DEIM2021 E21-5

FedMe：モデル交換に基づく連合学習手法

松田 光司† 堀 敬三† 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,hori.keizo,sasaki,chuanx,onizuka}@ist.osaka-u.ac.jp

あらまし 連合学習は複数のクライアントが中央のサーバと連携して，クライアントの持つデータを共有することなくモデルを学習する分散型の機械学習手法である．クライアントのもつデータの不均一性に対処するために様々な手法が提案されているが，クライアント毎に異なるモデル構造を保持することができないことやモデルの推論精度の低下といった問題がある．本稿では，クライアント毎に異なるモデル構造を保持できる高精度な連合学習手法，FedMe を提案する．FedMe で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．2 種類の実データを用いた評価実験にて，FedMe が既存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

近年では，スマートフォンやタブレットなどのデバイスの増加に伴い，これまでにないほどの大量の個人データが収集されている．それらのデータを用いた機械学習が様々なアプリケーションに応用されている．例えば，キーボードの入力履歴による次単語予測 [1] や，クライアントの音声データによるウェイクワード検出 [2]，スマートフォンの加速度センサやジャイロスコープの慣性データによる行動認識 [3] などがある．従来の機械学習では中央のサーバが全てのデータを保持しながら集中的にモデルを学習するため，サーバにデータを全て送る必要がある．しかし，デバイスから入手したデータは機密性やプライバシが高い．例えば，スマートフォンが保持するデータにはそのユーザの位置情報や顔写真などの個人情報が含まれることがある．そのため，それらのデータをサーバに集めることは個人情報漏洩のリスクを伴う．例えば，ヨーロッパでは EU 一般データ保護規則（GDPR）が，アメリカ・カリフォルニア州ではカリフォルニア消費者プライバシー法（CCPA）が制定されている．加えて，全てのデータをサーバに送ることはネットワーク帯域幅の制限があるため現実的ではない．

これらの問題に対処するために，データをサーバに送らずに，サーバとクライアントが持つデバイスが共同でモデルを学習する連合学習が提案された [4]．連合学習の手順は（1）クライアントが自身のデータセットであるローカルデータでモデルを学習するクライアント学習と（2）各クライアントが学習後のモデルをサーバに送り，サーバがそれらのモデルを使って新しくモデルを更新するモデル集約の 2 つのステップで構成される．この手順によって，各クライアントはデータを共有することなく各クライアントが持つデータをモデルの学習に利用できる．これにより，各クライアントが自身のデータのみを用いてモデルを学習した場合より，高精度に予測可能なモデルを構築することができる．

連合学習における課題の一つとして，データの不均一性の問題がある．各クライアントが持つデータの分布は異なり，非独立同分布として分散している．例えば，『I live in...』の後に続く単語は地域やクライアントによって異なる．加えて，各クライアントの持つローカルデータ数には差がある．すなわち全てのクライアントが同じモデルを用いて推論することは予測精度の低下につながる．そのため，各クライアントに適したモデルを作成する必要がある．また，クライアント毎に最も推論精度が高いモデルの構造は異なる．したがって，クライアント毎に異なる構造のモデルを保持できることが望ましい．しかし，多くの既存研究のモデル集約では各クライアントのモデル構造が同じであるという制約がある [5–9]．一方，クライアント毎に異なる構造のモデルを保持できる手法もいくつか存在するが，それらの手法では推論精度が低いといった問題がある [10, 11]．

本稿では上記の問題に対応するために，異種モデル交換に基づいた新しい連合学習手法である FedMe を提案する．FedMe では，各クライアントは他クライアントとローカルモデルを交換し，ローカルと交換モデル両方の学習を実施する．モデル交換により，全体でモデル集約する必要がないため，クライアント毎に異なる構造のモデルを保持可能である．交換モデルの選択方法は精度に影響し，類似したモデルを交換モデルとすることで精度は向上する．しかし，クライアント毎にモデル構造が異なる場合，モデルパラメータによる類似度は図ることができない．FedMe では，出力に基づいてモデルをクラスタリングすることで，類似したモデルを交換モデルとして選択可能にし，類似したモデル同士で深層相互学習 [12] を行うことで精度を向上させる．評価実験では（1）全てのクライアントが同じ構造のモデルを利用する実験と（2）クライアント毎に異なる構造のモデルを利用する実験の 2 つを行い，FedMe が既存手法と比べて高精度であることを示す．

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 章にて事前知識について説明する．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

連合学習は McMahan ら [4] によって導入された分散型の機械学習手法である．連合学習では，通信コストに関する研究 [13]，セキュリティに関する研究 [14,15] などが行われている．その他にも，Adam などの最適化手法の連合学習への応用 [16] や連合学習の産業への応用 [1,2]，フレームワークやライブラリ [17,18] など幅広く研究が行われている．また，これらの連合学習の最近の研究をまとめた論文もいくつか存在する [19,20]．

