連合学習入門

米谷 竜

機械学習、とりわけ深層ニューラルネットワークのような複雑なモデルを学習する場合において、大量かつ多様な学習データを収集することは重要である。これに対して連合学習は、通信ネットワークに接続された多数のクライアント計算機がローカルに保持するデータを各々個別に活用することによって、結果的に大規模な学習を実現するアプローチである。連合学習において、クライアントのデータは一箇所に集約されることはない。その代わりに、クライアントは学習をコーディネートするサーバが配布したモデルを自身のデータによって学習し、その学習済みモデルをサーバに共有する。そしてサーバは収集したクライアントのモデルを統合することによって、より高性能なモデルを獲得する。本稿ではこのような連合学習に関するモチベーションや基本的なアプローチ、そしていくつかの代表的な研究について紹介する。

Introduction to Federated Learning

YONETANI Ryo

Collecting a large-scale and diverse data collection is critical for machine learning especially when one needs to train a complex model like deep neural networks. Federated learning is an approach to enable a large-scale training by leveraging decentralized data stored locally by a population of client machines connected to the coordination server. Importantly, federated learning does not require clients to share their own data. Instead, clients will train a global model distributed from the coordination server using their own data and share the trained model with the server. A collection of the trained models is then aggregated to produce a higher-performance model. This article introduces the motivation, typical approaches, as well as some popular studies of the federated learning.

1. まえがき

スタンフォード大学の提供する Coursera によると、機械学習とは “the science of getting computers to act without being explicitly programmed” = 明示的にプログラミングすることなく計算機に（知的な）行動をさせるための科学（著者訳）である[1)]。いま、機械に何らかの意味で知的な行動をさせたいと考える。たとえば、ある写真にりんごが写っているかどうかを判断する機能を計算機上で実現したいとする。これを明示的なプログラミングにより行う場合、「画像中において RGB 値がこの範囲に入っている赤色のピクセルがこのような円形状の領域の一部として現れており、さらにその円領域の一部は光源の影響を受けてこのような RGB 値の範囲で色合いが変化し……」といったように、色や形状に関する特徴の定義とそのパラメタをすべて手動で指定することになる。りんごの品種やりんごの撮影された環境（たとえば屋外か屋内か）、あるいは撮影された状況（木になっているのか手にもたれているのか）が多様になるほど、このプログラミングが困難になることは容易に想像できる。これに対して機械学習は、このようなプログラミングを「大量の事例（学習データ）からの学習」に置き換える。すなわち、りんごがどのような色・形状をしており、それらが撮影環境や状況によってどのように変化するかという特徴の定義やそのパラメタを、大量の「りんごが映った画像」に基づいて自動的に獲得する。ニューラルネットや SVM といった機械学習モデルがこのようにしていったん “学習” されると、新たな画像にりんごが含まれるかどうかを自動的に判断できるようになる。深層学習がコモディティ化する以前は、ユーザ側があらかじめ特徴の抽出方法を定義し、そのパラメタを機械が学習するパラダイムが主流であった。一方深層学習では、特徴の抽出方法（特徴表現、feature representation と呼ばれる）自体もデータから学習可能となる。上では機械学習の中でも特に学習データにおいてモデルの入出力関係が陽に与えられる教師あり学習（より具体的には物体検出タスク）を例に挙げた

Contact : *YONETANI Ryo* ryo.yonetani@sinicx.com

が、ほかにも教師なし学習や強化学習など、機械学習にはさまざまなバリエーションがある。

それでは機械学習によって、我々は労することなく機械による知的な行動を実現できるようになったであろうか ― 実の所問題の難しさの種類が変わっただけであり、明示的なプログラミングの代わりに大量の学習データ（上記の例であれば大量のりんご画像）を収集することが必要となっている。機械学習の実応用においてしばしば問題となるように、学習データの質と量は、機械学習モデルの性能（何％の確率でりんごを見落とし・あるいは見過ぎてしまうか）に直結する。その結果、機械学習分野では「いかに大量・多様なデータを省コストで収集するか」「いかに少数のデータから高性能の機械学習モデルを学習するか」といった課題が継続的に取り組まれている。

