連合学習における知識転移グラフに関する研究

TP21002 岩田幸　　指導教授：山下隆義

1. はじめに

連合学習 [1] は複数のクライアントで学習する際に，データセットを共有せず，重みパラメータのみを共有して学習する手法である．データを所有する側をクライアント，クライアントが持つモデルの重みパラメータをまとめる役割を持つ側をサーバと呼ぶ．このとき，クライアント間でデータセットを共有しないため，各クライアントが持つデータのプライバシーを保護した状態で学習できる．各クライアントが保有するデータセットを共有しないことにより，全体のデータ分布を考慮した学習が困難となる問題がある．そのため，各クライアントが保有するデータのクラス分布やデータ量が異なる場合に 1 つのサーバモデルを学習することは難しい．

本研究では，サーバが配布した Meme モデルとクライアントが持つモデルの共同学習を行う．ここで，共同学習とはネットワークが知識を転移しながら学習することを指す．このとき，各クライアントで共同学習を行うことで獲得したサーバモデルを最適化することにより，データ分布の影響を低減した学習が期待できる．

2. 連合学習

連合学習 [1] は，複数のクライアントそれぞれで学習した重みを 1 つのサーバに集約して更新する手法である．学習結果を 1 箇所に集約するため，データセットが非同一分布である際に安定した学習が困難である．そのため，モデルの重みを直接集めるのではなく，クライアント独自のモデルを用意することにより，非同一分布のデータに対応する Federated Mutual Learning (FML) [2] が提案されている．FML は，クライアント独自のモデルと Meme モデルを相互学習 [3] する．クライアント独自のモデルをつくることにより，従来の連合学習と比べてクライアントごとに特化した学習が期待できる．しかし，サーバが配布したモデルとクライアントが保有するモデルの学習方法は限定的であり，データ分布が異なるクライアントを用いた学習に適しているとは限らない．また，クライアントごとのモデル構造を人手で選択する必要がある．

3. 提案手法

本研究では，FML のサーバが配布した Meme モデルとクライアントが保有するモデルの共同学習に知識転移を導入し，最適化することを提案する．損失関数に 3 種のゲート関数 [4] を組み込むことにより，各クライアント最適な学習方法の獲得を目指す．

3.1 知識転移を導入した FML

図 1 に提案手法の概念図を示す．サーバが保有するモデルはサーバモデルのみであり，Meme モデルとクライアントモデルはクライアントが保有している．Meme モデルとクライアントモデルの学習で知識転移する際の学習方法を多様な学習表現から最適化することにより，非同一分布に適した学習の獲得をする．以下に学習の流れを示す．

Step 1. Meme モデルとクライアントモデル間で知識転移する際のゲート関数やクライアントモデルをランダムに設定

Step 2. サーバモデルの重みをクライアントの Meme モデルに配布

Step 3. 各クライアントが保有するデータを用いて，Meme モデルとクライアントモデルを共同学習

Step 4. Meme モデルの重みをサーバに集約してサーバモデルを更新

Step 5. 学習が収束するまで Step 2 に戻る

Step 6. 事前に指定した探索回数になるまで Step 1 に戻る

Step 7. サーバモデルの検証精度が最大となる学習方法を選択

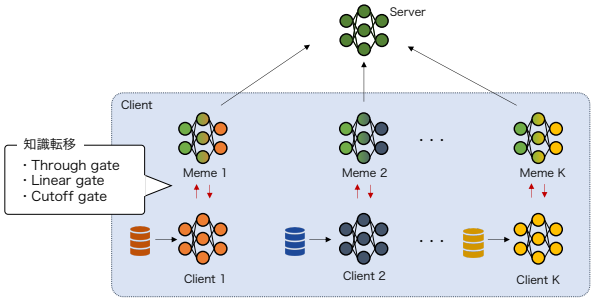

図 1：提案手法の概念図

3.2 クライアントモデルの共同学習

サーバが配布した Meme モデルとクライアントが保有するモデルの共同学習を行う．このとき，出力確率分布の損失は，Kullback-Leibler (KL) divergence を使用して算出する．クライアントモデルと Meme モデルの相互学習に使用する損失関数 $L_{s,t}$ は式 (1) のように求める．

$$L_{s,t} = \sum_{n}^{B} G_{s,t}(KL(\boldsymbol{p}_s(\boldsymbol{x}_n)||\boldsymbol{p}_t(\boldsymbol{x}_n))) + L_{C_t} \quad (1)$$