従来の機械学習と連合学習の大きな違いはクライアントのデータが共有されず，サーバや他のクライアントがアクセスすることができない点である．これにより，クライアントのデータのプライバシが保証される．McMahan らは，学習後のモデルを単に平均化することでモデル集約を行う FedAvg [4] を，Wang らはベイジアンノンパラメトリックに基づいた FedMA [21] を，Liu らはエッジを媒介として連合学習を行う HierFAVG [22] を提案した．一方，データ分布に偏りがある場合，モデルの推論精度が低下してしまう [23,24]．この問題に対応するために FedProx [25] や SCAFFOLD [26] などが提案されている．これらの手法では，各クライアントの学習後のモデルが理想とかけ離れるクライアントドリフトという問題を解決することで推論精度の低下を防ぐ．しかし，これらの手法は単一のモデルを作成することを目的としている．データの不均一性がある場合，全てのクライアントにとって推論精度が高いモデルと各クライアントにとって推論精度が高いモデルは異なる．そのため，単一のモデルを作成するよりもクライアント毎にモデルを作成する方が推論精度が向上する．

クライアント毎に異なるモデルパラメータを持つことでデータの不均一性に対応する研究も多くある．Mansour らはユーザクラスタリングを行う HypCluster とモデル補間を行う MAPPER を提案している [5]．HypCluster ではサーバが複数のモデルを作成する．各クライアントは最も推論精度が高いモデルを 1 つ選択し，選択したモデルを学習する．モデルを使用したユーザクラスタリングを行うことでデータの不均一性に対応する．MAPPER ではクライアント毎にモデルを作成する．各クライアントはサーバが作成したモデルとクライアントのモデルを加重平均して補間モデルを作成する．Arivazhagan らはモデルの一部の層（基礎層）のみをサーバに送り連合学習で学習して，残りの層（個人層）をクライアントが各自学習する FedPer [6] を提案している．Smith らはマルチタスク学習を利用した MOCHA [7] を提案している．メタ学習を FedAvg に組み込んだ手法として Fallah らは Per-FedAvg [8] を，Khodak らは FedAvg with ARUBA [9] を提案している．しかし，これらの手法はクライアント毎に同じモデル構造を必要とするため，クライアント毎に異なる構造のモデルを保持することができない．

クライアント毎に異なる構造のモデルを保持可能な手法もいくつか存在する．Li らは知識蒸留を組み込んだ FedMD [10] を提案した．FedMD は（1）クライアント毎にパブリックデータで学習した後ローカルデータで学習する転移学習と，(2）クライアントのモデルのパブリックデータに対する出力をサーバに送信し，平均化されたものを正解ラベルとする知識蒸留の 2 つのステップで学習を進める．パブリックデータは全てのクライアントやサーバがアクセスすることができるデータを指す．FedMD ではパブリックデータが少ない場合．推論精度が低下するという問題がある．加えて，パブリックデータが必要という問題もある．クライアントからデータを集めるとプライバシの問題に繋がるため，サーバがコストをかけて独自に収集する必要がある．Shen らは深層相互学習を組み込んだ Federated Mutual Learning（FML）[11] を提案した．FML は各クライアントのモデルとサーバが作成するモデル間を深層相互学習で学習し，サーバが作成するモデルのみをサーバに送信し集約する．FML では各クライアントのモデルの学習に用いるのは自身のローカルデータのみであるため，ローカルデータが少ない場合には推論精度が低下するという問題がある．

3 事前知識

3.1 連合学習

連合学習ではデータを共有することなく，サーバと多くのクライアントが協力しながら学習を行いモデルを作成する．サーバが作成し，全てのクライアントが共通して推論に使用するモデルはグローバルモデルと呼ばれる．クライアントが持つそれぞれのデータセットはローカルデータと呼ばれ，クライアント i のローカルデータを D_i，D_i のデータ数を n_i とする．クライアントの集合を S，全てのクライアントのデータの総数を n とすると，連合学習の最適化問題は以下の式で表せる．

$$\min_{w\in\mathbb{R}^d} \sum_{i=1}^{|S|} \frac{n_i}{n} f_i(w_g) \tag{1}$$

$$f_i(w) = \frac{1}{n_i} \sum_{(x_i,y_i)\in D_i} f_i(x_i, y_i, w) \tag{2}$$

f_i はクライアント i の損失関数，x_i と y_i はそれぞれローカルデータ D_i の入力データとラベル，w_g はグローバルモデルを表している．一般的に連合学習では，タイムステップ t でクライアントがモデルの学習を E 回繰り返した後，モデル集約を行いグローバルモデルを作成する．この手順をグローバルモデルが収束するまで R 回行う．全てのクライアントはグローバルモデルを使って推論を行う．連合学習では全てのクライアントが毎回学習に参加しない．タイムステップ t の参加クライアントの集合を S_t とする．

連合学習の代表的な手法である FedAvg では，タイムステップ t で各クライアントがグローバルモデル w_g^t をダウンロードして w_i^t を得る．そして，ローカルデータを用いてモデルを E 回学習することでローカルの目的関数を最適化する．クライアント $i \in S_t$ が以下のクライアント学習を行う．

$$\textbf{client training} : w_i^{t+1} \leftarrow w_i^t - \eta \nabla f_i(w_i^t) \tag{3}$$

η は学習率，$\nabla f_i(w_i^t)$ は $f_i(w_i^t)$ の勾配を表している．各クライアントは勾配降下法によってモデルを更新する．クライアント学習を E 回行った後，各クライアントはモデル w_i^{t+1} をサーバに送り，サーバはそれらを平均化して一つのグローバルモデルを作成する．

$$\textbf{model aggregation} : w_g^{t+1} \leftarrow \sum_{i \in S_t} \frac{1}{|S_t|} w_i^{t+1} \tag{4}$$

D_i が独立同分布である，すなわちデータが均一である場合，FedAvg は $D = \bigcup_{i=1}^{|S|} D_i$ で集中的に学習した理想のモデルに収束する．しかし，D_i が非独立同分布である，すなわちデータが不均一である場合，FedAvg は推論精度が低下することが示されている [23].