本稿で取り上げる連合学習[2)]は、大量・多様なデータを省コストで活用するアプローチの一つである。既存の機械学習の多くは、学習データがあらかじめ一箇所に集約されている、あるいはある単一の計算機から全学習データに容易にアクセスが可能である状況を想定する。これに対して本稿では、なんらかのネットワーク上でサーバと接続された多数のクライアント計算機が、それぞれローカルにデータを分散保持しているという状況を考える。連合学習ではこのような状況において、(1) サーバがランダムなクライアントに学習をしたいモデル（グローバルモデル、と呼ばれる）を配布し、(2) クライアントは配布されたモデルを自身のデータによって学習、サーバに返送する。その後、(3) サーバはクライアントから送られてきたモデルを統合することによってグローバルモデルを更新し、それをまた別のクライアントに配布する（図 1 も参照）。これらの手続きを繰り返すことにより、サーバは最終的にあたかも多数のクライアントのデータを集約して学習されたかのような、高性能なグローバルモデルを獲得することができるようになる。このような連合学習の枠組みにおいて、サーバは自身で大量のデータを集める必要がない。すなわち、「多数の他者にデータ収集と学習を依頼する」という形でデータ収集の問題を解決していると見ることができる。さらに連合学習は、クライアントは収集したデータそのものをサーバに提供する必要がないため、通信やサーバサイドのストレージ、さらにはプライバシーやセキュリティといった観点でも有望な手段である。

連合学習はさまざまな分野において実用に向けた検証が進められている。もともとはモバイルデバイスのキーボード入力における予測変換モデルの学習に利用されていた[3)]。これは、デバイスのハードウェアや OS が全クライアントで概ね共通である点で、連合学習を実施する比較的理想的な問題設定と見ることができる。また医療分野においては、複数の医療機関が患者のデータを直接共有することなく医療診断のための機械学習モデルを学習する手段として、連合学習が注目を集めている[4)]。

本稿では、このような連合学習についてその基礎を紹介するとともに、近年の研究で取り組まれている典型的な課題、さらには我々の研究事例も含め、実応用を見据えた取り組みのいくつかを紹介する。