ここで，$\boldsymbol{p}_s$ は知識転移元のモデルの確率分布，$\boldsymbol{p}_t$ は知識転移先のモデルの確率分布，B はバッチサイズ，$G(\cdot)$ は，ゲート関数，L_{C_t} はモデル t と教師ラベルのクロスエントロピー誤差である．得られた損失から勾配を求め，クライアントモデルと Meme モデルを更新する．

3.3 ゲート関数

損失関数をゲート関数で制御することにより，それぞれのネットワークでの知識に多様性を持たせる．ゲート関数として Through Gate， Cutoff Gate， Linear Gate の 3 種類の関数を定義する．

Through Gate は入力されたサンプルごとの損失に変化を加えないゲート関数である．式 (2) に Through Gate を示す．Through Gate により，FML が表現可能である．

$$G_{s,t}^{\mathrm{Through}}(a) = a \quad (2)$$

Cutoff Gate は損失計算を行わないゲート関数である．式 (3) に Cutoff Gate を示す．Cutoff Gate を用いることにより，知識の伝達を切断することができる．

$$G_{s,t}^{\mathrm{Cutoff}}(a) = 0 \quad (3)$$

Linear Gate は学習時間によって損失の重みを線形に変化させるゲート関数である．式 (4) に Linear Gate を示す．学習初期には重みを小さくし，学習が進むにつれて重みを大きくする．そのため，学習初期で精度が低いモデルの知識が伝達することを抑えることができる．

$$G_{s,t}^{\mathrm{Linear}}(a) = \frac{c}{c_{end}} \cdot a \quad (4)$$

ここで，c は累積更新回数であり，c_{end} は学習終了時における更新回数である．

3.4 サーバモデルの更新

更新したクライアントの Meme モデルをサーバに集約し，サーバモデルを更新する．サーバモデルの更新後の重み w_{t+1} はクライアント k に配布して更新した Meme モデル w_{t+1}^{k} を用いて式 (5) で求める．

$$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^{k} \quad (5)$$

ここで，K はクライアント数，n はクライアント全体のデータ数，n_k はクライアント k が保有するデータ数である．各

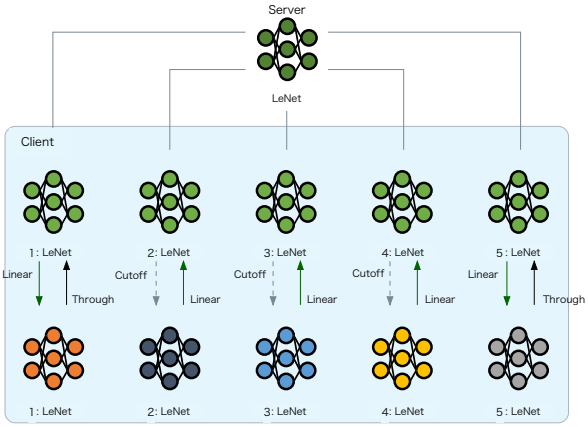

図 2：MNIST での最適化結果（Top1）

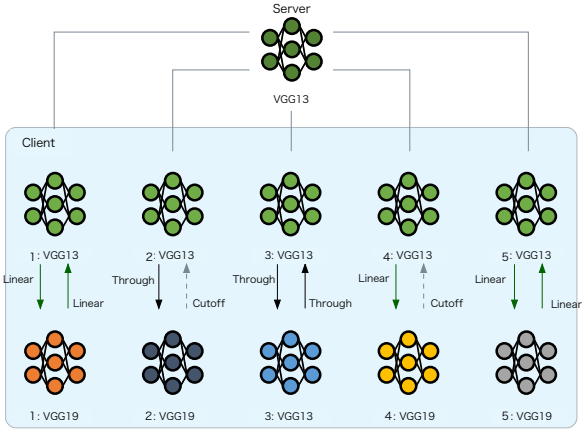

図 3：CIFAR-10 での最適化結果（Top1）

クライアントの重みは各クライアントが保有するデータ量によってサーバに対する影響が決定できる．各 Meme モデルを用いて更新したサーバモデルの重みを各クライアントに配布して Meme モデルを更新する．

3.5 学習の最適化

最適化対象のモデルをサーバモデルとする．サーバモデルの検証精度が最大となるゲート関数とモデルを選択する．使用する最適化手法はランダムサーチである．ゲート関数とモデルをランダムに選択して学習し，検証精度が最大となる学習方法を選択する．これにより，各クライアントのデータを用いて 1 つのモデルを学習する際に効果的な学習方法を獲得できる．