上記の最適化問題では 1 つのグローバルモデル w_g を作成することを目的としている．一方で，データの不均一性に対処するためにクライアント毎にモデルを作成する連合学習手法も存在する [5, 10, 11]．クライアント毎のモデルはローカルモデルと呼ばれ，クライアント i のローカルモデルを w_{l_i} とする．最適化問題は以下の式で表せる．

$$\min_{w_{l_1},\ldots,w_{l_{|S|}}} \sum_{i=1}^{|S|} \frac{n_i}{n} f_i(w_{l_i}) \tag{5}$$

一般的に，ローカルモデルはそれぞれのクライアント毎のローカルデータに最適化されるため精度が高くなる．

3.2 深層相互学習

本節では，本研究で用いる深層相互学習 [12] について述べる．深層相互学習は知識蒸留 [27] に基づいた学習方法であるため，まず知識蒸留について説明し，その後深層相互学習について説明する．

深層学習では一般的にモデルサイズが大きいほど推論精度が高いことが知られており，小さいサイズのモデルが大きいサイズのモデルと同程度の推論精度を達成するために知識蒸留 [27] が考案された．知識蒸留は，大きいサイズのモデル（教師モデルと呼ぶ）からより小さいサイズのモデル（生徒モデルと呼ぶ）へ精度を大きく低下させることなく知識を伝搬する手法である．知識蒸留では教師モデルは学習を行わず，生徒モデルのみが学習する．生徒モデルの損失関数には，教師モデルの出力と生徒モデルの出力間の損失であるソフトターゲット損失と，学習データの正解ラベルと生徒モデルの出力間の損失であるハードターゲット損失の 2 つを足し合わせたものを用いる．

$$\mathcal{L} = \mathcal{L}_H(p_s, y) + \mathcal{L}_S(p_t, p_s) \tag{6}$$

$$p_t = \frac{\exp(z/T)}{\sum_i \exp(z_i/T)} \tag{7}$$

$$p_s = \frac{\exp(v/T)}{\sum_i \exp(v_i/T)} \tag{8}$$

L_H と L_S それぞれハードターゲット損失とソフトターゲット損失を表している．一般的には L_H はクロスエントロピー誤差，L_S はカルバック・ライブラー・ダイバージェンスが使われる．y，p_t，p_s はそれぞれ正解ラベルと教師モデルと生徒モデルの予測値を表している．また，z は教師モデルの出力，v は生徒モデルの出力，T はハイパーパラメータである．知識蒸留ではワンホットラベルではなく，学習の手助けとなるような教師モデルのソフトターゲットで学習することで汎化性能が高まり，生徒モデルの推論精度が向上する．

本稿では，異なるモデル構造を持ったローカルモデル同士で学習を可能にするために知識蒸留をクライアント学習に組み込む．しかし，通常の知識蒸留は十分に学習された教師モデルが必要であり，ローカルモデルのみでは学習することができない．そこで，教師モデルを必要とせず，2 つの生徒モデルが知識蒸留を双方向で行う深層相互学習 [12] を提案手法のクライアント学習に利用する．深層相互学習では以下の損失関数を用いる．

$$\mathcal{L}_{w_1} = \mathcal{L}_{C_1} + D_{KL}(p_2||p_1) \tag{9}$$

$$\mathcal{L}_{w_2} = \mathcal{L}_{C_2} + D_{KL}(p_1||p_2) \tag{10}$$

$$p_1 = \frac{\exp(z)}{\sum_i \exp(z_i)} \tag{11}$$

$$p_2 = \frac{\exp(v)}{\sum_i \exp(v_i)} \tag{12}$$

L_{C_1}，L_{C_2} はそれぞれモデル 1，モデル 2 のハードターゲット損失を表しており，深層相互学習ではクロスエントロピー誤差が用いられている．D_{KL} はソフトターゲット損失を表しており，深層相互学習ではカルバック・ライブラー・ダイバージェンスが用いられている．p_1 と p_2 はそれぞれモデル 1 とモデル 2 の予測値を表している．この損失関数を最小にするようにモデル 1 とモデル 2 が学習する．深層相互学習によって学習されたモデルは，各モデルが単独に学習した時と比べて推論精度が上がることが知られている．

4 FedMe

本章では，提案手法である FedMe について説明する．FedMe では，グローバルモデルを作成する代わりに，各クライアントが個別にローカルモデルを作成する．また，各クライアントはモデル構造を自由に選択することで，各クライアントのローカルモデルの推論精度を向上させる．