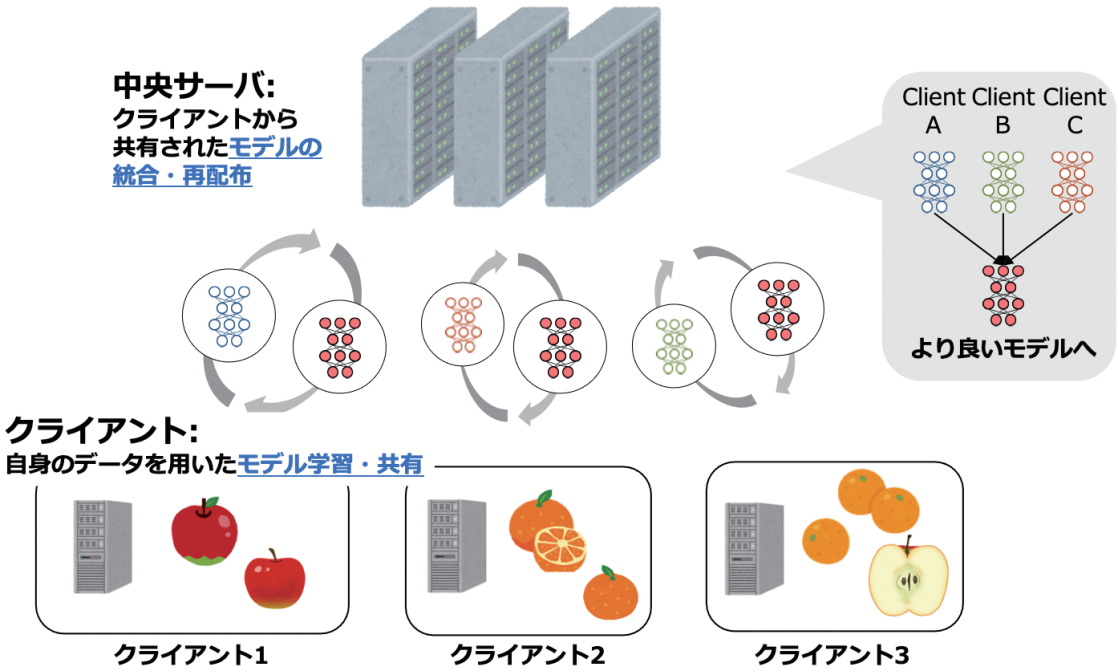

図 1　連合学習

2. 連合学習の基礎

2.1 アルゴリズムの概要

連合学習では、サーバと通信が可能な多数のクライアントデバイス（以降単純に“クライアント”と呼ぶ）を考える。それぞれのクライアントはローカルにデータを保持しており、また機械学習が可能な計算資源を備えていることを想定する。一方サーバは何かしら学習したいモデルを保持しており、これをグローバルモデルと呼ぶ。サーバはランダムに選択されたクライアントに対してグローバルモデルを配布し、クライアントは自身のデータを用いてそのモデルを学習する。その後クライアントは学習済みのモデルをサーバに返送する。ここでサーバは選択されたクライアントと同数のモデルを保持することになる。サーバはこれらのモデルを統合し、新たなグローバルモデルとする。これが連合学習の1ラウンドであり、このような手続きを複数ラウンド繰り返すことで、グローバルモデルの性能を向上させることを目指す。

以下では、連合学習のもっとも基本的なアプローチである Federated Averaging（FedAvg）[2] の具体的なアルゴリズムを示す。

アルゴリズム 1　Federated Averaging

K：クライアント数　　C：クライアント選択の割合
B：ミニバッチサイズ　　E：エポック数
η：学習率
P_k：k 番目のクライアントが持つデータ
n_k：k 番目のクライアントが持つデータのサンプル数
$\nabla L(w; b)$：バッチ b についての損失 $L(w; b)$ の勾配

サーバ：
1. Initialize w_1
2. for each round $t = 1, 2, \cdots$ do
3. $m \leftarrow \max(C \cdot K, 1)$
4. $S_t \leftarrow$(random set of $m = \max(C \cdot K, 1)$ clients)
   # K クライアントのうち C 割をランダムに選択
5. for each client $k \in S_t$; in parallel do
6. $w_{t+1}^{(k)} \leftarrow ClientUpdate(k, w_t)$
7. $w_{t+1} \leftarrow \Sigma_{k \in S_t} \frac{n_k}{n} w_{t+1}^{(k)}$　　$n = \Sigma_{k \in S_t} n_k$　　# モデルの統合

クライアント：
ClientUpdate(k, w):
8. $B \leftarrow$(split P_k into batches of size B)
9. for each local epoch $i = 1, 2, \cdots, E$ do
10. for batch $b \in B$ do
11. $w \leftarrow w - \eta \nabla L(w; b)$
12. return w to server

ここでは、合計 K 人のクライアントのうち、C 割のクライアントが毎ラウンドランダムに選択されることになる。モデルはラウンド t のインデクス付きのパラメタ w_t で表現される。選ばれたクライアントは ClientUpdate サブルーチンにおいてモデルを更新する。その結果得られるモデルはクライアントのデータに依存することになるため、クライアントのインデックスもついたパラメタ $w_{t+1}^{(k)}$ で表現されることになる。11 行目にあるとおり、この更新はいわゆるミニバッチ勾配降下法を用いて、B エポックだけ行われる。最後に、サーバはクライアントが更新したモデルを、クライアントそれぞれが持つデータサンプル数（n_k）で重み付けされた平均処理によって統合する（7 行目）。

2.2 連合学習の特徴

上記のアルゴリズムは、同一のモデルを複数の計算資源にコピーして学習する点で、いわゆるデータ並列を用いたモデルの分散学習と類似している。ただし、連合学習は以下の点で分散学習と大きく異なる。

アプローチの観点から見ると、単一のミニバッチを複数の計算資源に分配する分散学習と異なり、連合学習では各クライアントがミニバッチ勾配降下を複数エポック実行する。すなわち、各クライアントがサーバに返送するモデルは、分散学習の例と比較してよりクライアントのデータに対してより適合したものとなっている。実際、文献[2] ではこのエポック数を大きくすることがグローバルモデルの最終的なパフォーマンスに大きく寄与することが報告されている。一方で問題設定の観点から見ると、分散学習においてデータはあらかじめ一箇所に集約されており、学習において複数の計算資源に分散されるに過ぎない。一方で連合学習では、データはあらかじめ別々のクライアントによって独立に収集・保存されていることを想定する。このとき、クライアントによってデータの収集環境が大きく異なれば、クライアントごとのデータセットが持つ統計的性質もそれに応じて異なり、上記の FedAvg アルゴリズムを用いても効率的な学習が難しいことが知られている。これが連合学習における典型的な課題の一つであり、data non-iidness（データ分布の非独立・同一性）などと呼ばれ、その解決策が積極的に研究されている。

