しては畳み込み層と LSTM の層数を 1–4 層まで変更し，デフォルトは 2 層とした．VGG に対しては VGG11，VGG13，VGG16，VGG19 の 4 つを使用し，デフォルトは VGG13 とした．

学習と評価　実験では，クライアントの数は 20 とした．FEMNIST と Shakespeare では，クライアントにローカルデータを割り当てるために 20 個のサブデータをランダムに選択する．CIFAR-10 では，全てのデータをランダムに 20 個のローカルデータに分割し，各クライアントに割り当てた．全てのクライアントは最新研究 [15] に従って，学習に毎回参加する．各データセットから 1,000 個のラベルなしデータを選択する．このラベルなしデータは訓練用データとテスト用データから除外した．

グローバル通信ラウンド数は FEMNIST，CIFAR-10，Shakespeare でそれぞれ 300，500，100 とし，ローカル学習エポック E は全ての実験において 2 とした．異なるクライアント上で行った 5 回の実験における精度の平均と標準偏差を示す．

ベースラインとハイパーパラメータチューニング　FedMe を（1）非パーソナライズド連合学習手法，（2）パーソナライズド連合学習手法，そして（3）非連合学習手法の 3 種類の手法と比較する．（1）では FedAvg を使用し，（2）では HypCluster，MAPPER，FML，pFedMe を使用する．（3）では，クライアントがローカルデータのみを用いてモデルを構築する Local Data Only と，サーバが全クライアントからローカルデータセットを収集し学習する Centralized（Centralized は理想値と見なすことができる）を使用する．Centralized，FedAvg，HypCluster，そして FedMe では，モデルの構築後，各クライアントは自身のローカルデータで fine-tuning を行う．MAPPER と pFedMe では fini-tuning と似た学習手順をアルゴリズム中に含んでいるため，fine-tuning を行わない．また，pFedMe[2]以外の手法はコードが公開されていなかったため，実装した．

次にハイパーパラメータチューニングについて説明する．各手法の学習率は，$\eta \in \{10^{-3}, 10^{-2.5}, 10^{-2}, \ldots, 10^{0.5}\}$ の中からグリッドサーチにより最適化した．最適化手法は SGD を使用し，モーメンタムは 0.9，重み減衰は 10^{-4} とした．FEMNIST，CIFAR-10, Shakespeare のバッチサイズはそれぞれ 20，40，10 とした．Hypcluster では，2 つのグローバルモデルを構築す

2：https://github.com/CharlieDinh/pFedMe

表 3: 各モデル構造を選択したクライアントの平均数

	FEMNIST	CIFAR-10	Shakespeare
model1	6.8 ± 2.5	5.6 ± 1.5	7.4 ± 1.7
model2	7.4 ± 0.5	9.2 ± 1.6	6.6 ± 1.8
model3	4.4 ± 2.8	4.0 ± 2.0	5.0 ± 1.6
model4	1.4 ± 1.1	1.2 ± 0.4	1.0 ± 1.7

表 4: テスト用データに対する精度 (平均 ± 標準偏差).

	FEMNIST	CIFAR-10	Shakespeare
Local Data Only	64.71 ± 2.94	73.17 ± 1.55	24.77 ± 1.95
Centralized	79.35 ± 2.29	90.80 ± 0.92	48.43 ± 3.32
FedAvg	77.25 ± 3.99	89.59 ± 0.94	42.53 ± 2.19
HypCluster	76.29 ± 3.15	88.54 ± 1.42	41.10 ± 3.29
MAPPER	60.95 ± 3.04	61.29 ± 4.19	36.77 ± 1.58
FML	67.91 ± 2.53	79.89 ± 1.44	28.73 ± 1.78
pFedMe	72.92 ± 3.54	79.46 ± 2.08	40.33 ± 2.27
FedMe	$\mathbf{78.52 \pm 2.64}$	$\mathbf{89.76 \pm 0.90}$	$\mathbf{44.71 \pm 1.12}$

る．FedMe では，各クライアントはモデル 1–4 のうち，Local Data Only で最も精度の高いモデルを初期のモデル構造とする（表 3 を参照）．クラスタ数の範囲は 1–4 で初期値を 1 とし，FEMNIST，CIFAR-10，Shakespare のそれぞれについて，グローバル通信ラウンドが [150, 225, 275], [250, 375, 450], [50,75,90] のタイミングでクラスタ数を 1 ずつ増加させる．

5.2 実験結果

Q1．FedMe は最先端の手法と比較してどの程度精度が高いか？ 各手法の精度を表 4 と図 2 に示す．表 4 はテスト用データに対する平均精度と標準偏差，図 2 は各グローバル通信ラウンドでの検証用データに対する精度を示している．