FedMe の概要を図 1 示す．FedMe は（1）モデルクラスタリング，(2）深層相互学習，(3）ローカルモデル集約の 3 つのステップから構成される．他クライアントのモデルを交換モデルとしてサーバから受け取り，ローカルモデルと交換モデルの 2 つのモデルを深層相互学習によって学習する．交換モデルの決定方法として，ラベル無しデータを用いたモデルクラスタリングを行う．最後に，学習後のローカルモデルと交換モデルとして他クライアントが学習したモデルを，モデルパラメータの平均化によって 1 つのローカルモデルに集約する．

4.1 モデルクラスタリング

クライアント間でデータの不均一性があるため，ローカルモデルのモデルパラメータもクライアント毎に異なる．類似したモデル同士で深層相互学習を行うことで精度が向上すると考え

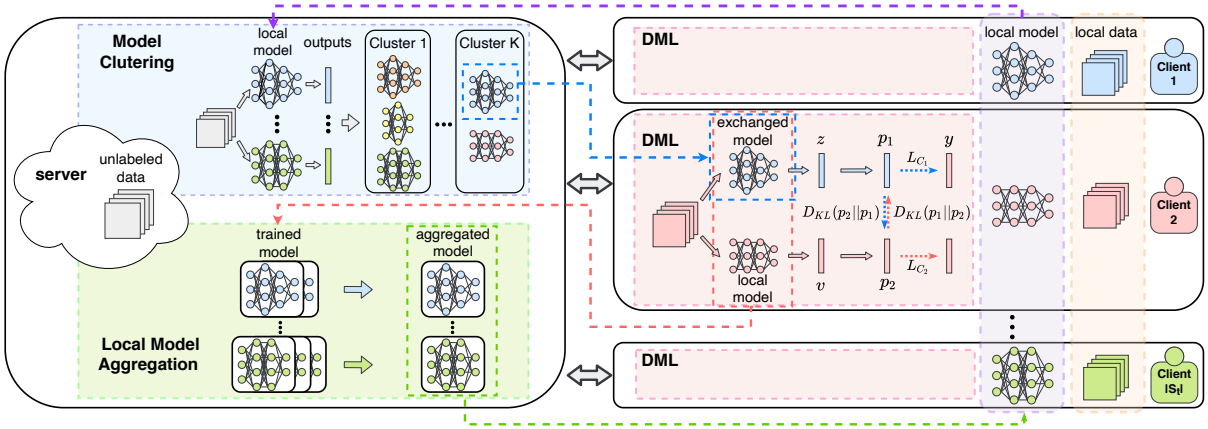


図 1: FedMe の概要図．FedMe では，(1) モデルをラベル無しデータに対する出力でクラスタリングし，(2) 深層相互学習によってモデルを学習し，(3) 各クライアントのローカルモデル毎にモデルパラメータを平均化し集約する．

られるため，FedMe ではモデルをクラスタリングし，各クライアントは類似したモデルを交換モデルとしてサーバから受け取る．連合学習のクラスタリングにモデルパラメータを使用した既存研究 [28] が存在するが，異なるモデル構造同士ではクラスタリングすることができない．そこで FedMe ではモデルの出力に基づいたクラスタリングを行う．クラスタリング法には Kmeans 法 [29] を使用する．しかし，連合学習ではクライアントのデータを共有しないため，入力にクライアントのデータを用いることができない．そこで FedMe ではワンショット連合学習 [30] のようにラベル無しデータにサーバがアクセスできることを想定し，ラベル無しデータ U を入力として利用する．パブリックデータにラベルがあるのに対し，ラベル無しデータにはラベルが存在しない．一般的にパブリックデータよりもラベル無しデータの方が取集が容易である．

モデルクラスタリングはサーバが行う．全てのクライアントのモデルを使いモデルクラスタリングを行うことが理想的だが，クライアント数が多い場合，全てのモデルをサーバが保持するのは現実的ではない．そこで，クライアント $i \in S_t$ のローカルモデルのみモデルクラスタリングの対象とする．まず，クライアント $i \in S_t$ は自身のローカルモデルをサーバに送る．サーバがラベル無しデータを入力としてモデルの出力を求める．ラベル無しデータを入力とした時のクライアント i のローカルモデルの出力を v_i，クラスタ j の中心点を c_j として，損失関数を以下のように定義する．

$$\mathcal{L} = \frac{1}{|S_t|} \sum_{j=1}^{K} \sum_{i=1}^{|S_t|} \delta_i^j ||v_i - c_j||^2 \tag{13}$$

ここで，δ_i^j はクラスタ割当てを表しており，以下の式で定義する．

$$\delta_i^j = \begin{cases} 1 & (if\ j = argmin_k ||v_i - c_k||^2 \\ 0 & (otherwise.) \end{cases} \tag{14}$$

クラスタ割当てに基づき，クライアントはモデル交換を行う．モデル交換では．各クライアントは自身のローカルモデルと同クラスタのモデルを，サーバから交換モデルとしてランダムに 1 つ受け取る．ただし，クラスタ内モデルが 1 つしかない場合，他クラスタのモデルを，サーバから交換モデルとしてランダムに 1 つ受け取る．