その他にも、クライアントによって計算資源や通信環境が異なる状況を想定することがある。連合学習では基本的に全クライアントのモデルがサーバに収集されたのち統合される（アルゴリズムの 7 行目）ため、学習やモデル送信により多くの時間を必要とするクライアントが含まれる場合、それが全体の連合学習を律速することになる。

3. 連合学習の最先端

2.2 節に述べた項目のみならず、連合学習にはいくつかの典型的な課題があり、それぞれ機械学習分野や通信分野において積極的に研究が進められている。本節ではそのいくつかを紹介する。

3.1 Data non-iidness への対応

連合学習ではクライアントがあらかじめ独立に収集したデータを活用してグローバルモデルを学習する。クライアントのデータ収集環境が多様であれば、最終的に学習されるグローバルモデルもより高い汎化性能を得ることが期待できる。しかしながら実際のところ、クライアントごとに学習データの分布が大きく異なると、学習の効率に悪影響を及ぼすことが知られている。例えば図 1 のように与えられた画像に映るりんごおよびみかんを検出するタスクにおいて、クライアント 1 はりんごの画像のみ、クライアント 2 はみかんの画像のみを保持していた状況を考える。このとき、FedAvg によってクライアント 1 が学習したモデルはりんごの検出に特化し、一方クライアント 2 のモデルはみかんの検出に特化することになる。それでは、これらのモデルのパラメタを単純に平均することで、りんごとみかんの両方を検出できるようになるであろうか。答えは否であり、両クライアントともにりんごとみかんの画像を保持している状況と比較して学習に必要なステップ数が増加する。

これを解決するアプローチの一つが FedProx[5] と呼ばれる手法である。FedProx における基本的な問題意識は、各クライアントが学習したモデルが、そのクライアントのデータに過適合してしまう点にある。これを防ぐために、同手法ではクライアントによるミニバッチ勾配降下において、降下後のパラメタが元のグローバルモデルのパラメタから変化しすぎないような制約を与えている。具体的には、アルゴリズム 1 における 11 行目の更新を以下のように変更する。

$$w \leftarrow w - \eta(\nabla F(w; b) + \mu(w - w_t)) \qquad (1)$$

ただし、w_t はこのラウンドにおいてクライアントが受け取ったグローバルモデルのパラメタであり、μ は制約の強さを示すハイパーパラメタである。図 2 に FedAvg と FedProx の幾何学的な比較を示す。FexProx の方がクライアント 1 に過適合しない形でグローバルモデルを更新していることが分かる。

3.2 効率的な通信

連合学習では、サーバと選択されたクライアントが、学習対象となるモデルを送り合う。このモデルのサイズは学習に必要となるデータよりは小さいことが多いものの、たとえばコンピュータビジョン分野でしばしば使われる ResNet101 モデルでは 171MB 程度となり、通信の効率化は重要である。これに対して、連合学習においてモデルのパフォーマンスを保ったままでできるだけクライアントからサーバへの通信量を減らすアプローチが多く研究されている。本節では、Konečný らによる代表的な手法[6] を紹介する（図 3 も参照）。