4. 評価実験

提案手法の有効性を調査するために評価実験を行う．従来手法には，連合学習の代表的な手法である Federated Averaging (FedAvg) [1] とゲート関数を全て Through Gate として学習に多様性を失った学習方法である FML を用いる．

4.1 実験条件

実験には 10 クラスの手書き数字の画像 MNIST と一般物体認識データセットである CIFAR-10 データセットを使用する．MNIST の訓練用 60,000 枚を学習用 50,000 枚，検証用 10,000 枚に分割，CIFAR-10 の場合は訓練用 50,000 枚を学習用 40,000 枚，検証用 10,000 枚に分割し，各クライアントで同程度の枚数となるよう分配して使用する．学習は各クライアントにおいてクラスバランスが崩れたデータセットを用い，評価時は評価用データセットを使用して，サーバモデルを評価する．非同一分布のデータセットはディリクレ分布 $\mathrm{Dir}(x|\alpha)$ を使用して作成する．α が小さいほど非同一分布となり，大きいほど同一分布に近くなる．

ネットワークモデルには，MNIST を使用する場合は LeNet，CIFAR-10 を使用する場合は VGG13 と VGG19 を用いる．各クライアントで 50 エポック学習した後にサーバへ集約し，集約する際の通信回数は 200 step である．また，サーバモデルの更新時に使用する使用するクライアント数は 5 であり，常時全てのクライアントを学習に使用する．探索は 1,000 回行い，Top1 の精度を用いて従来手法と比較する．

4.2 獲得した学習方法

ディリクレ分布のハイパーパラメータ α を 1000 に設定した際の MNIST を用いて探索した結果を図 2，CIFAR-10 を用いて探索した結果を図 3 に示す．MNIST と CIFAR-10 どちらにおいても 3 種類のゲート関数が選択されていることから，ゲート関数を用いてクライアントとサーバの学習方法に多様性を持たせることに有効性があるとわかる．

MNIST を用いて探索した結果から，クライアント 2, 3, 4 において Meme モデルは教師ラベルとクライアントモデルからの知識を使用するのに対し，クライアントモデルは教師ラベルのみを用いて学習すると良いことがわかる．このことから，教師ラベルを用いて学習した知識を徐々に学習に取り入れることが有効である．

表 1：非同一分布データセットにおける精度 [%]

データセット	手法名	$\alpha = 1000$	$\alpha = 10$	$\alpha = 0.1$
MINST	FedAvg	88.04	99.22	94.47
	FML	99.22	99.26	96.92
	Ours	**99.30**	**99.33**	**98.50**
CIFAR-10	FedAvg	37.33	35.88	35.22
	FML	40.37	38.76	37.23
	Ours	**42.69**	**39.35**	**38.65**

CIFAR-10 を用いて探索した結果から，クライアント 1, 3, 5 の知識を取り入れることによって学習していることがわかる．このことから，全てのクライアントの知識を使用する必要はないことがわかる．Cutoff Gate の導入により，不要な知識の伝達を防ぐことができた．また，VGG13 と 19 どちらも選択されていることから，両者ともに知識も学習に有効である．

4.3 非同一分布データセットにおける精度比較

非同一分布データセットにおける精度を従来手法と比較する．表 1 に非同一分布データセットにおける精度を示す．提案手法は，従来法である FML と比較して有効であることがわかる．また，MNIST において，非同一分布（α が小さい）に近づくにつれ，FML は 3.30 pt 精度が低下した一方，提案手法は 0.80 pt の精度低下に抑えることができた．このことから，提案手法により非同一分布に有効な学習ができたといえる．

5. おわりに

本研究では連合学習の学習方法を自動最適化することを提案した．評価実験により，提案手法を用いることで，ゲート関数を用いることにより効果的な学習ができることを確認した．また，非同一分布データセットを用いて従来手法と比較して有効性を示した．今後は，探索方法を改良することにより，一度の学習で適切な学習方法を得ることができるよう検討する．

参考文献

[1] H. B. Mcmahan, *et al.*, “Communication-efficient learning of deep networks from decentralized data”, AISTATS, 2017.

[2] T. Shen, *et al.*, “Federated Mutual Learning”, arXiv:2006.16765, 2020.

[3] Y. Zhang, *et al.*, “Deep Mutual Learning”, CVPR, 2018.

[4] S. Minami, *et al.*, “Knowledge Transfer Graph for Deep Collaborative Learning”, ACCV, 2020.

研究業績

[1] S. Iwata, *et al.*, “Refining Design Spaces in Knowledge Distillation for Deep Collaborative Learning”, ICPR, 2022.

(他 1 件)