4.2 深層相互学習

クライアント毎にモデル構造を自由に選択することで，クライアント間のデータの不均一性に対応することができる．単純にはクライアント毎にローカルデータのみで学習すれば良いが，ローカルデータが少ない場合は過学習を起こしてしまう．そこで，FedMe では各クライアントがローカルモデルを交換し合い，複数クライアントのローカルデータ上で学習することで過学習による推論精度の低下を防ぐ．各クライアントは自身のローカルモデルと，4.1 節でサーバから受け取った交換モデルの 2 つのモデルを学習する．

ローカルモデルと交換モデルの 2 つはモデル構造が異なるため，FedMe ではクライアント学習に深層相互学習を使用する．ローカルモデルと交換モデルの 2 つのモデルを深層相互学習で学習する．ローカルモデルと交換モデルの損失関数を以下で定義する．

$$\mathcal{L}_l = \mathcal{L}_{C_l} + D_{KL}(p_{ex}||p_l) \tag{15}$$

$$\mathcal{L}_{ex} = \mathcal{L}_{C_{ex}} + D_{KL}(p_l||p_{ex}) \tag{16}$$

$\mathcal{L}_{C_l}$ と $\mathcal{L}_{C_{ex}}$ はそれぞれローカルモデルと交換モデルのハードターゲット損失であるクロスエントロピー誤差を表している．また，p_l と p_{ex} はそれぞれローカルモデルと交換モデルの予測値を表している．上記の損失関数を最小にするようにクライアント i はローカルモデルと交換モデルを更新する．

$$w_{l_i}^t \leftarrow w_{l_i}^{t-1} - \eta \nabla \mathcal{L}_l \tag{17}$$

$$w_{ex_i}^t \leftarrow w_{ex_i}^{t-1} - \eta \nabla \mathcal{L}_{ex} \tag{18}$$

w_{l_i} と w_{ex_i} はそれぞれクライアント i のローカルモデルとクライアント i が受け取った交換モデルを表している．ローカルモデルと交換モデル間を深層相互学習で学習することで，単独で学習するよりも推論精度が向上する．加えて，4.1 節でモデルクラスタリングを行っているため，類似したモデル同士で深層

相互学習を行うことができる．

4.3 ローカルモデル集約

クライアント i はローカルモデル w_{l_i} を学習するが，他クライアントも w_{l_i} を交換モデルとして受け取り，同時に学習をする．そのため，それら全てを集約し一つのローカルモデルにする必要がある．FedMe では，FedAvg と同様にローカルモデルの集約をモデルパラメータの平均化によって行う．

$$w_{l_i}^t \leftarrow \frac{1}{m_i}(w_{l_i}^t + \sum_{j=1}^{S_t} u_{i,j} w_{ex_j}^t) \tag{19}$$

ここで，m_i は $w_{l_i}^t$ を学習したクライアントの総数を表す．また，$u_{i,j}$ はどのクライアントがローカルモデル $w_{l_i}^t$ を交換モデルとして受け取ったかを表す変数であり，以下の式で定義する．

$$u_{i,j} = \begin{cases} 1 & (if\ w_{ex_j}^{t-1} = w_{l_i}^{t-1}) \\ 0 & (otherwise.) \end{cases} \tag{20}$$

各クライアントのローカルモデル毎にモデルパラメータを平均化し集約するため，モデル構造の異種性に依存しない集約が可能である．

4.4 FedMe のアルゴリズム

FedMe のアルゴリズムを Algorithm 1 に示す．まず参加クライアントは自身のローカルモデルをサーバに送る（3–5 行）．サーバはラベル無しデータを用いて送られてきたモデルをクラスタリングし（6 行），各クライアントは同クラスタのモデルを交換モデルとしてサーバから受け取る（9–10 行）．そして各クライアントはローカルモデルと交換モデルを深層相互学習で学習し（11 行），2 つのモデルをサーバに送る（13 行）．サーバは各モデルをそれぞれ平均化することで集約し（16 行），各クライアントに返却する（18 行）．これらの手順を各ローカルモデルが収束するまで繰り返す．

Algorithm 1 FedMe

Input:learning rate η, number of global epochs R, number of local epochs E, set of clients S, number of participants m, Client datasets $\{D_i\}_i^{|S|}$, initial local model $\{w_{l_i}^0\}_1^{|S|}$, unlabeled dataset U, number of cluster K

1: **for** $t = 1, \cdots, R$ **do**
2: $S_t \leftarrow$ (random set of m clients)
3: **for** $i \in S_t$ **do**
4: Client i sends $w_{l_i}^{t-1}$ to server
5: **end for**
6: $\{c_1 \dots c_K\} \leftarrow$ **Model Clustering**$(\{w_{l_i}^{t-1}\}_{i \in S_t}, U, K)$ (subsection4. 1)
8: **for** $i \in S_t$ **do**
9: $w_{ex_i}^{t-1} \leftarrow w \in c_k$ *that includes* $w_{l_i}^{t-1}$
10: Server sends $w_{ex_i}^{t-1}$ to client i
11: **Deep Mutual Learning**$(w_{l_i}^{t-1}, w_{ex_i}^{t-1})$ (subsection4. 2)
13: Client i sends $w_{l_i}^t, w_{ex_i}^t$ to server
14: **end for**
15: **for** $i \in S_t$ **do**
16: **Model aggregation**$(\{w_{l_i}^t, w_{ex_i}^t\}_{i \in S_t})$ (subsection4. 3)
18: Server sends $w_{l_i}^t$ to client i
19: **end for**
20: **end for**