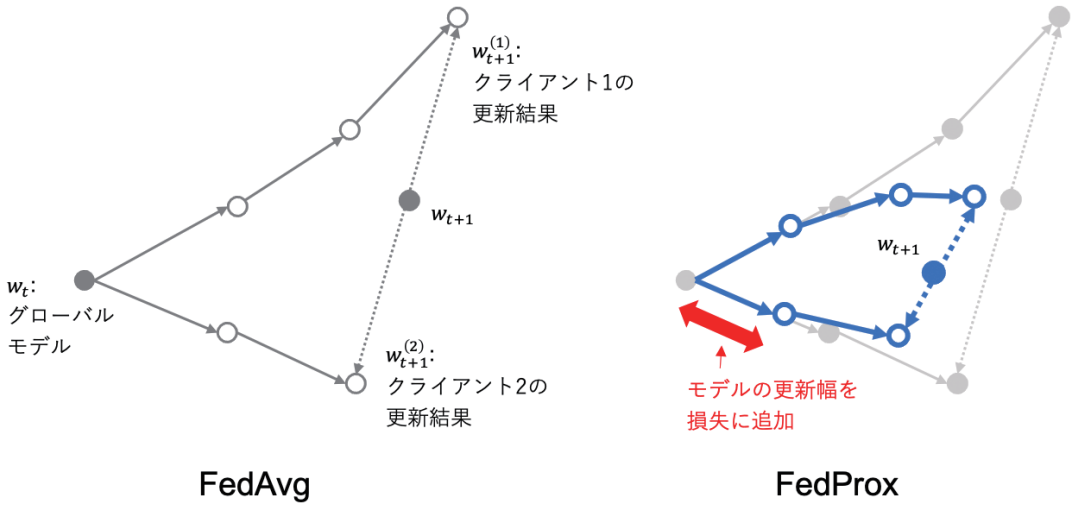

図 2　FedAvg と FedProx の比較

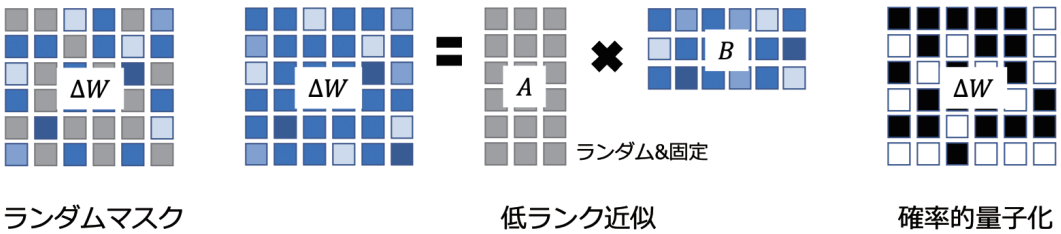

図 3　効率的な通信法

FedAvg アルゴリズムでは、クライアントはグローバルモデル w_t を受け取り、それを更新したモデル $w_t^{(k)}$ をサーバに返送する。このとき、サーバにとって w_t は既知であるため、サーバが受け取る新たな情報は、実質 $\Delta W = w_t^{(k)} - w_t$ という差分になる。したがって、クライアントはこの差分 ΔW をなんらかの手段で圧縮して送れば良いことになる。Konečný らのアプローチのうち最も単純なものはランダムマスク（図 3 左）であり、ΔW と同じサイズで少量のランダムな要素が 1、その他の要素がすべて 0 であるスパースなパターンを用意し、ΔW と掛け合わせたものをサーバに送信する。これは ΔW のうちランダムな数個の要素以外をマスクし、限られた要素のみをサーバに送信することと同義であり、転送量を大幅に低減できる。とはいえ、クライアントの学習結果のうち重要な情報が欠落する可能性もある。

より洗練されたアプローチとして、同論文では ΔW をよりサイズの小さい行列同士の積で近似する（低ランク近似）方法も提案されている（図 3 中）。いま、ΔW がサイズ $d_1 \times d_2$ の行列であるとする（たとえば d_1 次元の特徴ベクトルから d_2 次元の特徴ベクトルへ変換する全結合層を想像すれば良い）。このとき、適当な $k < d_1$ を用いて、$\Delta W = AB,\ A \in \mathbb{R}^{d_1 \times k},\ B \in \mathbb{R}^{k \times d_2}$ という形に ΔW を分解する。さらに、行列 A はランダムに生成し、学習中は固定するものとする。これにより、各クライアントがサーバに対して B のみを送信すれば良いこととなり、その転送量も k/d_1 まで削減されることになる。差分 ΔW における多くの要素を完全に捨ててしまう先のランダムマスクを用いたアプローチと異なり、本手法は ΔW の全ての要素を（近似しつつ）サーバに共有できるメリットがある。

ΔW の全ての要素を近似することで転送量を低減させるもう一つのアプローチとして、確率的量子化（図 3 右）がある。直感的には、深層学習フレームワークの中で通常 32 ビットの浮動小数点数型で表現される ΔW を 1 ビット（2 値）に 2 値化する。ただし、この際の手続きを確率的かつ各要素に対して適応的に行う。具体的には、ΔW の最大値を $\Delta w_{\max}$、最小値を $\Delta w_{\min}$ としたとき、ΔW 中のある要素 Δw を、$\frac{\Delta w - \Delta w_{\min}}{\Delta w_{\max} - \Delta w_{\min}}$ の確率で $\Delta w_{\max}$、$\frac{\Delta w_{\min} - \Delta w}{\Delta w_{\max} - \Delta w_{\min}}$ の確率で $\Delta w_{\min}$ に変換する。このとき、変換後の要素の期待値は元の Δw であることが簡単に確認できる。