表 4 から，FedMe は全てのデータセットにおいて連合学習手法の中で最も高い精度を達成し，その精度は Centralized と同等程度であることがわかる．なお FEMNIST と Shakespeare の標準偏差は，各実験で使用するクライアントが異なるため CIFAR-10 と比較して大きい．FedMe は全てのデータセットにおいて，連合学習手法の中で最も低い（あるいは次点）標準偏差を達成しており，FedMe が最も頑健であることがわかる．この結果は，FedMe の学習手順が有効であることを示している．

ベースライン内で比較すると，興味深いことに最もシンプルな FedAvg で fine-tuning したものがベースライン中で最も高い精度を達成している．この結果は，データの不均一性が fine-tuning によって十分に解決できることを示している．

図 2 より，FedAvg と FedMe はどちらも早期のラウンドで高い精度を達成していることがわかる．これは，FedMe が他のパーソナライズド連合学習手法と比較して早期に収束することを示している．

Q2. モデルチューニングはうまく動作しているか ？ここでは，FedMe が最適なモデル構造を自動的にチューニングできることを示す．表 5 は，モデル構造を固定した場合の FedMe の精度を示している．モデル 1–4 はそれぞれ，FEMNIST では畳み込み層の数が 1–4 の CNN，CIFAR-10 では (VGG11，VGG13，VGG16，VGG19)，Shakespeare では LSTM 層の数が 1–4 の LSTM である．全てのデータセットにおいて，オートチューニングはモデル 1–4 の中間程度の精度である．特に，FEMNIST では，オートチューニングの精度は，モデル構造固定した場合の最高精度に匹敵する．この結果は，モデル構造の自動チューニングが有効であることを示しており，モデル構造を手動でチューニングするコストを削除できることを示している．

表 5: モデル構造の自動チューニングの効果検証

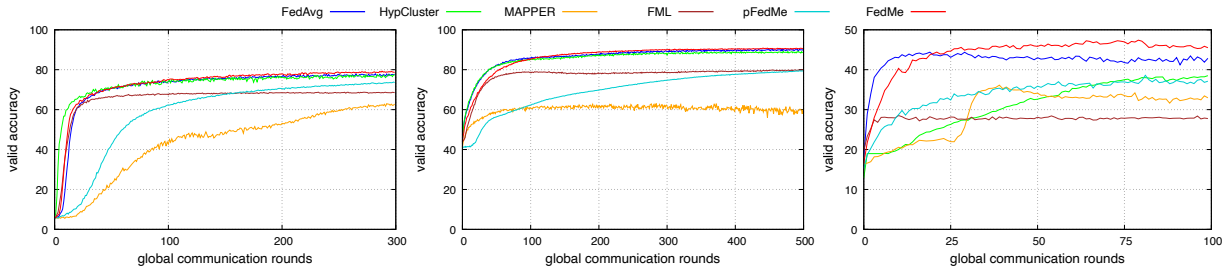
	FEMNIST	CIFAR-10	Shakespeare
model 1	74.80 ± 2.75	89.25 ± 0.74	45.31 ± 3.20
model 2	78.06 ± 3.00	90.96 ± 0.84	45.83 ± 2.48
model 3	77.85 ± 2.90	90.67 ± 0.47	46.01 ± 2.72
model 4	78.54 ± 2.92	90.45 ± 0.54	42.55 ± 5.12
auto-tuning	78.52 ± 2.64	89.76 ± 0.90	44.71 ± 1.12

Q3.FedMe は最先端の手法と比較してどの程度学習に時間がかかるか？ それぞれの手法において，学習にかかる実行時間をサーバ上とクライアント上でそれぞれ計測した．図 3 は，1 グローバル通信ラウンドにおけるサーバ上とクライアント上の平均実行時間である．この結果から，FedMe は各クライアントが 2 つのモデルを学習するが，クライアント上の実行時間は他の手法と同等程度であることがわかる．一方，FedMe はサーバが全てのパーソナライズドモデルの出力を取得しモデルクラスタリングを行うため，サーバ上での実行時間は他の手法より大きい．しかし，本実験ではサーバとクライアントに同じハードウェアを使用しているが，実際のシナリオでは一般にサーバの方が強力な計算資源を持っている．つまり，実世界のアプリケーションではサーバーの計算コストが小さくなる傾向がある．また，ラベルなしデータのサイズを変更することで，サーバーの計算コストを制御することも可能である．以上より，FedMe による精度向上を考慮すると，サーバーの計算コストは許容可能であると言える．