Output:local model $\{w_{l_i}^R\}_1^{|S|}$

5 評価実験

本章では不均一性のある 2 つのデータセットを使用し FedMe の学習可能性と推論精度を検証する．実験は（1）全てのクライアントで同じ構造のモデルを利用する実験と（2）クライアント毎に異なる構造のモデルを利用する実験の 2 つで行う．簡単化のために，Pytorch [31] 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

5.1.1 データセット

実験では Federated EMNIST-62 データセット（FEMNIST）[32] と Shakespeare データセットの 2 つを利用する．FEMNIST は TensorFlow Federated (TFF) で公開されているものを使用する [1]．データは 28×28 ピクセルの画像データであり，0–9,a–z,A–Z の計 62 種類のラベルから構成されている．ローカルデータはクライアント毎の手書き文字であり，画像データの特徴が異なる．FEMNIST には本来 3400 のクライアントが用意されているが，実験ではそのうち 100 のみを使用する．パブリックデータやラベル無しデータが必要な場合は，使用する 100 のクライアントのデータの 1% に相当する数のデータを，残りのクライアントのデータからランダムに選択する．

Shakespeare データセットは *The Complete Works of William Shakespeare* から作成されたもので，演劇中の役がそれぞれクライアントに割り当てられ，役毎のセリフがローカルデータである．Li ら [25] と同様のクライアントを作成し，前処理を行った．演劇中のセリフの次に来る文字（計 90 種類）を推論する．パブリックデータおよびラベル無しデータが必要な場合は全学習用データの 1% に相当する数のデータをランダムに選択し，選択されたデータはクライアントのローカルデータから削除して重複しないようにする．

2 つのデータセットの統計量を表 1 に示す．どちらのデータセットもクライアント毎のローカルデータの数に差がある．本実験では，学習用データを 7 : 3 に分割して新たに訓練用データと検証用データとする．

5.1.2 モデル

FEMNIST に対しては CNN を使用して多クラス分類を行う．全てのクライアントが同じ構造のモデルを利用する実験では 3×3 のカーネルを用いた畳み込み層を 2 層，プーリング層，

1：https//www.tensorflow.org/federated/api_docs/python/tff/simulation/datasets/emnist

表 1: データセットの統計量

データセット	クライアント数	総数	平均	標準偏差	最大数	最小数
FEMNIST（訓練用）	100	31825	318.25	50.70	393	118
FEMNIST（テスト用）	100	3621	36.21	5.64	45	14
Shakespeare（訓練用）	143	413629	2892.51	5446.75	33044	2
Shakespeare（テスト用）	143	103477	723.62	1361.69	8261	1

ドロップアウト，全結合層を 2 層で構成されたモデルを使用する．クライアント毎に異なる構造のモデルを利用する実験では畳み込み層の数を 1–4 に変化させる．Shakespeare に対しては RNN を使用して次文字予測をする．全てのクライアントが同じ構造のモデルを利用する実験では入力を 8 次元に埋め込む埋め込み層，256 個のノードを持った LSTM を 2 層，全結合層を 1 層で構成されたモデルを使用する．クライアント毎に異なる構造のモデルを利用する実験では．LSTM 層を 1–4 に変化させる．

5.1.3 ハイパーパラメータ

FEMNIST に対しては，バッチサイズを 20，各エポックの参加クライアントを 10，各エポック中のクライアント学習を 2 回とする．Shakespeare に対しては，バッチサイズを 10，各エポックの参加クライアントを 10，各エポック中のクライアント学習を 2 回とする．また，最適化手法は両データセットともに勾配降下法を利用し，各手法の学習率はグリッドサーチによって最適化する．また，HypCluster と FedMe のクラスタ数は 2 とする．

5.1.4 ベースライン

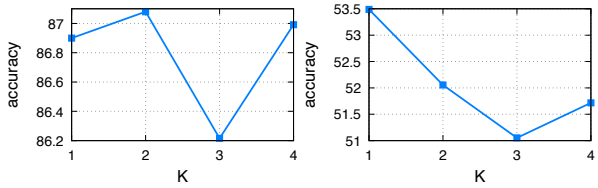
全てのクライアントが同じ構造のモデルを利用する実験のベースラインには，各クライアントが自身のローカルデータのみを使用して学習する local only，全データをサーバが持っているものとしてモデルを学習する Centralized，標準的な連合学習手法である FedAvg [4]，データの不均一性に対処する手法である HypCluster [5]，MAPPER [5]，FedMD [10]，FML [11] を使用する．Centralized，FedAvg，HypCluster は，各クライアントのローカルデータで再学習する fine-tune をベースラインに追加する．クライアント毎に異なる構造のモデルを利用する実験のベースラインには local only，FedMD，FML を使用する．