3.3 クライアント選択と公平性

連合学習の各ラウンドにおいて、サーバは（1）ランダムなクライアントを選択し、グローバルモデルを配布する。そして、（2）選択されたクライアントが学習したモデルを全て収集し、平均統合することにより、新たなグローバルモデルを獲得する。このとき、（1）において選択したクライアントの一部が限られた計算資源しか保持しない場合や不安定な通信環境にいる場合、サーバへのモデル返送が遅れることがある。その結果、サーバは（2）の平均統合処理を実施できず、学習全体が遅れることとなる。

このような問題を解決する一つの手段が、適応的なクライアント選択[7] である。このアプローチにおいて、サーバはあらかじめ全クライアントについてのモデル学習および送信にかかる時間の見積もりを知ることができるものとする。そして、たとえば 10 分間など限られた時間の中で、できるだけ多くのクライアントがモデルを送信できるように、クライアントの組合せを適応的に選択する。これにより、ランダムにクライアントを選択する場合と比較して、モデルが所望の性能に到達するまでの時間を大幅に短縮できることが実験的に示されている。

一方で、連合学習に参加するクライアントが偏ることで、公平性上の課題があることや、データに偏りがある際にモデルの性能が悪化することも知られている。Agnostic Federated Learning（AFL）[8] はこのような課題に対応するためのアプローチである。いま、あるクライアントによるモデル w の更新を、損失関数 $L_k(w)$ の最小化として定義する。すなわち、$L_k(w)$ に基づいてミニバッチ b で勾配降下する式が アルゴリズム 1 における 11 行目の手続きである。すると、複数クライアントが学習したモデルの統合は、以下の最小化によって表現することができる。

$$\min_w L(w) = \min_w \frac{1}{N} \sum_k L_k(w) \tag{2}$$

これに対して、AFL では以下のような minmax 問題を考える。

$$\min_w \max_\lambda L(w, \lambda) = \min_w \max_\lambda \frac{1}{N} \sum_k \lambda_k L_k(w) \tag{3}$$

上式において λ に関する最大化は、「現状のモデルで損失が大きいクライアントについての損失をより大きく重み付けする」という効果がある。そのようにして重み付けされた損失を w に関して最小化することにより、どのクライアントに対しても公平にモデルが学習されることになる。

3.4 クライアント特化モデル

上記の公平性とは少し異なった観点で、「個々のクライアントごとに特化（パーソナライズ）されたモデルを学習したい」という要求がある。連合学習は通常、単一のグローバルモデルを学習することが最終的な目標となるため、それとは異なるアプローチが必要である。

これに対して文献[9] は、Model-agnostic meta learning（MAML）と呼ばれるメタ学習のアプローチに着想したクライアント特化モデルの連合学習手法を提案している。同手法では、連合学習における最小化問題を以下のように変更する。

$$\min_w L(w) = \min_w \frac{1}{N} \sum_k L_k(w - \alpha \nabla L_k(w)) \tag{4}$$

この式は以下のように解釈できる：各クライアントは損失関数 L_k を「最終的なグローバルモデルのパラメタ w から

1ステップ $\alpha\nabla L_k(w)$ だけ勾配降下すると最小化される」ように最小化する。このようにして得られたパラメタ w は全クライアントにわたって共通であるものの、各クライアントはローカルでモデルを $w-\alpha\nabla L_k(w)$ という形で各自更新することで、自身のデータに関して最もよく当てはまる（つまり、L_k を最小化する）モデルを獲得することができる。これは言い換えれば、クライアントに特化したモデルを容易に獲得できるということを意味している。

3.5 暗号化

連合学習の持つ特徴の一つは、クライアントがサーバに対して自身のデータを直接共有しなくていいという点である。しかし、これは「クライアントのデータの機密が完全に守られる」ということは必ずしも意味しない。また、サーバ側はクライアントがどのようなデータを使って学習を行なったのかは必ずしも分からない、というのも重要な点である。これらの点をついて連合学習に対する攻撃を検証する研究も取り組まれている。たとえば、Inference attack は学習済みモデルから学習データの典型例を推測・生成したり、ある特定のサンプルが学習データに含まれていたかを推測したりする攻撃である。