6 まとめ

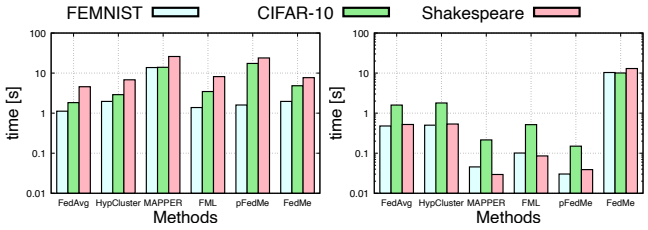
本稿では，我々の先行研究である Federated Learning via Model exchange (FedMe) を拡張し，モデル構造のチューニングを可能とした．FedMe は，モデル交換と深層相互学習により，クライアント毎に異なる構造のモデルを学習することができる．本拡張によってモデルパラメータを学習しながらモデル構造を自動的にチューニングし，クライアント毎にモデルを構築する．評価実験では，FedMe が最先端の手法よりも精度が高く，モデル構造を自動でチューニングできることを示した．

今後の課題として，我々はモデルチューニングを拡張し，モデル構造をより柔軟にチューニングできるようにする予定である．FedMe はモデル構造を自動的にチューニングするが，その候補はあくまでクライアントが事前に設計したモデル構造に限定される．そのため，最適なモデル構造が設計されていない場合，FedMe の自動チューニングはうまく動作しない可能性がある．そこで，構造自動探索（NAS）などを用いた，より柔軟なモデルチューニング方法を考案する必要がある．



(a) FEMNIST (b) CIFAR-10 (c) Shakespeare

図 2: 各手法の検証用データに対する精度の推移



(a) Client (b) Server

図 3: 1 ラウンドあたりの平均実行時間

謝　　辞

本研究は JSPS 科学研究費 JP17H06099 および JP20H00584 の支援によって行われた.

文　　献

[1] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *ESANN*, Vol. 3, pp. 437–442, 2013.

[2] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv*, 2018.

[3] David Leroy, Alice Coucke, Thibaut Lavril, Thibault Gisselbrecht, and Joseph Dureau. Federated learning for keyword spotting. In *ICASSP*, pp. 6341–6345, 2019.

[4] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, pp. 1273–1282, 2017.

[5] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *MLSys*, Vol. 2, pp. 429–450, 2020.

[6] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *arXiv*, 2020.

[7] Tao Shen, Jie Zhang, Xinkang Jia, Fengda Zhang, Gang Huang, Pan Zhou, Kun Kuang, Fei Wu, and Chao Wu. Federated mutual learning. *arXiv*, 2020.

[8] Canh T. Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. In *NIPS*, pp. 21394–21405, 2020.

[9] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M. Alvarez. Personalized federated learning with first order model optimization. In *ICLR*, 2021.

[10] Chaoyang He, Murali Annavaram, and Salman Avestimehr. Fednas: Federated deep learning via neural architecture search. In *CVPR*, 2020.

[11] 松田光司, 堀敬三, 佐々木勇和, 肖川, 鬼塚真. Fedme：モデル交換に基づく連合学習手法. 第 13 回データ工学と情報マネジメントに関するフォーラム, 2021.

[12] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu. Deep mutual learning. In *CVPR*, pp. 4320–4328, 2018.

[13] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv*, 2019.

[14] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Commun. Surv. Tutor.*, 2020.

[15] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *ICLR*, 2020.

[16] Lumin Liu, Jun Zhang, SH Song, and Khaled Ben Letaief. Edge-assisted hierarchical federated learning with non-iid data. *arXiv*, 2019.

[17] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv*, 2019.

[18] Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. In *ICLR*, 2017.

[19] Koji Matsuda, Yuya Sasaki, Chuan Xiao, and Makoto Onizuka. Fedme: Federated learning via model exchange. In *SDM*, 2022.

[20] Hong-You Chen and Wei-Lun Chao. Fedbe: Making bayesian model ensemble applicable to federated learning. In *ICLR*, 2021.

[21] James MacQueen, et al. Some methods for classification and analysis of multivariate observations. In *BSMSP*, Vol. 1(14), pp. 281–297, 1967.

[22] Jiyang Gao, Zhen Li, Ram Nevatia, et al. Knowledge concentration: Learning 100k object classifiers in a single cnn. *arXiv*, 2017.

[23] Neel Guha, Ameet Talwalkar, and Virginia Smith. One-shot federated learning. *arXiv*, 2019.

[24] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv*, 2018.

[25] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *technical report*, 2009.

[26] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv*, 2020.