5.2 全てのクライアントが同じ構造のモデルを利用する実験

ベースラインと FedMe を比較する．FedMe のモデルクラスタリングによる影響を検証するため $K = 1$，すなわち交換モデルをランダムに選択する場合も比較する．表 2 に実験結果を示す．Centralized が最も精度が高く，local only が最も精度が低い．2 つのデータセットには不均一性があるため，連合学習手法の精度は Centralized よりも低い．一方で，連合学習手法は local only よりも精度が高い．ローカルデータのみの学習では過学習を起こしてしまい，連合学習手法の有効性がわかる．既存手法の中では FEMNIST と Shakespeare に対しては FedAvg (fine-tune) と HypCluster がそれぞれ最も精度が高い．HypCluster では，FEMNIST に対してはほとんど FedAvg と

表 2: 同じモデル構造を利用する場合の実験結果

手法	FEMNIST	Shakespeare
local only	66.59	24.18
FedAvg	86.30	48.80
FedAvg (fine-tune)	86.84	48.90
HypCluster	86.31	51.32
HypCluster (fine-tune)	86.25	51.06
MAPPER	82.07	42.96
FedMD	71.09	41.10
FML	71.34	30.22
FedMe (K=1)	86.90	**53.49**
FedMe (K=2)	**87.08**	52.06
Centralized	86.60	55.13
Centralized (fine-tune)	87.23	55.29



(a) FEMNIST (b) Shakespeare

図 2: クラスタ数 K の与える影響

精度が変わらないが，Shakespeare に対しては FedAvg よりも精度が高い．これは FEMNIST よりも Shakespeare の方がデータの不均一性があり，ユーザクラスタリングの効果が高いためだと考えられる．FedMD と FML は，local only よりも精度が高いが他の手法に比べて精度が低い．FedMD と FML の精度が低いのは，どちらもローカルモデルが各クライアントのローカルデータ上でしか学習しないためである．

連合学習手法の中では，2 つのデータセットで FedMe が最も精度が高い．このことより，モデル交換や深層相互学習することの有効性がわかる．しかし，Shakespeare に対しては $K = 1$ と比べて $K = 2$ の方が精度が低い．クラスタ数が精度に与える影響を調べるため，2 つのデータセットに対して FedMe のクラスタ数 K を 1 から 4 に変化させて実験を行う．実験結果を図 2 に示す．FEMNIST では $K = 2$ の場合が最も精度が高いため，最適なクラスタ数は 2 であることがわかる．Shakespeare では $K = 1$ の場合が最も精度が高く，モデルクラスタリングを行うことで精度が低下している．これは，モデルクラスタリングを行うことで類似したモデル間の深層相互学習が多くなり，モデルの汎用性が失われたためだと考えられる．

表 3: 各モデルを選択したクライアントの総数

モデル	FEMNIST	Shakespeare
Conv/LSTM1 層	18	86
Conv/LSTM2 層	29	36
Conv/LSTM3 層	32	13
Conv/LSTM4 層	21	8

表 4: 異なるモデル構造を利用する場合の実験結果

手法	FEMNIST	Shakespeare
local only	67.94	27.72
FedMD	70.59	36.00
FML	72.58	30.59
FedMe (K=1)	**86.51**	51.77
FedMe (K=2)	86.26	**52.23**

5.3 クライアント毎に異なる構造のモデルを利用する実験

実験 1 と同様にベースライン，FedMe ($K = 1$)，FedMe ($K = 2$) を比較する．まず，畳み込み層と LSTM 層を 1–4 に変化させて local only の実験を行い，各クライアントはそれぞれ最も精度が高いモデルを選択した．各モデルを選択したクライアントの総数を表 3 に示す．表 3 より，クライアント毎に適したモデル構造が異なることがわかる．各クライアントは，それぞれ選択したモデルを使用して他の手法の実験を行った．

表 4 に実験結果を示す．local only では，クライアント毎にモデル構造を変化させた方が精度が高く，ローカルデータに合わせてモデル構造を選択することの有効性がわかる．

他の手法の実験結果を表 4 に示す．全てのクライアントが同じ構造のモデルを利用する実験と同様に，local only が最も精度が低く，FedMD と FML と比べて FedMe は精度が高い．一方で，ほとんどの手法において，同じモデル構造を利用する場合と比べて，異なるモデル構造を利用することで精度が低下している．これは，local only における最適なモデル構造と連合学習における最適なモデル構造が異なっているためである．

図 3 に FedMD，FML，FedMe のエポック毎の参加クライアントの検証用データに対する精度の平均を示す．無印が全てのクライアントが同じ構造のモデルを利用する実験を表し，MH がクライアント毎に異なる構造のモデルを利用する実験を表している．2 つのデータセットにおいて FedMD の初期精度が高いのは，あらかじめパブリックデータによる転移学習を行っているためである．Shakespeare で精度の平均が安定していないのは，Shakespeare ではクライアント毎の精度に差があるためである．図 3 からも，FedMD や FML と比べて FedMe が高精度であることがわかる．

6 ま と め

本稿では，データの不均一性に対処するため，クライアント毎に異なる構造のモデルを保持できる連合学習手法，FedMe を提案した．FedMe ではモデルをクライアント間で交換し合い，それらのモデルを深層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．評価実験では 2 つのデータセットに対して FedMe と既存手法の比較を行い，FedMe が既存手法と比べて高精度であることを示した．