このような攻撃を防ぐ方法の一つは、サーバに対してクライアントがモデルを共有する際に、モデルパラメタの具体的値を隠蔽することである。クライアントごとのサンプル数が同一であるという簡単化をすると、連合学習におけるモデル統合は、結局のところ $w_{t+1}=\frac{1}{N}\Sigma_k w_{t+1}^{(k)}$ という平均処理である。これに対して、$b=\Sigma_k a_k$ という加算を a_k の値を隠蔽したままで実現するセキュアな統合プロトコルが多く研究されており、連合学習のシナリオにも適用可能である。たとえば Bonawitz ら[10] が紹介している masking with one-time pads と呼ばれるアプローチでは、モデルの平均統合に先立って、各クライアント k は他クライアント j に対してランダムなベクトル $s_{k,j}\in[0,R)^k$ を生成、交換する。そして、クライアントは $w_{t+1}^{(k)}$ そのものではなく、$x_{t+1}^{(k)}=w_{t+1}^{(k)}+\Sigma_j(s_{k,j}-s_{j,k})$ というランダムベクトルを加算したものを送信する。この時点でサーバは $x_{t+1}^{(k)}$ から元の値 $w_{t+1}^{(k)}$ を知ることはできない。しかしながら、$x_{t+1}^{(k)}$ を全クライアントにわたって加算すると、

$$\sum_k x_{t+1}^{(k)}=\sum_k w_{t+1}^{(k)}+\sum_k\sum_j(s_{k,j}-s_{j,k})=$$

$$\sum_k w_{t+1}^{(k)}+\sum_k\sum_j s_{k,j}-\sum_k\sum_j s_{j,k}=\sum_k w_{t+1}^{(k)} \qquad (5)$$

となり、$\Sigma_k w_{t+1}^{(k)}$ という正しい統合結果を得ることができる。

3.6 教師なし学習への応用

連合学習に関する既存研究のほとんどは、ここまでに紹介したような教師あり学習タスクを想定している。一方冒頭に挙げた通り、機械学習には教師なし学習や強化学習などさまざまなタスクのバリエーションが存在する。ここでは我々の研究事例の一つとして、敵対的生成ネットワーク（Generative Adversarial Networks; GAN）を連合学習の枠組みで学習するアプローチ[11] を紹介する。