今後の研究として 3 つの課題に取り組む．まず，FedMe ではモデルクラスタリングすることで推論精度が低下することがある．これは，モデルの汎用性が失われることが原因と考えられるため，学習初期では類似していないモデルを交換モデルとし，学習が進むにつれて徐々に類似したモデルを交換モデルとするように段階的に学習をする必要がある．次に，FedMe では各クライアントのモデル構造の決定方法について定めていない．モデル構造は推論精度に大きく影響するため，モデル構造を動的に決定する必要がある．最後に，FedMe では新規参入したクライアントに対応できない．これは，ローカルモデルは十分に学習されないため，新規参入したクライアントの推論精度が低下してしまうのが原因である．連合学習においてクライアントが新規参入することは十分に考えられるため，新規参入したクライアントのローカルモデルのための効率の良い学習アルゴリズムの考案が必要である．

謝　　辞

本研究は科学研究費（JP20H00584）の支援によって行われた．実験には産総研の AI 橋渡しクラウド (ABCI) を利用した．ここに記して謝意を表す．

文　　献

[1] 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 preprint arXiv:1811.03604*, 2018.

[2] David Leroy, Alice Coucke, Thibaut Lavril, Thibault Gisselbrecht, and Joseph Dureau. Federated learning for keyword spotting. In *International Conference on Acoustics, Speech and Signal Processing*, pp. 6341–6345, 2019.

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

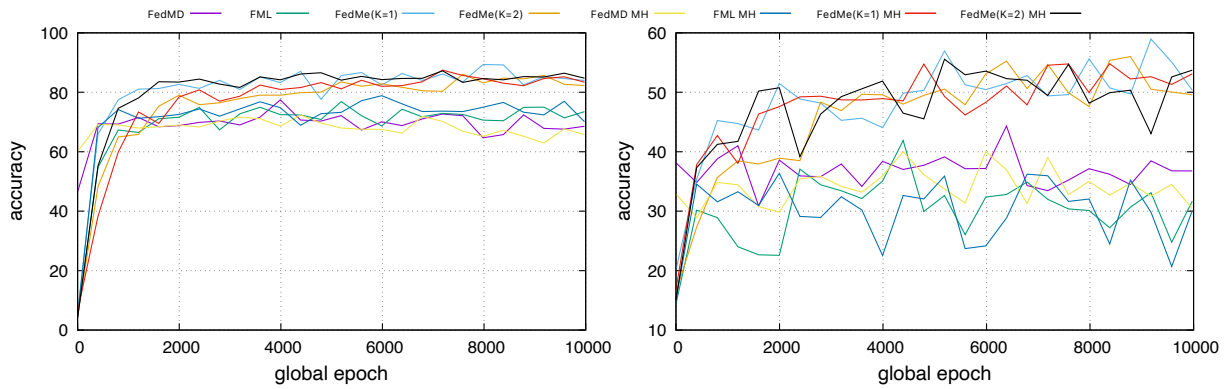
[4] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pp. 1273–1282, 2017.

[5] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020.

[6] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

[7] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. *Advances in Neural Information Processing Systems*, pp. 4424–4434, 2017.

[8] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*, 2020.



(a) FEMNIST (b) Shakespeare

図 3: 学習曲線

[9] Mikhail Khodak, Maria-Florina F Balcan, and Ameet S Talwalkar. Adaptive gradient-based meta-learning methods. In *Advances in Neural Information Processing Systems*, pp. 5917–5928, 2019.

[10] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

[11] Tao Shen, Jie Zhang, Xinkang Jia, Fengda Zhang, Gang Huang, Pan Zhou, Kun Kuang, Fei Wu, and Chao Wu. Federated mutual learning. *arXiv preprint arXiv:2006.16765*, 2020.

[12] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu. Deep mutual learning. In *Conference on Computer Vision and Pattern Recognition*, pp. 4320–4328, 2018.

[13] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *Artificial Intelligence and Statistics*, pp. 2021–2031, 2020.

[14] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*, pp. 634–643, 2019.

[15] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *Artificial Intelligence and Statistics*, pp. 2938–2948, 2020.

[16] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

[17] Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Titouan Parcollet, and Nicholas D Lane. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*, 2020.

[18] 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. *arXiv preprint arXiv:2007.13518*, 2020.

[19] 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 Communications Surveys & Tutorials*, 2020.

[20] 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 preprint arXiv:1912.04977*, 2019.

[21] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020.

[22] Lumin Liu, Jun Zhang, SH Song, and Khaled Ben Letaief. Edge-assisted hierarchical federated learning with non-iid data. *arXiv preprint arXiv:1905.06641*, 2019.

[23] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2019.

[24] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

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

[26] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143, 2020.

[27] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[28] Ming Xie, Guodong Long, Tao Shen, Tianyi Zhou, Xianzhi Wang, and Jing Jiang. Multi-center federated learning. *arXiv preprint arXiv:2005.01026*, 2020.

[29] James MacQueen, et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1, pp. 281–297, 1967.

[30] Neel Guha, Ameet Talwalkar, and Virginia Smith. One-shot federated learning. *arXiv preprint arXiv:1902.11175*, 2019.

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pp. 8026–8037, 2019.

[32] 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 preprint arXiv:1812.01097*, 2018.