いま、N 人のクライアントがそれぞれ確率分布 $p_n(x)$ に従うデータセット X_n を保持している状況を考える（図 4）。

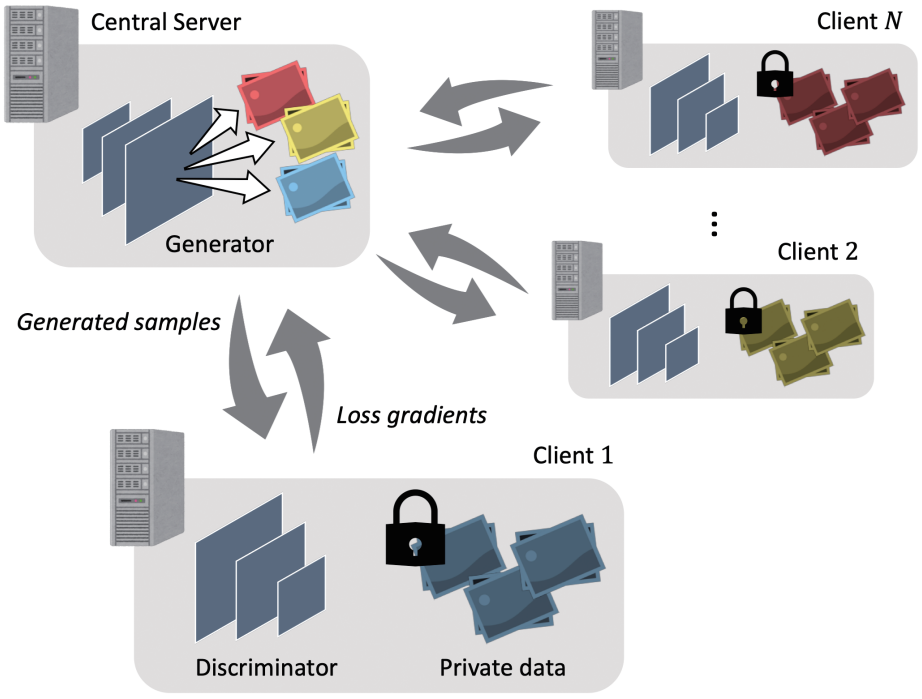

図 4　連合学習の枠組みでの GAN の学習

このとき、これらのデータセットに直接アクセスすることなく、全データを生成可能な分布 $p_{\max}(x)=\frac{1}{z}\max_n p_n(x)\ \ z=\int_x \max_n p_n(x)\mathrm{dx}$ に対応する生成モデル（generator）$G(x)$を学習することが本研究の目的である。提案するアプローチにおいてクライアントは、X_nのかわりに、そのデータと生成モデルが生成したデータを区別する識別器（discriminator）$D_n(x)$を学習する。具体的には、$D_n(x)$ はX_nからサンプリングされたデータについては1、$G(x)$からのデータについては0を出力するように勾配法で学習される。サーバは各クライアントから$D_n(x)$を受け取り、$D_{\max}(x)=\max_n D_n(x)$を計算し、その出力が1に近づくように$G(x)$を更新する。これを繰り返すことにより、$G(x)$の大域的最適解が$p_{\max}(x)$となることが理論的に証明されている。

我々の提案するこのようなアプローチは、例えばファクトリーオートメーションの外観検査において、各現場間で不良品データに偏りがあり、個々の現場のみでは性能向上が困難な場合においても「各現場間でデータを直接やりとりすることなく」、検査アルゴリズムの性能をアップデートすることを可能にする。

4. むすび

ネットワークで接続された多数クライアントが保持するデータを活用した機械学習の一つとして連合学習を紹介した。連合学習ではクライアントはサーバに対して学習済みモデルを共有するのみであり、自身のデータをローカルに留めたままにできるという利点がある。一方、クライアントによって保持するデータの統計的性質や計算・通信資源が異なる場合に学習が非効率化するという課題から、さまざまなアプローチが研究されている。

連合学習は機械学習および通信分野で近年とりわけ研究の活発なトピックの一つであり、ICML, NeurIPS, ICLR, AAAI, IJCAI, ICC, GLOBECOMといった国際会議において多くの研究論文が発表されている。近年の研究動向はKairouzらのサーベイ[12]によくまとめられているので、興味のある読者は参照されたい。

参考文献

1）Stanford Online. “機 械 学 習”, https://www.coursera.org/learn/machine-learning,（参照 2021-07-21）.

2）McMahan, H. B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B. A. y. “Communication-Efficient Learning of Deep Networks from Decentralized Data”. Proceedings of International Conference on Artificial Intelligence and Statistics（AISTATS）, 2017, p.1273-1282.

3）Hard, A. et al. Federated learning for mobile keyboard prediction. arXiv. 2018, preprint arXiv:1811.03604.

4）Rieke, N. et al. The future of digital health with federated learning. NPJ Digital Medicine. 2020, Vol.3, 119.

5）Li, T. et al. “Federated Optimization in Heterogeneous Networks”. Conference on Machine Learning and Systems（MLSys）. 2020, p.1-16.

6）Konečný, J. et al. “Federated Learning: Strategies for Improving Communication Efficiency”. NIPS Workshop on Private Multi-Party Machine Learning. 2016, p.1-10.

7）Nishio, T.; Yonetani, R. “Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge”. IEEE International Conference on Communications（ICC）. 2019, p.1-7.

8）Mohri, M.; Sivek, G.; Suresh, A. T. “Agnostic Federated Learning”, International Conference on Machine Learning（ICML）, PMLR. 2019, 97, p.4615-4625.

9）Fallah, A.; Mokhtari, A.; Ozdaglar, A. “Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta Learning Approach”, Annual Conference on Neural Information Processing Systems（NeurIPS）. 2020, p.1-12

10）Bonawitz, K. et al. “Practical Secure Aggregation for Federated Learning on User-Held Data”, NIPS Workshop on Private Multi-Party Machine Learning. 2016, p.1-5.

11）Yonetani, R.; Takahashi, T.; Hashimoto, A; Ushiku, Y. Decentralized Learning of Generative Adversarial Networks from Non-iid Data, arXiv. 2019, preprint arXiv:1905.09684.

12）Kairouz, P. et al. “Advances and Open Problems in Federated Learning”, Foundations and Trends in Machine Learning. 2021, Vol.14, No.1-2, p.1-210.

執筆者紹介

米谷　竜　YONETANI Ryo
オムロンサイニックエックス株式会社
リサーチアドミニストレイティブディビジョン
専門：コンピュータビジョン、機械学習
所属学会：情報処理学会、電子情報通信学会、IEEE
博士（情報学）